EPSON
EXCEED YOUR VISION

CMOS 16-BIT SINGLE CHIP MICROCONTROLLER

S1C17F57
Technical Manual

SEIKO EPSON CORPORATION

Rev. 1.1

Evaluation board/kit and Development tool important notice

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

NOTICE : PLEASE READ CAREFULLY BELOW 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.) and applications individually listed in this document (“General Purpose”). Epson products are NOT intended for any use beyond the General 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. Therefore, you are advised to use Epson products only for General Purpose uses. Should you desire to buy and use Epson products for a particular purpose other than a General Purpose use, 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.

   【Particular purpose】

   Space equipment (artificial satellites, rockets, etc.)
   Transportation vehicles and their control equipment (automobiles, aircraft, trains, ships, etc.)
   Medical equipment (other than applications individually listed in this document) / 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
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.

(Rev. e1.0, 2021.9)

– Contents –

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 CPU 2-1
- 2.1 Features of the S1C17 Core 2-1
- 2.2 CPU Registers 2-2
- 2.3 Instruction Set 2-2
- 2.4 Reading PSR 2-5
- 2.5 Processor Information 2-6

3 Memory Map, Bus Control 3-1
- 3.1 Bus Cycle 3-1
  - 3.1.1 Restrictions on Access Size 3-2
  - 3.1.2 Restrictions on Instruction Execution Cycles 3-2
- 3.2 Flash Area 3-2
  - 3.2.1 Embedded Flash Memory 3-2
  - 3.2.2 Flash Programming 3-2
  - 3.2.3 Protect Bits 3-2
  - 3.2.4 Flash Memory Read Wait Cycle Setting 3-3
    - FLASHC Read Wait Control Register (FLASHC_WAIT) 3-3
- 3.3 Internal RAM Area 3-3
  - 3.3.1 Embedded RAM 3-3
    - IRAM Size Register (MISC_IRAMSZ) 3-4
- 3.4 Internal Peripheral Area 3-4
  - 3.4.1 Internal Peripheral Area 1 (0x4000–) 3-4
  - 3.4.2 Internal Peripheral Area 2 (0x5000–) 3-4
- 3.5 S1C17 Core I/O Area 3-5

4 Power Supply 4-1
- 4.1 Power Supply Voltage (V_{DD}) 4-1
- 4.2 Flash Programming Power Supply Voltage (V_{PP}) 4-1
- 4.3 Internal Power Supply Circuit 4-1
  - 4.3.1 V_{D1} and V_{OSC} Regulators 4-2
  - 4.3.2 EPD Power Supply Circuit 4-2
  - 4.3.3 EPD Contrast Adjustment 4-3
  - 4.3.4 Heavy Load Protection Mode 4-4
  - 4.3.5 Internal Power Supply Circuit Control Procedures 4-4
- 4.4 Control Register Details 4-5
  - EPD Doubler Clock Control Register (EPD_DCLK) 4-5
  - EPD Booster Clock Control Register (EPD_BCLK) 4-6
  - V_{D1} Control Register (VD1_CTL) 4-7
  - EPD Power Control Register 0 (EPD_PWR0) 4-8
  - EPD Power Control Register 1 (EPD_PWR1) 4-9

5 Initial Reset 5-1
- 5.1 Initial Reset Sources 5-1
  - 5.1.1 #RESET Pin 5-1
  - 5.1.2 P0 Port Key-Entry Reset 5-1

5.1.3 Resetting by the Watchdog Timer 5-1
5.2 Initial Reset Sequence 5-2
5.3 Initial Settings After an Initial Reset 5-2

6 Interrupt Controller (ITC) 6-1
6.1 ITC Module Overview 6-1
6.2 Vector Table 6-2
Vector Table Address Low/High Registers (MISC_TTBRL, MISC_TTBRH) 6-3
6.3 Control of Maskable Interrupts 6-3
6.3.1 Interrupt Control Bits in Peripheral Modules 6-3
6.3.2 ITC Interrupt Request Processing 6-3
6.3.3 Interrupt Processing by the S1C17 Core 6-4
6.4 NMI 6-5
6.5 Software Interrupts 6-5
6.6 HALT and SLEEP Mode Cancellation 6-5
6.7 Control Register Details 6-5
Interrupt Level Setup Register *x* (ITC_LV*x*) 6-6

7 Clock Generator (CLG) 7-1
7.1 CLG Module Overview 7-1
7.2 CLG Input/Output Pins 7-2
7.3 Oscillators 7-2
7.3.1 OSC3B Oscillator 7-2
7.3.2 OSC3A Oscillator 7-4
7.3.3 OSC1 Oscillator 7-4
7.4 System Clock Switching 7-7
7.5 CPU Core Clock (CCLK) Control 7-8
7.6 Peripheral Module Clock (PCLK) Control 7-8
7.7 Clock External Output (FOUTA, FOUTB) 7-9
7.8 Control Register Details 7-10
Clock Source Select Register (CLG_SRC) 7-11
Oscillation Control Register (CLG_CTL) 7-12
FOUTA Control Register (CLG_FOUTA) 7-13
FOUTB Control Register (CLG_FOUTB) 7-14
Oscillation Stabilization Wait Control Register (CLG_WAIT) 7-15
PCLK Control Register (CLG_PCLK) 7-17
CCLK Control Register (CLG_CCLK) 7-18

8 Theoretical Regulation (TR) 8-1
8.1 TR Module Overview 8-1
8.2 TR Output Pin 8-1
8.3 Theoretical Regulation Control 8-1
8.3.1 Setting Regulation Values 8-1
8.3.2 Executing Theoretical Regulation 8-2
8.3.3 Regulated Clock External Monitor 8-2
8.4 Control Register Details 8-3
TR Control Register (TR_CTL) 8-3
TR Value Register (TR_VAL) 8-3

9 Real-Time Clock (RTC) 9-1
9.1 RTC Module Overview 9-1
9.2 RTC Counters 9-1
9.3 RTC Control 9-3
9.3.1 Operating Clock Control 9-3

9.3.2 12-hour/24-hour mode selection 9-3
9.3.3 RTC Start/Stop 9-3
9.3.4 Counter Settings 9-3
9.3.5 Counter Read 9-4
9.4 RTC Interrupts 9-5
9.5 Control Register Details 9-5
RTC Control Register (RTC_CTL) 9-5
RTC Interrupt Enable Register (RTC_IEN) 9-6
RTC Interrupt Flag Register (RTC_IFLG) 9-7
RTC Minute/Second Counter Register (RTC_MS) 9-8
RTC Hour Counter Register (RTC_H) 9-9

10 I/O Ports (P) 10-1

10.1 P Module Overview 10-1
10.2 Input/Output Pin Function Selection (Port MUX) 10-2
10.3 Data Input/Output 10-2
10.4 Pull-up Control 10-3
10.5 Port Input Interrupt 10-4
10.6 P0, P2 Port Chattering Filter Function 10-4
10.7 P0 Port Key-Entry Reset 10-5
10.8 Control Register Details 10-5
P*x* Port Input Data Registers (P*x*_IN) 10-6
P*x* Port Output Data Registers (P*x*_OUT) 10-6
P*x* Port Output Enable Registers (P*x*_OEN) 10-7
P*x* Port Pull-up Control Registers (P*x*_PU) 10-7
P*x* Port Interrupt Mask Registers (P*x*_IMSK) 10-7
P*x* Port Interrupt Edge Select Registers (P*x*_EDGE) 10-8
P*x* Port Interrupt Flag Registers (P*x*_IFLG) 10-8
P*x* Port Chattering Filter Control Registers (P*x*_CHAT) 10-9
P0 Port Key-Entry Reset Configuration Register (P0_KRST) 10-10
P*x* Port Input Enable Registers (P*x*_IEN) 10-10
P0[3:0] Port Function Select Register (P00_03PMUX) 10-11
P0[7:4] Port Function Select Register (P04_07PMUX) 10-12
P1[3:0] Port Function Select Register (P10_13PMUX) 10-13
P1[7:4] Port Function Select Register (P14_17PMUX) 10-14
P2[3:0] Port Function Select Register (P20_23PMUX) 10-15
P2[7:4] Port Function Select Register (P24_27PMUX) 10-16
P3[3:0] Port Function Select Register (P30_33PMUX) 10-17
P34 Port Function Select Register (P34PMUX) 10-18

11 8-bit Timers (T8) 11-1

11.1 T8 Module Overview 11-1
11.2 Count Clock 11-2
11.3 Count Mode 11-2
11.4 Reload Data Register and Underflow Cycle 11-2
11.5 Timer Reset 11-3
11.6 Timer RUN/STOP Control 11-3
11.7 T8 Output Signals 11-4
11.8 T8 Interrupts 11-4
11.9 Control Register Details 11-5
T8 Ch.*x* Count Clock Select Registers (T8_CLK*x*) 11-5
T8 Ch.*x* Reload Data Registers (T8_TR*x*) 11-5
T8 Ch.*x* Counter Data Registers (T8_TC*x*) 11-6
T8 Ch.*x* Control Registers (T8_CTL*x*) 11-6
T8 Ch.*x* Interrupt Control Registers (T8_INT*x*) 11-7

12 16-bit PWM Timers (T16A2) 12-1
12.1 T16A2 Module Overview 12-1
12.2 T16A2 Input/Output Pins 12-2
12.3 Count Clock 12-2
12.4 T16A2 Operating Modes 12-4
12.4.1 Comparator Mode and Capture Mode 12-4
12.4.2 Repeat Mode and One-Shot Mode 12-5
12.4.3 Normal Channel Mode and Multi-Comparator/Capture Mode 12-6
12.4.4 Normal Clock Mode and Half Clock Mode 12-7
12.5 Counter Control 12-7
12.5.1 Counter Reset 12-7
12.5.2 Counter RUN/STOP Control 12-7
12.5.3 Reading Counter Values 12-8
12.5.4 Counter Operation and Interrupt Timing Charts 12-8
12.6 Timer Output Control 12-8
12.7 T16A2 Interrupts 12-10
12.8 Control Register Details 12-12
T16A Clock Control Registers Ch.*x* (T16A_CLK*x*) 12-12
T16A Counter Ch.*x* Control Registers (T16A_CTL*x*) 12-14
T16A Counter Ch.*x* Data Registers (T16A_TC*x*) 12-15
T16A Comparator/Capture Ch.*x* Control Registers (T16A_CCCTL*x*) 12-16
T16A Comparator/Capture Ch.*x* A Data Registers (T16A_CCA*x*) 12-18
T16A Comparator/Capture Ch.*x* B Data Registers (T16A_CCB*x*) 12-18
T16A Comparator/Capture Ch.*x* Interrupt Enable Registers (T16A_IEN*x*) 12-19
T16A Comparator/Capture Ch.*x* Interrupt Flag Registers (T16A_IFLG*x*) 12-20

13 Clock Timer (CT) 13-1
13.1 CT Module Overview 13-1
13.2 Operation Clock 13-1
13.3 Timer Reset 13-1
13.4 Timer RUN/STOP Control 13-2
13.5 CT Interrupts 13-2
13.6 Control Register Details 13-3
Clock Timer Control Register (CT_CTL) 13-3
Clock Timer Counter Register (CT_CNT) 13-4
Clock Timer Interrupt Mask Register (CT_IMSK) 13-4
Clock Timer Interrupt Flag Register (CT_IFLG) 13-4

14 Stopwatch Timer (SWT) 14-1
14.1 SWT Module Overview 14-1
14.2 Operation Clock 14-1
14.3 BCD Counters 14-1
14.4 Timer Reset 14-2
14.5 Timer RUN/STOP Control 14-2
14.6 SWT Interrupts 14-3
14.7 Control Register Details 14-4
Stopwatch Timer Control Register (SWT_CTL) 14-4
Stopwatch Timer BCD Counter Register (SWT_BCNT) 14-4
Stopwatch Timer Interrupt Mask Register (SWT_IMSK) 14-5
Stopwatch Timer Interrupt Flag Register (SWT_IFLG) 14-5

15 Watchdog Timer (WDT) 15-1
15.1 WDT Module Overview 15-1
15.2 Operation Clock 15-1

15.3 WDT Control 15-1
15.3.1 NMI/Reset Mode Selection 15-1
15.3.2 WDT Run/Stop Control 15-2
15.3.3 WDT Reset 15-2
15.3.4 Operations in HALT and SLEEP Modes 15-2
15.4 Control Register Details 15-2
Watchdog Timer Control Register (WDT_CTL) 15-2
Watchdog Timer Status Register (WDT_ST) 15-3

16 UART 16-1
16.1 UART Module Overview 16-1
16.2 UART Input/Output Pins 16-2
16.3 Baud Rate Generator 16-2
16.4 Transfer Data Settings 16-4
16.5 Data Transfer Control 16-5
16.6 Receive Errors 16-7
16.7 UART Interrupts 16-8
16.8 IrDA Interface 16-9
16.9 Control Register Details 16-10
UART Ch.*x* Status Register (UART_ST*x*) 16-10
UART Ch.*x* Transmit Data Register (UART_TXD*x*) 16-12
UART Ch.*x* Receive Data Register (UART_RXD*x*) 16-12
UART Ch.*x* Mode Register (UART_MOD*x*) 16-12
UART Ch.*x* Control Register (UART_CTL*x*) 16-13
UART Ch.*x* Expansion Register (UART_EXP*x*) 16-14
UART Ch.*x* Baud Rate Register (UART_BR*x*) 16-14
UART Ch.*x* Fine Mode Register (UART_FMD*x*) 16-15
UART Ch.*x* Clock Control Register (UART_CLK*x*) 16-15

17 SPI 17-1
17.1 SPI Module Overview 17-1
17.2 SPI Input/Output Pins 17-1
17.3 SPI Clock 17-2
17.4 Data Transfer Condition Settings 17-2
17.5 Data Transfer Control 17-3
17.6 SPI Interrupts 17-5
17.7 Control Register Details 17-6
SPI Ch.*x* Status Register (SPI_ST*x*) 17-6
SPI Ch.*x* Transmit Data Register (SPI_TXD*x*) 17-7
SPI Ch.*x* Receive Data Register (SPI_RXD*x*) 17-7
SPI Ch.*x* Control Register (SPI_CTL*x*) 17-7

18 I^2C Master (I2CM) 18-1
18.1 I2CM Module Overview 18-1
18.2 I2CM Input/Output Pins 18-1
18.3 Synchronization Clock 18-2
18.4 Settings Before Data Transfer 18-2
18.5 Data Transfer Control 18-2
18.6 I2CM Interrupts 18-6
18.7 Control Register Details 18-7
I^2C Master Enable Register (I2CM_EN) 18-8
I^2C Master Control Register (I2CM_CTL) 18-8
I^2C Master Data Register (I2CM_DAT) 18-9
I^2C Master Interrupt Control Register (I2CM_ICTL) 18-10

19 I²C Slave (I2CS) 19-1
19.1 I2CS Module Overview 19-1
19.2 I2CS Input/Output Pins 19-1
19.3 Operation Clock 19-2
19.4 Initializing I2CS 19-2
19.4.1 Reset 19-2
19.4.2 Setting Slave Address 19-3
19.4.3 Optional Functions 19-3
19.5 Data Transfer Control 19-3
19.6 I2CS Interrupts 19-8
19.7 Control Register Details 19-9
I²C Slave Transmit Data Register (I2CS_TRNS) 19-10
I²C Slave Receive Data Register (I2CS_RECV) 19-10
I²C Slave Address Setup Register (I2CS_SADRS) 19-11
I²C Slave Control Register (I2CS_CTL) 19-11
I²C Slave Status Register (I2CS_STAT) 19-13
I²C Slave Access Status Register (I2CS_ASTAT) 19-15
I²C Slave Interrupt Control Register (I2CS_ICTL) 19-16

20 EPD Controller/Driver (EPD) 20-1
20.1 EPD Module Overview 20-1
20.2 EPD Output Pins 20-1
20.3 Power Supply for EPD Driver 20-2
20.4 EPD Clock 20-2
20.5 Operating Mode 20-2
20.6 Display Waveform Memory 20-3
20.7 Display Data Memory 20-4
20.8 Display Control in Wave Mode 20-4
20.9 Display Control in Direct Mode 20-6
20.10 Outputs for External Driver 20-7
20.11 EPD Interrupt 20-7
20.12 Control Register Details 20-7
EPD Timing Clock Control Register (EPD_TCLK) 20-8
EPD Display Control Register (EPD_CTL) 20-9
EPD Interrupt Control Register (EPD_INT) 20-10
EPD Top/Back Plane Data Register (EPD_PLNDAT) 20-10
EPD Display Data Registers 0–3 (EPD_SEGDAT0–EPD_SEGDAT3) 20-11
EPD Waveform Timing Set *x* Registers (EPD_WAVE*x*) 20-12

21 Sound Generator (SND) 21-1
21.1 SND Module Overview 21-1
21.2 SND Output Pins 21-1
21.3 SND Operating Clock 21-1
21.4 Buzzer Frequency and Volume Settings 21-2
21.4.1 Buzzer Frequency 21-2
21.4.2 Volume level 21-2
21.5 Buzzer Mode and Output Control 21-3
21.5.1 Buzzer Mode Selection 21-3
21.5.2 Output Control in Normal Mode 21-3
21.5.3 Output Control in One-shot Mode 21-3
21.5.4 Output Control in Envelope Mode 21-4
21.6 Control Register Details 21-5
SND Clock Control Register (SND_CLK) 21-5

SND Control Register (SND_CTL) 21-5
Buzzer Frequency Control Register (SND_BZFQ) 21-7
Buzzer Duty Ratio Control Register (SND_BZDT) 21-7

22 R/F Converter (RFC) 22-1

22.1 RFC Module Overview 22-1
22.2 RFC Input/Output Pins 22-2
22.3 Operation Clock 22-2
22.4 Operating Modes 22-3
22.4.1 Oscillation Mode 22-3
22.4.2 External Clock Input Mode (Event Counter Mode) 22-4
22.4.3 Functions for Measuring CR Oscillation Clock Frequency 22-4
22.5 RFC Counters 22-5
22.6 Conversion Operations 22-5
22.6.1 Initial Settings 22-5
22.6.2 Reference Oscillation Control 22-6
22.6.3 Sensor Oscillation Control 22-7
22.6.4 Forced Termination 22-7
22.6.5 Conversion Error 22-7
22.7 RFC Interrupts 22-8
22.8 Control Register Details 22-9
RFC Clock Control Registers (RFC_CLK) 22-9
RFC Control Register (RFC_CTL) 22-10
RFC Oscillation Trigger Register (RFC_TRG) 22-11
RFC Measurement Counter Low and High Registers (RFC_MCL, RFC_MCH) 22-12
RFC Time Base Counter Low and High Registers (RFC_TCL, RFC_TCH) 22-12
RFC Interrupt Mask Register (RFC_IMSK) 22-12
RFC Interrupt Flag Register (RFC_IFLG) 22-13

23 Temperature Detection Circuit (TEM) 23-1

23.1 TEM Module Overview 23-1
23.2 Operation Clock 23-2
23.3 TEM Control 23-3
23.4 TEM Interrupt 23-5
23.5 Control Register Details 23-5
TEM Clock Control Register (TEM_CLK) 23-5
TEM Comparison Time Setting Register (TEM_TIME) 23-6
TEM Control Register (TEM_CTL) 23-6
TEM Status Register (TEM_STAT) 23-7
TEM Conversion Result Register (TEM_RSLT) 23-7

24 Supply Voltage Detection Circuit (SVD) 24-1

24.1 SVD Module Overview 24-1
24.2 Comparison Voltage Setting 24-1
24.3 SVD Control 24-2
24.4 Control Register Details 24-2
SVD Enable Register (SVD_EN) 24-2
SVD Comparison Voltage Register (SVD_CMP) 24-3
SVD Detection Result Register (SVD_RSLT) 24-4

25 On-chip Debugger (DBG) 25-1

25.1 Resource Requirements and Debugging Tools 25-1
25.2 Debug Break Operation Status 25-1
25.3 Additional Debugging Function 25-2

25.4 Control Register Details 25-2
Debug Mode Control Register 1 (MISC_DMODE1) 25-2
Debug Mode Control Register 2 (MISC_DMODE2) 25-3
IRAM Size Select Register (MISC_IRAMSZ) 25-3
Debug RAM Base Register (DBRAM) 25-4
Debug Control Register (DCR) 25-4
Instruction Break Address Register 2 (IBAR2) 25-5
Instruction Break Address Register 3 (IBAR3) 25-5
Instruction Break Address Register 4 (IBAR4) 25-6

26 Multiplier/Divider (COPRO) 26-1
26.1 Overview 26-1
26.2 Operation Mode and Output Mode 26-1
26.3 Multiplication 26-2
26.4 Division 26-3
26.5 MAC 26-4
26.6 Reading Operation Results 26-6

27 Electrical Characteristics 27-1
27.1 Absolute Maximum Ratings 27-1
27.2 Recommended Operating Conditions 27-1
27.3 Current Consumption 27-2
27.4 Oscillation Characteristics 27-4
27.5 External Clock Input Characteristics 27-5
27.6 Input/Output Pin Characteristics 27-5
27.7 SPI Characteristics 27-6
27.8 I^2C Characteristics 27-7
27.9 EPD Driver Characteristics 27-7
27.10 R/F Converter Characteristics 27-11
27.11 Temperature Detection Circuit Characteristics 27-13
27.12 SVD Circuit Characteristics 27-14
27.13 Flash Memory Characteristics 27-14

28 Basic External Connection Diagram 28-1

29 Chip 29-1
29.1 Pad/Bump Configuration 29-1
29.2 Gold Bump Specifications 29-3

Appendix A List of I/O Registers AP-A-1
0x4100–0x4107, 0x506c UART (with IrDA) Ch.0 ..... AP-A-4
0x4240–0x4248 8-bit Timer Ch.0 ..... AP-A-5
0x4260–0x4268 8-bit Timer Ch.1 ..... AP-A-5
0x4306–0x4318 Interrupt Controller AP-A-6
0x4320–0x4326 SPI Ch.0 ..... AP-A-6
0x4340–0x4346 I^2C Master AP-A-7
0x4360–0x436c I^2C Slave AP-A-7
0x5000–0x5003 Clock Timer AP-A-8
0x5020–0x5023 Stopwatch Timer AP-A-8
0x5040–0x5041 Watchdog Timer AP-A-8
0x5060–0x5081 Clock Generator AP-A-8
0x5078–0x5079 Theoretical Regulation Circuit AP-A-10
0x5100–0x5102 SVD Circuit AP-A-10
0x5120 Power Generator AP-A-10

0x506e, 0x5180–0x5182	Sound Generator	AP-A-10
0x506f, 0x51a0–0x51a3	Temperature Detection Circuit	AP-A-11
0x5200–0x52a7	P Port & Port MUX	AP-A-11
0x4020, 0x5322–0x532c	MISC Registers	AP-A-15
0x5067, 0x53a0–0x53ae	R/F Converter	AP-A-16
0x5068, 0x5400–0x540c	16-bit PWM Timer Ch.0	AP-A-17
0x5069, 0x5420–0x542c	16-bit PWM Timer Ch.1	AP-A-19
0x54b0	Flash Controller	AP-A-20
0x5070–0x5072, 0x5600–0x567e	EPD Controller/Driver	AP-A-20
0x56c0–0x56c8	Real-time Clock	AP-A-22
0xffff84–0xffffd0	S1C17 Core I/O	AP-A-22

Appendix B Power Saving AP-B-1

B.1 Clock Control Power Saving AP-B-1

B.2 Reducing Power Consumption via Power Supply Control AP-B-3

B.3 Other Power Saving Methods AP-B-3

Appendix C Mounting Precautions AP-C-1

Appendix D Measures Against Noise AP-D-1

Appendix E Initialization Routine AP-E-1

Revision History

1 Overview

1.1 Features

The main features of the S1C17F57 are listed below.

Table 1.1.1 Features

CPU	
CPU core	Seiko Epson original 16-bit RISC CPU core S1C17
Multiplier/Divider (COPRO)	• 16-bit × 16-bit multiplier • 16-bit × 16-bit + 32-bit multiply and accumulation unit • 16-bit ÷ 16-bit divider
Embedded Flash memory	
Capacity	32K bytes (for both instructions and data)
Erase/program count	Three times
Other	• Read/program protection function • A programming power supply (V_{PP}) is required. • Allows on-board programming using a debugging tool such as ICDmini.
Embedded RAM	
Capacity	2K bytes
Clock generator	
System clock source	3 sources (OSC3B/OSC3A/OSC1)
OSC3B oscillator circuit	2M/1M/500k Hz (typ.) internal oscillator circuit
OSC3A oscillator circuit	4.2 MHz (max.) crystal or ceramic oscillator circuit
OSC1B oscillator circuit	32 kHz (typ.) internal oscillator circuit
OSC1A oscillator circuit	32.768 kHz (typ.) crystal oscillator circuit
	Oscillation adjustment by theoretical regulation
Other	• Core clock frequency control • Peripheral module clock supply control
EPD controller/driver	
Number of driver outputs	Segment output: 64 pins Top plane output: 2 pins Back plane output: 2 pins
Drive voltage	0 V/15 V (V_{SS}/V_{EPD})
Other	• Includes a drive power supply. • Includes a display data memory. • Output drive waveforms can be programmed. • Supports pin output direct control.
I/O ports	
Number of general-purpose I/O ports	Max. 29 bits (Pins are shared with the peripheral I/O.)
Other	• Schmitt input • Pull-up control function • Port input interrupt: 8 bits × 2 channels
Serial interfaces	
SPI	1 channel
I²C master (I2CM) *1	1 channel
I²C slave (I2CS) *1	1 channel
UART	1 channel (IrDA1.0 supported)
Timers/Counters	
8-bit timer (T8)	2 channels (Generates the SPI Ch.0 and I2CM clocks.)
16-bit PWM timer (T16A2)	2 channels (PWM output, event counter, and count capture functions)
Watchdog timer (WDT)	1 channel (Generates NMI/reset.)
Clock functions	
Real-time clock (RTC)	1 channel (Hour, minute, and second counters) with theoretical regulation support
Clock timer (CT)	1 channel (128 Hz to 1 Hz counters) with theoretical regulation support
Stopwatch timer (SWT)	1 channel (1/100 second and 1/10 second counters) with theoretical regulation support
Theoretical regulation function (TR)	Time adjustment function in +16/32768 to -15/32768 second units
Sound generator	
Buzzer frequency	8 frequencies selectable
Volume control	8 steps adjustable
Other	• One-shot buzzer • Auto envelope function

Analog circuits		
R/F converter (RFC)	2 channels (24-bit CR oscillation type. Supports DC-bias resistive sensors and AC-bias resistive sensors.)	
Temperature detection circuit (TEM)	1 channel (Measurement range: 0°C to 50°C)	
Supply voltage detection circuit (SVD)	1 channel (Detection voltage: 13 levels)	
Interrupts		
Reset interrupt	#RESET pin/watchdog timer	
NMI	Watchdog timer	
Programmable interrupts	16 systems (8 levels)	
Power supply voltage		
Operating voltage (V_{DD})	2.0 V to 3.6 V	
Flash programming/erasing voltage (V_{PP})	7 V/7.5 V	
Operating temperature		
Operating temperature range	-40°C to 85°C	
Current consumption (Typ value, V_{DD} = 2.0 V to 3.6 V)		
SLEEP state *2	100 nA	(OSC1 = Off, RTC = Off, OSC3B = Off, OSC3A = Off)
HALT state	0.55 µA	(OSC1 = 32 kHz (OSC1A), RTC = Off, OSC3B = Off, OSC3A = Off)
	0.5 µA	(OSC1 = 32 kHz (OSC1A), RTC = On, OSC3B = Off, OSC3A = Off)
Run state	12 µA	(OSC1 = 32 kHz (OSC1A), RTC = Off, OSC3B = Off, OSC3A = Off)
	1440 µA	(OSC1 = Off, RTC = Off, OSC3B = Off, OSC3A = 4 MHz ceramic)
	770 µA	(OSC1 = Off, RTC = Off, OSC3B = 2 MHz, OSC3A = Off)
Shipping form		
1	Aluminum pad chip	
2	Gold bump chip	

*1 The input filters in I2CM/I2CS (SDA and SCL inputs) do not comply with the standard for removing noise spikes less than 50 ns.
*2 The RAM retains data even in SLEEP mode.

1.2 Block Diagram

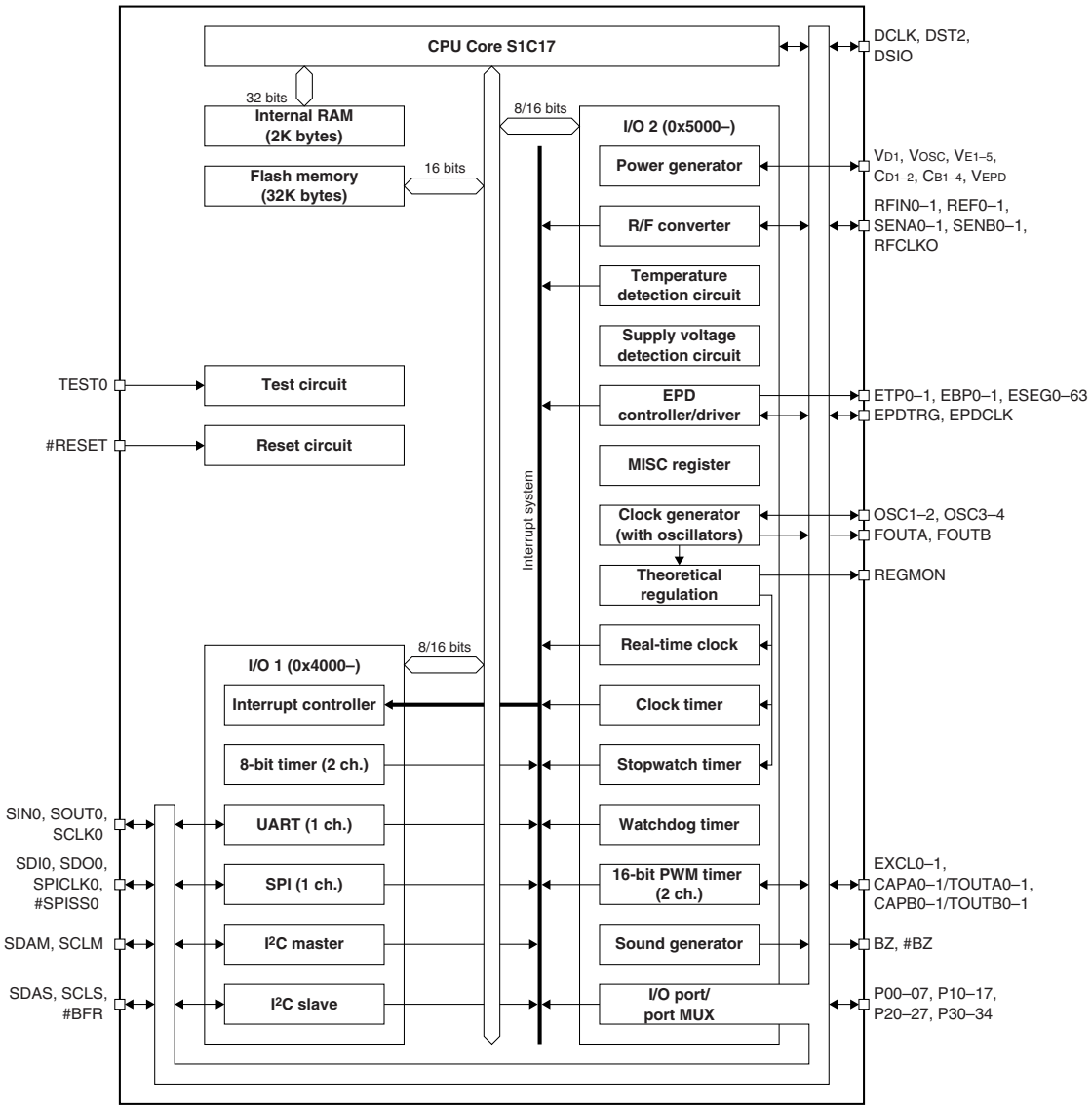


Figure 1.2.1 S1C17F57 Block Diagram

1.3 Pins

1.3.1 Pin Configuration Diagram

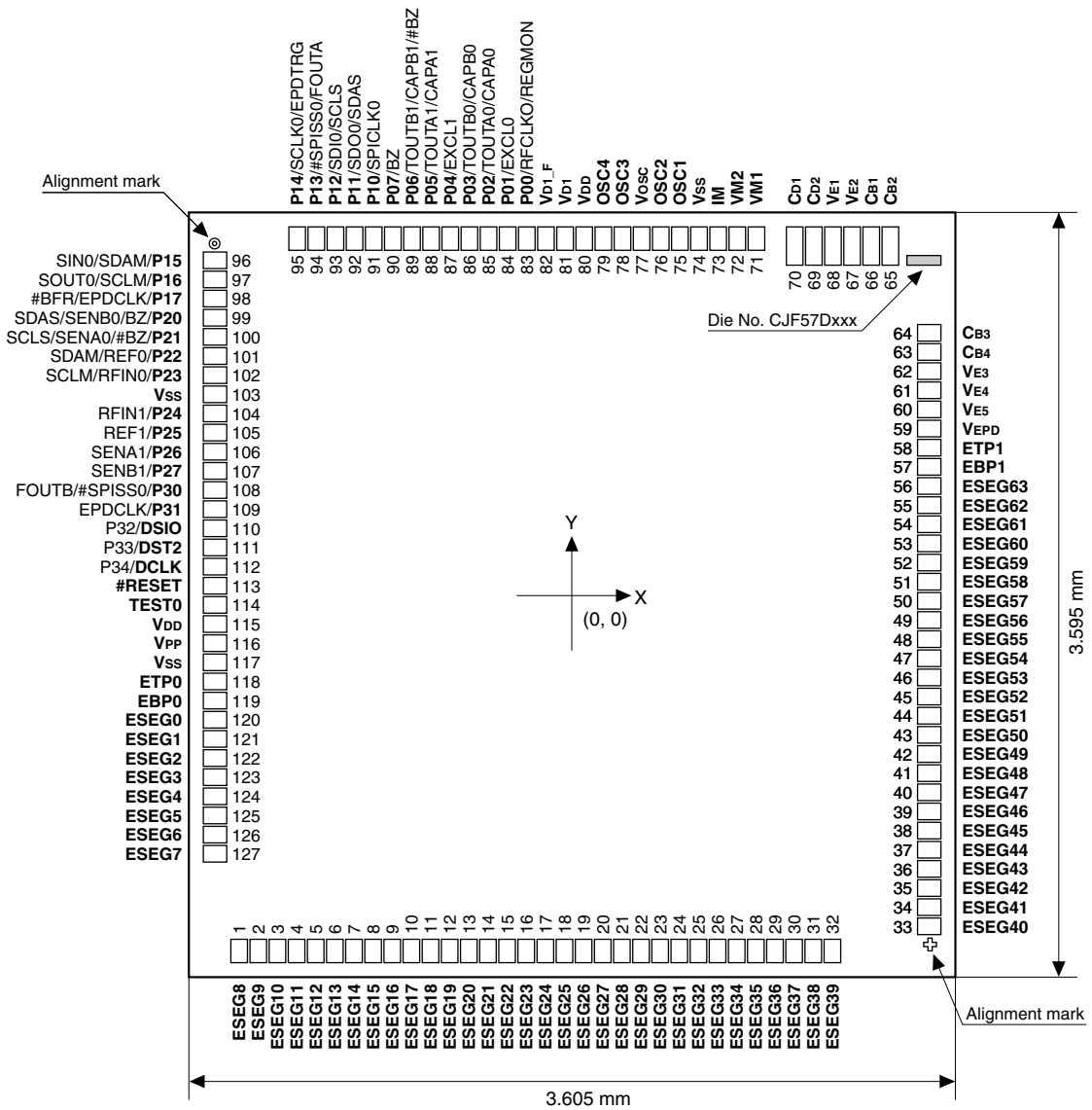


Figure 1.3.1.1 S1C17F57 Pad/Bump Configuration Diagram

1.3.2 Pin Descriptions

Note: The pin names described in boldface type are default settings.

Table 1.3.2.1 Pin Descriptions

Pin No.	Name	I/O	Default status	Function
1–56	**ESEG8–ESEG63**	O	O (Hi-Z)	EPD segment output pins
57	**EBP1**	O	O (Hi-Z)	EPD back plane output pin
58	**ETP1**	O	O (Hi-Z)	EPD top plane output pin
59	**V_{EPD}**	–	–	EPD system power supply circuit output pin
60	**V_{E5}**	–	–	EPD system power supply circuit output pin
61	**V_{E4}**	–	–	EPD system power supply circuit output pin
62	**V_{E3}**	–	–	EPD system power supply circuit output pin
63	**C_{B4}**	–	–	Voltage boost capacitor connecting pin for EPD system power supply circuit
64	**C_{B3}**	–	–	Voltage boost capacitor connecting pin for EPD system power supply circuit
65	**C_{B2}**	–	–	Voltage boost capacitor connecting pin for EPD system power supply circuit
66	**C_{B1}**	–	–	Voltage boost capacitor connecting pin for EPD system power supply circuit
67	**V_{E2}**	–	–	EPD system power supply circuit output pin
68	**V_{E1}**	–	–	EPD system power supply circuit output pin
69	**C_{D2}**	–	–	Voltage boost capacitor connecting pin for EPD system power supply circuit
70	**C_{D1}**	–	–	Voltage boost capacitor connecting pin for EPD system power supply circuit
71	**VM1**	–	–	Test output pin (Leave the pin open during normal operation.)
72	**VM2**	–	–	Test output pin (Leave the pin open during normal operation.)
73	**IM**	–	–	Test output pin (Leave the pin open during normal operation.)
74	**V_{SS}**	–	–	GND pin
75	**OSC1**	I	I	OSC1A oscillation input pin
76	**OSC2**	O	O	OSC1A oscillation output pin
77	**V_{OSC}**	–	–	Oscillation system voltage regulator output pin
78	**OSC3**	I	I	OSC3A oscillation input pin
79	**OSC4**	O	O	OSC3A oscillation output pin
80	**V_{DD}**	–	–	Power supply pin (2.0 to 3.6 V)
81	**V_{D1}**	–	–	Internal logic system voltage regulator output pin
82	**V_{D1_F}**	–	–	Flash system voltage regulator output pin
83	**P00**	I/O	I (Pull-up)	I/O port pin (with port input interrupt function)
	RFCLKO	O		R/F clock monitor output pin
	REGMON	O		Theoretical regulation clock monitor output pin
84	**P01**	I/O	I (Pull-up)	I/O port pin (with port input interrupt function)
	EXCL0	I		T16A2 Ch.0 external clock input pin
85	**P02**	I/O	I (Pull-up)	I/O port pin (with port input interrupt function)
	TOUTA0	O		T16A2 Ch.0 TOUT A signal output pin
	CAPA0	I		T16A2 Ch.0 capture A trigger signal input pin
86	**P03**	I/O	I (Pull-up)	I/O port pin (with port input interrupt function)
	TOUTB0	O		T16A2 Ch.0 TOUT B signal output pin
	CAPB0	I		T16A2 Ch.0 capture B trigger signal input pin
87	**P04**	I/O	I (Pull-up)	I/O port pin (with port input interrupt function)
	EXCL1	I		T16A2 Ch.1 external clock input pin
88	**P05**	I/O	I (Pull-up)	I/O port pin (with port input interrupt function)
	TOUTA1	O		T16A2 Ch.1 TOUT A signal output pin
	CAPA1	I		T16A2 Ch.1 capture A trigger signal input pin
89	**P06**	I/O	I (Pull-up)	I/O port pin (with port input interrupt function)
	TOUTB1	O		T16A2 Ch.1 TOUT B signal output pin
	CAPB1	I		T16A2 Ch.1 capture B trigger signal input pin
	#BZ	O		Buzzer inverted output pin
90	**P07**	I/O	I (Pull-up)	I/O port pin (with port input interrupt function)
	BZ	O		Buzzer output pin
91	**P10**	I/O	I (Pull-up)	I/O port pin
	SPICLK0	I/O		SPI Ch.0 clock input/output pin
92	**P11**	I/O	I (Pull-up)	I/O port pin
	SDO0	O		SPI Ch.0 data output pin
	SDAS	I/O		I^2C slave data input/output pin
93	**P12**	I/O	I (Pull-up)	I/O port pin
	SDI0	I		SPI Ch.0 data input pin
	SCLS	I/O		I^2C slave SCL input/output pin

Pin No.	Name	I/O	Default status	Function
94	**P13**	I/O	I (Pull-up)	I/O port pin
	#SPISS0	I		SPI Ch.0 slave select signal input pin
	FOUTA	O		Clock output pin
95	**P14**	I/O	I (Pull-up)	I/O port pin
	SCLK0	I		UART Ch.0 external clock input pin
	EPDTRG	O		EPD trigger signal output pin
96	**P15**	I/O	I (Pull-up)	I/O port pin
	SIN0	I		UART Ch.0 data input pin
	SDAM	I/O		I²C master data input/output pin
97	**P16**	I/O	I (Pull-up)	I/O port pin
	SOUT0	O		UART Ch.0 data output pin
	SCLM	I/O		I²C master SCL input/output pin
98	**P17**	I/O	I (Pull-up)	I/O port pin
	#BFR	I		I²C slave bus free request input pin
	EPDCLK	O		EPD clock output pin
99	**P20**	I/O	I (Pull-up)	I/O port pin (with port input interrupt function)
	SDAS	I/O		I²C slave data input/output pin
	SENB0	I/O		R/F converter Ch.0 sensor B oscillation control pin
	BZ	O		Buzzer output pin
100	**P21**	I/O	I (Pull-up)	I/O port pin (with port input interrupt function)
	SCLS	I/O		I²C slave SCL input/output pin
	SENA0	I/O		R/F converter Ch.0 sensor A oscillation control pin
	#BZ	O		Buzzer inverted output pin
101	**P22**	I/O	I (Pull-up)	I/O port pin (with port input interrupt function)
	SDAM	I/O		I²C master data input/output pin
	REF0	I/O		R/F converter Ch.0 reference oscillation control pin
102	**P23**	I/O	I (Pull-up)	I/O port pin (with port input interrupt function)
	SCLM	I/O		I²C master SCL input/output pin
	RFIN0	I/O		R/F converter Ch.0 RFCLK input and oscillation control pin
103	**VSS**	–	–	GND pin
104	**P24**	I/O	I (Pull-up)	I/O port pin (with port input interrupt function)
	RFIN1	I/O		R/F converter Ch.1 RFCLK input and oscillation control pin
105	**P25**	I/O	I (Pull-up)	I/O port pin (with port input interrupt function)
	REF1	I/O		R/F converter Ch.1 reference oscillation control pin
106	**P26**	I/O	I (Pull-up)	I/O port pin (with port input interrupt function)
	SENA1	I/O		R/F converter Ch.1 sensor A oscillation control pin
107	**P27**	I/O	I (Pull-up)	I/O port pin (with port input interrupt function)
	SENB1	I/O		R/F converter Ch.1 sensor B oscillation control pin
108	**P30**	I/O	I (Pull-up)	I/O port pin
	FOUTB	O		Clock output pin
	#SPISS0	I		SPI Ch.0 slave select signal input pin
109	**P31**	I/O	I (Pull-up)	I/O port pin
	EPDCLK	O		EPD clock output pin
110	**DSIO**	I/O	I (Pull-up)	On-chip debugger data input/output pin
	P32	I/O		I/O port pin
111	**DST2**	O	O (L)	On-chip debugger status output pin
	P33	I/O		I/O port pin
112	**DCLK**	O	O (H)	On-chip debugger clock output pin
	P34	I/O		I/O port pin
113	**#RESET**	I	I (Pull-up)	Initial reset input pin
114	**TEST0**	I	I (Pull-down)	Test input pin (Connect to VSS for normal operation.)
115	**VDD**	–	–	Power supply pin (2.0 to 3.6 V)
116	**VPP**	–	–	Flash programming/erasing power supply pin (7/7.5 V) (Leave the pin open during normal operation.)
117	**VSS**	–	–	GND pin
118	**ETP0**	O	O (Hi-Z)	EPD top plane output pin
119	**EBP0**	O	O (Hi-Z)	EPD back plane output pin
120–127	**ESEG0–ESEG7**	O	O (Hi-Z)	EPD segment output pins

2 CPU

The S1C17F57 contains the S1C17 Core as its core processor.
The S1C17 Core is a Seiko Epson original 16-bit RISC-type processor.
It features low power consumption, high-speed operation, large address space, main instructions executable in one clock cycle, and a small sized design. The S1C17 Core is suitable for embedded applications such as controllers and sequencers for which an eight-bit CPU is commonly used.
For details of the S1C17 Core, refer to the "S1C17 Family S1C17 Core Manual."

2.1 Features of the S1C17 Core

Processor type

- Seiko Epson original 16-bit RISC processor
- 0.35–0.15 μm low power CMOS process technology

Instruction set

- Code length: 16-bit fixed length
- Number of instructions: 111 basic instructions (184 including variations)
- Execution cycle: Main instructions executed in one cycle
- Extended immediate instructions: Immediate extended up to 24 bits
- Compact and fast instruction set optimized for development in C language

Register set

- Eight 24-bit general-purpose registers
- Two 24-bit special registers
- One 8-bit special register

Memory space and bus

- Up to 16M bytes of memory space (24-bit address)
- Harvard architecture using separated instruction bus (16 bits) and data bus (32 bits)

Interrupts

- Reset, NMI, and 32 external interrupts supported
- Address misaligned interrupt
- Debug interrupt
- Direct branching from vector table to interrupt handler routine
- Programmable software interrupts with a vector number specified (all vector numbers specifiable)

Power saving

- HALT (`halt` instruction)
- SLEEP (`slp` instruction)

Coprocessor interface

- 16-bit × 16-bit multiplier
- 16-bit ÷ 16-bit divider
- 16-bit × 16-bit + 32-bit multiply and accumulation unit

2.2 CPU Registers

The S1C17 Core contains eight general-purpose registers and three special registers.

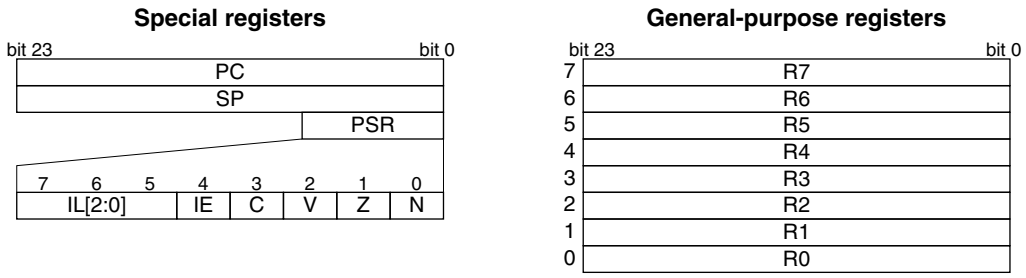


Figure 2.2.1 Registers

2.3 Instruction Set

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

Table 2.3.1 List of S1C17 Core Instructions

Classification	Mnemonic		Function
Data transfer	ld.b	%*rd*,%*rs*	General-purpose register (byte) → general-purpose register (sign-extended)
		%*rd*,[%*rb*]	Memory (byte) → general-purpose register (sign-extended)
		%*rd*,[%*rb*]+	Memory address post-increment, post-decrement, and pre-decrement
		%*rd*,[%*rb*]–	functions can be used.
		%*rd*,–[%*rb*]	
		%*rd*,[%sp+*imm7*]	Stack (byte) → general-purpose register (sign-extended)
		%*rd*,[*imm7*]	Memory (byte) → general-purpose register (sign-extended)
		[%*rb*],%*rs*	General-purpose register (byte) → memory
		[%*rb*]+,%*rs*	Memory address post-increment, post-decrement, and pre-decrement
		[%*rb*]–,%*rs*	functions can be used.
		–[%*rb*],%*rs*	
		[%sp+*imm7*],%*rs*	General-purpose register (byte) → stack
		[*imm7*],%*rs*	General-purpose register (byte) → memory
	ld.ub	%*rd*,%*rs*	General-purpose register (byte) → general-purpose register (zero-extended)
		%*rd*,[%*rb*]	Memory (byte) → general-purpose register (zero-extended)
		%*rd*,[%*rb*]+	Memory address post-increment, post-decrement, and pre-decrement
		%*rd*,[%*rb*]–	functions can be used.
		%*rd*,–[%*rb*]	
		%*rd*,[%sp+*imm7*]	Stack (byte) → general-purpose register (zero-extended)
		%*rd*,[*imm7*]	Memory (byte) → general-purpose register (zero-extended)
	ld	%*rd*,%*rs*	General-purpose register (16 bits) → general-purpose register
		%*rd*,*sign7*	Immediate → general-purpose register (sign-extended)
		%*rd*,[%*rb*]	Memory (16 bits) → general-purpose register
		%*rd*,[%*rb*]+	Memory address post-increment, post-decrement, and pre-decrement
		%*rd*,[%*rb*]–	functions can be used.
		%*rd*,–[%*rb*]	
		%*rd*,[%sp+*imm7*]	Stack (16 bits) → general-purpose register
		%*rd*,[*imm7*]	Memory (16 bits) → general-purpose register
		[%*rb*],%*rs*	General-purpose register (16 bits) → memory
		[%*rb*]+,%*rs*	Memory address post-increment, post-decrement, and pre-decrement
		[%*rb*]–,%*rs*	functions can be used.
		–[%*rb*],%*rs*	
		[%sp+*imm7*],%*rs*	General-purpose register (16 bits) → stack
		[*imm7*],%*rs*	General-purpose register (16 bits) → memory
	ld.a	%*rd*,%*rs*	General-purpose register (24 bits) → general-purpose register
		%*rd*,*imm7*	Immediate → general-purpose register (zero-extended)

Classification	Mnemonic		Function
Data transfer	ld.a	%*rd*,[%*rb*]	Memory (32 bits) → general-purpose register (∗1)
		%*rd*,[%*rb*]+	Memory address post-increment, post-decrement, and pre-decrement functions can be used.
		%*rd*,[%*rb*]-	
		%*rd*,-[%*rb*]	
		%*rd*,[%sp+*imm7*]	Stack (32 bits) → general-purpose register (∗1)
		%*rd*,[*imm7*]	Memory (32 bits) → general-purpose register (∗1)
		[%*rb*],%*rs*	General-purpose register (32 bits, zero-extended) → memory (∗1)
		[%*rb*]+,%*rs*	Memory address post-increment, post-decrement, and pre-decrement functions can be used.
		[%*rb*]-,%*rs*	
		-[%*rb*],%*rs*	
		[%sp+*imm7*],%*rs*	General-purpose register (32 bits, zero-extended) → stack (∗1)
		[*imm7*],%*rs*	General-purpose register (32 bits, zero-extended) → memory (∗1)
		%*rd*,%sp	SP → general-purpose register
		%*rd*,%pc	PC → general-purpose register
		%*rd*,[%sp]	Stack (32 bits) → general-purpose register (∗1)
		%*rd*,[%sp]+	Stack pointer post-increment, post-decrement, and pre-decrement functions can be used.
		%*rd*,[%sp]-	
		%*rd*,-[%sp]	
		[%sp],%*rs*	General-purpose register (32 bits, zero-extended) → stack (∗1)
		[%sp]+,%*rs*	Stack pointer post-increment, post-decrement, and pre-decrement functions can be used.
		[%sp]-,%*rs*	
		-[%sp],%*rs*	
		%sp,%*rs*	General-purpose register (24 bits) → SP
		%sp,*imm7*	Immediate → SP
Integer arithmetic operation	add	%*rd*,%*rs*	16-bit addition between general-purpose registers Supports conditional execution (/c: executed if C = 1, /nc: executed if C = 0).
	add/c		
	add/nc		
	add	%*rd*,*imm7*	16-bit addition of general-purpose register and immediate
	add.a	%*rd*,%*rs*	24-bit addition between general-purpose registers Supports conditional execution (/c: executed if C = 1, /nc: executed if C = 0).
	add.a/c		
	add.a/nc		
	add.a	%sp,%*rs*	24-bit addition of SP and general-purpose register
		%*rd*,*imm7*	24-bit addition of general-purpose register and immediate
		%sp,*imm7*	24-bit addition of SP and immediate
	adc	%*rd*,%*rs*	16-bit addition with carry between general-purpose registers Supports conditional execution (/c: executed if C = 1, /nc: executed if C = 0).
	adc/c		
	adc/nc		
	adc	%*rd*,*imm7*	16-bit addition of general-purpose register and immediate with carry
	sub	%*rd*,%*rs*	16-bit subtraction between general-purpose registers Supports conditional execution (/c: executed if C = 1, /nc: executed if C = 0).
	sub/c		
	sub/nc		
	sub	%*rd*,*imm7*	16-bit subtraction of general-purpose register and immediate
	sub.a	%*rd*,%*rs*	24-bit subtraction between general-purpose registers Supports conditional execution (/c: executed if C = 1, /nc: executed if C = 0).
	sub.a/c		
	sub.a/nc		
	sub.a	%sp,%*rs*	24-bit subtraction of SP and general-purpose register
		%*rd*,*imm7*	24-bit subtraction of general-purpose register and immediate
		%sp,*imm7*	24-bit subtraction of SP and immediate
	sbc	%*rd*,%*rs*	16-bit subtraction with carry between general-purpose registers Supports conditional execution (/c: executed if C = 1, /nc: executed if C = 0).
	sbc/c		
	sbc/nc		
	sbc	%*rd*,*imm7*	16-bit subtraction of general-purpose register and immediate with carry
	cmp	%*rd*,%*rs*	16-bit comparison between general-purpose registers Supports conditional execution (/c: executed if C = 1, /nc: executed if C = 0).
	cmp/c		
	cmp/nc		
	cmp	%*rd*,*sign7*	16-bit comparison of general-purpose register and immediate
	cmp.a	%*rd*,%*rs*	24-bit comparison between general-purpose registers Supports conditional execution (/c: executed if C = 1, /nc: executed if C = 0).
	cmp.a/c		
	cmp.a/nc		
	cmp.a	%*rd*,*imm7*	24-bit comparison of general-purpose register and immediate
	cmc	%*rd*,%*rs*	16-bit comparison with carry between general-purpose registers Supports conditional execution (/c: executed if C = 1, /nc: executed if C = 0).
	cmc/c		
	cmc/nc		
	cmc	%*rd*,*sign7*	16-bit comparison of general-purpose register and immediate with carry

Classification	Mnemonic		Function
Logical operation	and	%rd,%rs	Logical AND between general-purpose registers
	and/c		Supports conditional execution (/c: executed if C = 1, /nc: executed if C = 0).
	and/nc		
	and	%rd,sign7	Logical AND of general-purpose register and immediate
	or	%rd,%rs	Logical OR between general-purpose registers
	or/c		Supports conditional execution (/c: executed if C = 1, /nc: executed if C = 0).
	or/nc		
	or	%rd,sign7	Logical OR of general-purpose register and immediate
	xor	%rd,%rs	Exclusive OR between general-purpose registers
	xor/c		Supports conditional execution (/c: executed if C = 1, /nc: executed if C = 0).
	xor/nc		
	xor	%rd,sign7	Exclusive OR of general-purpose register and immediate
	not	%rd,%rs	Logical inversion between general-purpose registers (1's complement)
	not/c		Supports conditional execution (/c: executed if C = 1, /nc: executed if C = 0).
	not/nc		
	not	%rd,sign7	Logical inversion of general-purpose register and immediate (1's complement)
Shift and swap	sr	%rd,%rs	Logical shift to the right with the number of bits specified by the register
		%rd,imm7	Logical shift to the right with the number of bits specified by immediate
	sa	%rd,%rs	Arithmetic shift to the right with the number of bits specified by the register
		%rd,imm7	Arithmetic shift to the right with the number of bits specified by immediate
	sl	%rd,%rs	Logical shift to the left with the number of bits specified by the register
		%rd,imm7	Logical shift to the left with the number of bits specified by immediate
	swap	%rd,%rs	Bytewise swap on byte boundary in 16 bits
Immediate extension	ext	imm13	Extend operand in the following instruction
Conversion	cv.ab	%rd,%rs	Converts signed 8-bit data into 24 bits
	cv.as	%rd,%rs	Converts signed 16-bit data into 24 bits
	cv.al	%rd,%rs	Converts 32-bit data into 24 bits
	cv.la	%rd,%rs	Converts 24-bit data into 32 bits
	cv.ls	%rd,%rs	Converts 16-bit data into 32 bits
Branch	jpr	sign10	PC relative jump
	jpr.d	%rb	Delayed branching possible
	jpa	imm7	Absolute jump
	jpa.d	%rb	Delayed branching possible
	jrgt	sign7	PC relative conditional jump Branch condition: !Z & !(N ^ V)
	jrgt.d		Delayed branching possible
	jrge	sign7	PC relative conditional jump Branch condition: !(N ^ V)
	jrge.d		Delayed branching possible
	jrlt	sign7	PC relative conditional jump Branch condition: N ^ V
	jrlt.d		Delayed branching possible
	jrle	sign7	PC relative conditional jump Branch condition: Z | N ^ V
	jrle.d		Delayed branching possible
	jrugt	sign7	PC relative conditional jump Branch condition: !Z & !C
	jrugt.d		Delayed branching possible
	jruge	sign7	PC relative conditional jump Branch condition: !C
	jruge.d		Delayed branching possible
	jrult	sign7	PC relative conditional jump Branch condition: C
	jrult.d		Delayed branching possible
	jrule	sign7	PC relative conditional jump Branch condition: Z | C
	jrule.d		Delayed branching possible
	jreq	sign7	PC relative conditional jump Branch condition: Z
	jreq.d		Delayed branching possible
	jrne	sign7	PC relative conditional jump Branch condition: !Z
	jrne.d		Delayed branching possible
	call	sign10	PC relative subroutine call
	call.d	%rb	Delayed call possible
	calla	imm7	Absolute subroutine call
	calla.d	%rb	Delayed call possible
	ret		Return from subroutine
	ret.d		Delayed return possible
	int	imm5	Software interrupt
	intl	imm5,imm3	Software interrupt with interrupt level setting
	reti		Return from interrupt handling
	reti.d		Delayed call possible
	brk		Debug interrupt

Classification	Mnemonic		Function
Branch	retd		Return from debug processing
System control	nop		No operation
	halt		HALT mode
	slp		SLEEP mode
	ei		Enable interrupts
	di		Disable interrupts
Coprocessor control	ld.cw	%rd,%rs	Transfer data to coprocessor
		%rd,imm7	
	ld.ca	%rd,%rs	Transfer data to coprocessor and get results and flag statuses
		%rd,imm7	
	ld.cf	%rd,%rs	Transfer data to coprocessor and get flag statuses
		%rd,imm7	

*1 The ld.a instruction accesses memories in 32-bit length. During data transfer from a register to a memory, the 32-bit data in which the eight high-order bits are set to 0 is written to the memory. During reading from a memory, the eight high-order bits of the read data are ignored.

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

Table 2.3.2 Symbol Meanings

Symbol	Description
%rs	General-purpose register, source
%rd	General-purpose register, destination
[%rb]	Memory addressed by general-purpose register
[%rb]+	Memory addressed by general-purpose register with address post-incremented
[%rb]-	Memory addressed by general-purpose register with address post-decremented
-[%rb]	Memory addressed by general-purpose register with address pre-decremented
%sp	Stack pointer
[%sp],[%sp+imm7]	Stack
[%sp]+	Stack with address post-incremented
[%sp]-	Stack with address post-decremented
-[%sp]	Stack with address pre-decremented
imm3,imm5,imm7,imm13	Unsigned immediate (numerals indicating bit length)
sign7,sign10	Signed immediate (numerals indicating bit length)

2.4 Reading PSR

The S1C17F57 includes the MISC_PSR register for reading the contents of the PSR (Processor Status Register) in the S1C17 Core. Reading the contents of this register makes it possible to check the contents of the PSR using the application software. Note that data cannot be written to the PSR.

PSR Register (MISC_PSR)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
PSR Register (MISC_PSR)	0x532c (16 bits)	D15–8	–	reserved	–				–	–	0 when being read.
		D7–5	PSRIL[2:0]	PSR interrupt level (IL) bits	0x0 to 0x7				0x0	R	
		D4	PSRIE	PSR interrupt enable (IE) bit	1	1 (enable)	0	0 (disable)	0	R	
		D3	PSRC	PSR carry (C) flag	1	1 (set)	0	0 (cleared)	0	R	
		D2	PSRV	PSR overflow (V) flag	1	1 (set)	0	0 (cleared)	0	R	
		D1	PSRZ	PSR zero (Z) flag	1	1 (set)	0	0 (cleared)	0	R	
		D0	PSRN	PSR negative (N) flag	1	1 (set)	0	0 (cleared)	0	R	

D[15:8] Reserved

D[7:5] PSRIL[2:0]: PSR Interrupt Level (IL) Bits

The value of the PSR IL (interrupt level) bits can be read out. (Default: 0x0)

D4 PSRIE: PSR Interrupt Enable (IE) Bit

The value of the PSR IE (interrupt enable) bit can be read out.

1 (R): 1 (interrupt enabled)

0 (R): 0 (interrupt disabled) (default)

D3 **PSRC: PSR Carry (C) Flag Bit**

The value of the PSR C (carry) flag can be read out.

1 (R): 1

0 (R): 0 (default)

D2 **PSRV: PSR Overflow (V) Flag Bit**

The value of the PSR V (overflow) flag can be read out.

1 (R): 1

0 (R): 0 (default)

D1 **PSRZ: PSR Zero (Z) Flag Bit**

The value of the PSR Z (zero) flag can be read out.

1 (R): 1

0 (R): 0 (default)

D0 **PSRN: PSR Negative (N) Flag Bit**

The value of the PSR N (negative) flag can be read out.

1 (R): 1

0 (R): 0 (default)

2.5 Processor Information

The S1C17F57 has the IDIR register shown below that allows the application software to identify CPU core type.

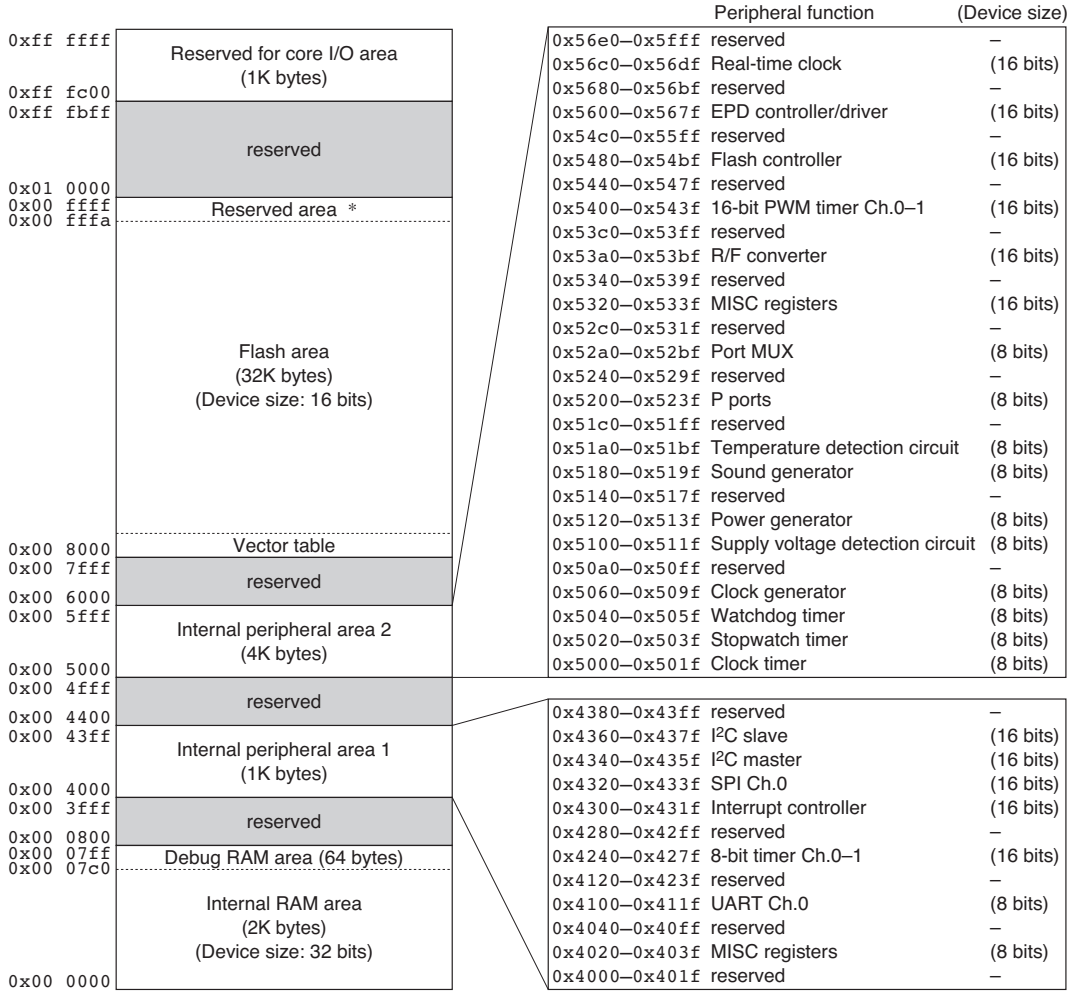
Processor ID Register (IDIR)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Processor ID Register (IDIR)	**0xffff84** (8 bits)	D7–0	**IDIR[7:0]**	Processor ID 0x10: S1C17 Core	0x10	0x10	R	

This is a read-only register that contains the ID code to represent a processor model. The S1C17 Core's ID code is 0x10.

3 Memory Map, Bus Control

Figure 3.1 shows the S1C17F57 memory map.



* 0xfffa–0xfffb: Reserved for the theoretical regulation function (See the "Setting Regulation Values" section in the "Theoretical Regulation" chapter.)
0xfffc–0xffff: Flash protect bit area (See the "Protect Bits" section in this chapter.)

Figure 3.1 S1C17F57 Memory Map

3.1 Bus Cycle

The CPU uses the system clock for bus access operations. For more information on the system clock, see "System Clock Switching" in the "Clock Generator (CLG)" chapter. Accessing in one bus cycle requires one system clock in all areas. Furthermore, the number of bus accesses depends on the CPU instruction (access size) and device size.

Table 3.1.1 Number of Bus Accesses

Device size	CPU access size	Number of bus accesses
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

* Handling the eight high-order bits during 32-bit accesses
  The size of the S1C17 Core general-purpose registers is 24 bits.
  During writing, the eight high-order bits are written as 0. During reading from a memory, the eight high-order bits are ignored. However, the stack operation in an interrupt handling reads/writes 32-bit data that consists of the PSR value as the high-order 8 bits and the return address as the low order 24 bits.
  For more information, refer to the "S1C17 Core Manual."

3.1.1 Restrictions on Access Size

The peripheral modules can be accessed with an 8-bit, 16-bit, or 32-bit instruction. However, reading for an unnecessary register may change the peripheral module status and it may cause a problem. Therefore, use the appropriate instructions according to the device size.

3.1.2 Restrictions on Instruction Execution Cycles

An instruction fetch and a data access are not performed simultaneously under one of the conditions listed below. This prolongs the instruction fetch cycle for the number of data area bus cycles.

- When the S1C17F57 executes the instruction stored in the Flash area and accesses data in the Flash area
- When the S1C17F57 executes the instruction stored in the internal RAM area and accesses data in the internal RAM area

3.2 Flash Area

3.2.1 Embedded Flash Memory

The 32K-byte area from address 0x8000 to address 0xffff contains a Flash memory (4K bytes × 8 sectors) for storing application programs and data. Address 0x8000 is defined as the vector table base address, therefore a vector table (see "Vector Table" in the "Interrupt Controller (ITC)" chapter) must be placed from the beginning of the area. The vector table base address can be modified with the MISC_TTBRL/MISC_TTBRH registers.

3.2.2 Flash Programming

The S1C17F57 supports on-board programming of the Flash memory, it makes it possible to program the Flash memory with the application programs/data by using the debugger through an ICDmini.

3.2.3 Protect Bits

In order to protect the memory contents, the Flash memory provides two protection features, write protection and data read protection, that can be configured for every 4K-byte areas. The write protection disables writing data to the configured area and erasing the sectors (except the sector that includes the protect bits). The data-read protection disables reading data from the configured area (the read value is always 0x0000). However, it does not disable the instruction fetch operation by the CPU. The Flash memory provides the protect bits listed below. Program the protect bit corresponding to the area to be protected to 0. The protection can only be disabled using the debugger.

Flash Protect Bits

Address	Bit	Function	Setting				Init.	R/W	Remarks
0xfffc (16 bits)	D15–8	reserved	–				–	–	
	D7	Flash write-protect bit for 0xf000–0xffff	1	Writable	0	Protected	1	R/W	
	D6	Flash write-protect bit for 0xe000–0xefff	1	Writable	0	Protected	1	R/W	
	D5	Flash write-protect bit for 0xd000–0xdfff	1	Writable	0	Protected	1	R/W	
	D4	Flash write-protect bit for 0xc000–0xcfff	1	Writable	0	Protected	1	R/W	
	D3	Flash write-protect bit for 0xb000–0xbfff	1	Writable	0	Protected	1	R/W	
	D2	Flash write-protect bit for 0xa000–0xafff	1	Writable	0	Protected	1	R/W	
	D1	Flash write-protect bit for 0x9000–0x9fff	1	Writable	0	Protected	1	R/W	
	D0	Flash write-protect bit for 0x8000–0x8fff	1	Writable	0	Protected	1	R/W	
0xfffe (16 bits)	D15–8	reserved	–				–	–	
	D7	Flash data-read-protect bit for 0xf000–0xffff	1	Readable	0	Protected	1	R/W	
	D6	Flash data-read-protect bit for 0xe000–0xefff	1	Readable	0	Protected	1	R/W	
	D5	Flash data-read-protect bit for 0xd000–0xdfff	1	Readable	0	Protected	1	R/W	
	D4	Flash data-read-protect bit for 0xc000–0xcfff	1	Readable	0	Protected	1	R/W	
	D3	Flash data-read-protect bit for 0xb000–0xbfff	1	Readable	0	Protected	1	R/W	
	D2	Flash data-read-protect bit for 0xa000–0xafff	1	Readable	0	Protected	1	R/W	
	D1	Flash data-read-protect bit for 0x9000–0x9fff	1	Readable	0	Protected	1	R/W	
	D0	reserved	1				1	R/W	Always set to 1.

Notes: • Be sure not to locate the area with data-read protection into the .data and .rodata sections.

• Be sure to set D0 of address 0xfffe to 1. If it is set to 0, the program cannot be booted.

3.2.4 Flash Memory Read Wait Cycle Setting

In order to read data from the Flash memory properly, set the appropriate number of wait cycles according to the system clock frequency using the RDWAIT[1:0]/FLASHC_WAIT register.

FLASHC Read Wait Control Register (FLASHC_WAIT)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
FLASHC Read Wait Control Register (FLASHC_ WAIT)	**0x54b0** (16 bits)	D15–8	**–**	reserved	–		–	–	0 when being read.
		D7	**–**	reserved	–		X	–	X when being read.
		D6-2	**–**	reserved	–		–	–	0 when being read.
		D1–0	**RDWAIT [1:0]**	Flash read wait cycle	RDWAIT[1:0]	Wait	0x3	R/W	
					0x3	3 wait			
					0x2	2 wait			
					0x1	1 wait			
					0x0	No wait			

D[1:0] RDWAIT[1:0]: Flash Read Wait Cycle Bits

Sets the number of wait cycles for reading from the Flash memory. One wait insertion prolongs bus cycles by one system clock cycle.

Note: Set RDWAIT[1:0] to 0x0 to achieve the best performance.

3.3 Internal RAM Area

3.3.1 Embedded RAM

The S1C17F57 contains a RAM in the 2K-byte area from address 0x0 to address 0x7ff. The RAM allows high-speed execution of the instruction codes copied into it as well as storing variables and other data.

Note: The 64-byte area at the end of the RAM (0x7c0–0x7ff) is reserved for the on-chip debugger. When using the debug functions under application development, do not access this area from the application program.
This area can be used for applications of mass-produced devices that do not need debugging.

The S1C17F57 enables the RAM size used to apply restrictions to 2KB, 1KB, or 512B. For example, when using the S1C17F57 to develop an application for a built-in ROM model, you can set the RAM size to match that of the target model, preventing creating programs that seek to access areas outside the RAM areas of the target product. The RAM size is selected using IRAMSZ[2:0]/MISC_IRAMSZ register.

IRAM Size Register (MISC_IRAMSZ)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
IRAM Size Register (MISC_IRAMSZ)	0x5326 (16 bits)	D15–9	–	reserved	–	–	–	0 when being read.
		D8	DBADR	Debug base address select	1 0x0 / 0 0xfffc00	0	R/W	
		D7	–	reserved	–	–	–	0 when being read.
		D6–4	IRAMACTSZ[2:0]	IRAM actual size	0x3 (= 2KB)	0x3	R	
		D3	–	reserved	–	–	–	0 when being read.
		D2–0	IRAMSZ[2:0]	IRAM size select	IRAMSZ[2:0] / Size: 0x5 512B; 0x4 1KB; 0x3 2KB; Other reserved	0x3	R/W	

D[6:4] IRAMACTSZ[2:0]: IRAM Actual Size Bits

Indicates the actual internal RAM size embedded. (Default: 0x3)

D[2:0] IRAMSZ[2:0]: IRAM Size Select Bits

Selects the internal RAM size used.

Table 3.3.1.1 Selecting Internal RAM Size

IRAMSZ[2:0]	Internal RAM size
0x5	512B
0x4	1KB
0x3	2KB
Other	Reserved

(Default: 0x3)

Note: The MISC_IRAMSZ register is write-protected. The write-protection must be overridden by writing 0x96 to the MISC_PROT register. Note that the MISC_PROT register should normally be set to a value other than 0x96, except when writing to the MISC_IRAMSZ register. Unnecessary programs may result in system malfunctions.

3.4 Internal Peripheral Area

The I/O and control registers for the internal peripheral modules are located in the 1K-byte area beginning with address 0x4000 and the 4K-byte area beginning with address 0x5000.
For details of each control register, see the I/O register list in Appendix or description for each peripheral module.

3.4.1 Internal Peripheral Area 1 (0x4000–)

The internal peripheral area 1 beginning with address 0x4000 contains the I/O memory for the peripheral functions listed below.

- MISC register (MISC, 8-bit device)
- UART (UART, 8-bit device)
- 8-bit timers (T8, 16-bit device)
- Interrupt controller (ITC, 16-bit device)
- SPI (SPI, 16-bit device)
- I^2C master (I2CM, 16-bit device)
- I^2C slave (I2CS, 16-bit device)

3.4.2 Internal Peripheral Area 2 (0x5000–)

The internal peripheral area 2 beginning with address 0x5000 contains the I/O memory for the peripheral functions listed below.

- Clock timer (CT, 8-bit device)
- Stopwatch timer (SWT, 8-bit device)
- Watchdog timer (WDT, 8-bit device)
- Clock generator (CLG, 8-bit device)

- Supply voltage detection circuit (SVD, 8-bit device)
- Power generator (VD1, 8-bit device)
- Sound generator (SND, 8-bit device)
- Temperature detection circuit (TEM, 8-bit device)
- I/O port & port MUX (P, 8-bit device)
- MISC register (MISC, 16-bit device)
- R/F converter (RFC, 16-bit device)
- 16-bit PWM timers (T16A2, 16-bit device)
- Flash controller (FLASHC, 16-bit device)
- EPD controller/driver (EPD, 16-bit device)
- Real-time clock (RTC, 16-bit device)

3.5 S1C17 Core I/O Area

The 1K-byte area from address 0xfffc00 to address 0xffffff is the I/O area for the CPU core in which the I/O registers listed in the table below are located.

Table 3.5.1 I/O Map (S1C17 Core I/O Area)

Peripheral	Address	Register name		Function
S1C17 Core I/O	0xffff84	IDIR	Processor ID Register	Indicates the processor ID.
	0xffff90	DBRAM	Debug RAM Base Register	Indicates the debug RAM base address.
	0xffffa0	DCR	Debug Control Register	Debug control
	0xffffb4	IBAR1	Instruction Break Address Register 1	Instruction break address #1 setting
	0xffffb8	IBAR2	Instruction Break Address Register 2	Instruction break address #2 setting
	0xffffbc	IBAR3	Instruction Break Address Register 3	Instruction break address #3 setting
	0xffffd0	IBAR4	Instruction Break Address Register 4	Instruction break address #4 setting

See “Processor Information” in the “CPU” chapter for more information on IDIR. See the “On-chip Debugger (DBG)” chapter for more information on other registers.
This area includes the S1C17 Core registers, in addition to those described above. For more information on these registers, refer to the “S1C17 Core Manual.”

4 Power Supply

4.1 Power Supply Voltage (V_{DD})

The S1C17F57 operates with a voltage supplied between the V_{DD} and V_{SS} pins. Supply a voltage within the range shown below to the V_{DD} pins with the V_{SS} pins as the GND level.

V_{DD} = 2.0 V to 3.6 V (V_{SS} = GND)

The S1C17F57 provides two or more V_{DD} and V_{SS} pins. Do not leave any power supply pins open and be sure to connect them to + power source and GND.

4.2 Flash Programming Power Supply Voltage (V_{PP})

The V_{PP} voltage is used for programming/erasing the embedded Flash memory. Supply a voltage shown below to the V_{PP} pin with the V_{SS} pins as the GND level to program the Flash memory.

V_{PP} = 7 V (V_{SS} = GND) for programming
V_{PP} = 7.5 V (V_{SS} = GND) for erasing

Note: Leave the V_{PP} pin open during normal operation.

4.3 Internal Power Supply Circuit

The S1C17F57 has a built-in power supply circuit shown in Figure 4.3.1 to generate the operating voltages required for the internal circuits.

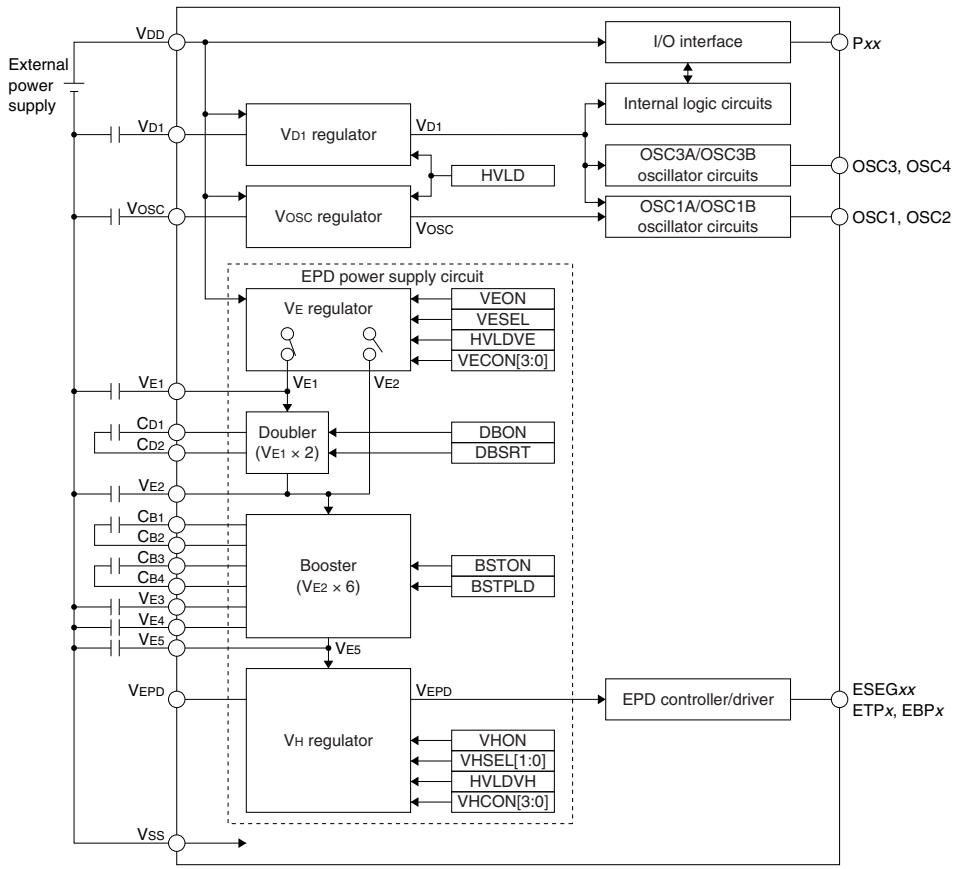


Figure 4.3.1 Configuration of Internal Power Supply Circuit

The internal power supply circuit consists of a V_{D1} regulator, a V_{OSC} regulator, and an EPD power supply circuit.

Note: Be sure to avoid using the outputs of the internal power supply circuit for drive external devices.

4.3.1 V_{D1} and V_{OSC} Regulators

The V_{D1} and V_{OSC} regulators generate the operating voltages for the internal logic and oscillator circuits. This regulator always operates.

4.3.2 EPD Power Supply Circuit

The EPD power supply circuit generates the EPD drive voltage V_{EPD}. V_{EPD} is supplied to the EPD driver to generate EPD drive waveforms.
The EPD power supply circuit consists of a V_E regulator, two voltage boosters (doubler and booster), and a V_H regulator that can be controlled via software individually.
See the "Electrical Characteristics" chapter for the EPD drive voltage (V_{EPD}) value.

V_E regulator

The V_E regulator generates the reference voltage for boosting (V_{E1} or V_{E2}) from V_{DD}.
Either V_{E1} or V_{E2} can be generated and one of them should be selected according to the V_{DD} value using VESEL/EPD_PWR0 register.

Table 4.3.2.1 V_E Regulator Output Selection

Power supply voltage V_{DD}	VESEL	V_E regulator output
$V_{DD} \geq V_{E1} + 0.3$ V	0 (default)	V_{E1}
$V_{DD} \geq V_{E2} + 0.2$ V	1	V_{E2}

This selection determines the path to generate V_{EPD} as follows:

1. Reference voltage = V_{E1}
   V_{DD} → [V_E regulator] → V_{E1} (reference voltage) → [Doubler] → V_{E2} (= $2V_{E1}$) → [Booster]
   → V_{E5} (= $6V_{E2}$) → [V_H regulator] → V_{EPD}
2. Reference voltage = V_{E2}
   V_{DD} → [V_E regulator] → V_{E2} (reference voltage) → [Booster] → V_{E5} (= $6V_{E2}$) → [V_H regulator]
   → V_{EPD}

The V_E regulator is deactivated at initial reset and is activated by setting VEON/EPD_PWR0 register to 1.

Doubler

The doubler generates V_{E2} by doubling the V_{E1} generated by the V_E regulator.
The doubler is deactivated at initial reset and is activated by setting DBON/EPD_PWR0 register to 1.
When V_{E2} is generated by the V_E regulator, set DBON to 0 as the doubler is not required.
Between the doubler input and output can be short-circuited so that the doubler circuit cannot affect the V_E regulator output when V_{E2} is generated. When the doubler is not used, short between the input and output by setting DBSRT/EPD_PWR0 register to 1.

Booster

The booster multiplies the V_{E2} input from the V_E regulator or doubler by a factor of six to generate V_{E5}.
The booster is deactivated at initial reset and is activated by setting BSTON/EPD_PWR1 register to 1.
The booster output can be pulled down to V_{SS} to set the EPD drive voltage to an off level. To pull down the output, set BSTPLD/EPD_PWR1 to 1.

V_H regulator

The V_H regulator inputs the V_{E5} generated by the booster and generates the EPD drive voltage V_{EPD}. V_{EPD} is supplied to the EPD driver to generate EPD drive waveforms.
The V_H regulator is deactivated at initial reset and is activated by setting VHON/EPD_PWR1 register to 1.
The V_{EPD} output value can be selected from the three levels shown below using VHSEL[1:0]/EPD_PWR1 register. Select one in accordance with the EPD specification.

Table 4.3.2.2 V_{EPD} Voltage Value

VHSEL[1:0]	V_{EPD} voltage
0x3	Reserved
0x2	9 V type
0x1	12 V type
0x0	15 V type

(Default: 0x0)

Doubler and booster clocks

The doubler and booster use a clock for boosting voltage. When using the doubler and booster, make the settings shown below to supply the clock to them.

Clock source selection

Select the clock source from OSC3B, OSC3A, and OSC1 using EPDDCLKSRC[1:0]/EPD_DCLK register for the doubler and EPDBCLKSRC[1:0]/EPD_BCLK register for booster.

Table 4.3.2.3 Clock Source Selection

EPDDCLKSRC[1:0] EPDBCLKSRC[1:0]	Clock source
0x3	Reserved
0x2	OSC3A
0x1	OSC1
0x0	OSC3B

(Default: 0x0)

Clock division ratio selection

Select the division ratio using EPDDCLKD[2:0]/EPD_DCLK register for the doubler and EPDBCLKD[2:0]/EPD_BCLK register for the booster. Set a clock frequency within the range shown below.
Doubler clock: 8 kHz to 32 kHz Booster clock: 4 kHz to 16 kHz

Table 4.3.2.4 Clock Division Ratio Selection

EPDDCLKD[2:0] EPDBCLKD[2:0]	Division ratio		
	Clock source = OSC3B	Clock source = OSC3A	Clock source = OSC1
0x7 to 0x6	Reserved	Reserved	Reserved
0x5	1/128	1/256	
0x4	1/64	1/128	
0x3	1/32	1/64	1/8
0x2	1/16	1/32	1/4
0x1	Reserved	1/16	1/2
0x0		Reserved	1/1

(Default: 0x0)

Clock enable

The doubler clock supply is enabled with EPDDCLKE/EPD_DCLK register. The booster clock supply is enabled with EPDBCLKE/EPD_BCLK register. The EPDDCLKE and EPDBCLKE default settings are 0, which stop the clock. Setting EPDDCLKE or EPDBCLKE to 1 feeds the clock generated as above to the doubler or booster. If no doubler or booster operation is required, stop the clock to reduce current consumption.

4.3.3 EPD Contrast Adjustment

The V_E regulator and V_H regulator allow software to switch the output voltage in 16 steps for adjusting the EPD contrast. Use VECON[3:0]/EPD_PWR0 register to set the V_E regulator and VHCON[3:0]/EPD_PWR1 register to set the V_H regulator.

Table 4.3.3.1 Setting V_E Regulator Output Level (EPD Contrast Adjustment Function)

VECON[3:0]	V_E regulator output level
0xf	Level 15 (high contrast)
:	:
0x7	Level 7 (default)
:	:
0x0	Level 0 (low contrast)

Table 4.3.3.2 Setting V_H Regulator Output Level (EPD Contrast Adjustment Function)

VHCON[3:0]	V_H regulator output level
0xf	Level 15 (high contrast)
:	:
0x5	Level 5 (default)
:	:
0x0	Level 0 (low contrast)

See the "Electrical Characteristics" chapter for the voltage values.

4.3.4 Heavy Load Protection Mode

In order to ensure a stable circuit behavior and EPD display quality even if the power supply voltage fluctuates due to driving an external load, the regulators have a heavy load protection function.
The table below lists the control bits used for setting heavy load protection mode.

Table 4.3.4.1 Heavy Load Protection Mode Control Bits

Regulator	Control bit
V_{D1} regulator	HVLD/VD1_CTL register
V_{OSC} regulator	
V_E regulator	HVLDVE/EPD_PWR0 register
V_H regulator	HVLDVH/EPD_PWR1 register

When the control bit is set to 1, the regulator ensures stable output.
The V_{D1} and V_{OSC} regulators should be placed into heavy load protection mode before driving a heavy load such as a lamp or buzzer with a port output. The V_E and V_H regulators should be placed into heavy load protection mode when the display has inconsistencies in density.

Note: Current consumption increases in heavy load protection mode, therefore do not set heavy load protection mode with software if unnecessary.

4.3.5 Internal Power Supply Circuit Control Procedures

V_{D1} and V_{OSC} regulators

On/off control is not required. Set them into heavy load protection mode (HVLD/VD1_CTL register) if necessary.

EPD Power Supply Circuit

Control the EPD power supply circuit as the procedure shown below.

1. Set and enable the doubler and booster clocks. (EPD_DCLK and EPD_BCLK registers)
2. Set the following conditions using the EPD_PWR0 and EPD_PWR1 registers:
   - V_E regulator output (V_{E1} or V_{E2}) (VESEL/EPD_PWR0 register)
   - V_{EPD} voltage value (9/12/15 V type) (VHSEL[1:0]/EPD_PWR1 register)
   - Initial contrast setting (VECON[3:0]/EPD_PWR0 register, VHCON[3:0]/EPD_PWR1 register)
3. Turn the V_E regulator on. (VEON/EPD_PWR0 register = 1)
4. When the reference voltage is V_{E1}: Turn the doubler on. (DBON/EPD_PWR0 register = 1)

When the reference voltage is V_{E2}: Turn the doubler off. (DBON/EPD_PWR0 register = 0)
Short between the doubler input and output.
(DBSRT/EPD_PWR0 register = 1)

5. Turn the booster on. (BSTON/EPD_PWR1 register = 1)

Steps 2 to 5 should be performed in the sequence above or simultaneously.

6. Wait at least 35 ms (booster output stabilization wait time).
7. Turn the V_H regulator on. (VHON/EPD_PWR1 register = 1)
8. Wait at least 5 ms (V_{EPD} output stabilization wait time).

After the above procedure has been finished, display on the EPD can be started.
Set the regulators into heavy load protection mode (HVLDVE/EPD_PWR0 register and HVLDVH/EPD_PWR1 register) if necessary.

4.4 Control Register Details

Table 4.4.1 List of Power Control Registers

Address	Register name		Function
0x5071	EPD_DCLK	EPD Doubler Clock Control Register	Controls the EPD doubler clock.
0x5072	EPD_BCLK	EPD Booster Clock Control Register	Controls the EPD booster clock.
0x5120	VD1_CTL	V_{D1} Control Register	Controls the V_{D1} regulator heavy load protection mode.
0x5600	EPD_PWR0	EPD Power Control Register 0	Controls the V_E regulator and doubler.
0x5602	EPD_PWR1	EPD Power Control Register 1	Controls the V_H regulator and booster.

The power control registers are described in detail below.

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

EPD Doubler Clock Control Register (EPD_DCLK)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
EPD Doubler Clock Control Register (EPD_DCLK)	**0x5071** (8 bits)	D7	**–**	reserved	–	–	–	0 when being read.
		D6–4	**EPDDCLKD [2:0]**	EPD doubler clock division ratio select	EPDD CLKD [2:0] / OSC3B / OSC3A / OSC1 0x5 / 1/128 / 1/256 / reserved 0x4 / 1/64 / 1/128 / reserved 0x3 / 1/32 / 1/64 / 1/8 0x2 / 1/16 / 1/32 / 1/4 0x1 / reserved / 1/16 / 1/2 0x0 / reserved / reserved / 1/1 Other / reserved	0x0	R/W	
		D3–2	**EPDDCLK SRC[1:0]**	EPD doubler clock source select	EPDDCLK SRC[1:0] / Clock source 0x3 / reserved 0x2 / OSC3A 0x1 / OSC1 0x0 / OSC3B	0x0	R/W	
		D1	**–**	reserved	–	–	–	0 when being read.
		D0	**EPDDCLKE**	EPD doubler clock enable	1 Enable / 0 Disable	0	R/W	

Note: The doubler clock must be set up when the doubler is used (when V_{E1} is generated by the V_E regulator).

D7 Reserved

D[6:4] EPDDCLKD[2:0]: EPD Doubler Clock Division Ratio Select Bits
Selects the division ratio for generating the doubler clock.

Table 4.4.2 Clock Division Ratio Selection

EPDDCLKD[2:0]	Division ratio		
	Clock source = OSC3B	Clock source = OSC3A	Clock source = OSC1
0x7 to 0x6	Reserved	Reserved	Reserved
0x5	1/128	1/256	
0x4	1/64	1/128	
0x3	1/32	1/64	1/8
0x2	1/16	1/32	1/4
0x1	Reserved	1/16	1/2
0x0		Reserved	1/1

(Default: 0x0)

D[3:2] EPDDCLKSRC[1:0]: EPD Doubler Clock Source Select Bits

Selects the doubler clock source.

Table 4.4.3 Clock Source Selection

EPDDCLKSRC[1:0]	Clock source
0x3	Reserved
0x2	OSC3A
0x1	OSC1
0x0	OSC3B

(Default: 0x0)

D1 Reserved

D0 EPDDCLKE: EPD Doubler Clock Enable Bit

Enables or disables the clock supply to the doubler.

1 (R/W): Enabled (on)

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

The EPDDCLKE default setting is 0, which disables the clock supply. Setting EPDDCLKE to 1 sends the clock selected as above to the doubler. If doubler operation is not required, disable the clock supply to reduce current consumption.

EPD Booster Clock Control Register (EPD_BCLK)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
EPD Booster Clock Control Register (EPD_BCLK)	0x5072 (8 bits)	D7	–	reserved	–	–	–	0 when being read.
		D6–4	EPDBCLKD [2:0]	EPD booster clock division ratio select	EPDB CLKD [2:0] / OSC3B / OSC3A / OSC1 0x5 / 1/128 / 1/256 / reserved 0x4 / 1/64 / 1/128 / reserved 0x3 / 1/32 / 1/64 / 1/8 0x2 / 1/16 / 1/32 / 1/4 0x1 / reserved / 1/16 / 1/2 0x0 / reserved / reserved / 1/1 Other / reserved	0x0	R/W	
		D3–2	EPDBCLK SRC[1:0]	EPD booster clock source select	EPDBCLK SRC[1:0] / Clock source 0x3 / reserved 0x2 / OSC3A 0x1 / OSC1 0x0 / OSC3B	0x0	R/W	
		D1	–	reserved	–	–	–	0 when being read.
		D0	EPDBCLKE	EPD booster clock enable	1 Enable / 0 Disable	0	R/W	

D7 Reserved

D[6:4] EPDBCLKD[2:0]: EPD Booster Clock Division Ratio Select Bits

Selects the division ratio for generating the booster clock.

Table 4.4.4 Clock Division Ratio Selection

EPDBCLKD[2:0]	Division ratio		
	Clock source = OSC3B	Clock source = OSC3A	Clock source = OSC1
0x7 to 0x6	Reserved	Reserved	Reserved
0x5	1/128	1/256	
0x4	1/64	1/128	
0x3	1/32	1/64	1/8
0x2	1/16	1/32	1/4
0x1	Reserved	1/16	1/2
0x0		Reserved	1/1

(Default: 0x0)

D[3:2] EPDBCLKSRC[1:0]: EPD Booster Clock Source Select Bits

Selects the booster clock source.

Table 4.4.5 Clock Source Selection

EPDBCLKSRC[1:0]	Clock source
0x3	Reserved
0x2	OSC3A
0x1	OSC1
0x0	OSC3B

(Default: 0x0)

D1 Reserved

D0 EPDBCLKE: EPD Booster Clock Enable Bit

Enables or disables the clock supply to the booster.
1 (R/W): Enabled (on)
0 (R/W): Disabled (off) (default)

The EPDBCLKE default setting is 0, which disables the clock supply. Setting EPDBCLKE to 1 sends the clock selected as above to the booster. If booster operation is not required, disable the clock supply to reduce current consumption.

V_{D1} Control Register (VD1_CTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
V_{D1} Control Register (VD1_CTL)	0x5120 (8 bits)	D7–6	–	reserved	–	–	–	0 when being read.
		D5	HVLD	V_{D1} heavy load protection mode	1 On 0 Off	0	R/W	
		D4–0	–	reserved	–	–	–	0 when being read.

D[7:6] Reserved

D5 HVLD: V_{D1} Heavy Load Protection Mode Bit

Sets the V_{D1} and V_{OSC} regulators into heavy load protection mode.
1 (R/W): Heavy load protection On
0 (R/W): Heavy load protection Off (default)

The V_{D1} and V_{OSC} regulators enter heavy load protection mode by writing 1 to HVLD and they ensure stable V_{D1} and V_{OSC} outputs. Use the heavy load protection function when a heavy load such as a lamp or buzzer is driven with a port output. Current consumption increases in heavy load protection mode, therefore do not set if unnecessary.

D[4:0] Reserved

EPD Power Control Register 0 (EPD_PWR0)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
EPD Power Control Register 0 (EPD_PWR0)	0x5600 (16 bits)	D15–10	–	reserved	–	–	–	0 when being read.
		D9	DBSRT	V_{E1}–V_{E2} doubler short	1 Short 0 Open	0	R/W	
		D8	DBON	V_{E1} doubler on/off	1 On 0 Off	0	R/W	
		D7–4	VECON[3:0]	V_E regulator contrast setting	VECON[3:0]: 0xf : 0x0 / Contrast: High : Low	0x7	R/W	
		D3	–	reserved	–	–	–	0 when being read.
		D2	HVLDVE	V_E heavy load protection mode	1 On 0 Off	0	R/W	
		D1	VESEL	Reference voltage select	1 V_{E2} 0 V_{E1}	0	R/W	
		D0	VEON	V_E regulator on/off	1 On 0 Off	0	R/W	

D[15:10] Reserved

D9 DBSRT: V_{E1}–V_{E2} Doubler Short Bit

Bridges the doubler input and the output.
1 (R/W): Short
0 (R/W): Open (default)

Between the doubler input and output can be short-circuited so that the doubler circuit cannot affect the V_E regulator output when V_{E2} is generated. When the doubler is not used, short between the input and output by setting DBSRT to 1.
When the doubler is activated, the input and output are not short-circuited even if DBSRT is set to 1.

D8 DBON: V_{E1} Doubler On/Off Bit

Turns the doubler on or off.
1 (R/W): On
0 (R/W): Off (default)

Set DBON to 1 to turn the doubler on when V_{E1} is generated by the V_E regulator. The doubler generates V_{E2} by doubling V_{E1}.
When V_{E2} is generated by the V_E regulator, set DBON to 0 as the doubler is not required.

D[7:4] VECON[3:0]: V_E Regulator Contrast Setting Bits

Switches the V_E regulator output voltage value to adjust the EPD contrast.

Table 4.4.6 Setting V_E Regulator Output Level (EPD Contrast Adjustment Function)

VECON[3:0]	V_E regulator output level
0xf	Level 15 (high contrast)
:	:
0x7	Level 7 (default)
:	:
0x0	Level 0 (low contrast)

See the "Electrical Characteristics" chapter for the voltage values.

D3 Reserved

D2 HVLDVE: V_E Heavy Load Protection Mode Bit

Sets the V_E regulator into heavy load protection mode.
1 (R/W): Heavy load protection On
0 (R/W): Heavy load protection Off (default)

The V_E regulator enters heavy load protection mode by writing 1 to HVLDVE and it ensures stable output. Use the heavy load protection function when the display has inconsistencies in density. Current consumption increases in heavy load protection mode, therefore do not set if unnecessary.

D1 VESEL: Reference Voltage Select Bit

Selects the V_E regulator output voltage (reference voltage for boosting).
1 (R/W): V_{E2}
0 (R/W): V_{E1} (default)

Select either V_{E1} or V_{E2} to be generated by the V_E regulator according to the V_{DD} value.

Table 4.4.7 V_E Regulator Output Selection

Power supply voltage V_{DD}	VESEL	V_E regulator output
$V_{DD} \geq V_{E1} + 0.3$ V	0 (default)	V_{E1}
$V_{DD} \geq V_{E2} + 0.2$ V	1	V_{E2}

D0 VEON: V_E Regulator On/Off Bit

Turns the V_E regulator on or off.
1 (R/W): On
0 (R/W): Off (default)

The V_E regulator is activated by setting VEON to 1 and it generates the reference voltage V_{E1} or V_{E2} for boosting.

EPD Power Control Register 1 (EPD_PWR1)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
EPD Power Control Register 1 (EPD_PWR1)	0x5602 (16 bits)	D15–12	VHCON [3:0]	V_H regulator contrast setting	VHCON[3:0] Contrast 0xf High : : 0x0 Low	0x5	R/W	
		D11–10	VHSEL[1:0]	V_H regulator range select	VHSEL[1:0] Voltage 0x3 reserved 0x2 9 V 0x1 12 V 0x0 15 V	0x0	R/W	
		D9	HVLDVH	V_H heavy load protection mode	1 On 0 Off	0	R/W	
		D8	VHON	V_H regulator on/off	1 On 0 Off	0	R/W	
		D7–2	–	reserved	–	–	–	0 when being read.
		D1	BSTPLD	Booster pull-down on/off	1 On 0 Off	0	R/W	
		D0	BSTON	Booster on/off	1 On 0 Off	0	R/W	

D[15:12] VHCON[3:0]: V_H Regulator Contrast Setting Bits

Switches the V_H regulator output voltage value to adjust the EPD contrast.

Table 4.4.8 Setting V_H Regulator Output Level (EPD Contrast Adjustment Function)

VHCON[3:0]	V_H regulator output level
0xf	Level 15 (high contrast)
:	:
0x5	Level 5 (default)
:	:
0x0	Level 0 (low contrast)

See the "Electrical Characteristics" chapter for the voltage values.

D[11:10] VHSEL[1:0]: V_H Regulator Range Select Bits

Selects the V_H regulator output voltage value (V_{EPD}) in accordance with the EPD specification.

Table 4.4.9 V_{EPD} Voltage Value

VHSEL[1:0]	V_{EPD} voltage
0x3	Reserved
0x2	9 V type
0x1	12 V type
0x0	15 V type

(Default: 0x0)

D9 HVLDVH: V_H Heavy Load Protection Mode Bit

Sets the V_H regulator into heavy load protection mode.
1 (R/W): Heavy load protection On
0 (R/W): Heavy load protection Off (default)

The V_H regulator enters heavy load protection mode by writing 1 to HVLDVH and it ensures stable output. Use the heavy load protection function when the display has inconsistencies in density. Current consumption increases in heavy load protection mode, therefore do not set if unnecessary.

D8 VHON: V_H Regulator On/Off Bit

Turns the V_H regulator on or off.
1 (R/W): On
0 (R/W): Off (default)

The V_H regulator is activated by setting VHON to 1 and the EPD drive voltage V_{EPD} is supplied to the EPD driver.

Note: After the V_H regulator is turned on, the V_{EPD} output voltage requires about 5 ms to stabilize. Do not start display on the EPD in this unstable period.

D[7:2] Reserved

D1 BSTPLD: Booster Pull-Down On/Off Bit

Pulls down the booster output to V_{SS}.
1 (R/W): On
0 (R/W): Off (default)

This bit allows software to set the EPD drive voltage to an off level.

D0 BSTON: Booster On/Off Bit

Turns the booster on or off.
1 (R/W): On
0 (R/W): Off (default)

The booster is activated by setting BSTON to 1 and it generates the V_{E5} to be input to the V_H regulator.

Note: After the booster is turned on, the V_{E5} output voltage requires about 35 ms to stabilize. Do not activate the V_H regulator in this unstable period.

5 Initial Reset

5.1 Initial Reset Sources

The S1C17F57 has three initial reset sources that initialize the internal circuits.

(1) #RESET pin (external initial reset)
(2) Key-entry reset using the P0 ports (P00–P03 pins) (software selectable external initial reset)
(3) Watchdog timer (software selectable internal initial reset)

Figure 5.1.1 shows the configuration of the initial reset circuit.

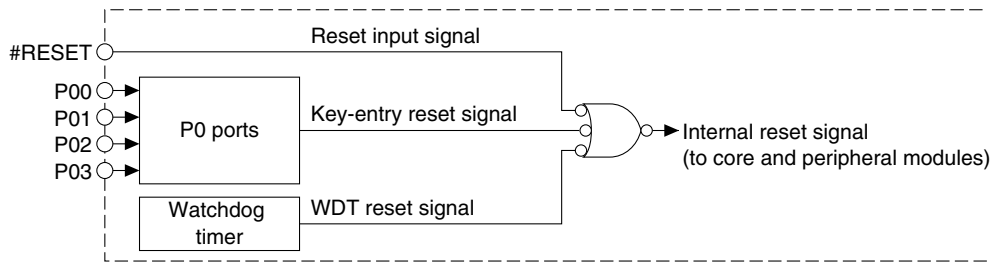


Figure 5.1.1 Configuration of Initial Reset Circuit

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

5.1.1 #RESET Pin

By setting the #RESET pin to low level, the S1C17F57 enters initial reset state. In order to initialize the S1C17F57 for sure, the #RESET pin must be held at low for more than the prescribed time (see "Input/Output Pin Characteristics" in the "Electrical Characteristics" chapter) after the power supply voltage is supplied.
When the #RESET pin at low level is set to high level, the CPU starts executing the initial reset sequence.
The #RESET pin is equipped with a pull-up resistor.

5.1.2 P0 Port Key-Entry Reset

Entering low level simultaneously to the ports (P00–P03) selected with software triggers an initial reset. For details of the key-entry reset function, see the "I/O Ports (P)" chapter.

Note: The P0 port key-entry reset function cannot be used for power-on reset as it must be enabled with software.

5.1.3 Resetting by the Watchdog Timer

The S1C17F57 has a built-in watchdog timer to detect runaway of the CPU. The watchdog timer overflows if it is not reset with software (due to CPU runaway) in four-second cycles. The overflow signal can generate either NMI or reset. Write 1 to the WDTMD/WDT_ST register to generate reset (NMI occurs when WDTMD = 0).
For details of the watchdog timer, see the "Watchdog Timer (WDT)" chapter.

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

5.2 Initial Reset Sequence

Even if the #RESET pin input negates the reset signal after power is turned on, the CPU cannot boot up until the oscillation stabilization waiting time (64/OSC3B clock frequency) and the internal reset hold period (32/OSC3B clock frequency) have elapsed.
Figure 5.2.1 shows the operating sequence following cancellation of initial reset.
The CPU starts operating in synchronization with the OSC3B (internal oscillator) clock after reset state is canceled.

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

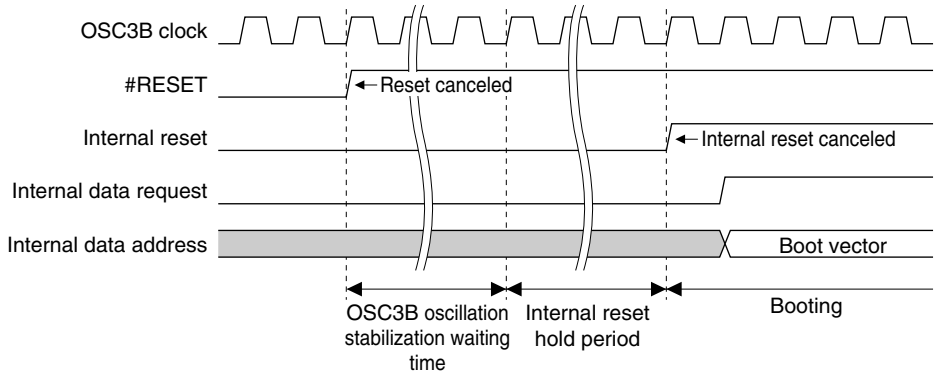


Figure 5.2.1 Operation Sequence Following Cancellation of Initial Reset

5.3 Initial Settings After an Initial Reset

The CPU internal registers are initialized as follows at initial reset.
R0–R7: 0x0
PSR: 0x0 (interrupt level = 0, interrupt disabled)
SP: 0x0
PC: Reset vector stored at the beginning of the vector table is loaded by the reset handling.

The internal RAM should be initialized with software as it is not initialized at initial reset.
The internal peripheral modules are initialized to the default values (except some undefined registers). Change the settings with software if necessary. For the default values set at initial reset, see the list of I/O registers in Appendix or descriptions for each peripheral module.

6 Interrupt Controller (ITC)

6.1 ITC Module Overview

The interrupt controller (ITC) honors interrupt requests from the peripheral modules and outputs the interrupt request, interrupt level and vector number signals to the S1C17 Core according to the priority and interrupt levels.
The features of the ITC module are listed below.

- Supports 16 maskable interrupt systems.
  1. P00–P07 input interrupt (8 types)
  2. P20–P27 input interrupt (8 types)
  3. Stopwatch timer interrupt (3 types)
  4. Clock timer interrupt (4 types)
  5. RTC interrupt (10 types)
  6. EPD interrupt (1 type)
  7. 16-bit PWM timer Ch.0 interrupt (6 types)
  8. 16-bit PWM timer Ch.1 interrupt (6 types)
  9. 8-bit timer Ch.0 interrupt (1 type)
  10. 8-bit timer Ch.1 interrupt (1 type)
  11. UART Ch.0 interrupt (4 types)
  12. SPI Ch.0 interrupt (2 types)
  13. I^2C master interrupt (2 types)
  14. I^2C slave interrupt (3 types)
  15. Temperature detection circuit interrupt (1 type)
  16. R/F converter interrupt (5 types)
- Supports eight interrupt levels to prioritize the interrupt sources.

The ITC enables the interrupt level (priority) for determining the processing sequence when multiple interrupts occur simultaneously to be set for each interrupt system separately.
Each interrupt system includes the number of interrupt causes indicated in parentheses above. Settings to enable or disable interrupt for different causes are set by the respective peripheral module registers.
For specific information on interrupt causes and their control, refer to the peripheral module explanations.
Figure 6.1.1 shows the structure of the interrupt system.

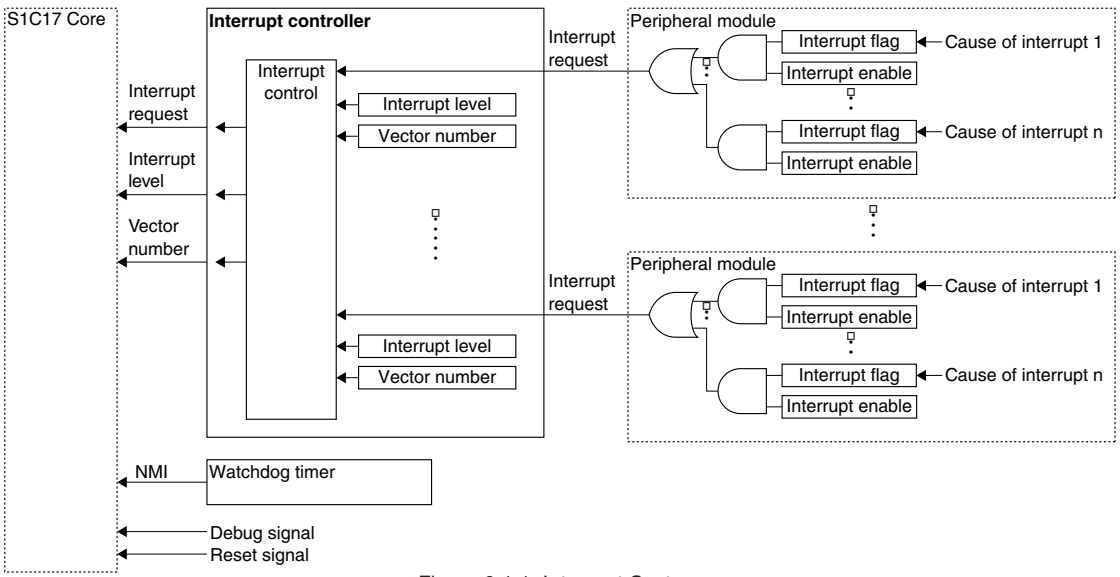


Figure 6.1.1 Interrupt System

6.2 Vector Table

The vector table contains the vectors to the interrupt handler routines (handler routine start address) that will be read by the S1C17 Core to execute the handler when an interrupt occurs.
Table 6.2.1 shows the vector table of the S1C17F57.

Table 6.2.1 Vector Table

Vector No. Software interrupt No.	Vector address	Hardware interrupt name	Cause of hardware interrupt	Priority
0 (0x00)	TTBR + 0x00	Reset	• Low input to the #RESET pin • Watchdog timer overflow *2	1
1 (0x01)	TTBR + 0x04	Address misaligned interrupt	Memory access instruction	2
–	(0xfffc00)	Debugging interrupt	`brk` instruction, etc.	3
2 (0x02)	TTBR + 0x08	NMI	Watchdog timer overflow *2	4
3 (0x03)	TTBR + 0x0c	Reserved for C compiler	–	–
4 (0x04)	TTBR + 0x10	P0 port interrupt	P00–P07 port inputs	High *1
5 (0x05)	TTBR + 0x14	P2 port interrupt	P20–P27 port inputs	↑
6 (0x06)	TTBR + 0x18	Stopwatch timer interrupt	• 100 Hz timer signal • 10 Hz timer signal • 1 Hz timer signal	
7 (0x07)	TTBR + 0x1c	Clock timer interrupt	• 32 Hz timer signal • 8 Hz timer signal • 2 Hz timer signal • 1 Hz timer signal	
8 (0x08)	TTBR + 0x20	RTC interrupt	• 32 Hz, 8 Hz, 4 Hz, 1 Hz • 10 s, 1 m, 10 m, 1 h • Half-day, one day	
9 (0x09)	TTBR + 0x24	reserved	–	
10 (0x0a)	TTBR + 0x28	EPD interrupt	Display update	
11 (0x0b)	TTBR + 0x2c	16-bit PWM timer Ch.0 interrupt	• Compare A/B • Capture A/B • Capture A/B overwrite	
12 (0x0c)	TTBR + 0x30	reserved	–	
13 (0x0d)	TTBR + 0x34			
14 (0x0e)	TTBR + 0x38	8-bit timer Ch. 0 interrupt	Timer underflow	
15 (0x0f)	TTBR + 0x3c	8-bit timer Ch. 1 interrupt	Timer underflow	
16 (0x10)	TTBR + 0x40	UART Ch.0 interrupt	• Transmit buffer empty • End of transmission • Receive buffer full • Receive error	
17 (0x11)	TTBR + 0x44	I^2C Slave interrupt	• Transmit buffer empty • Receive buffer full • Bus status	
18 (0x12)	TTBR + 0x48	SPI Ch.0 interrupt	• Transmit buffer empty • Receive buffer full	
19 (0x13)	TTBR + 0x4c	I^2C Master interrupt	• Transmit buffer empty • Receive buffer full	
20 (0x14)	TTBR + 0x50	reserved	–	
21 (0x15)	TTBR + 0x54	16-bit PWM timer Ch.1 interrupt	• Compare A/B • Capture A/B • Capture A/B overwrite	
22 (0x16)	TTBR + 0x58	Temperature detection circuit interrupt	Conversion completion	
23 (0x17)	TTBR + 0x5c	R/F converter interrupt	• Reference oscillation completion • Sensor A oscillation completion • Sensor B oscillation completion • Time base counter overflow error • Measurement counter overflow error	
24 (0x18) : 31 (0x1f)	TTBR + 0x60 : TTBR + 0x7c	reserved : reserved	– : –	↓ Low *1

*1 When the same interrupt level is set
*2 Either reset or NMI can be selected as the watchdog timer interrupt with software.

Vector numbers 4 to 8, 10 to 11, 14 to 19, and 21 to 23 are assigned to the maskable interrupts supported by the S1C17F57.

Vector table base address

The S1C17F57 allows the base (starting) address of the vector table to be set using the MISC_TTBRL and MISC_TTBRH registers. "TTBR" described in Table 6.2.1 means the value set to these registers. After an initial reset, the MISC_TTBRL and MISC_TTBRH registers are set to 0x8000. Therefore, even when the vector table location is changed, it is necessary that at least the reset vector be written to the above address. Bits 7 to 0 in the MISC_TTBRL register are fixed at 0, so the vector table starting address always begins with a 256-byte boundary address.

Vector Table Address Low/High Registers (MISC_TTBRL, MISC_TTBRH)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Vector Table Address Low Register (MISC_TTBRL)	**0x5328** (16 bits)	D15–8	**TTBR[15:8]**	Vector table base address A[15:8]	0x0–0xff	0x80	R/W	
		D7–0	**TTBR[7:0]**	Vector table base address A[7:0] (fixed at 0)	0x0	0x0	R	
Vector Table Address High Register (MISC_TTBRH)	**0x532a** (16 bits)	D15–8	**–**	reserved	–	–	–	0 when being read.
		D7–0	**TTBR[23:16]**	Vector table base address A[23:16]	0x0–0xff	0x0	R/W	

Note: The MISC_TTBRL and MISC_TTBRH registers are write-protected. Before these registers can be rewritten, write protection must be removed by writing data 0x96 to the MISC_PROT register. Note that since unnecessary rewrites to the MISC_TTBRL and MISC_TTBRH registers could lead to erratic system operation, the MISC_PROT register should be set to other than 0x96 unless the Vector Table Base Registers must be rewritten.

6.3 Control of Maskable Interrupts

6.3.1 Interrupt Control Bits in Peripheral Modules

The peripheral module that generates interrupts includes an interrupt enable bit and an interrupt flag for each interrupt cause. The interrupt flag is set to 1 when the cause of interrupt occurs. By setting the interrupt enable bit to 1 (interrupt enabled), the flag state will be sent to the ITC as an interrupt request signal, generating an interrupt request to the S1C17 Core.

The corresponding interrupt enable bits should be set to 0 for those causes for which interrupts are not desired. In this case, although the interrupt flag is set to 1 if the interrupt cause occurs, the interrupt request signal sent to the ITC will not be asserted.

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

Note: To prevent recurrence of the interrupt due to the same cause of interrupt, always reset the interrupt flag in the peripheral module before enabling the interrupt, resetting the PSR, or executing the `reti` instruction.

6.3.2 ITC Interrupt Request Processing

On receiving an interrupt signal from a peripheral module, the ITC sends the interrupt request, interrupt level, and vector number signals to the S1C17 Core.

Vector numbers are determined by the ITC internal hardware for each interrupt cause, as shown in Table 6.2.1.

The interrupt level is a value used by the S1C17 Core to compare with the IL bits (PSR). This interrupt level is used in the S1C17 Core to disable subsequently occurring interrupts with the same or lower level. (See Section 6.3.3.)

The default ITC settings are level 0 for all maskable interrupts. Interrupt requests are not accepted by the S1C17 Core if the level is 0.

The ITC includes control bits for selecting the interrupt level, and the level can be set to between 0 (low) and 7 (high) interrupt levels for each interrupt type.

If interrupt requests are input to the ITC simultaneously from two or more peripheral modules, the ITC outputs the interrupt request with the highest priority to the S1C17 Core in accordance with the following conditions.

1. The interrupt with the highest interrupt level takes precedence.
2. If multiple interrupt requests are input with the same interrupt level, the interrupt with the lowest vector number takes precedence.

The other interrupts occurring at the same time are held until all interrupts with higher priority levels have been accepted by the S1C17 Core.
If an interrupt cause with higher priority occurs while the ITC is outputting an interrupt request signal to the S1C17 Core (before being accepted by the S1C17 Core), the ITC alters the vector number and interrupt level signals to the setting information on the more recent interrupt. The previously occurring interrupt is held. The held interrupt is canceled and no interrupt is generated if the interrupt flag in the peripheral module is reset with software.

Table 6.3.2.1 Interrupt Level Setting Bits

Hardware interrupt	Interrupt level setting bits	Register address
P0 port interrupt	ILV0[2:0] (D[2:0]/ITC_LV0 register)	0x4306
P2 port interrupt	ILV1[2:0] (D[10:8]/ITC_LV0 register)	0x4306
Stopwatch timer interrupt	ILV2[2:0] (D[2:0]/ITC_LV1 register)	0x4308
Clock timer interrupt	ILV3[2:0] (D[10:8]/ITC_LV1 register)	0x4308
RTC interrupt	ILV4[2:0] (D[2:0]/ITC_LV2 register)	0x430a
EPD interrupt	ILV6[2:0] (D[2:0]/ITC_LV3 register)	0x430c
16-bit PWM timer Ch.0 interrupt	ILV7[2:0] (D[10:8]/ITC_LV3 register)	0x430c
8-bit timer Ch.0 interrupt	ILV10[2:0] (D[2:0]/ITC_LV5 register)	0x4310
8-bit timer Ch.1 interrupt	ILV11[2:0] (D[10:8]/ITC_LV5 register)	0x4310
UART Ch.0 interrupt	ILV12[2:0] (D[2:0]/ITC_LV6 register)	0x4312
I^2C slave interrupt	ILV13[2:0] (D[10:8]/ITC_LV6 register)	0x4312
SPI Ch.0 interrupt	ILV14[2:0] (D[2:0]/ITC_LV7 register)	0x4314
I^2C master interrupt	ILV15[2:0] (D[10:8]/ITC_LV7 register)	0x4314
16-bit PWM timer Ch.1 interrupt	ILV17[2:0] (D[10:8]/ITC_LV8 register)	0x4316
Temperature detection circuit interrupt	ILV18[2:0] (D[2:0]/ITC_LV9 register)	0x4318
R/F converter interrupt	ILV19[2:0] (D[10:8]/ITC_LV9 register)	0x4318

6.3.3 Interrupt Processing by the S1C17 Core

A maskable interrupt to the S1C17 Core occurs when all of the following conditions are met:

- The interrupt is enabled by the interrupt control bit inside the peripheral module.
- The IE (Interrupt Enable) bit of the PSR (Processor Status Register) in the S1C17 Core has been set to 1.
- The cause of interrupt that has occurred has a higher interrupt level than the value set in the IL field of the PSR.
- No other cause of interrupt having higher priority, such as NMI, has occurred.

If an interrupt cause that has been enabled in the peripheral module occurs, the corresponding interrupt flag is set to 1, and this state is maintained until it is reset by the program. This means that the interrupt cause is not cleared even if the conditions listed above are not met when the interrupt cause occurs. An interrupt occurs if the above conditions are met.
If multiple maskable interrupt causes occurs simultaneously, the interrupt cause with the highest interrupt level and lowest vector number becomes the subject of the interrupt request to the S1C17 Core. Interrupts with lower levels are held until the above conditions are subsequently met.

The S1C17 Core samples interrupt requests for each cycle. On accepting an interrupt request, the S1C17 Core switches to interrupt processing immediately after execution of the current instruction has been completed.
Interrupt processing involves the following steps:
(1) The PSR and current program counter (PC) values are saved to the stack.
(2) The PSR IE bit is reset to 0 (disabling subsequent maskable interrupts).
(3) The PSR IL bits are set to the received interrupt level. (The NMI does not affect the IL bits.)
(4) The vector for the interrupt occurred is loaded to the PC to execute the interrupt handler routine.

When an interrupt is accepted, (2) prevents subsequent maskable interrupts. Setting the IE bit to 1 in the interrupt handler routine allows handling of multiple interrupts. In this case, since IL is changed by (3), only an interrupt with a higher level than that of the currently processed interrupt will be accepted.
Ending interrupt handler routines using the `reti` instruction returns the PSR to the state before the interrupt has occurred. The program resumes processing following the instruction being executed at the time the interrupt occurred.

6.4 NMI

In the S1C17F57, the watchdog timer can generate a non-maskable interrupt (NMI). The vector number for NMI is 2, with the vector address set to the vector table's starting address + 8 bytes.
This interrupt takes precedence over other interrupts and is unconditionally accepted by the S1C17 Core.

For detailed information on generating NMI, see the "Watchdog Timer (WDT)" chapter.

6.5 Software Interrupts

The S1C17 Core provides the "`int` *`imm5`*" and "`intl` *`imm5,imm3`*" instructions allowing the software to generate any interrupts. The operand *imm5* specifies a vector number (0–31) in the vector table. In addition to this, the `intl` instruction has the operand *imm3* to specify the interrupt level (0–7) to be set to the IL field in the PSR.
The processor performs the same interrupt processing as that of the hardware interrupt.

6.6 HALT and SLEEP Mode Cancellation

HALT and SLEEP modes are cleared by the following signals, which start the CPU.
- Interrupt request signal sent to the CPU from the ITC
- NMI signal output by the watchdog timer
- Debug interrupt signal
- Reset signal

Notes: • If the CPU is able to receive interrupts when HALT or SLEEP mode has been cleared by an interrupt request for the CPU from the ITC, processing branches to the interrupt handler routine immediately after cancellation. In all other cases, the program is executed following the `halt` or `slp` instruction.

• HALT or SLEEP mode clearing due to interrupt requests cannot be masked (prohibited) using ITC interrupt level settings.

For more information, see "Power Saving by Clock Control" in the appendix chapter. For the oscillator circuit and system clock statuses after HALT or SLEEP mode is canceled, see the "Clock Generator (CLG)" chapter.

6.7 Control Register Details

Table 6.7.1 List of ITC Registers

Address	Register name		Function
0x4306	ITC_LV0	Interrupt Level Setup Register 0	Sets the P0 and P2 interrupt levels.
0x4308	ITC_LV1	Interrupt Level Setup Register 1	Sets the SWT and CT interrupt levels.
0x430a	ITC_LV2	Interrupt Level Setup Register 2	Sets the RTC interrupt level.
0x430c	ITC_LV3	Interrupt Level Setup Register 3	Sets the EPD and T16A2 Ch.0 interrupt levels.
0x4310	ITC_LV5	Interrupt Level Setup Register 5	Sets the T8 Ch.0 and Ch.1 interrupt levels.
0x4312	ITC_LV6	Interrupt Level Setup Register 6	Sets the UART Ch.0 and I2CS interrupt levels.
0x4314	ITC_LV7	Interrupt Level Setup Register 7	Sets the SPI Ch.0 and I2CM interrupt levels.
0x4316	ITC_LV8	Interrupt Level Setup Register 8	Sets the T16A2 Ch.1 interrupt level.
0x4318	ITC_LV9	Interrupt Level Setup Register 9	Sets the TEM and RFC interrupt levels.

The ITC registers are described in detail below.

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

Interrupt Level Setup Register *x* (ITC_LV*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Interrupt Level Setup Register *x* (ITC_LV*x*)	0x4306 | 0x4318 (16 bits)	D15–11	–	reserved	–	–	–	0 when being read.
		D10–8	ILV*n*[2:0]	*INTn* (1, 3, ... 19) interrupt level	0 to 7	0x0	R/W	
		D7–3	–	reserved	–	–	–	0 when being read.
		D2–0	ILV*n*[2:0]	*INTn* (0, 2, ... 18) interrupt level	0 to 7	0x0	R/W	

D[15:11], D[7:3]

Reserved

D[10:8], D[2:0]

ILV*n*[2:0]: *INTn* Interrupt Level Bits (*n* = 0–19)

Sets the interrupt level (0 to 7) of each interrupt. (Default: 0x0)

The S1C17 Core does not accept interrupts with a level set lower than the PSR IL value.

The ITC uses the interrupt level when multiple interrupt requests occur simultaneously.

If multiple interrupt requests enabled by the interrupt enable bit occur simultaneously, the ITC sends the interrupt request with the highest level set by the ITC_LV*x* registers (0x4306 to 0x4318) to the S1C17 Core.

If multiple interrupt requests with the same interrupt level occur simultaneously, the interrupt with the lowest vector number is processed first.

The other interrupts are held until all interrupts of higher priority have been accepted by the S1C17 Core.

If an interrupt requests of higher priority occurs while the ITC outputs an interrupt request signal to the S1C17 Core (before acceptance by the S1C17 Core), the ITC alters the vector number and interrupt level signals to the setting details of the most recent interrupt. The immediately preceding interrupt is held.

Table 6.7.2 Interrupt Level Bits

Register	Bit	Interrupt
ITC_LV0(0x4306)	ILV0[2:0] (D[2:0])	P0 port interrupt
	ILV1[2:0] (D[10:8])	P2 port interrupt
ITC_LV1(0x4308)	ILV2[2:0] (D[2:0])	Stopwatch timer interrupt
	ILV3[2:0] (D[10:8])	Clock timer interrupt
ITC_LV2(0x430a)	ILV4[2:0] (D[2:0])	RTC interrupt
	(ILV5[2:0] (D[10:8]))	Reserved
ITC_LV3(0x430c)	ILV6[2:0] (D[2:0])	EPD interrupt
	ILV7[2:0] (D[10:8])	16-bit PWM timer Ch.0 interrupt
(ITC_LV4(0x430e))	(ILV8[2:0] (D[2:0]))	Reserved
	(ILV9[2:0] (D[10:8]))	Reserved
ITC_LV5(0x4310)	ILV10[2:0] (D[2:0])	8-bit timer Ch.0 interrupt
	ILV11[2:0] (D[10:8])	8-bit timer Ch.1 interrupt
ITC_LV6(0x4312)	ILV12[2:0] (D[2:0])	UART Ch.0 interrupt
	ILV13[2:0] (D[10:8])	I^2C slave interrupt
ITC_LV7(0x4314)	ILV14[2:0] (D[2:0])	SPI Ch.0 interrupt
	ILV15[2:0] (D[10:8])	I^2C master interrupt
ITC_LV8(0x4316)	(ILV16[2:0] (D[2:0]))	Reserved
	ILV17[2:0] (D[10:8])	16-bit PWM timer Ch.1 interrupt
ITC_LV9(0x4318)	ILV18[2:0] (D[2:0])	Temperature detection circuit interrupt
	ILV19[2:0] (D[10:8])	R/F converter interrupt

7 Clock Generator (CLG)

7.1 CLG Module Overview

The clock generator (CLG) controls the internal oscillators and the system clocks to be supplied to the S1C17 Core, on-chip peripheral modules, and external devices.
The features of the CLG module are listed below.

- Generates the operating clocks with the built-in oscillators.
  - OSC3B oscillator circuit: 2 MHz/1 MHz/500 kHz (typ.) internal oscillator circuit
  - OSC3A oscillator circuit: 4.2 MHz (max.) crystal or ceramic oscillator circuit
  - OSC1B oscillator circuit: 32 kHz (typ.) internal oscillator circuit
  - OSC1A oscillator circuit: 32.768 kHz (typ.) crystal oscillator circuit
- Switches the system clock. The system clock source can be selected from OSC3B, OSC3A, and OSC1 via software.
- Generates the CPU core clock (CCLK) and controls the clock supply to the core block. The CCLK frequency can be selected from system clock × 1/1, 1/2, 1/4, and 1/8.
- Controls the clock supply to the peripheral modules.
- Turns the clocks on and off according to the CPU operating status (RUN, HALT, or SLEEP).
- Supports quick-restart processing from SLEEP mode.
  Turns OSC3B on forcibly and switches the system clock to OSC3B when SLEEP mode is canceled.
- Controls two clock outputs to external devices.

Figure 7.1.1 shows the clock system and CLG module configuration.

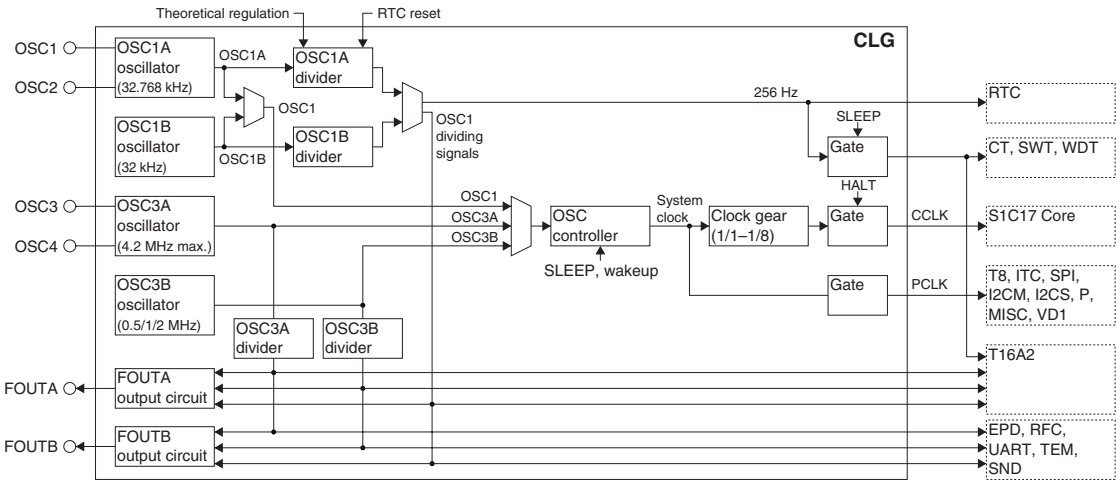


Figure 7.1.1 CLG Module Configuration

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

7.2 CLG Input/Output Pins

Table 7.2.1 lists the input/output pins for the CLG module.

Table 7.2.1 List of CLG Pins

Pin name	I/O	Qty	Function
OSC1	I	1	OSC1A oscillator input pin Connect a crystal resonator (32.768 kHz) and a gate capacitor.
OSC2	O	1	OSC1A oscillator output pin Connect a crystal resonator (32.768 kHz).
OSC3	I	1	OSC3A oscillator input pin Connect a crystal or ceramic resonator (max. 4.2 MHz), and a gate capacitor.
OSC4	O	1	OSC3A oscillator output pin Connect a crystal or ceramic resonator (max. 4.2 MHz), and a drain capacitor.
FOUTA	O	1	FOUTA clock output pin Outputs a divided OSC3B, OSC3A, or OSC1 clock.
FOUTB	O	1	FOUTB clock output pin Outputs a divided OSC3B, OSC3A, or OSC1 clock.

The CLG output pins (FOUTA, FOUTB) are shared with I/O ports and are initially set as general purpose I/O port pins. The pin functions must be switched using the port function select bits to use the general purpose I/O port pins as the CLG output pins. For detailed information on pin function switching, see the "I/O Ports (P)" chapter.

7.3 Oscillators

The CLG module contains four internal oscillator circuits (OSC3B, OSC3A, OSC1B, and OSC1A). The OSC3B and OSC3A oscillators generate the main clock for high-speed operation of the S1C17 Core and peripheral circuits. The OSC1B or OSC1A oscillator generates a sub-clock for timers and low-power operations. The OSC3B clock is selected as the system clock after an initial reset. Oscillator on/off switching and system clock selection (from OSC3B, OSC3A, and OSC1) are controlled with software. Either the OSC1B or OSC1A oscillator can be selected as the OSC1 clock source.

7.3.1 OSC3B Oscillator

The OSC3B oscillator initiates high-speed oscillation without external components. It initiates oscillation when power is turned on. The S1C17 Core and peripheral circuits operate with this oscillation clock after an initial reset.

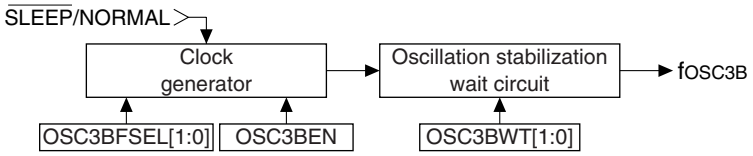


Figure 7.3.1.1 OSC3B Oscillator Circuit

OSC3B oscillation frequency selection

The OSC3B oscillation frequency can be selected from three types shown below using OSC3BFSEL[1:0]/CLG_SRC register.

Table 7.3.1.1 OSC3B Oscillation Frequency Setting

OSC3BFSEL[1:0]	OSC3B oscillation frequency (typ.)
0x3	Reserved
0x2	500 kHz
0x1	1 MHz
0x0	2 MHz

(Default: 0x0)

OSC3B oscillation on/off

The OSC3B oscillator stops oscillating when OSC3BEN/CLG_CTL register is set to 0 and starts oscillating when set to 1. The OSC3B oscillator stops oscillating in SLEEP mode.
After an initial reset, OSC3BEN is set to 1, and the OSC3B oscillator goes on. Since the OSC3B clock is used as the system clock, the S1C17 Core starts operating using the OSC3B clock.

Stabilization wait time at start of OSC3B oscillation

The OSC3B oscillator circuit includes an oscillation stabilization wait circuit to prevent malfunctions due to unstable clock operations at the start of OSC3B oscillation—e.g., when the OSC3B oscillator is turned on with software. Figure 7.3.1.2 shows the relationship between the oscillation start time and the oscillation stabilization wait time.

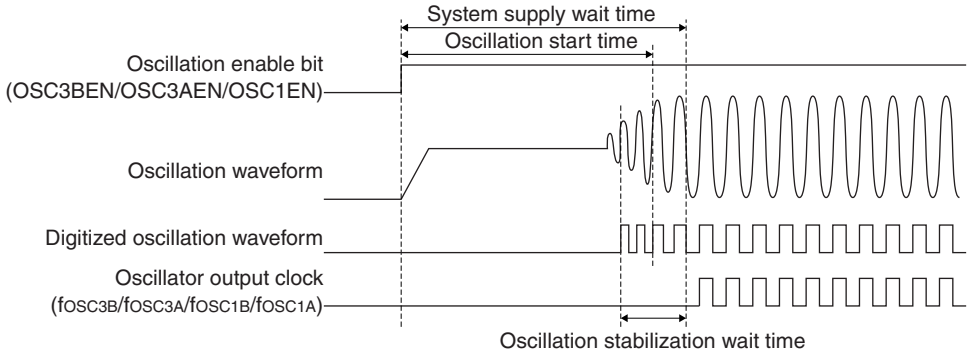


Figure 7.3.1.2 Oscillation Start Time and Oscillation Stabilization Wait Time

The OSC3B clock is not supplied to the system until the time set for this circuit has elapsed.
Use OSC3BWT[1:0]/CLG_WAIT register to select one of four oscillation stabilization wait times.

Table 7.3.1.2 OSC3B Oscillation Stabilization Wait Time Settings

OSC3BWT[1:0]	Oscillation stabilization wait time
0x3	8 cycles
0x2	16 cycles
0x1	32 cycles
0x0	64 cycles

(Default: 0x0)

This is set to 64 cycles (OSC3B clock) after an initial reset. This means the CPU can start operating when the CPU operation start time at initial reset indicated below (at a maximum) has elapsed after the reset state is canceled. For the oscillation start time, see the "Electrical Characteristics" chapter.

CPU operation start time at initial reset ≤ OSC3B oscillation start time (max.) + OSC3B oscillation stabilization wait time (64 cycles)

When the system clock is switched to OSC3B immediately after turning the OSC3B oscillator on, the OSC3B clock is supplied to the system after the OSC3B clock system supply wait time indicated below (at a maximum) has elapsed. If the power supply voltage V_{DD} has stabilized sufficiently, OSC3BWT[1:0] can be set to 0x3 to reduce the oscillation stabilization wait time.

OSC3B clock system supply wait time ≤ OSC3B oscillation start time (max.) + OSC3B oscillation stabilization wait time

7.3.2 OSC3A Oscillator

The OSC3A oscillator is a high-precision, high-speed oscillator circuit that uses either a crystal resonator or a ceramic resonator. It can be switched for use with the OSC3B oscillator. Figure 7.3.2.1 shows the OSC3A oscillator configuration.

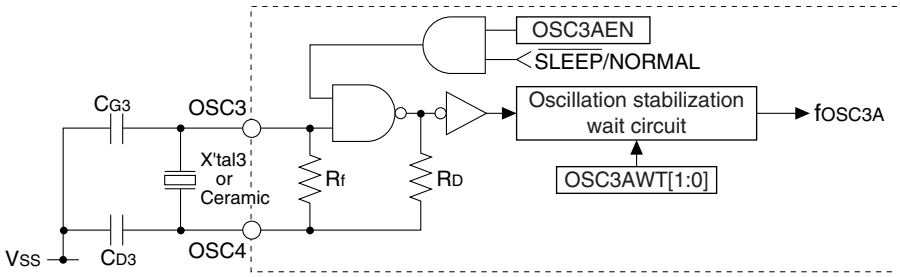


Figure 7.3.2.1 OSC3A Oscillator Circuit

A crystal resonator (X'tal3) or ceramic resonator (Ceramic) should be connected between the OSC3 and OSC4 pins. Additionally, two capacitors (C_{G3} and C_{D3}) should be connected between the OSC3/OSC4 pins and V_{SS}.
For the effective frequency range and oscillation characteristics, see the "Electrical Characteristics" chapter.

OSC3A oscillation on/off

The OSC3A oscillator circuit starts oscillating when OSC3AEN/CLG_CTL register is set to 1 and stops oscillating when set to 0. The OSC3A oscillator circuit stops oscillating in SLEEP mode.
After an initial reset, OSC3AEN is set to 0, and the OSC3A oscillator circuit is halted.

Stabilization wait time at start of OSC3A oscillation

The OSC3A oscillator circuit includes an oscillation stabilization wait circuit to prevent malfunctions due to unstable clock operations at the start of OSC3A oscillation—e.g., when the OSC3A oscillator is turned on with software. The OSC3A clock is not supplied to the system until the time set for this circuit has elapsed. Use OSC3AWT[1:0]/CLG_WAIT register to select one of four oscillation stabilization wait times. For the oscillation start time, see the "Electrical Characteristics" chapter.

Table 7.3.2.1 OSC3A Oscillation Stabilization Wait Time Settings

OSC3AWT[1:0]	Oscillation stabilization wait time
0x3	128 cycles
0x2	256 cycles
0x1	512 cycles
0x0	1024 cycles

(Default: 0x0)

This is set to 1,024 cycles (OSC3A clock) after an initial reset.
When the system clock is switched to OSC3A immediately after the OSC3A oscillator circuit is turned on, the OSC3A clock is supplied to the system after the OSC3A clock system supply wait time indicated below (at a maximum) has elapsed. For the oscillation start time, see the "Electrical Characteristics" chapter.

OSC3A clock system supply wait time ≤ OSC3A oscillation start time (max.) + OSC3A oscillation stabilization wait time

Note: Oscillation stability will vary, depending on the resonator and other external components. Carefully consider the OSC3A oscillation stabilization wait time before reducing the time.

7.3.3 OSC1 Oscillator

The S1C17F57 has two low-speed oscillator circuits (OSC1A and OSC1B) and either one can be selected as the OSC1 oscillator.
The OSC1 clock is generally used as the timer operation clock (for the real-time clock, clock timer, stopwatch timer, watchdog timer, and 16-bit PWM timer) and an operation clock for the R/F converter, UART, sound generator, temperature detection circuit, and EPD controller/driver. It can be used as the system clock instead of the OSC3B or OSC3A clock to reduce power consumption when no high-speed processing is required.

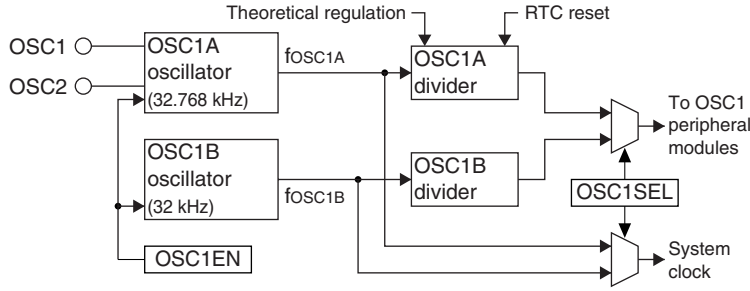


Figure 7.3.3.1 OSC1 Oscillator Configuration

OSC1A oscillator

The OSC1A oscillator is a high-precision, low-speed oscillator circuit that uses a 32.768 kHz crystal resonator. Figure 7.3.3.2 shows the OSC1A oscillator configuration.

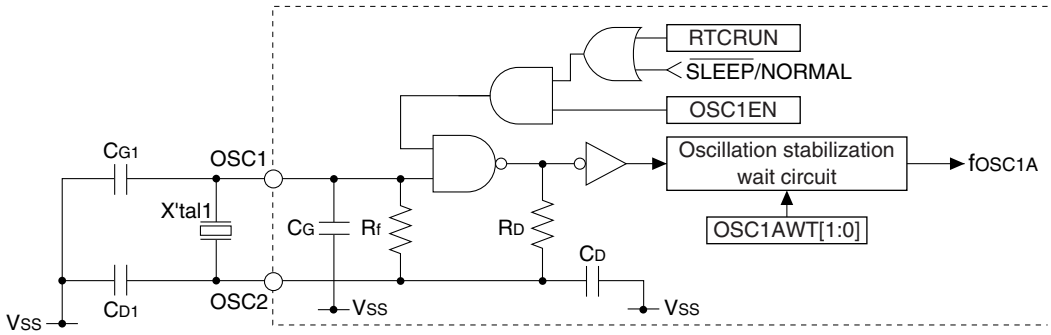


Figure 7.3.3.2 OSC1A Oscillator Circuit

A crystal resonator (X'tal1, typ. 32.768 kHz) should be connected between the OSC1 and OSC2 pins. Additionally, two capacitors (C_{G1} and C_{D1}) should be connected between the OSC1/OSC2 pins and V_{SS}.

Note: The OSC1A divider output clock may be adjusted for frequency correction by the theoretical regulation function. Also the divider is reset by staring the RTC. These operation modifies the 256 Hz output clock cycle at that point, and this affects the count cycle of the timers that use the 256 Hz clock (CT, SWT, WDT, T16A2).

OSC1B oscillator

The OSC1B oscillator generates about 32 kHz clock without external components.

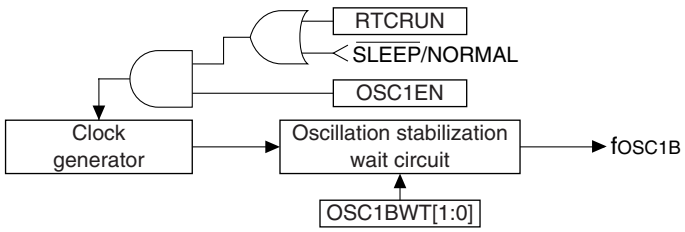


Figure 7.3.3.3 OSC1B Oscillator Circuit

OSC1A/OSC1B oscillator selection

Either OSC1A or OSC1B can be selected as the OSC1 oscillator using OSC1SEL/CLG_SRC register. When OSC1SEL is 1 (default), OSC1B is selected. Setting OSC1SEL to 0 selects OSC1A. The OSC1 oscillation control bits are effective only for the oscillator selected here.

OSC1 oscillation on/off

The OSC1 oscillator starts oscillating when OSC1EN/CLG_CTL register is set to 1 and stops oscillating when set to 0.

When RTCRUN and OSC1EN are both set to 1, the OSC1 oscillator continues operating if the system enters SLEEP mode.
When RTCRUN = 0, the OSC1 stops in SLEEP mode regardless of how OSC1EN is set.
After an initial reset, OSC1EN and RTCRUN are both set to 0, and the OSC1 oscillator circuit is halted.

Table 7.3.3.1 OSC1 Oscillator Operating Status (normal operation)

OSC1EN	RTCRUN	OSC1 oscillator
1	1	On
1	0	On
0	1	Off
0	0	Off

Table 7.3.3.2 OSC1 Oscillator Operating Status (SLEEP mode)

OSC1EN	RTCRUN	OSC1 oscillator
1	1	On
1	0	Off
0	1	Off
0	0	Off

Stabilization wait time at start of OSC1 oscillation

The OSC1 oscillator circuit includes an oscillation stabilization wait circuit to prevent malfunctions due to unstable clock operations at the start of OSC1 oscillation—e.g., when the OSC1 oscillator is turned on with software. The OSC1 clock is not supplied to the system until the time set for this circuit has elapsed. Use OSC1AWT[1:0]/CLG_WAIT register to select one of four OSC1A oscillation stabilization wait times. Use OSC1BWT[1:0]/CLG_WAIT register for OSC1B. For the oscillation start time, see the "Electrical Characteristics" chapter.

Table 7.3.3.3 OSC1A Oscillation Stabilization Wait Time Settings

OSC1AWT[1:0]	Oscillation stabilization wait time
0x3	2048 cycles
0x2	4096 cycles
0x1	8192 cycles
0x0	16384 cycles

(Default: 0x0)

Table 7.3.3.4 OSC1B Oscillation Stabilization Wait Time Settings

OSC1BWT[1:0]	Oscillation stabilization wait time
0x3	8 cycles
0x2	16 cycles
0x1	32 cycles
0x0	64 cycles

(Default: 0x0)

This is set to 16384 cycles (OSC1 clock) when OSC1A is selected or 64 cycles when OSC1B is selected after an initial reset.
When the system clock is switched to OSC1 immediately after the OSC1 oscillator circuit is turned on, the OSC1 clock is supplied to the system after the OSC1 clock system supply wait time indicated below (at a maximum) has elapsed. For the oscillation start time, see the "Electrical Characteristics" chapter.

OSC1 clock system supply wait time ≤ OSC1 oscillation start time (max.) + OSC1 oscillation stabilization wait time

Notes:
- Oscillation stability will vary, depending on the resonator and other external components. Carefully consider the OSC1A oscillation stabilization wait time before reducing the time.
- Be sure to avoid turning the OSC1A or OSC1B oscillator off for at least four seconds from start of oscillation after the oscillator is turned on. For the oscillation start time, see the "Electrical Characteristics" chapter.
- The OSC1B oscillation frequency will be higher than the value that is described in the "Electrical Characteristics" chapter for about 3 ms immediately after turning the OSC1B oscillator on.

7.4 System Clock Switching

The figure below shows the system clock selector.

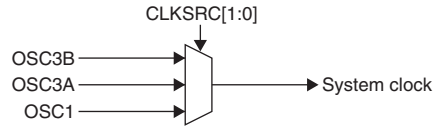


Figure 7.4.1 System Clock Selector

The S1C17F57 has three system clock sources (OSC3B, OSC3A, and OSC1) and it start operating with the OSC3B clock after an initial reset. The system clock can be switched to the OSC3A clock when a high-speed clock is required for the processing, or to the OSC1 clock for power saving. Use CLKSRC[1:0]/CLG_SRC register for this switching. Oscillator circuits other than those selected as the system clock source and not used for running peripheral circuits can be shut down to reduce current consumption.

Table 7.4.1 System Clock Selection

CLKSRC[1:0]	System clock source
0x3	Reserved
0x2	OSC3A
0x1	OSC1
0x0	OSC3B

(Default: 0x0)

The following shows system clock switching procedures:

Switching the system clock to OSC3A from OSC3B or OSC1

1. Set the OSC3A oscillation stabilization wait time if necessary. (OSC3AWT[1:0])
2. Turn the OSC3A oscillator on if it is off. (OSC3AEN = 1)
3. Select the OSC3A clock as the system clock. (CLKSRC[1:0] = 0x2)
4. Turn the OSC3B or OSC1 oscillator off if peripheral modules and FOUTA/B output circuits have not used the OSC3B or OSC1 clock.

Switching the system clock to OSC1 from OSC3B or OSC3A

1. Set the OSC1A or OSC1B oscillation stabilization wait time if necessary. (OSC1AWT[1:0]/OSC1BWT[1:0])
2. Turn the OSC1 oscillator on if it is off. (OSC1EN = 1)
3. Select the OSC1 clock as the system clock. (CLKSRC[1:0] = 0x1)
4. Turn the OSC3B or OSC3A oscillator off if peripheral modules and FOUTA/B output circuits have not used the OSC3B or OSC3A clock.

Switching the system clock to OSC3B from OSC3A or OSC1

1. Set the OSC3B oscillation stabilization wait time if necessary. (OSC3BWT[1:0])
2. Turn the OSC3B oscillator on if it is off. (OSC3BEN = 1)
3. Select the OSC3B clock as the system clock. (CLKSRC[1:0] = 0x0)
4. Turn the OSC3A or OSC1 oscillator off if peripheral modules and FOUTA/B output circuits have not used the OSC3A or OSC1 clock.

Notes: • The oscillator to be used as the system clock source must be operated before switching the system clock. Otherwise, the CLG will not switch the system clock source, even if CLKSRC[1:0] is written to, and the CLKSRC[1:0] value will remain unchanged.
The table below lists the combinations of clock operating status and register settings enabling system clock selection.

Table 7.4.2 System Clock Switching Conditions

OSC3BEN	OSC3AEN	OSC1EN	System clock
1	1	1	OSC3B, OSC3A, or OSC1
1	1	0	OSC3B or OSC3A
1	0	1	OSC3B or OSC1
0	1	1	OSC3A or OSC1

- The oscillator circuit selected as the system clock source cannot be turned off.
- Continuous write/read access to CLKSRC[1:0] is prohibited. At least one instruction unrelated to CLKSRC[1:0] access must be inserted between the write and read instructions.
- When SLEEP mode is canceled, the OSC3B oscillator circuit is turned on (OSC3BEN = 1) and is used as the system clock source (CLKSRC[1:0] = 0x0) regardless of the system clock configured before the chip entered SLEEP mode.
  Canceling HALT mode does not change the clock status configured before the chip entered HALT mode.

7.5 CPU Core Clock (CCLK) Control

The CLG module includes a clock gear to slow down the system clock to send to the S1C17 Core. To reduce current consumption, operate the S1C17 Core with the slowest possible clock speed. The `halt` instruction can be executed to stop the clock supply from the CLG to the S1C17 Core for power savings.

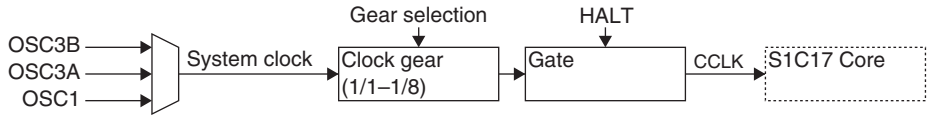


Figure 7.5.1 CCLK Supply System

Clock gear settings

CCLKGR[1:0]/CLG_CCLK register is used to select the gear ratio to reduce system clock speeds.

Table 7.5.1 CCLK Gear Ratio Selection

CCLKGR[1:0]	Gear ratio
0x3	1/8
0x2	1/4
0x1	1/2
0x0	1/1

(Default: 0x0)

Clock supply control

The CCLK clock supply is stopped by executing the `halt` instruction. Since this does not stop the system clock, peripheral modules will continue to operate.

HALT mode is cleared by resetting, NMI, or other interrupts. The CCLK supply resumes when HALT mode is cleared.

Executing the `slp` instruction suspends system clock supply to the CLG, thereby halting the CCLK supply as well. Clearing SLEEP mode with an external interrupt restarts the system clock supply and the CCLK supply.

7.6 Peripheral Module Clock (PCLK) Control

The CLG module also controls the clock supply to peripheral modules.
The system clock is used unmodified for the peripheral module clock (PCLK).

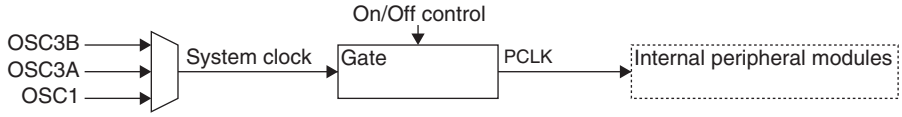


Figure 7.6.1 Peripheral Module Clock Control Circuit

Clock supply control

PCLK supply is controlled by PCKEN[1:0]/CLG_PCLK register.

Table 7.6.1 PCLK Control

PCKEN[1:0]	PCLK supply
0x3	Enabled (on)
0x2	Setting prohibited
0x1	Setting prohibited
0x0	Disabled (off)

(Default: 0x3)

The default setting is 0x3, which enables the clock supply. Stop the clock supply to reduce current consumption unless all peripheral modules (modules listed below) within the internal peripheral circuit area need to be running.

Note: Do not set PCKEN[1:0]/CLG_PCLK register to 0x2 or 0x1, since doing so will stop the operation of certain peripheral modules.

Table 7.6.2 Peripheral Modules and Operating Clocks

Peripheral modules	Operating clock	Remarks
Interrupt controller	PCLK	The PCLK supply cannot be disabled if one or more peripheral modules in these list must be operated. The PCLK supply can be disabled if all the peripheral circuits in these list can be stopped.
8-bit timer		
SPI		
I²C master		
I²C slave		
Power generator		
P port & port MUX		
MISC registers		
Real-time clock	Divided OSC1 clock	The OSC1 oscillator circuit cannot be disabled if one or more peripheral modules in these list must be operated. The PCLK supply can be disabled.
Clock timer		
Stopwatch timer		
Watchdog timer		
EPD controller/driver	Clock selected by software (divided OSC3B/OSC3A/ OSC1 clock)	The oscillator circuit used as the clock source cannot be disabled (see Section 7.7 or each peripheral module chapter). The PCLK supply can be disabled.
Sound generator		
Temperature detection circuit		
R/F converter		
16-bit PWM timer		
UART		
FOUTA/FOUTB outputs		

7.7 Clock External Output (FOUTA, FOUTB)

Divided OSC3B, OSC3A, or OSC1 clocks can be output to external devices.

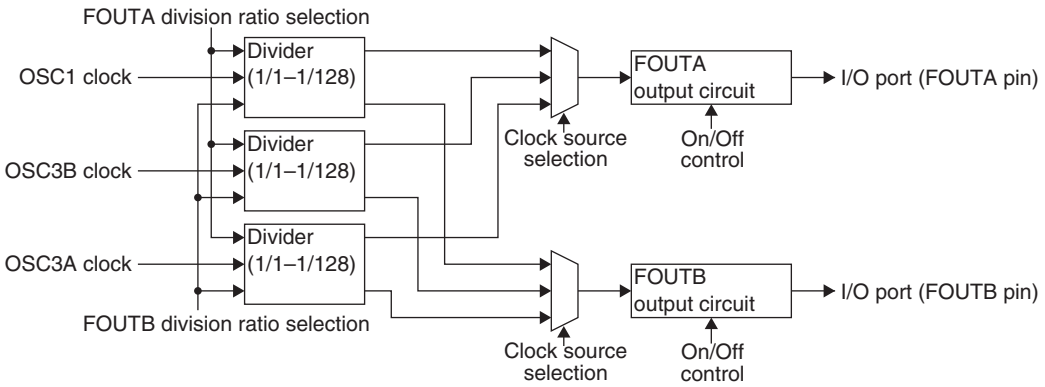


Figure 7.7.1 Clock Output Circuit

There are two output systems available: FOUTA and FOUTB. The FOUTA and FOUTB output circuits have the same functions.

Output pin setting

The FOUTA and FOUTB output pins are shared with I/O ports. The pin is configured for the I/O port by default, so the pin function should be changed using the port function select bit before the clock output can be used. See the "I/O Ports (P)" chapter for the FOUTA/FOUTB pins and selecting pin functions.

Clock source selection

The clock source can be selected from OSC3B, OSC3A, and OSC1 using FOUTASRC[1:0]/CLG_FOUTA register or FOUTBSRC[1:0]/CLG_FOUTB register.

Table 7.7.1 Clock Source Selection

FOUTASRC[1:0]/FOUTBSRC[1:0]	Clock source
0x3	Reserved
0x2	OSC3A
0x1	OSC1
0x0	OSC3B

(Default: 0x0)

Clock frequency selection

Eight different clock output frequencies can be selected. Select the division ratio for the source clock using FOUTAD[2:0]/CLG_FOUTA register or FOUTBD[2:0]/CLG_FOUTB register.

Table 7.7.2 Clock Division Ratio Selection

FOUTAD[2:0]/FOUTBD[2:0]	Division ratio
0x7	1/128
0x6	1/64
0x5	1/32
0x4	1/16
0x3	1/8
0x2	1/4
0x1	1/2
0x0	1/1

(Default: 0x0)

Clock output control

The clock output is controlled using FOUTAE/CLG_FOUTA register or FOUTBE/CLG_FOUTB register. Setting FOUTAE/FOUTBE to 1 outputs the FOUTA/FOUTB clock from the FOUTA/FOUTB pin. Setting it to 0 disables output.

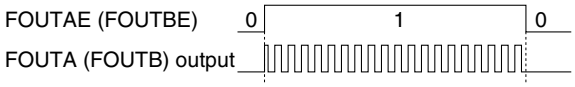


Figure 7.7.2 FOUTA/FOUTB Output

Notes: • Since the FOUTA/FOUTB signal is not synchronized with FOUTAE/FOUTBE writing, switching output on or off will generate certain hazards.

• There may be a time lag between setting FOUTAE/FOUTBE to 1 and start of FOUTA/FOUTB signal output due to the oscillation stabilization wait time and other conditions.

7.8 Control Register Details

Table 7.8.1 List of CLG Registers

Address	Register name		Function
0x5060	CLG_SRC	Clock Source Select Register	Selects the clock source.
0x5061	CLG_CTL	Oscillation Control Register	Controls oscillation.
0x5064	CLG_FOUTA	FOUTA Control Register	Controls FOUTA clock output.
0x5065	CLG_FOUTB	FOUTB Control Register	Controls FOUTB clock output.

Address	Register name		Function
0x507d	CLG_WAIT	Oscillation Stabilization Wait Control Register	Controls oscillation stabilization waiting time.
0x5080	CLG_PCLK	PCLK Control Register	Controls the PCLK supply.
0x5081	CLG_CCLK	CCLK Control Register	Configures the CCLK division ratio.

The CLG module registers are described in detail below.

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

Clock Source Select Register (CLG_SRC)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Clock Source Select Register (CLG_SRC)	**0x5060** (8 bits)	D7–6	**OSC3B FSEL[1:0]**	OSC3B frequency select	OSC3BFSEL[1:0] Frequency 0x3 reserved 0x2 500 kHz 0x1 1 MHz 0x0 2 MHz	0x0	R/W	
		D5	**–**	reserved	–	–	–	0 when being read.
		D4	**OSC1SEL**	OSC1 source select	1 OSC1B 0 OSC1A	1	R/W	
		D3–2	**–**	reserved	–	–	–	0 when being read.
		D1–0	**CLKSRC[1:0]**	System clock source select	CLKSRC[1:0] Clock source 0x3 reserved 0x2 OSC3A 0x1 OSC1 0x0 OSC3B	0x0	R/W	

D[7:6] OSC3BFSEL[1:0]: OSC3B Frequency Select Bits

Selects the OSC3B oscillation frequency.

Table 7.8.2 OSC3B Oscillation Frequency Setting

OSC3BFSEL[1:0]	OSC3B oscillation frequency (typ.)
0x3	Reserved
0x2	500 kHz
0x1	1 MHz
0x0	2 MHz

(Default: 0x0)

D5 Reserved

D4 OSC1SEL: OSC1 Source Select Bit

Selects the OSC1 clock source.
1 (R/W): OSC1B (default)
0 (R/W): OSC1A

D[3:2] Reserved

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

Selects the system clock source.

Table 7.8.3 System Clock Selection

CLKSRC[1:0]	System clock source
0x3	Reserved
0x2	OSC3A
0x1	OSC1
0x0	OSC3B

(Default: 0x0)

Select OSC3B or OSC3A for normal (high-speed) operations. If no high-speed clock is required, OSC1 can be set as the system clock and OSC3B and OSC3A stopped to reduce current consumption.

Notes: • The oscillator to be used as the system clock source must be operated before switching the system clock. Otherwise, the CLG will not switch the system clock source, even if CLKSRC[1:0] is written to, and the CLKSRC[1:0] value will remain unchanged.
The table below lists the combinations of clock operating status and register settings enabling system clock selection.

Table 7.8.4 System Clock Switching Conditions

OSC3BEN	OSC3AEN	OSC1EN	System clock
1	1	1	OSC3B, OSC3A, or OSC1
1	1	0	OSC3B or OSC3A
1	0	1	OSC3B or OSC1
0	1	1	OSC3A or OSC1

- The oscillator circuit selected as the system clock source cannot be turned off.
- Continuous write/read access to CLKSRC[1:0] is prohibited. At least one instruction unrelated to CLKSRC[1:0] access must be inserted between the write and read instructions.
- When SLEEP mode is canceled, the OSC3B oscillator circuit is turned on (OSC3BEN = 1) and is used as the system clock source (CLKSRC[1:0] = 0x0) regardless of the system clock configured before the chip entered SLEEP mode.
  Canceling HALT mode does not change the clock status configured before the chip entered HALT mode.

Oscillation Control Register (CLG_CTL)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Oscillation Control Register (CLG_CTL)	0x5061 (8 bits)	D7–3	–	reserved	–				–	–	0 when being read.
		D2	OSC3BEN	OSC3B enable	1	Enable	0	Disable	1	R/W	
		D1	OSC1EN	OSC1 enable	1	Enable	0	Disable	0	R/W	
		D0	OSC3AEN	OSC3A enable	1	Enable	0	Disable	0	R/W	

D[7:3] Reserved

D2 OSC3BEN: OSC3B Enable Bit

Enables or disables OSC3B oscillator operations.
1 (R/W): Enabled (on) (default)
0 (R/W): Disabled (off)

Note: The OSC3B oscillator cannot be stopped if the OSC3B clock is being used as the system clock.

D1 OSC1EN: OSC1 Enable Bit

Enables or disables OSC1 oscillator operations.
1 (R/W): Enabled (on)
0 (R/W): Disabled (off) (default)

Notes:
- Be sure to select the OSC1 clock source (OSC1A or OSC1B) using OSC1SEL/CLG_SRC register before starting OSC1 oscillation.
- The OSC1 oscillator cannot be stopped if the OSC1 clock is being used as the system clock.

D0 OSC3AEN: OSC3A Enable Bit

Enables or disables OSC3A oscillator operations.
1 (R/W): Enabled (on)
0 (R/W): Disabled (off) (default)

Note: The OSC3A oscillator cannot be stopped if the OSC3A clock is being used as the system clock.

FOUTA Control Register (CLG_FOUTA)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
FOUTA Control Register (CLG_FOUTA)	0x5064 (8 bits)	D7	–	reserved	–		–	–	0 when being read.
		D6–4	FOUTAD [2:0]	FOUTA clock division ratio select	FOUTAD[2:0]	Division ratio	0x0	R/W	
					0x7	1/128			
					0x6	1/64			
					0x5	1/32			
					0x4	1/16			
					0x3	1/8			
					0x2	1/4			
					0x1	1/2			
					0x0	1/1			
		D3–2	FOUTASRC [1:0]	FOUTA clock source select	FOUTASRC[1:0]	Clock source	0x0	R/W	
					0x3	reserved			
					0x2	OSC3A			
					0x1	OSC1			
					0x0	OSC3B			
		D1	–	reserved	–		–	–	0 when being read.
		D0	FOUTAE	FOUTA output enable	1 Enable	0 Disable	0	R/W	

D7 Reserved

D[6:4] FOUTAD[2:0]: FOUTA Clock Division Ratio Select Bits

Selects the source clock division ratio to set the FOUTA clock frequency.

Table 7.8.5 Clock Division Ratio Selection

FOUTAD[2:0]	Division ratio
0x7	1/128
0x6	1/64
0x5	1/32
0x4	1/16
0x3	1/8
0x2	1/4
0x1	1/2
0x0	1/1

(Default: 0x0)

D[3:2] FOUTASRC[1:0]: FOUTA Clock Source Select Bits

Selects the FOUTA clock source.

Table 7.8.6 FOUTA Clock Source Selection

FOUTASRC[1:0]	Clock source
0x3	Reserved
0x2	OSC3A
0x1	OSC1
0x0	OSC3B

(Default: 0x0)

D1 Reserved

D0 FOUTAE: FOUTA Output Enable Bit

Enables or disables FOUTA clock external output.

1 (R/W): Enabled (on)

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

Setting FOUTAE to 1 outputs the FOUTA clock from the FOUTA pin. Setting it to 0 stops the output.

FOUTB Control Register (CLG_FOUTB)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
FOUTB Control Register (CLG_FOUTB)	0x5065 (8 bits)	D7	–	reserved	–	–	–	0 when being read.
		D6–4	FOUTBD [2:0]	FOUTB clock division ratio select	FOUTBD[2:0] / Division ratio: 0x7 1/128; 0x6 1/64; 0x5 1/32; 0x4 1/16; 0x3 1/8; 0x2 1/4; 0x1 1/2; 0x0 1/1	0x0	R/W	
		D3–2	FOUTBSRC [1:0]	FOUTB clock source select	FOUTBSRC[1:0] / Clock source: 0x3 reserved; 0x2 OSC3A; 0x1 OSC1; 0x0 OSC3B	0x0	R/W	
		D1	–	reserved	–	–	–	0 when being read.
		D0	FOUTBE	FOUTB output enable	1 Enable; 0 Disable	0	R/W	

D7 Reserved

D[6:4] FOUTBD[2:0]: FOUTB Clock Division Ratio Select Bits

Selects the source clock division ratio to set the FOUTB clock frequency.

Table 7.8.7 Clock Division Ratio Selection

FOUTBD[2:0]	Division ratio
0x7	1/128
0x6	1/64
0x5	1/32
0x4	1/16
0x3	1/8
0x2	1/4
0x1	1/2
0x0	1/1

(Default: 0x0)

D[3:2] FOUTBSRC[1:0]: FOUTB Clock Source Select Bits

Selects the FOUTB clock source.

Table 7.8.8 FOUTB Clock Source Selection

FOUTBSRC[1:0]	Clock source
0x3	Reserved
0x2	OSC3A
0x1	OSC1
0x0	OSC3B

(Default: 0x0)

D1 Reserved

D0 FOUTBE: FOUTB Output Enable Bit

Enables or disables FOUTB clock external output.

1 (R/W): Enabled (on)

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

Setting FOUTBE to 1 outputs the FOUTB clock from the FOUTB pin. Setting it to 0 stops the output.

Oscillation Stabilization Wait Control Register (CLG_WAIT)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
Oscillation Stabilization Wait Control Register (CLG_WAIT)	0x507d (8 bits)	D7–6	OSC3BWT [1:0]	OSC3B stabilization wait cycle select	OSC3BWT[1:0]	Wait cycle	0x0	R/W	
					0x3	8 cycles			
					0x2	16 cycles			
					0x1	32 cycles			
					0x0	64 cycles			
		D5–4	OSC3AWT [1:0]	OSC3A stabilization wait cycle select	OSC3AWT[1:0]	Wait cycle	0x0	R/W	
					0x3	128 cycles			
					0x2	256 cycles			
					0x1	512 cycles			
					0x0	1024 cycles			
		D3–2	OSC1BWT [1:0]	OSC1B stabilization wait cycle select	OSC1BWT[1:0]	Wait cycle	0x0	R/W	
					0x3	8 cycles			
					0x2	16 cycles			
					0x1	32 cycles			
					0x0	64 cycles			
		D1–0	OSC1AWT [1:0]	OSC1A stabilization wait cycle select	OSC1AWT[1:0]	Wait cycle	0x0	R/W	
					0x3	2048 cycles			
					0x2	4096 cycles			
					0x1	8192 cycles			
					0x0	16384 cycles			

D[7:6] OSC3BWT[1:0]: OSC3B Stabilization Wait Cycle Select Bits

An oscillation stabilization wait time is set to prevent malfunctions due to unstable clock operations at the start of OSC3B oscillation. The OSC3B clock is not supplied to the system immediately after OSC3B oscillation starts until the time set here has elapsed.

Table 7.8.9 OSC3B Oscillation Stabilization Wait Time Settings

OSC3BWT[1:0]	Oscillation stabilization wait time
0x3	8 cycles
0x2	16 cycles
0x1	32 cycles
0x0	64 cycles

(Default: 0x0)

This is set to 64 cycles (OSC3B clock) after an initial reset. This means the CPU can start operating when the CPU operation start time at initial reset indicated below (at a maximum) has elapsed after the reset state is canceled.

CPU operation start time at initial reset ≤ OSC3B oscillation start time (max.) + OSC3B oscillation stabilization wait time (64 cycles)

When the system clock is switched to OSC3B immediately after turning the OSC3B oscillator on, the OSC3B clock is supplied to the system after the OSC3B clock system supply wait time indicated below (at a maximum) has elapsed. If the power supply voltage V_{DD} has stabilized sufficiently, OSC3BWT[1:0] can be set to 0x3 to reduce the oscillation stabilization wait time.

OSC3B clock system supply wait time ≤ OSC3B oscillation start time (max.) + OSC3B oscillation stabilization wait time

D[5:4] OSC3AWT[1:0]: OSC3A Stabilization Wait Cycle Select Bits

An oscillation stabilization wait time is set to prevent malfunctions due to unstable clock operation at the start of OSC3A oscillation. The OSC3A clock is not supplied to the system immediately after OSC3A oscillation starts until the time set here has elapsed.

Table 7.8.10 OSC3A Oscillation Stabilization Wait Time Settings

OSC3AWT[1:0]	Oscillation stabilization wait time
0x3	128 cycles
0x2	256 cycles
0x1	512 cycles
0x0	1024 cycles

(Default: 0x0)

This is set to 1,024 cycles (OSC3A clock) after an initial reset.
When the system clock is switched to OSC3A immediately after the OSC3A oscillator circuit is turned on, the OSC3A clock is supplied to the system after the OSC3A clock system supply wait time indicated below (at a maximum) has elapsed.

OSC3A clock system supply wait time ≤ OSC3A oscillation start time (max.) + OSC3A oscillation stabilization wait time

Note: Oscillation stability will vary, depending on the resonator and other external components. Carefully consider the OSC3A oscillation stabilization wait time before reducing the time.

D[3:2] OSC1BWT[1:0]: OSC1B Stabilization Wait Cycle Select Bits

An oscillation stabilization wait time is set to prevent malfunctions due to unstable clock operation at the start of OSC1B oscillation. The OSC1 clock is not supplied to the system immediately after OSC1B oscillation starts until the time set here has elapsed.

Table 7.8.11 OSC1B Oscillation Stabilization Wait Time Settings

OSC1BWT[1:0]	Oscillation stabilization wait time
0x3	8 cycles
0x2	16 cycles
0x1	32 cycles
0x0	64 cycles

(Default: 0x0)

This is set to 64 cycles (OSC1 clock) after an initial reset.
When the system clock is switched to OSC1 immediately after the OSC1B oscillator circuit is turned on, the OSC1 clock is supplied to the system after the OSC1 clock system supply wait time indicated below (at a maximum) has elapsed.

OSC1 clock system supply wait time ≤ OSC1B oscillation start time (max.) + OSC1B oscillation stabilization wait time

D[1:0] OSC1AWT[1:0]: OSC1A Stabilization Wait Cycle Select Bits

An oscillation stabilization wait time is set to prevent malfunctions due to unstable clock operation at the start of OSC1A oscillation. The OSC1 clock is not supplied to the system immediately after OSC1A oscillation starts until the time set here has elapsed.

Table 7.8.12 OSC1A Oscillation Stabilization Wait Time Settings

OSC1AWT[1:0]	Oscillation stabilization wait time
0x3	2048 cycles
0x2	4096 cycles
0x1	8192 cycles
0x0	16384 cycles

(Default: 0x0)

This is set to 16384 cycles (OSC1 clock) after an initial reset.
When the system clock is switched to OSC1 immediately after the OSC1A oscillator circuit is turned on, the OSC1 clock is supplied to the system after the OSC1 clock system supply wait time indicated below (at a maximum) has elapsed.

OSC1 clock system supply wait time ≤ OSC1A oscillation start time (max.) + OSC1A oscillation stabilization wait time

Note: Oscillation stability will vary, depending on the resonator and other external components. Carefully consider the OSC1A oscillation stabilization wait time before reducing the time.

PCLK Control Register (CLG_PCLK)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
PCLK Control Register (CLG_PCLK)	0x5080 (8 bits)	D7–2	–	reserved	–		–	–	0 when being read.
		D1–0	PCKEN[1:0]	PCLK enable	PCKEN[1:0]	PCLK supply	0x3	R/W	
					0x3	Enable			
					0x2	Not allowed			
					0x1	Not allowed			
					0x0	Disable			

D[7:2] Reserved

D[1:0] PCKEN[1:0]: PCLK Enable Bits

Enables or disables clock (PCLK) supply to the internal peripheral modules.

Table 7.8.13 PCLK Control

PCKEN[1:0]	PCLK supply
0x3	Enabled (on)
0x2	Setting prohibited
0x1	Setting prohibited
0x0	Disabled (off)

(Default: 0x3)

The PCKEN[1:0] default setting is 0x3, which enables clock supply.

Peripheral modules that use PCLK

- Interrupt controller
- 8-bit timer Ch.0 and Ch.1
- SPI Ch.0
- I^2C master
- I^2C slave
- Power generator
- P port & port MUX
- MISC registers

The PCLK supply cannot be disabled if one or more peripheral modules in these list must be operated. The PCLK supply can be disabled if all the peripheral circuits in these list can be stopped. Stop the PCLK supply to reduce current consumption if all the peripheral modules listed above are not required.

Peripheral modules/functions that do not use PCLK

- Real-time clock
- Clock timer
- Stopwatch timer
- Watchdog timer
- EPD controller/driver
- Sound generator
- SVD circuit
- Temperature detection circuit
- R/F converter
- 16-bit PWM timer Ch.0 and Ch.1
- UART Ch.0
- FOUTA/FOUTB outputs

These peripheral modules/functions can operate even if PCLK is stopped.

Note: Do not set PCKEN[1:0] to 0x2 or 0x1, since doing so will stop the operation of certain peripheral modules.

CCLK Control Register (CLG_CCLK)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
CCLK Control Register (CLG_CCLK)	**0x5081** (8 bits)	D7–2	**–**	reserved	–		–	–	0 when being read.
		D1–0	**CCLKGR[1:0]**	CCLK clock gear ratio select	CCLKGR[1:0]	Gear ratio	0x0	R/W	
					0x3	1/8			
					0x2	1/4			
					0x1	1/2			
					0x0	1/1			

D[7:2] Reserved

D[1:0] CCLKGR[1:0]: CCLK Clock Gear Ratio Select Bits

Selects the gear ratio for reducing system clock speed and sets the CCLK clock speed for operating the S1C17 Core. To reduce current consumption, operate the S1C17 Core using the slowest possible clock speed.

Table 7.8.14 CCLK Gear Ratio Selection

CCLKGR[1:0]	Gear ratio
0x3	1/8
0x2	1/4
0x1	1/2
0x0	1/1

(Default: 0x0)

8 Theoretical Regulation (TR)

8.1 TR Module Overview

The S1C17F57 has a theoretical regulation function that theoretically corrects time clock errors due to deviation in oscillation frequencies.

- Adjusts the OSC1A clock (32.768 kHz Typ.) (Note that the OSC1B clock cannot be adjusted.)
- Adjustable range: -15/32768 to +16/32768 [second] in a correction operation
- Peripheral modules that use the regulated clock (F256)
  1. Real-time clock (RTC)
  2. Clock timer (CT)
  3. Stopwatch timer (SWT)
  4. Watchdog timer (WDT)
  5. 16-bit PWM timer (T16A2) ∗ Only when F256 is selected as the count clock
- Software can execute theoretical regulation at any time

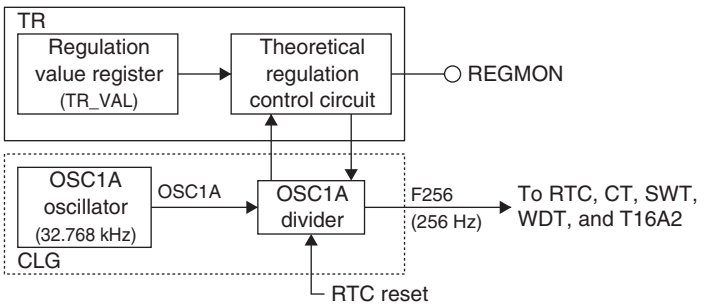


Figure 8.1.1 TR Module Configuration

8.2 TR Output Pin

Table 8.2.1 shows the TR output pin.

Table 8.2.1 TR Output Pin

Pin name	I/O	Qty	Function
REGMON	O	1	Theoretical regulation monitor output pin This pin outputs a regulated clock (F256 (256 Hz) or F1 (1 Hz)) for monitoring the theoretical regulation results.

The TR output pin (REGMON) is shared with an I/O port and is initially set as a general-purpose I/O port pin. The pin function must be switched using the port function select bit to use the general-purpose I/O port pin as the TR output pin.
For detailed information on pin function switching, see the "I/O Ports (P)" chapter.

8.3 Theoretical Regulation Control

8.3.1 Setting Regulation Values

The correction value (-15/32768 to +16/32768) for theoretical regulation is specified using TRIM[4:0]/TR_VAL register.

Table 8.3.1.1 Regulation Value Settings

TRIM[4:0]	Amount of correction/ one adjustment (n/32768)	Rate * (Seconds/Day)	TRIM[4:0]	Amount of correction/ one adjustment (n/32768)	Rate * (Seconds/Day)
0x10	-15	+3.955	0x00	+1	-0.264
0x11	-14	+3.691	0x01	+2	-0.527
0x12	-13	+3.428	0x02	+3	-0.791
0x13	-12	+3.164	0x03	+4	-1.055
0x14	-11	+2.900	0x04	+5	-1.318
0x15	-10	+2.637	0x05	+6	-1.582
0x16	-9	+2.373	0x06	+7	-1.846
0x17	-8	+2.109	0x07	+8	-2.109
0x18	-7	+1.846	0x08	+9	-2.373
0x19	-6	+1.582	0x09	+10	-2.637
0x1a	-5	+1.318	0x0a	+11	-2.900
0x1b	-4	+1.055	0x0b	+12	-3.164
0x1c	-3	+0.791	0x0c	+13	-3.428
0x1d	-2	+0.527	0x0d	+14	-3.691
0x1e	-1	+0.264	0x0e	+15	-3.955
0x1f	0	0	0x0f	+16	-4.219

* Rates when theoretical regulation is executed in 10-second cycles (Default: 0x0)

Addresses 0xfffa to 0xfffb in the Flash memory are reserved for storing the correction value. The correction value should be programmed in this area by the user and use it for setting TRIM[4:0]. The IC will be shipped with this area emptied, therefore, do not place any program code or data in these addresses.

8.3.2 Executing Theoretical Regulation

Writing 1 to REGTRIG/TR_CTL register starts theoretical regulation that is performed in the OSC1A clock (32.768 kHz) divider. This operation extends or reduces the cycle time of the 256 Hz clock output by the OSC1A divider for the regulation value specified by TRIM[4:0]. Theoretical regulation is performed only once by writing 1 to REGTRIG. To perform theoretical regulation periodically, use a timer interrupt handler to write 1 to REGTRIG.
Note that a maximum 16.6 ms of delay occurs before theoretical regulation actually starts after writing to REGTRIG. Writing 1 to REGTRIG in this period is ineffective, so to write 1 to REGTRIG successively, an interval at least 16.6 ms is necessary between writings.
The regulated clock (F256) will be supplied to the OSC1 peripheral circuits such as the clock timer.

Note: Use an interrupt from a peripheral timer module that runs with the regulated clock (F256) to execute theoretical regulation. An interrupt from the timer that runs all the time should be used to reduce current consumption.

8.3.3 Regulated Clock External Monitor

Either the 256 Hz (F256) or 1 Hz (F1) regulated clock can be output from the REGMON pin for monitoring. RCLKFSEL/TR_CTL register is used to select the clock to be monitored from F256 and F1. When RCLKFSEL is 0 (default), F256 is selected; when RCLKFSEL is set to 1, F1 is selected.
The selected clock is output from the REGMON pin by setting RCLKMON to 1. Setting RCLKMON to 0 stops the clock output and the REGMON pin goes low (Vss) level.

Notes:
- Before the 256 Hz regulated clock can be monitored, any of the clock timer (CT), stopwatch timer (SWT), or watchdog timer (WDT) must be turned on. Or turn the 16-bit PWM timer (T16A2) on after F256 (256 Hz regulated clock) is selected as its clock.
- Before the 1 Hz regulated clock can be monitored, the real-time clock (RTC) must be turned on.

8.4 Control Register Details

Table 8.4.1 List of TR Registers

Address	Register name		Function
0x5078	TR_CTL	TR Control Register	Controls theoretical regulation.
0x5079	TR_VAL	TR Value Register	Sets a regulation value.

The TR module registers are described in detail below.

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

TR Control Register (TR_CTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
TR Control Register (TR_CTL)	0x5078 (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.
		D3	RCLKFSEL	Monitor clock frequency select	1 1 Hz 0 256 Hz	0	R/W	
		D2	RCLKMON	Regulated clock monitor enable	1 Enable 0 Disable	0	R/W	
		D1	–	reserved	–	–	–	0 when being read.
		D0	REGTRIG	Regulation trigger	1 Trigger 0 Ignored	0	W	

D[7:4] Reserved

D3 RCLKFSEL: Monitor Clock Frequency Select Bit

Selects the regulated clock to be output from the REGMON pin for monitoring.

1 (R/W): F1 (1 Hz)

0 (R/W): F256 (256 Hz) (default)

D2 RCLKMON: Regulated Clock Monitor Enable Bit

Controls the clock monitor output from the REGMON pin.

1 (R/W): Enabled (On)

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

Setting RCLKMON to 1 outputs the clock selected by RCLKFSEL from the REGMON pin.

D1 Reserved

D0 REGTRIG: Regulation Trigger Bit

Executes theoretical regulation.

1 (W): Trigger

0 (W): Ignored (default)

Theoretical regulation is performed only once by writing 1 to REGTRIG.

Note that a maximum 16.6 ms of delay occurs before theoretical regulation actually starts after writing to REGTRIG. Writing 1 to REGTRIG in this period is ineffective, so to write 1 to REGTRIG successively, an interval at least 16.6 ms is necessary between writings.

TR Value Register (TR_VAL)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
TR Value Register (TR_VAL)	0x5079 (8 bits)	D7–5	–	reserved	–		–	–	0 when being read.
		D4–0	TRIM[4:0]	Regulation value	TRIM[4:0]	Regulation value	0x0	R/W	
					0xf	+16			
					0xe	+15			
					:	:			
					0x1	+2			
					0x0	+1			
					0x1f	0			
					0x1e	-1			
					:	:			
					0x11	-14			
					0x10	-15			

D[7:5] Reserved

D[4:0] TRIM[4:0]: Regulation Value Bits

Specifies the correction value (-15/32768 to +16/32768) for theoretical regulation.

Table 8.4.2 Regulation Value Settings

TRIM[4:0]	Amount of correction/ one adjustment (n/32768)	Rate * (Seconds/Day)	TRIM[4:0]	Amount of correction/ one adjustment (n/32768)	Rate * (Seconds/Day)
0x10	-15	+3.955	0x00	+1	-0.264
0x11	-14	+3.691	0x01	+2	-0.527
0x12	-13	+3.428	0x02	+3	-0.791
0x13	-12	+3.164	0x03	+4	-1.055
0x14	-11	+2.900	0x04	+5	-1.318
0x15	-10	+2.637	0x05	+6	-1.582
0x16	-9	+2.373	0x06	+7	-1.846
0x17	-8	+2.109	0x07	+8	-2.109
0x18	-7	+1.846	0x08	+9	-2.373
0x19	-6	+1.582	0x09	+10	-2.637
0x1a	-5	+1.318	0x0a	+11	-2.900
0x1b	-4	+1.055	0x0b	+12	-3.164
0x1c	-3	+0.791	0x0c	+13	-3.428
0x1d	-2	+0.527	0x0d	+14	-3.691
0x1e	-1	+0.264	0x0e	+15	-3.955
0x1f	0	0	0x0f	+16	-4.219

* Rates when theoretical regulation is executed in 10-second cycles (Default: 0x0)

9 Real-Time Clock (RTC)

9.1 RTC Module Overview

The S1C17F57 incorporates a real-time clock (RTC).
The main features of the RTC are outlined below.

- Contains time counters (seconds, minutes, and hours).
- The RTC operates with the OSC1A oscillator circuit even in SLEEP mode.
- Either binary or BCD data can be read from and written to the counters.
- Capable of controlling the starting and stopping of time clocks.
- 24-hour or 12-hour mode can be selected.
- Periodic interrupts (32 Hz, 8 Hz, 4 Hz, 1 Hz, 10 second, 1 minute, 10 minutes, 1 hour, 10 hours, half-day, and 1 day) are possible.

Figure 9.1.1 shows a block diagram of the RTC.

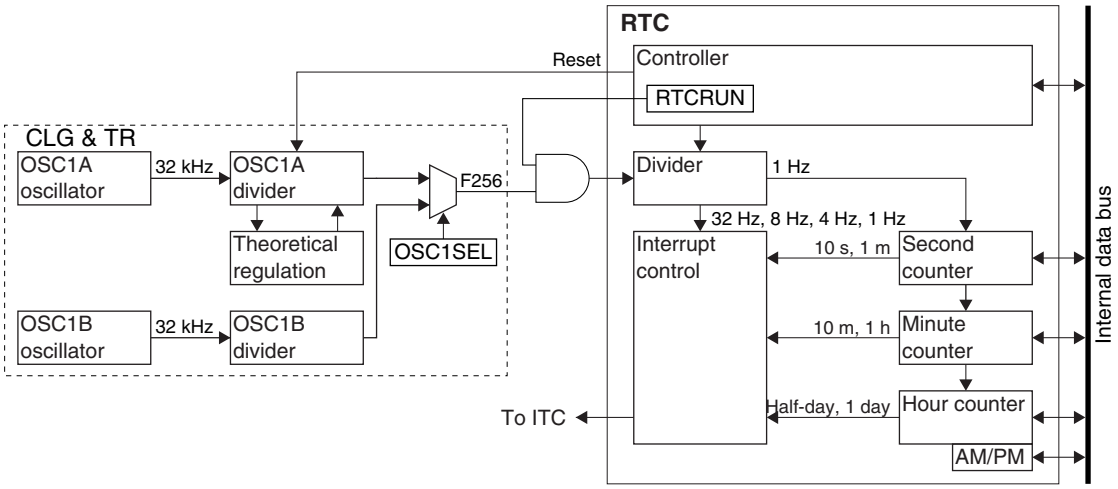


Figure 9.1.1 RTC Block Diagram

9.2 RTC Counters

The RTC contains the following three counters, whose count values can be read out as either binary or BCD data from the respective registers. Each counter can also be set to any desired time by writing data to the respective register.

Second counter

This 7-bit binary counter counts from 0 to 59 seconds synchronously with the 1 Hz signal derived from the divider. This counter is also used as a 3-bit (0 to 5) + 4-bit (0 to 9) BCD counter by setting BCDMD/RTC_CTL register to 1. The counter data can be read/written using RTCSEC[6:0]/RTC_MS register. The counter is reset to 0 when it reaches 60 seconds and outputs a carry over of 1 to the minute counter.

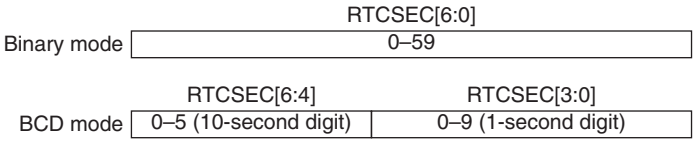


Figure 9.2.1 Second Counter

Minute counter

This 7-bit binary counter counts from 0 to 59 minutes with 1 carried over from the second counter. This counter is also used as a 3-bit (0 to 5) + 4-bit (0 to 9) BCD counter by setting BCDMD/RTC_CTL register to 1. The counter data can be read/written using RTCMIN[6:0]/RTC_MS register. The counter is reset to 0 when it reaches 60 minutes and outputs a carry over of 1 to the hour counter.

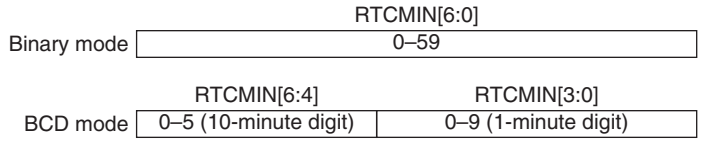


Figure 9.2.2 Minute Counter

Hour counter

This 6-bit binary counter counts from 0 to 23 o'clock (24-hour mode) or from 1 to 12 o'clock (12-hour mode) with 1 carried over from the minute counter. This counter is also used as a 2-bit (0 to 2 or 0 to 1) + 4-bit (0 to 9) BCD counter by setting BCDMD/RTC_CTL register to 1. The counter data can be read/written using RTCHOUR[5:0]/RTC_H register.

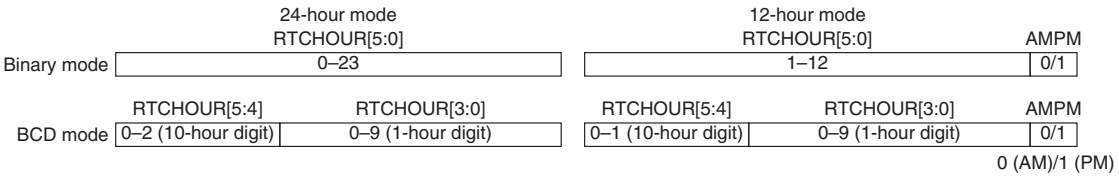


Figure 9.2.3 Hour Counter

Table 9.2.1 Hour Counter Values

Time	24-hour mode		12-hour mode		
	RTCHOUR[5:0] (binary)	RTCHOUR[5:0] (BCD)	RTCHOUR[5:0] (binary)	RTCHOUR[5:0] (BCD)	AMPM
0 o'clock (12am)	0x0	0x00	0xc	0x12	0
1 o'clock (1am)	0x1	0x01	0x1	0x01	0
2 o'clock (2am)	0x2	0x02	0x2	0x02	0
3 o'clock (3am)	0x3	0x03	0x3	0x03	0
4 o'clock (4am)	0x4	0x04	0x4	0x04	0
5 o'clock (5am)	0x5	0x05	0x5	0x05	0
6 o'clock (6am)	0x6	0x06	0x6	0x06	0
7 o'clock (7am)	0x7	0x07	0x7	0x07	0
8 o'clock (8am)	0x8	0x08	0x8	0x08	0
9 o'clock (9am)	0x9	0x09	0x9	0x09	0
10 o'clock (10am)	0xa	0x10	0xa	0x10	0
11 o'clock (11am)	0xb	0x11	0xb	0x11	0
12 o'clock (12pm)	0xc	0x12	0xc	0x12	1
13 o'clock (1pm)	0xd	0x13	0x1	0x01	1
14 o'clock (2pm)	0xe	0x14	0x2	0x02	1
15 o'clock (3pm)	0xf	0x15	0x3	0x03	1
16 o'clock (4pm)	0x10	0x16	0x4	0x04	1
17 o'clock (5pm)	0x11	0x17	0x5	0x05	1
18 o'clock (6pm)	0x12	0x18	0x6	0x06	1
19 o'clock (7pm)	0x13	0x19	0x7	0x07	1
20 o'clock (8pm)	0x14	0x20	0x8	0x08	1
21 o'clock (9pm)	0x15	0x21	0x9	0x09	1
22 o'clock (10pm)	0x16	0x22	0xa	0x10	1
23 o'clock (11pm)	0x17	0x23	0xb	0x11	1

Initial counter values

An initial reset does not initialize the counter values. Be sure to initialize the counters via software.

9.3 RTC Control

9.3.1 Operating Clock Control

The RTC module uses the 256 Hz clock output by the CLG module as the operation clock (normally, RTC is clocked by the F256 clock (regulated 256 Hz clock) derived from the OSC1A divider). Therefore, the OSC1 oscillator must be turned on before starting the RTC. However, the clock is not supplied to the RTC module while RTC is stopped even if the OSC1 oscillator is on. For detailed information on clock control, see the "Clock Generator (CLG)" and "Theoretical Regulation (TR)" chapters.

Notes:
- The RTC module input clock frequency is 256 Hz only when the OSC1 clock frequency is 32.768 kHz. The frequency described in this chapter will vary accordingly for other OSC1 clock frequencies.
- The RTC module can also be operated with the OSC1B divider clock (about 256 Hz) even if OSC1B is selected as the OSC1 clock source in the CLG. However, the RTC cannot be used as an accurate clock.
- The OSC1A divider is reset when the RTC starts running (when 1 is written to RTCRUN/RTC_CTL register). This affects the count operations of the timer modules (CT, SWT, WDT, and T16A2), as new 256 Hz cycle begins from that point.
- After an initial reset, RTCRUN is set to 0 and the RTC idles. The OSC1 oscillator circuit is also idle. Therefore, resetting the IC suspends the RTC operation for the period shown below.

  RTC idle time = [#RESET = low period] +
  [OSC3B oscillation stabilization time] +
  [Time until OSC1 is started] +
  [OSC1 oscillation stabilization time] +
  [Time until RTC is restarted]

9.3.2 12-hour/24-hour mode selection

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

RTC24H = 1: 12-hour mode
RTC24H = 0: 24-hour mode

The count range of the hour counter changes with this selection.
Basically, this setting should be changed while the counters are idle. RTC24H is allocated to the same address as the control bits that start the counters. Therefore, 12-hour mode or 24-hour mode can be selected at the same time the counters are started.

Checking A.M./P.M. with 12-hour mode selected

When 12-hour mode is selected, AMPM/RTC_H register that indicates A.M. or P.M. is enabled.
AMPM = 0: A.M.
AMPM = 1: P.M.

For 24-hour mode, AMPM is fixed to 0.
When setting the time of day, write either of the values above to this bit to specify A.M. or P.M.

9.3.3 RTC Start/Stop

The RTC starts counting when RTCRUN/RTC_CTL register is set to 1, and stops counting when this bit is set to 0. The OSC1A divider in the CLG module is reset by writing 1 to RTCRUN and it starts division of the OSC1A clock.

9.3.4 Counter Settings

Counter values should be set in the procedure shown below.

1. Stop the RTC by writing 0 to RTCRUN/RTC_CTL register.
2. Wait until RTCST/RTC_CTL register is reset to 0 (the RTC actually stops operating).

3. Write the counter values to the RTC_MS and RTC_H registers.
4. Start the RTC by writing 1 to RTCRUN/RTC_CTL register.

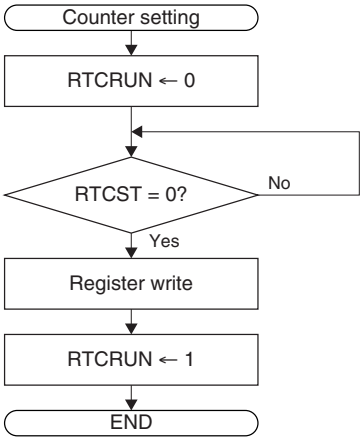


Figure 9.3.4.1 Procedure for Setting Counters

Notes: • Do not set the counters while the RTC is running, as proper settings to the counters cannot be guaranteed.
• Counter values to be set must be within the effective range according to binary/BCD mode. The counter will be undefined if a value out of the range is written.
• Depending on the value set, an interrupt may occur immediately after starting the RTC.

9.3.5 Counter Read

If 1 is being carried over while the counters are being read, correct time may not be read.
Counter values should be read in the procedure shown below.

Read procedure 1

1. Read the RTC_MS and RTC_H registers.
2. Read the RTC_MS and RTC_H registers again.
3. If the same value is read out in Steps 1 and 2, use it as the correct time.
   If different values are read out, try again from Step 1.

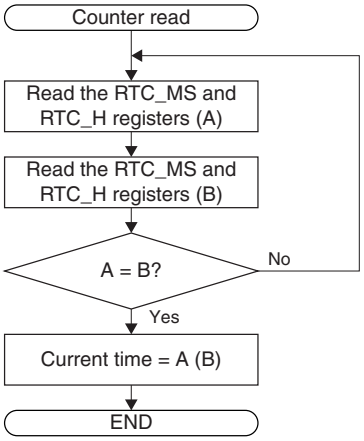


Figure 9.3.5.1 Procedure for Reading Counters

Read procedure 2

After a 1 Hz interrupt (or 10-second to 1 day interrupt) occurs, read the RTC_MS and RTC_H registers within one second.

9.4 RTC Interrupts

The RTC can generate interrupts in 10 different cycles listed in Table 9.4.1. To generate interrupts, set the interrupt enable bits for the interrupt cycles to 1. If an interrupt enable bit is set to 0 (default), interrupt requests for the cause will not be sent to the ITC.

Table 9.4.1 Interrupt Cycles and Interrupt Control Bits

Interrupt cycle	Interrupt timing	Interrupt flag (RTC_IFLG register)	Interrupt enable bit (RTC_IEN register)
One day	Hour counter = 23→0 (24-hour mode) Hour counter = 11pm→12am (12-hour mode)	INT1D	INT1DEN
Half-day	Hour counter = 11→12, 23→0 (24-hour mode) Hour counter = 11am→12pm, 11pm→12am (12-hour mode)	INTHD	INTHDEN
1 hour	Minute counter = 59→0	INT1H	INT1HEN
10 minutes	Minute counter = 9→10, 19→20, 29→30, 39→40, 49→50, 59→0	INT10M	INT10MEN
1 minute	Second counter = 59→0	INT1M	INT1MEN
10 seconds	Second counter = 9→10, 19→20, 29→30, 39→40, 49→50, 59→0	INT10S	INT10SEN
1 Hz	Divider 1 Hz signal cycles	INT1HZ	INT1HZEN
4 Hz	Divider 4 Hz signal cycles	INT4HZ	INT4HZEN
8 Hz	Divider 8 Hz signal cycles	INT8HZ	INT8HZEN
32 Hz	Divider 32 Hz signal cycles	INT32HZ	INT32HZEN

When the interrupt enable bit is set to 1, the corresponding interrupt flag will be set to 1 in the timing shown above and the interrupt request will be sent to the ITC.
Since the RTC is active even in SLEEP mode, RTC interrupt requests may be used to cancel SLEEP mode. For example, the RTC interrupt can be used for executing periodical theoretical regulation processing when the theoretical regulation type OSC1A oscillator is used.

For more information on interrupt processing, see the "Interrupt Controller (ITC)" chapter.

Notes: • To prevent interrupt recurrences, the interrupt flag must be reset in the interrupt handler routine after an RTC interrupt has occurred. The interrupt flag is reset by writing 1.

• To prevent unwanted interrupts, reset the interrupt flags before enabling interrupts with the interrupt enable bits.

9.5 Control Register Details

Table 9.5.1 List of RTC Registers

Address	Register name		Function
0x56c0	RTC_CTL	RTC Control Register	Controls the RTC.
0x56c2	RTC_IEN	RTC Interrupt Enable Register	Enables/disables interrupts.
0x56c4	RTC_IFLG	RTC Interrupt Flag Register	Displays/sets interrupt occurrence status.
0x56c6	RTC_MS	RTC Minute/Second Counter Register	Minute/second counter data
0x56c8	RTC_H	RTC Hour Counter Register	Hour counter data

The following describes each RTC register.

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

RTC Control Register (RTC_CTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RTC Control Register (RTC_CTL)	**0x56c0** (16 bits)	D15–9	–	reserved	–	–	–	0 when being read.
		D8	**RTCST**	RTC run/stop status	1 Running 0 Stop	0	R	
		D7–6	–	reserved	–	–	–	0 when being read.
		D5	**BCDMD**	BCD mode select	1 BCD mode 0 Binary mode	0	R/W	
		D4	**RTC24H**	24H/12H mode select	1 12H 0 24H	0	R/W	
		D3–1	–	reserved	–	–	–	0 when being read.
		D0	**RTCRUN**	RTC run/stop control	1 Run 0 Stop	0	R/W	

D[15:9] Reserved

D8 RTCST: RTC Run/Stop Status Bit

Indicates the RTC operating status.
1 (R): Running
0 (R): Stop (default)

RTCST goes 1 when the RTC starts running by writing 1 to RTCRUN. RTCST reverts to 0 when the count operation is actually stopped after 0 is written to RTCRUN. When setting counter values, write 0 to RTCRUN and make sure that RTCST is reset to 0 before writing data.

D[7:6] Reserved

D5 BCDMD: BCD Mode Select Bit

Sets the second, minute, and hour counters into BCD mode.
1 (R/W): BCD mode
0 (R/W): Binary mode (default)

By default, each counter operates as a binary counter and a binary value is read or written as counter data. Setting BCDMD to 1 configures the counter so that two-digit BCD value can be read or written. See Section 9.2 for the configuration of the counter in each mode.

D4 RTC24H: 24H/12H Mode Select Bit

Selects whether to use the hour counter in 24-hour or 12-hour mode.
1 (R/W): 24-hour mode
0 (R/W): 12-hour mode (default)

The count range of the hour counter changes with this selection. Basically, this setting should be changed while the counters are idle. Since this register is assigned a control bit (D0) to start the counters, 12-hour or 24-hour mode may be selected when starting the counters.

D[3:1] Reserved

D0 RTCRUN: RTC Run/Stop Control Bit

Starts or stops the RTC.
1 (R/W): Start
0 (R/W): Stop (default)

The RTCRUN default setting is 0, which stops the RTC. Setting RTCRUN to 1 enables the CLG to send the clock to the RTC. When RTCRUN is set to 1, the OSC1A oscillator circuit does not stop even if the IC enters SLEEP mode (the OSC1 clock will be supplied to the RTC only).
Writing 1 to RTCRUN resets the OSC1A divider in the CLG module.

RTC Interrupt Enable Register (RTC_IEN)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RTC Interrupt Enable Register (RTC_IEN)	0x56c2 (16 bits)	D15–10	–	reserved	–	–	–	0 when being read.
		D9	INT1DEN	1-day interrupt enable	1 Enable 0 Disable	0	R/W	
		D8	INTHDEN	Half-day interrupt enable	1 Enable 0 Disable	0	R/W	
		D7	INT1HEN	1-hour interrupt enable	1 Enable 0 Disable	0	R/W	
		D6	INT10MEN	10-minute interrupt enable	1 Enable 0 Disable	0	R/W	
		D5	INT1MEN	1-minute interrupt enable	1 Enable 0 Disable	0	R/W	
		D4	INT10SEN	10-second interrupt enable	1 Enable 0 Disable	0	R/W	
		D3	INT1HZEN	1 Hz interrupt enable	1 Enable 0 Disable	0	R/W	
		D2	INT4HZEN	4 Hz interrupt enable	1 Enable 0 Disable	0	R/W	
		D1	INT8HZEN	8 Hz interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	INT32HZEN	32 Hz interrupt enable	1 Enable 0 Disable	0	R/W	

This register is used to enable/disable RTC interrupts. When the interrupt enable bit for an interrupt cycle is set to 1, the corresponding interrupt flag will be set to 1 in the interrupt cycles and the interrupt request will be sent to the ITC. If an interrupt enable bit is set to 0, the interrupt request will not be sent to the ITC.

1 (R/W): Interrupt enabled
0 (R/W): Interrupt disabled (default)

D[15:10] Reserved

D9 INT1DEN: 1-Day Interrupt Enable Bit
Enables or disables 1-day interrupt requests to the ITC.

D8 INTHDEN: Half-Day Interrupt Enable Bit
Enables or disables half-day interrupt requests to the ITC.

D7 INT1HEN: 1-Hour Interrupt Enable Bit
Enables or disables 1-hour interrupt requests to the ITC.

D6 INT10MEN: 10-Minute Interrupt Enable Bit
Enables or disables 10-minute interrupt requests to the ITC.

D5 INT1MEN: 1-Minute Interrupt Enable Bit
Enables or disables 1-minute interrupt requests to the ITC.

D4 INT10SEN: 10-Second Interrupt Enable Bit
Enables or disables 10-second interrupt requests to the ITC.

D3 INT1HZEN: 1 Hz Interrupt Enable Bit
Enables or disables 1 Hz interrupt requests to the ITC.

D2 INT4HZEN: 4 Hz Interrupt Enable Bit
Enables or disables 4 Hz interrupt requests to the ITC.

D1 INT8HZEN: 8 Hz Interrupt Enable Bit
Enables or disables 8 Hz interrupt requests to the ITC.

D0 INT32HZEN: 32 Hz Interrupt Enable Bit
Enables or disables 32 Hz interrupt requests to the ITC.

RTC Interrupt Flag Register (RTC_IFLG)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
RTC Interrupt Flag Register (RTC_IFLG)	0x56c4 (16 bits)	D15–10	–	reserved	–				–	–	0 when being read.
		D9	INT1D	1-day interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D8	INTHD	Half-day interrupt flag					0	R/W	
		D7	INT1H	1-hour interrupt flag					0	R/W	
		D6	INT10M	10-minute interrupt flag					0	R/W	
		D5	INT1M	1-minute interrupt flag					0	R/W	
		D4	INT10S	10-second interrupt flag					0	R/W	
		D3	INT1HZ	1 Hz interrupt flag					0	R/W	
		D2	INT4HZ	4 Hz interrupt flag					0	R/W	
		D1	INT8HZ	8 Hz interrupt flag					0	R/W	
		D0	INT32HZ	32 Hz interrupt flag					0	R/W	

This register indicates RTC interrupt cause occurrence status. When the corresponding interrupt enable bit is set to 1, the interrupt flag will be set to 1 in the interrupt cycles and the interrupt request will be sent to the ITC. The interrupt flags are reset to 0 by writing 1.

1 (R): Cause of interrupt has occurred
0 (R): No cause of interrupt has occurred (default)
1 (W): Flag is reset
0 (W): Ignored

D[15:10] Reserved

D9 INT1D: 1-Day Interrupt Flag Bit
Indicates 1-day interrupt cause occurrence status. INT1D is set to 1 at the same time the hour counter changes from 23 to 0 (in 24-hour mode) or 11pm to 12am (in 12-hour mode).

D8 INTHD: Half-Day Interrupt Flag Bit
Indicates half-day interrupt cause occurrence status. INTHD is set to 1 at the same time the hour counter changes from 11 to 12, 23 to 0 (in 24-hour mode), or 11am to 12pm, 11pm to 12am (in 12-hour mode).

D7 INT1H: 1-Hour Interrupt Flag Bit

Indicates 1-hour interrupt cause occurrence status. INT1H is set to 1 at the same time the minute counter changes from 59 to 0.

D6 INT10M: 10-Minute Interrupt Flag Bit

Indicates 10-minute interrupt cause occurrence status. INT10M is set to 1 at the same time the minute counter changes from 9 to 10, 19 to 20, 29 to 30, 39 to 40, 49 to 50, or 59 to 0.

D5 INT1M: 1-Minute Interrupt Flag Bit

Indicates 1-minute interrupt cause occurrence status. INT1M is set to 1 at the same time the second counter changes from 59 to 0.

D4 INT10S: 10-Second Interrupt Flag Bit

Indicates 10-second interrupt cause occurrence status. INT10S is set to 1 at the same time the second counter changes from 9 to 10, 19 to 20, 29 to 30, 39 to 40, 49 to 50, or 59 to 0.

D3 INT1HZ: 1 Hz Interrupt Flag Bit

Indicates 1 Hz interrupt cause occurrence status. INT1HZ is set to 1 in the divider 1 Hz signal cycles.

D2 INT4HZ: 4 Hz Interrupt Flag Bit

Indicates 4 Hz interrupt cause occurrence status. INT4HZ is set to 1 in the divider 4 Hz signal cycles.

D1 INT8HZ: 8 Hz Interrupt Flag Bit

Indicates 8 Hz interrupt cause occurrence status. INT8HZ is set to 1 in the divider 8 Hz signal cycles.

D0 INT32HZ: 32 Hz Interrupt Flag Bit

Indicates 32 Hz interrupt cause occurrence status. INT32HZ is set to 1 in the divider 32 Hz signal cycles.

RTC Minute/Second Counter Register (RTC_MS)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RTC Minute/Second Counter Register (RTC_MS)	0x56c6 (16 bits)	D15	–	reserved	–	–	–	0 when being read.
		D14–8	RTCMIN [6:0]	Minute counter	0x0 to 0x3b (binary mode) 0x00 to 0x59 (BCD mode)	X	R/W	
		D7	–	reserved	–	–	–	0 when being read.
		D6–0	RTCSEC [6:0]	Second counter	0x0 to 0x3b (binary mode) 0x00 to 0x59 (BCD mode)	X	R/W	

D15 Reserved

D[14:8] RTCMIN[6:0]: Minute Counter Bits

These bits are used to read and write data from/to the minute counter. (Default: undefined)
The effective range of the read/setting values are as follows:

- RTCMIN[6:0] = 0x0 to 0x3b (0 to 59 minutes) in binary mode (BCDMD = 0)
- RTCMIN[6:4] = 0x0 to 0x5 (10-minute digit) and RTCMIN[3:0] = 0x0 to 0x9 (1-minute digit) in BCD mode (BCDMD = 1)

D7 Reserved

D[6:0] RTCSEC[6:0]: Second Counter Bits

These bits are used to read and write data from/to the second counter. (Default: undefined)
The effective range of the read/setting values are as follows:

- RTCSEC[6:0] = 0x0 to 0x3b (0 to 59 seconds) in binary mode (BCDMD = 0)
- RTCSEC[6:4] = 0x0 to 0x5 (10-second digit) and RTCSEC[3:0] = 0x0 to 0x9 (1-second digit) in BCD mode (BCDMD = 1)

Notes: • For the counter read and write procedures, see Section 9.3.5, “Counter Read,” and Section 9.3.4, “Counter Settings.”

• Do not set the counters while the RTC is running, as proper settings to the counters cannot be guaranteed.

- Counter values to be set must be within the effective range according to binary/BCD mode. The counter will be undefined if a value out of the range is written.
- Depending on the value set, an interrupt may occur immediately after starting the RTC.

RTC Hour Counter Register (RTC_H)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RTC Hour Counter Register (RTC_H)	0x56c8 (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7	AMPM	AM/PM	1 PM / 0 AM	X	R/W	
		D6	–	reserved	–	–	–	0 when being read.
		D5–0	RTCHOUR [5:0]	Hour counter	0x0 to 0x17 (binary mode) 0x00 to 0x23 (BCD mode)	X	R/W	

D[15:8] Reserved

D7 AMPM: AM/PM Bit

Indicates A.M. or P.M. when 12-hour mode is selected. (Default: undefined)
1 (R/W): P.M.
0 (R/W): A.M.

This bit is only effective when RTC24H/RTC_CTL register is set to 1 (12-hour mode).
When 24-hour mode is selected, this bit is fixed to 0. In this case, do not write 1 to AMPM.

Note: The AMPM bit will be fixed at 0 immediately after RTC24H/RTC_CTL register is changed from 12-hour mode to 24-hour mode.

D6 Reserved

D[5:0] RTCHOUR[5:0]: Hour Counter Bits

These bits are used to read and write data from/to the hour counter. (Default: undefined)
The effective range of the read/setting values in binary mode (BCDMD = 0) are as follows:

- RTCHOUR[5:0] = 0x0 to 0x17 (0 to 23 o'clock) in 24-hour mode
- RTCHOUR[5:0] = 0x1 to 0xc (1 to 12 o'clock) in 12-hour mode

The effective range of the read/setting values in BCD mode (BCDMD = 1) are as follows:

- RTCHOUR[5:4] = 0x0 to 0x2 (10-hour digit) and RTCHOUR[3:0] = 0x0 to 0x9 (1-hour digit) in 24-hour mode
- RTCHOUR[5:4] = 0x0 to 0x1 (10-hour digit) and RTCHOUR[3:0] = 0x0 to 0x9 (1-hour digit) in 12-hour mode

Notes:
- For the counter read and write procedures, see Section 9.3.5, "Counter Read," and Section 9.3.4, "Counter Settings."
- Do not set the counters while the RTC is running, as proper settings to the counters cannot be guaranteed.
- Counter values to be set must be within the effective range according to binary/BCD mode. The counter will be undefined if a value out of the range is written.
- Depending on the value set, an interrupt may occur immediately after starting the RTC.

10 I/O Ports (P)

10.1 P Module Overview

The P ports are general-purpose digital inputs/outputs that allow software to control the input/output direction and pull-up resistor. These ports are shared with internal peripheral module inputs/outputs, and the pin functions can be switched by setting the registers. A number of port groups can generate interrupts caused by a transition of the input signal.
The following shows the features of the P module:

- Maximum 29 I/O ports (P0[7:0], P1[7:0], P2[7:0], P3[4:0]) are available.
  * The number of ports for general-purpose use depends on the peripheral functions used.
- Each port has a pull-up resistor that can be enabled with software.
- Input interface level: CMOS Schmitt
- The P0 and P2 ports can generate input interrupts at the signal edge selected with software.
- The P0 and P2 ports include a chattering filter.
- Can generate an initial reset by entering low level simultaneously to the P0 ports selected with software.
- All port provide a port function select bit to configure the pin function (for GPIO or peripheral functions).

Figure 10.1.1 shows the I/O port configuration.

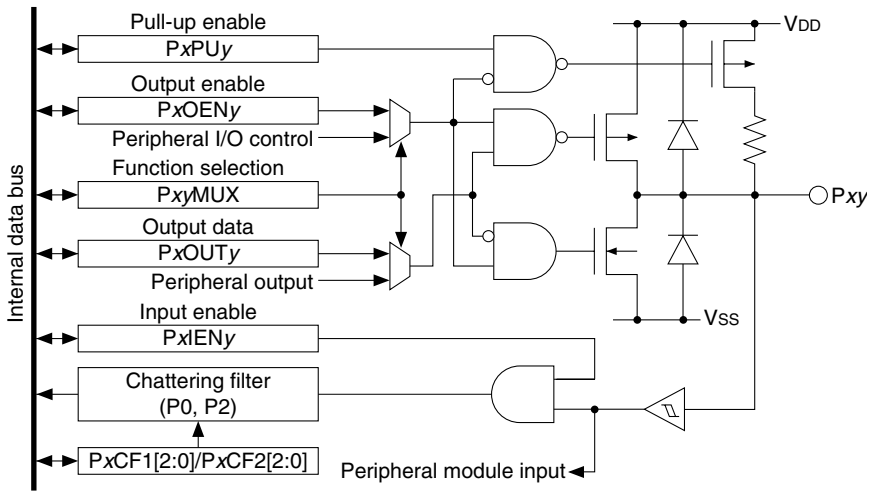


Figure 10.1.1 I/O Port Configuration

Notes: • The PCLK clock must be supplied from the clock generator to access the I/O port. The PCLK clock is also needed to operate the P0 and P2 chattering filters.

• The "*xy*" in the register and bit names refers to the port number (P*xy*, *x* = 0 to 3, *y* = 0 to 7).
Example: PxINy/Px_IN register
P00: P0IN0/P0_IN register
P17: P1IN7/P1_IN register

10.2 Input/Output Pin Function Selection (Port MUX)

The I/O port pins share peripheral module input/output pins. Each pin can be configured for use as an I/O port or for a peripheral module function via the corresponding port function-select bits. Pins not used for peripheral modules can be used as general-purpose I/O ports.

Table 10.2.1 Input/Output Pin Function Selection

Pin function 1 P*xy*MUX[1:0] = 0x0	Pin function 2 P*xy*MUX[1:0] = 0x1	Pin function 3 P*xy*MUX[1:0] = 0x2	Pin function 4 P*xy*MUX[1:0] = 0x3	Port function select bits
P00	RFCLKO (RFC)	REGMON (TR)	–	P00MUX[1:0]/P00_03PMUX register
P01/EXCL0 (T16A2)	–	–	–	P01MUX[1:0]/P00_03PMUX register
P02	TOUTA0/CAPA0 (T16A2)	–	–	P02MUX[1:0]/P00_03PMUX register
P03	TOUTB0/CAPB0 (T16A2)	–	–	P03MUX[1:0]/P00_03PMUX register
P04/EXCL1 (T16A2)	–	–	–	P04MUX[1:0]/P04_07PMUX register
P05	TOUTA1/CAPA1 (T16A2)	–	–	P05MUX[1:0]/P04_07PMUX register
P06	TOUTB1/CAPB1 (T16A2)	#BZ (SND)	–	P06MUX[1:0]/P04_07PMUX register
P07	BZ (SND)	–	–	P07MUX[1:0]/P04_07PMUX register
P10	SPICLK0 (SPI)	–	–	P10MUX[1:0]/P10_13PMUX register
P11	SDO0 (SPI)	SDAS (I2CS)	–	P11MUX[1:0]/P10_13PMUX register
P12	SDI0 (SPI)	SCLS (I2CS)	–	P12MUX[1:0]/P10_13PMUX register
P13	#SPISS0 (SPI)	FOUTA (CLG)	–	P13MUX[1:0]/P10_13PMUX register
P14	SCLK0 (UART)	EPDTRG (EPD)	–	P14MUX[1:0]/P14_17PMUX register
P15	SIN0 (UART)	SDAM (I2CM)	–	P15MUX[1:0]/P14_17PMUX register
P16	SOUT0 (UART)	SCLM (I2CM)	–	P16MUX[1:0]/P14_17PMUX register
P17	#BFR (I2CS)	EPDCLK (EPD)	–	P17MUX[1:0]/P14_17PMUX register
P20	SDAS (I2CS)	SENB0 (RFC)	BZ (SND)	P20MUX[1:0]/P20_23PMUX register
P21	SCLS (I2CS)	SENA0 (RFC)	#BZ (SND)	P21MUX[1:0]/P20_23PMUX register
P22	SDAM (I2CM)	REF0 (RFC)	–	P22MUX[1:0]/P20_23PMUX register
P23	SCLM (I2CM)	RFIN0 (RFC)	–	P23MUX[1:0]/P20_23PMUX register
P24	RFIN1 (RFC)	–	–	P24MUX[1:0]/P24_27PMUX register
P25	REF1 (RFC)	–	–	P25MUX[1:0]/P24_27PMUX register
P26	SENA1 (RFC)	–	–	P26MUX[1:0]/P24_27PMUX register
P27	SENB1 (RFC)	–	–	P27MUX[1:0]/P24_27PMUX register
P30	FOUTB (CLG)	#SPISS0 (SPI)	–	P30MUX[1:0]/P30_33PMUX register
P31	EPDCLK (EPD)	–	–	P31MUX[1:0]/P30_33PMUX register
DSIO (DBG)	P32	–	–	P32MUX[1:0]/P30_33PMUX register
DST2 (DBG)	P33	–	–	P33MUX[1:0]/P30_33PMUX register
DCLK (DBG)	P34	–	–	P34MUX[1:0]/P34PMUX register

At initial reset, each I/O port pin (P*xy*) is initialized for the default function ("Pin function 1" in Table 10.2.1).

Pins P01 and P04 can also be used as 16-bit PWM timer external clock input pins by setting them to input mode. However, general-purpose input port function is also effective in this case.

For information on functions other than the I/O ports, see the descriptions of the peripheral modules indicated in parentheses. The sections below describe port functions with the pins set as general-purpose I/O ports.

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

10.3 Data Input/Output

Data input/output control

The I/O ports allow selection of the data input/output direction for each bit using P*x*OEN*y*/P*x*_OEN register and P*x*IEN*y*/P*x*_IEN register. P*x*OEN*y* enables and disables data output, while P*x*IEN*y* enables and disables data input.

Table 10.3.1 Data Input/Output Status

PxOENy output control	PxIENy input control	PxPUy pull-up control	Port status
0	1	0	Functions as an input port (pull-up off). The port pin (external input signal) value can be read out from PxINy (input data). Output is disabled.
0	1	1	Functions as an input port (pull-up on). (Default) The port pin (external input signal) value can be read out from PxINy (input data). Output is disabled.
1	0	1 or 0	Functions as an output port (pull-up off). Input is disabled. The value read from PxINy (input data) is 0.
1	1	1 or 0	Functions as an output port (pull-up off). Input is also enabled. The port pin value (output value) can be read out from PxINy (input data).
0	0	0	The pin is placed into high-impedance status (pull-up off). Output and input are both disabled. The value read from PxINy (input data) is 0.
0	0	1	The pin is placed into high-impedance status (pull-up on). Output and input are both disabled. The value read from PxINy (input data) is 0.

The input/output direction of ports with a peripheral module function selected is controlled by the peripheral module. P*x*OEN*y* and P*x*IEN*y* settings are ignored.

Data input

To input the port pin status and read out the value, enable input by setting P*x*IEN*y* to 1 (default).
To input an external signal, P*x*OEN*y* should also be set to 0 (default). The I/O port is placed into high-impedance status and it functions as an input port (input mode). The port is pulled up if pull-up is enabled by P*x*PU*y*/P*x*_PU register.
In input mode, the input pin status can be read out directly from P*x*IN*y*/P*x*_IN register. The value read will be 1 when the input pin is at High (V_{DD}) level and 0 when it is at Low (V_{SS}) level.
The port pin status is always input when P*x*IEN*y* is 1, even if output is enabled (P*x*OEN*y* = 1) (output mode). In this case, the value actually output from the port can be read out from P*x*IN*y*.
When P*x*IEN*y* is set to 0, input is disabled, and 0 will be read out from P*x*IN*y*.

Data output

To output data from the port pin, enable output by setting P*x*OEN*y* to 1 (set to output mode). The I/O port then functions as an output port, and the value set in the P*x*OUT*y*/P*x*_OUT register is output from the port pin. The port pin outputs High (V_{DD}) level when P*x*OUT*y* is set to 1 and Low (V_{SS}) level when set to 0. Note that the port will not be pulled up in output mode, even if pull-up is enabled by P*x*PU*y*.
Writing to P*x*OUT*y* is possible without affecting pin status, even in input mode.

10.4 Pull-up Control

The I/O port contains a pull-up resistor that can be enabled or disabled individually for each bit using P*x*PU*y*/P*x*_PU register. Setting P*x*PU*y* to 1 (default) enables the pull-up resistor and pulls up the port pin in input mode. It will not be pulled up if set to 0. The P*x*PU*y* setting is ignored and not pulled up in output mode, regardless of how the P*x*IEN*y* is set.
I/O ports that are not used should be set with pull-up enabled.
The P*x*PU*y* setting is also ignored if a pin function other than P*xy* I/O port is selected. In this case, the pull-up resistor is automatically enabled/disabled according to the pin function selected.
A delay will occur in the waveform rising edge depending on time constants such as pull-up resistance and pin load capacitance if the port pin is switched from Low level to High level through the internal pull-up resistor. An appropriate wait time must be set for the I/O port loading. The wait time set should be a value not less than that calculated from the following equation.

Wait time = R_{IN} × (C_{IN} + load capacitance on board) × 1.6 [s]
R_{IN}: pull-up resistance maximum value, C_{IN}: pin capacitance maximum value

10.5 Port Input Interrupt

The P0 and P2 ports include input interrupt functions.
Select which of the 16 ports are to be used for interrupts based on requirements. You can also select whether interrupts are generated for either the rising edge or falling edge of the input signals.
Figure 10.5.1 shows the port input interrupt circuit configuration.

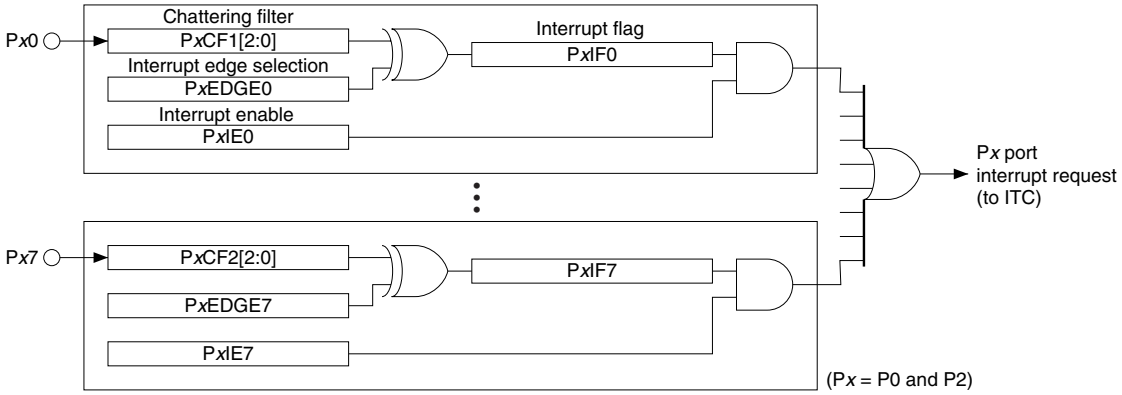


Figure 10.5.1 Port Input Interrupt Circuit Configuration

Interrupt port selection

Select the port generating an interrupt using P*x*IE*y*/P*x*_IMSK register.
Setting P*x*IE*y* to 1 enables interrupt generation by the corresponding port. Setting to 0 (default) disables interrupt generation.

Interrupt edge selection

Port input interrupts can be generated at either the rising edge or falling edge of the input signal. Select the edge used to generate interrupts using P*x*EDGE*y*/P*x*_EDGE register.
Setting P*x*EDGE*y* to 1 generates port input interrupts at the input signal falling edge. Setting it to 0 (default) generates interrupts at the rising edge.

Interrupt flags

The ITC is able to accept two interrupt requests from the P0 and P2 ports, and the P port module contains interrupt flags P*x*IF*y*/P*x*_IFLG register corresponding to the individual 16 ports to enable individual control of the 16 P0[7:0] and P2[7:0] port interrupts. P*x*IF*y* is set to 1 at the specified edge (rising or falling edge) of the input signal. If the corresponding P*x*IE*y* has been set to 1, an interrupt request signal is also output to the ITC at the same time. An interrupt is generated if the ITC and S1C17 Core interrupt conditions are satisfied.
P*x*IF*y* is reset by writing 1.
For specific information on interrupt processing, see the "Interrupt Controller (ITC)" chapter.

Notes: • The P port module interrupt flag P*x*IF*y* must be reset in the interrupt handler routine after a port interrupt has occurred to prevent recurring interrupts.

• To prevent generating unnecessary interrupts, reset the relevant P*x*IF*y* before enabling interrupts for the required port using P*x*IE*y*.

10.6 P0, P2 Port Chattering Filter Function

The P0 and P2 ports include a chattering filter circuit for key entry that can be disabled or enabled with a check time specified individually for the four P*x*[3:0] and P*x*[7:4] ports using P*x*CF1[2:0]/P*x*_CHAT register and P*x*CF2[2:0]/P*x*_CHAT register, respectively.

Table 10.6.1 Chattering Filter Function Settings

PxCF1[2:0]/PxCF2[2:0]	Check time *
0x7	16384/fPCLK (8 ms)
0x6	8192/fPCLK (4 ms)
0x5	4096/fPCLK (2 ms)
0x4	2048/fPCLK (1 ms)
0x3	1024/fPCLK (512 µs)
0x2	512/fPCLK (256 µs)
0x1	256/fPCLK (128 µs)
0x0	No check time (off)

(Default: 0x0, * when PCLK = 2 MHz)

Notes: • An unexpected interrupt may occur after SLEEP status is canceled if the slp instruction is executed while the chattering filter function is enabled. The chattering filter must be disabled before placing the CPU into SLEEP status.

• The chattering filter check time refers to the maximum pulse width that can be filtered. Generating an input interrupt requires an input time of twice the check time.

• The P*x* port interrupt must be disabled before setting the P*x*_CHAT register. Setting the register while the interrupt is enabled may generate inadvertent P*x* port interrupt. Also the chattering filter circuit requires a maximum of twice the check time for stabilizing the operation status. Before enabling the interrupt, make sure that the stabilization time has elapsed.

10.7 P0 Port Key-Entry Reset

Entering low level simultaneously to the ports (P00–P03) selected with software triggers an initial reset. The ports used for the reset function can be selected with the P0KRST[1:0]/P0_KRST register.

Table 10.7.1 Configuration of P0 Port Key-Entry Reset

P0KRST[1:0]	Port used for resetting
0x3	P00, P01, P02, P03
0x2	P00, P01, P02
0x1	P00, P01
0x0	Not used

(Default: 0x0)

For example, if P0KRST[1:0] is set to 0x3, an initial reset will take place when the four ports P00–P03 are set to low level at the same time.

Notes: • The P0 port key-entry reset function cannot be used for power-on reset as it must be enabled with software.

• When using the P0 port key-entry reset function, make sure that the designated input ports will not be simultaneously set to low level while the application program is running.

10.8 Control Register Details

Table 10.8.1 List of I/O Port Control Registers

Address	Register name		Function
0x5200	P0_IN	P0 Port Input Data Register	P0 port input data
0x5201	P0_OUT	P0 Port Output Data Register	P0 port output data
0x5202	P0_OEN	P0 Port Output Enable Register	Enables P0 port outputs.
0x5203	P0_PU	P0 Port Pull-up Control Register	Controls the P0 port pull-up resistor.
0x5205	P0_IMSK	P0 Port Interrupt Mask Register	Enables P0 port interrupts.
0x5206	P0_EDGE	P0 Port Interrupt Edge Select Register	Selects the signal edge for generating P0 port interrupts.
0x5207	P0_IFLG	P0 Port Interrupt Flag Register	Indicates/resets the P0 port interrupt occurrence status.
0x5208	P0_CHAT	P0 Port Chattering Filter Control Register	Controls the P0 port chattering filter.
0x5209	P0_KRST	P0 Port Key-Entry Reset Configuration Register	Configures the P0 port key-entry reset function.
0x520a	P0_IEN	P0 Port Input Enable Register	Enables P0 port inputs.
0x5210	P1_IN	P1 Port Input Data Register	P1 port input data

Address	Register name		Function
0x5211	P1_OUT	P1 Port Output Data Register	P1 port output data
0x5212	P1_OEN	P1 Port Output Enable Register	Enables P1 port outputs.
0x5213	P1_PU	P1 Port Pull-up Control Register	Controls the P1 port pull-up resistor.
0x521a	P1_IEN	P1 Port Input Enable Register	Enables P1 port inputs.
0x5220	P2_IN	P2 Port Input Data Register	P2 port input data
0x5221	P2_OUT	P2 Port Output Data Register	P2 port output data
0x5222	P2_OEN	P2 Output Enable Register	Enables P2 port outputs.
0x5223	P2_PU	P2 Port Pull-up Control Register	Controls the P2 port pull-up resistor.
0x5225	P2_IMSK	P2 Port Interrupt Mask Register	Enables P2 port interrupts.
0x5226	P2_EDGE	P2 Port Interrupt Edge Select Register	Selects the signal edge for generating P2 port interrupts.
0x5227	P2_IFLG	P2 Port Interrupt Flag Register	Indicates/resets the P2 port interrupt occurrence status.
0x5228	P2_CHAT	P2 Port Chattering Filter Control Register	Controls the P2 port chattering filter.
0x522a	P2_IEN	P2 Port Input Enable Register	Enables P2 port inputs.
0x5230	P3_IN	P3 Port Input Data Register	P3 port input data
0x5231	P3_OUT	P3 Port Output Data Register	P3 port output data
0x5232	P3_OEN	P3 Port Output Enable Register	Enables P3 port outputs.
0x5233	P3_PU	P3 Port Pull-up Control Register	Controls the P3 port pull-up resistor.
0x523a	P3_IEN	P3 Port Input Enable Register	Enables P3 port inputs.
0x52a0	P00_03PMUX	P0[3:0] Port Function Select Register	Selects the P0[3:0] port functions.
0x52a1	P04_07PMUX	P0[7:4] Port Function Select Register	Selects the P0[7:4] port functions.
0x52a2	P10_13PMUX	P1[3:0] Port Function Select Register	Selects the P1[3:0] port functions.
0x52a3	P14_17PMUX	P1[7:4] Port Function Select Register	Selects the P1[7:4] port functions.
0x52a4	P20_23PMUX	P2[3:0] Port Function Select Register	Selects the P2[3:0] port functions.
0x52a5	P24_27PMUX	P2[7:4] Port Function Select Register	Selects the P2[7:4] port functions.
0x52a6	P30_33PMUX	P3[3:0] Port Function Select Register	Selects the P3[3:0] port functions.
0x52a7	P34PMUX	P34 Port Function Select Register	Selects the P34 port functions.

The I/O port registers are described in detail below.

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

P*x* Port Input Data Registers (P*x*_IN)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
P*x* Port Input Data Register (P*x*_IN)	**0x5200** **0x5210** **0x5220** **0x5230** (8 bits)	D7–0	**PxIN[7:0]**	P*x*[7:0] port input data	1	1 (H)	0	0 (L)	×	R	

Note: P3IN[4:0] only are available for the P3 ports. Other bits are reserved and always read as 0.

D[7:0] PxIN[7:0]: P*x*[7:0] Port Input Data Bits

The port pin status can be read out. (Default: external input status)

1 (R): High level

0 (R): Low level

P*x*IN*y* corresponds directly to the P*xy* pin. The pin voltage level can be read out when input is enabled (P*x*IEN*y* = 1) (even if output is also enabled (P*x*OEN*y* = 1)). The value read out will be 1 when the pin voltage is High and 0 when Low.

The value read out is 0 when input is disabled (P*x*IEN*y* = 0).

Writing operations to the read-only P*x*IN*y* is disabled.

P*x* Port Output Data Registers (P*x*_OUT)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
P*x* Port Output Data Register (P*x*_OUT)	**0x5201** **0x5211** **0x5221** **0x5231** (8 bits)	D7–0	**PxOUT[7:0]**	P*x*[7:0] port output data	1	1 (H)	0	0 (L)	0	R/W	

Note: P3OUT[4:0] only are available for the P3 ports. Other bits are reserved and always read as 0.

D[7:0] PxOUT[7:0]: Px[7:0] Port Output Data Bits

Sets the data to be output from the port pin.
1 (R/W): High level
0 (R/W): Low level (default)

PxOUTy corresponds directly to the Pxy pins. The data written will be output unchanged from the port pins when output is enabled (PxOENy = 1). The port pin will be High when the data bit is set to 1 and Low when set to 0.
Port data can also be written when output is disabled (PxOENy = 0) (the pin status is unaffected).

Px Port Output Enable Registers (Px_OEN)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Px Port Output Enable Register (Px_OEN)	0x5202 0x5212 0x5222 0x5232 (8 bits)	D7–0	PxOEN[7:0]	Px[7:0] port output enable	1	Enable	0	Disable	0	R/W	

Note: P3OEN[4:0] only are available for the P3 ports. Other bits are reserved and always read as 0.

D[7:0] PxOEN[7:0]: Px[7:0] Port Output Enable Bits

Enables or disables port outputs.
1 (R/W): Enabled
0 (R/W): Disabled (default)

PxOENy is the output enable bit that corresponds directly to Pxy port. Setting to 1 enables output and the data set in PxOUTy is output from the port pin. Output is disabled when PxOENy is set to 0, and the port pin is set into high-impedance status. The peripheral module determines whether output is enabled or disabled when the port is used for a peripheral module function.
Refer to Table 10.3.1 for more information on input/output status for ports, including settings other than for the PxOEN register.

Px Port Pull-up Control Registers (Px_PU)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Px Port Pull-up Control Register (Px_PU)	0x5203 0x5213 0x5223 0x5233 (8 bits)	D7–0	PxPU[7:0]	Px[7:0] port pull-up enable	1	Enable	0	Disable	1 (0xff)	R/W	

Note: P3PU[4:0] only are available for the P3 ports. Other bits are reserved and always read as 0.

D[7:0] PxPU[7:0]: Px[7:0] Port Pull-up Enable Bits

Enables or disables the pull-up resistor included in each port.
1 (R/W): Enabled (default)
0 (R/W): Disabled

PxPUy is the pull-up control bit that corresponds directly to the Pxy port. Setting to 1 enables the pull-up resistor and the port pin will be pulled up when output is disabled (PxOENy = 0). When PxPUy is set to 0, the pin will not be pulled up.
When output is enabled (PxOENy = 1), the PxPUy setting is ignored, and the pin is not pulled up.
I/O ports that are not used should be set with pull-up enabled.
The PxPUy setting is also ignored if a pin function other than Pxy I/O port is selected. In this case, the pull-up resistor is automatically enabled/disabled according to the pin function selected.

Px Port Interrupt Mask Registers (Px_IMSK)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Px Port Interrupt Mask Register (Px_IMSK)	0x5205 0x5225 (8 bits)	D7–0	PxIE[7:0]	Px[7:0] port interrupt enable	1	Enable	0	Disable	0	R/W	

Note: This register is available for the P0 and P2 ports.

D[7:0] PxIE[7:0]: Px[7:0] Port Interrupt Enable Bits

Enables or disables each port interrupt.
1 (R/W): Interrupt enabled
0 (R/W): Interrupt disabled (default)

Setting PxIEy to 1 enables the corresponding Pxy port input interrupt, while setting to 0 disables the interrupt. Status changes for the input pins with interrupt disabled do not affect interrupt occurrence.

Px Port Interrupt Edge Select Registers (Px_EDGE)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Px Port Interrupt Edge Select Register (Px_EDGE)	0x5206 0x5226 (8 bits)	D7–0	PxEDGE[7:0]	Px[7:0] port interrupt edge select	1	Falling edge	0	Rising edge	0	R/W	

Note: This register is available for the P0 and P2 ports.

D[7:0] PxEDGE[7:0]: Px[7:0] Port Interrupt Edge Select Bits

Selects the input signal edge for generating each port interrupt.
1 (R/W): Falling edge
0 (R/W): Rising edge (default)

Port interrupts are generated at the input signal falling edge when PxEDGEy is set to 1 and at the rising edge when set to 0.

Px Port Interrupt Flag Registers (Px_IFLG)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Px Port Interrupt Flag Register (Px_IFLG)	0x5207 0x5227 (8 bits)	D7–0	PxIF[7:0]	Px[7:0] port interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.

Note: This register is available for the P0 and P2 ports.

D[7:0] PxIF[7:0]: Px[7:0] Port Interrupt Flag Bits

These are interrupt flags indicating the interrupt cause occurrence status.
1 (R): Interrupt cause occurred
0 (R): No interrupt cause occurred (default)
1 (W): Reset flag
0 (W): Ignored

PxIFy is the interrupt flag corresponding to the individual 16 ports of P0 and P2. It is set to 1 at the specified edge (rising or falling edge) of the input signal. When the corresponding PxIEy/Px_IMSK register has been set to 1, a port interrupt request signal is also output to the ITC at the same time. An interrupt is generated if the ITC and S1C17 Core interrupt conditions are satisfied.
PxIFy is reset by writing 1.

Notes:
- The P port module interrupt flag PxIFy must be reset in the interrupt handler routine after a port interrupt has occurred to prevent recurring interrupts.
- To prevent generating unnecessary interrupts, reset the relevant PxIFy before enabling interrupts for the required port using PxIEy/Px_IMSK register.

Px Port Chattering Filter Control Registers (Px_CHAT)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
Px Port Chattering Filter Control Register (Px_CHAT)	0x5208 0x5228 (8 bits)	D7	–	reserved	–		–	–	0 when being read.
		D6–4	PxCF2[2:0]	Px[7:4] chattering filter time select	PxCF2[2:0]	Filter time	0x0	R/W	
					0x7	16384/fPCLK			
					0x6	8192/fPCLK			
					0x5	4096/fPCLK			
					0x4	2048/fPCLK			
					0x3	1024/fPCLK			
					0x2	512/fPCLK			
					0x1	256/fPCLK			
					0x0	None			
		D3	–	reserved	–		–	–	0 when being read.
		D2–0	PxCF1[2:0]	Px[3:0] chattering filter time select	PxCF1[2:0]	Filter time	0x0	R/W	
					0x7	16384/fPCLK			
					0x6	8192/fPCLK			
					0x5	4096/fPCLK			
					0x4	2048/fPCLK			
					0x3	1024/fPCLK			
					0x2	512/fPCLK			
					0x1	256/fPCLK			
					0x0	None			

Note: This register is available for the P0 and P2 ports.

D7 Reserved

D[6:4] PxCF2[2:0]: Px[7:4] Chattering Filter Time Select Bits

Configures the chattering filter circuit for the P*x*[7:4] ports.

D3 Reserved

D[2:0] PxCF1[2:0]: Px[3:0] Chattering Filter Time Select Bits

Configures the chattering filter circuit for the P*x*[3:0] ports.

The P0 and P2 ports include a chattering filter circuit for key entry that can be disabled or enabled with a check time specified individually for the four P*x*[3:0] and P*x*[7:4] ports using P*x*CF1[2:0] and P*x*CF2[2:0], respectively.

Table 10.8.2 Chattering Filter Function Settings

PxCF1[2:0]/PxCF2[2:0]	Check time *
0x7	16384/fPCLK (8 ms)
0x6	8192/fPCLK (4 ms)
0x5	4096/fPCLK (2 ms)
0x4	2048/fPCLK (1 ms)
0x3	1024/fPCLK (512 µs)
0x2	512/fPCLK (256 µs)
0x1	256/fPCLK (128 µs)
0x0	No check time (off)

(Default: 0x0, * when PCLK = 2 MHz)

Notes:
- An unexpected interrupt may occur after SLEEP status is canceled if the slp instruction is executed while the chattering filter function is enabled. The chattering filter must be disabled before placing the CPU into SLEEP status.
- The chattering filter check time refers to the maximum pulse width that can be filtered. Generating an input interrupt requires an input time of twice the check time.
- The P*x* port interrupt must be disabled before setting the P*x*_CHAT register. Setting the register while the interrupt is enabled may generate inadvertent P*x* interrupt. Also the chattering filter circuit requires a maximum of twice the check time for stabilizing the operation status. Before enabling the interrupt, make sure that the stabilization time has elapsed.

P0 Port Key-Entry Reset Configuration Register (P0_KRST)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
P0 Port Key-Entry Reset Configuration Register (P0_KRST)	0x5209 (8 bits)	D7–2	–	reserved	–		–	–	0 when being read.
		D1–0	P0KRST[1:0]	P0 port key-entry reset configuration	P0KRST[1:0]	Configuration	0x0	R/W	
					0x3	P0[3:0]			
					0x2	P0[2:0]			
					0x1	P0[1:0]			
					0x0	Disable			

D[7:2] Reserved

D[1:0] P0KRST[1:0]: P0 Port Key-Entry Reset Configuration Bits

Selects the port combination used for P0 port key-entry reset.

Table 10.8.3 P0 Port Key-Entry Reset Settings

P0KRST[1:0]	Ports used for resetting
0x3	P00, P01, P02, P03
0x2	P00, P01, P02
0x1	P00, P01
0x0	Not used

(Default: 0x0)

The key-entry reset function performs an initial reset by inputting Low level simultaneously to the ports selected here. For example, if P0KRST[1:0] is set to 0x3, an initial reset is performed when the four ports P00 to P03 are simultaneously set to Low level.
Set P0KRST[1:0] to 0x0 when this reset function is not used.

Notes: • The P0 port key-entry reset function is disabled at initial reset and cannot be used for power-on reset.

• When using the P0 port key-entry reset function, make sure that the designated input ports will not be simultaneously set to low level while the application program is running.

P*x* Port Input Enable Registers (P*x*_IEN)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
P*x* Port Input Enable Register (P*x*_IEN)	0x520a 0x521a 0x522a 0x523a (8 bits)	D7–0	PxIEN[7:0]	P*x*[7:0] port input enable	1	Enable	0	Disable	1 (0xff)	R/W	

Note: P3IEN[4:0] only are available for the P3 ports. Other bits are reserved and always read as 0.

D[7:0] P*x*IEN[7:0]: P*x*[7:0] Port Input Enable Bits

Enables or disables port inputs.
1 (R/W): Enable (default)
0 (R/W): disable

P*x*IEN*y* is the input enable bit that corresponds directly to the P*xy* port. Setting to 1 enables input and the corresponding port pin input or output signal level can be read out from the P*x*_IN register. Setting to 0 disables input.
Refer to Table 10.3.1 for more information on port input/output status, including settings other than for the P*x*_IEN register.

P0[3:0] Port Function Select Register (P00_03PMUX)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
P0[3:0] Port Function Select Register (P00_03PMUX)	0x52a0 (8 bits)	D7–6	P03MUX[1:0]	P03 port function select	P03MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	TOUTB0/CAPB0			
					0x0	P03			
		D5–4	P02MUX[1:0]	P02 port function select	P02MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	TOUTA0/CAPA0			
					0x0	P02			
		D3–2	P01MUX[1:0]	P01 port function select	P01MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	reserved			
					0x0	P01/EXCL0			
		D1–0	P00MUX[1:0]	P00 port function select	P00MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	REGMON			
					0x1	RFCLKO			
					0x0	P00			

The P00 to P03 port pins are shared with the peripheral module pins. This register is used to select how the pins are used.

D[7:6] P03MUX[1:0]: P03 Port Function Select Bits

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): TOUTB0 (T16A2 Ch.0 comparator mode) or CAPB0 (T16A2 Ch.0 capture mode)
0x0 (R/W): P03 port (default)

D[5:4] P02MUX[1:0]: P02 Port Function Select Bits

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): TOUTA0 (T16A2 Ch.0 comparator mode) or CAPA0 (T16A2 Ch.0 capture mode)
0x0 (R/W): P02 port (default)

D[3:2] P01MUX[1:0]: P01 Port Function Select Bits

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): Reserved
0x0 (R/W): P01 port/EXCL0 (T16A2 Ch.0) (default)

To use the P01 pin for EXCL0 input, P0OEN1/P0_OEN register must be set to 0 and P0IEN1/P0_IEN register must be set to 1.

D[1:0] P00MUX[1:0]: P00 Port Function Select Bits

0x3 (R/W): Reserved
0x2 (R/W): REGMON (TR)
0x1 (R/W): RFCLKO (RFC)
0x0 (R/W): P00 port (default)

P0[7:4] Port Function Select Register (P04_07PMUX)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
P0[7:4] Port Function Select Register (P04_07PMUX)	0x52a1 (8 bits)	D7–6	P07MUX[1:0]	P07 port function select	P07MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	BZ			
					0x0	P07			
		D5–4	P06MUX[1:0]	P06 port function select	P06MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	#BZ			
					0x1	TOUTB1/CAPB1			
					0x0	P06			
		D3–2	P05MUX[1:0]	P05 port function select	P05MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	TOUTA1/CAPA1			
					0x0	P05			
		D1–0	P04MUX[1:0]	P04 port function select	P04MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	reserved			
					0x0	P04/EXCL1			

The P04 to P07 port pins are shared with the peripheral module pins. This register is used to select how the pins are used.

D[7:6] P07MUX[1:0]: P07 Port Function Select Bits

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): BZ (SND)
0x0 (R/W): P07 port (default)

D[5:4] P06MUX[1:0]: P06 Port Function Select Bits

0x3 (R/W): Reserved
0x2 (R/W): #BZ (SND)
0x1 (R/W): TOUTB1 (T16A2 Ch.1 comparator mode) or CAPB1 (T16A2 Ch.1 capture mode)
0x0 (R/W): P06 port (default)

D[3:2] P05MUX[1:0]: P05 Port Function Select Bits

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): TOUTA1 (T16A2 Ch.1 comparator mode) or CAPA1 (T16A2 Ch.1 capture mode)
0x0 (R/W): P05 port (default)

D[1:0] P04MUX[1:0]: P04 Port Function Select Bits

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): Reserved
0x0 (R/W): P04 port/EXCL1 (T16A2 Ch.1) (default)

To use the P04 pin for EXCL1 input, P0OEN4/P0_OEN register must be set to 0 and P0IEN4/P0_IEN register must be set to 1.

P1[3:0] Port Function Select Register (P10_13PMUX)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
P1[3:0] Port Function Select Register (P10_13PMUX)	**0x52a2** (8 bits)	D7–6	**P13MUX[1:0]**	P13 port function select	P13MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	FOUTA			
					0x1	#SPISS0			
					0x0	P13			
		D5–4	**P12MUX[1:0]**	P12 port function select	P12MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	SCLS			
					0x1	SDI0			
					0x0	P12			
		D3–2	**P11MUX[1:0]**	P11 port function select	P11MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	SDAS			
					0x1	SDO0			
					0x0	P11			
		D1–0	**P10MUX[1:0]**	P10 port function select	P10MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	SPICLK0			
					0x0	P10			

The P10 to P13 port pins are shared with the peripheral module pins. This register is used to select how the pins are used.

D[7:6] P13MUX[1:0]: P13 Port Function Select Bits

0x3 (R/W): Reserved
0x2 (R/W): FOUTA (CLG)
0x1 (R/W): #SPISS0 (SPI Ch.0)
0x0 (R/W): P13 port (default)

D[5:4] P12MUX[1:0]: P12 Port Function Select Bits

0x3 (R/W): Reserved
0x2 (R/W): SCLS (I2CS)
0x1 (R/W): SDI0 (SPI Ch.0)
0x0 (R/W): P12 port (default)

D[3:2] P11MUX[1:0]: P11 Port Function Select Bits

0x3 (R/W): Reserved
0x2 (R/W): SDAS (I2CS)
0x1 (R/W): SDO0 (SPI Ch.0)
0x0 (R/W): P11 port (default)

D[1:0] P10MUX[1:0]: P10 Port Function Select Bits

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): SPICLK0 (SPI Ch.0)
0x0 (R/W): P10 port (default)

P1[7:4] Port Function Select Register (P14_17PMUX)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
P1[7:4] Port Function Select Register (P14_17PMUX)	0x52a3 (8 bits)	D7–6	P17MUX[1:0]	P17 port function select	P17MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	EPDCLK			
					0x1	#BFR			
					0x0	P17			
		D5–4	P16MUX[1:0]	P16 port function select	P16MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	SCLM			
					0x1	SOUT0			
					0x0	P16			
		D3–2	P15MUX[1:0]	P15 port function select	P15MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	SDAM			
					0x1	SIN0			
					0x0	P15			
		D1–0	P14MUX[1:0]	P14 port function select	P14MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	EPDTRG			
					0x1	SCLK0			
					0x0	P14			

The P14 to P17 port pins are shared with the peripheral module pins. This register is used to select how the pins are used.

D[7:6] P17MUX[1:0]: P17 Port Function Select Bits

0x3 (R/W): Reserved
0x2 (R/W): EPDCLK (EPD)
0x1 (R/W): #BFR (I2CS)
0x0 (R/W): P17 port (default)

D[5:4] P16MUX[1:0]: P16 Port Function Select Bits

0x3 (R/W): Reserved
0x2 (R/W): SCLM (I2CM)
0x1 (R/W): SOUT0 (UART Ch.0)
0x0 (R/W): P16 port (default)

D[3:2] P15MUX[1:0]: P15 Port Function Select Bits

0x3 (R/W): Reserved
0x2 (R/W): SDAM (I2CM)
0x1 (R/W): SIN0 (UART Ch.0)
0x0 (R/W): P15 port (default)

D[1:0] P14MUX[1:0]: P14 Port Function Select Bits

0x3 (R/W): Reserved
0x2 (R/W): EPDTRG (EPD)
0x1 (R/W): SCLK0 (UART Ch.0)
0x0 (R/W): P14 port (default)

P2[3:0] Port Function Select Register (P20_23PMUX)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
P2[3:0] Port Function Select Register (P20_23PMUX)	**0x52a4** (8 bits)	D7–6	**P23MUX[1:0]**	P23 port function select	P23MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	RFIN0			
					0x1	SCLM			
					0x0	P23			
		D5–4	**P22MUX[1:0]**	P22 port function select	P22MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	REF0			
					0x1	SDAM			
					0x0	P22			
		D3–2	**P21MUX[1:0]**	P21 port function select	P21MUX[1:0]	Function	0x0	R/W	
					0x3	#BZ			
					0x2	SENA0			
					0x1	SCLS			
					0x0	P21			
		D1–0	**P20MUX[1:0]**	P20 port function select	P20MUX[1:0]	Function	0x0	R/W	
					0x3	BZ			
					0x2	SENB0			
					0x1	SDAS			
					0x0	P20			

The P20 to P23 port pins are shared with the peripheral module pins. This register is used to select how the pins are used.

D[7:6] P23MUX[1:0]: P23 Port Function Select Bits

0x3 (R/W): Reserved
0x2 (R/W): RFIN0 (RFC)
0x1 (R/W): SCLM (I2CM)
0x0 (R/W): P23 port (default)

D[5:4] P22MUX[1:0]: P22 Port Function Select Bits

0x3 (R/W): Reserved
0x2 (R/W): REF0 (RFC)
0x1 (R/W): SDAM (I2CM)
0x0 (R/W): P22 port (default)

D[3:2] P21MUX[1:0]: P21 Port Function Select Bits

0x3 (R/W): #BZ (SND)
0x2 (R/W): SENA0 (RFC)
0x1 (R/W): SCLS (I2CS)
0x0 (R/W): P21 port (default)

D[1:0] P20MUX[1:0]: P20 Port Function Select Bits

0x3 (R/W): BZ (SND)
0x2 (R/W): SENB0 (RFC)
0x1 (R/W): SDAS (I2CS)
0x0 (R/W): P20 port (default)

P2[7:4] Port Function Select Register (P24_27PMUX)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
P2[7:4] Port Function Select Register (P24_27PMUX)	0x52a5 (8 bits)	D7–6	P27MUX[1:0]	P27 port function select	P27MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	SENB1			
					0x0	P27			
		D5–4	P26MUX[1:0]	P26 port function select	P26MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	SENA1			
					0x0	P26			
		D3–2	P25MUX[1:0]	P25 port function select	P25MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	REF1			
					0x0	P25			
		D1–0	P24MUX[1:0]	P24 port function select	P24MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	RFIN1			
					0x0	P24			

The P24 to P27 port pins are shared with the peripheral module pins. This register is used to select how the pins are used.

D[7:6] P27MUX[1:0]: P27 Port Function Select Bits

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): SENB1 (RFC)
0x0 (R/W): P27 port (default)

D[5:4] P26MUX[1:0]: P26 Port Function Select Bits

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): SENA1 (RFC)
0x0 (R/W): P26 port (default)

D[3:2] P25MUX[1:0]: P25 Port Function Select Bits

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): REF1 (RFC)
0x0 (R/W): P25 port (default)

D[1:0] P24MUX[1:0]: P24 Port Function Select Bits

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): RFIN1 (RFC)
0x0 (R/W): P24 port (default)

P3[3:0] Port Function Select Register (P30_33PMUX)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
P3[3:0] Port Function Select Register (P30_33PMUX)	**0x52a6** (8 bits)	D7–6	**P33MUX[1:0]**	P33 port function select	P33MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	P33			
					0x0	DST2			
		D5–4	**P32MUX[1:0]**	P32 port function select	P32MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	P32			
					0x0	DSIO			
		D3–2	**P31MUX[1:0]**	P31 port function select	P31MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	EPDCLK			
					0x0	P31			
		D1–0	**P30MUX[1:0]**	P30 port function select	P30MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	#SPISS0			
					0x1	FOUTB			
					0x0	P30			

The P30 to P33 port pins are shared with the peripheral module pins. This register is used to select how the pins are used.

D[7:6] P33MUX[1:0]: P33 Port Function Select Bits

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): P33 port
0x0 (R/W): DST2 (DBG) (default)

D[5:4] P32MUX[1:0]: P32 Port Function Select Bits

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): P32 port
0x0 (R/W): DSIO (DBG) (default)

D[3:2] P31MUX[1:0]: P31 Port Function Select Bits

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): EPDCLK (EPD)
0x0 (R/W): P31 port (default)

D[1:0] P30MUX[1:0]: P30 Port Function Select Bits

0x3 (R/W): Reserved
0x2 (R/W): #SPISS0 (SPI Ch.0)
0x1 (R/W): FOUTB (CLG)
0x0 (R/W): P30 port (default)

P34 Port Function Select Register (P34PMUX)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
P34 Port Function Select Register (P34PMUX)	0x52a7 (8 bits)	D7–2	–	reserved	–		–	–	0 when being read.
		D1–0	P34MUX[1:0]	P34 port function select	P34MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	P34			
					0x0	DCLK			

The P34 port pin is shared with the peripheral module pin. This register is used to select how the pin is used.

D[7:2] Reserved

D[1:0] P34MUX[1:0]: P34 Port Function Select Bits

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): P34 port
0x0 (R/W): DCLK (DBG) (default)

11 8-bit Timers (T8)

11.1 T8 Module Overview

The S1C17F57 includes an 8-bit timer module (T8).
The features of the T8 module are listed below.

- Consists of two timer channels (T8 Ch.0 and Ch.1).
- 8-bit presettable down counter with an 8-bit reload data register for setting the preset value
- Generates the SPI and I²C master operating clocks from the counter underflow signals.
- Generates underflow interrupt signals to the interrupt controller (ITC).
- Any desired time intervals and serial transfer rates can be programmed by selecting an appropriate count clock and preset value.

Figure 11.1.1 shows the T8 configuration.

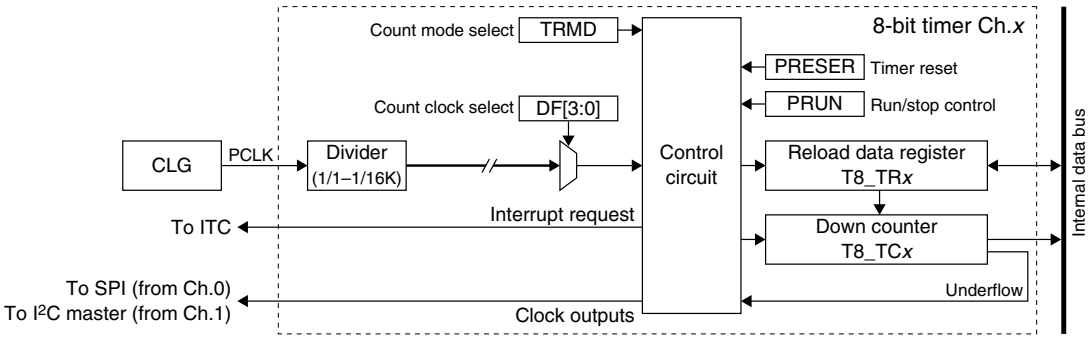


Figure 11.1.1 T8 Configuration (1 Channel)

Each channel of the T8 module consists of an 8-bit presettable down counter and an 8-bit reload data register holding the preset value. The timer counts down from the initial value set in the reload data register and outputs an underflow signal when the counter underflows. The underflow signal is used to generate an interrupt and an internal serial interface clock. The underflow cycle can be programmed by selecting the count clock and reload data, enabling the application program to obtain time intervals and serial transfer rates as required.

Note: The letter '*x*' in register names refers to a channel number (0 or 1).
Example: T8_CTL*x* register
Ch.0: T8_CTL0 register
Ch.1: T8_CTL1 register

11.2 Count Clock

The count clock is generated by dividing the PCLK clock into 1/1 to 1/16K. The division ratio can be selected from the 15 types shown below using DF[3:0]/T8_CLK*x* register.

Table 11.2.1 PCLK Division Ratio Selection

DF[3:0]	Division ratio	DF[3:0]	Division ratio
0xf	Reserved	0x7	1/128
0xe	1/16384	0x6	1/64
0xd	1/8192	0x5	1/32
0xc	1/4096	0x4	1/16
0xb	1/2048	0x3	1/8
0xa	1/1024	0x2	1/4
0x9	1/512	0x1	1/2
0x8	1/256	0x0	1/1

(Default: 0x0)

Notes: • The clock generator (CLG) must be configured to supply PCLK to the peripheral modules before running the timer.

• Make sure the counter is halted before setting the count clock.

For detailed information on the CLG control, see the "Clock Generator (CLG)" chapter.

11.3 Count Mode

The T8 module features two count modes: repeat mode and one-shot mode. These modes are selected using TRMD/T8_CTL*x* register.

Repeat mode (TRMD = 0, default)

Setting TRMD to 0 sets T8 to repeat mode.

In this mode, once the count starts, the timer continues running until stopped by the application program. When the counter underflows, the timer presets the reload data register value into the counter and continues the count. Thus, the timer periodically outputs an underflow pulse. T8 should be set to this mode to generate periodic interrupts at desired intervals or to generate a serial transfer clock.

One-shot mode (TRMD = 1)

Setting TRMD to 1 sets T8 to one-shot mode.

In this mode, the timer stops automatically as soon as the counter underflows. This means only one interrupt can be generated after the timer starts. Note that the timer presets the reload data register value to the counter, then stops after an underflow has occurred. T8 should be set to this mode to set a specific wait time.

11.4 Reload Data Register and Underflow Cycle

The reload data register T8_TR*x* is used to set the initial value for the down counter.

The initial counter value set in the reload data register is preset to the down counter if the timer is reset or the counter underflows. If the timer is started after resetting, it counts down from the reload value (initial value). This means that the reload value and the input clock frequency determine the time elapsed from the point at which the timer starts until the underflow occurs (or between underflows). The time determined is used to obtain the specified wait time, the intervals between periodic interrupts, and the programmable serial interface transfer clock.

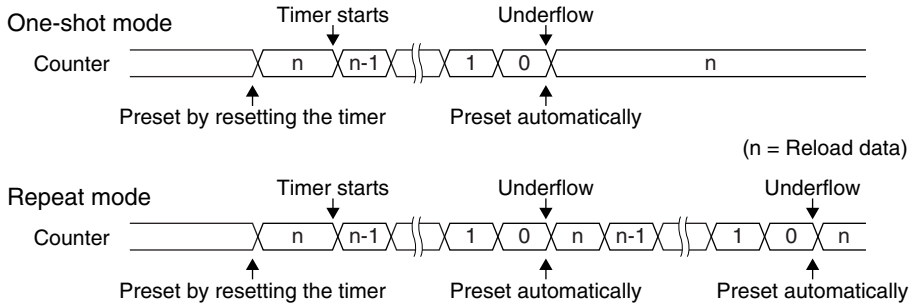


Figure 11.4.1 Preset Timing

The underflow cycle can be calculated as follows:

$$\text{Underflow interval} = \frac{TR + 1}{ct\_clk}\ [s] \qquad \text{Underflow cycle} = \frac{ct\_clk}{TR + 1}\ [Hz]$$

ct_clk: Count clock frequency [Hz]
TR: Reload data (0–255)

11.5 Timer Reset

The timer is reset by writing 1 to PRESER/T8_CTL*x* register. The reload data is preset and the counter is initialized.

11.6 Timer RUN/STOP Control

Make the following settings before starting the timer.
(1) Select the count clock. See Section 11.2.
(2) Set the count mode (one-shot or repeat). See Section 11.3.
(3) Calculate the initial counter value and set it to the reload data register. See Section 11.4.
(4) Reset the timer to preset the counter to the initial value. See Section 11.5.
(5) When using timer interrupts, set the interrupt level and enable interrupts for the relevant timer channel. See Section 11.8.

To start the timer, write 1 to PRUN/T8_CTL*x* register.
The timer starts counting down from the initial value or from the current counter value if no initial value was preset. When the counter underflows, the timer outputs an underflow pulse and presets the counter to the initial value. An interrupt request is sent simultaneously to the interrupt controller (ITC).
In one-shot mode, the timer stops counting.
In repeat mode, the timer continues counting from the reloaded initial value.

Write 0 to PRUN to stop the timer via the application program. The counter stops counting and retains the current counter value until either the timer is reset or restarted. To restart the count from the initial value, the timer should be reset before writing 1 to PRUN.

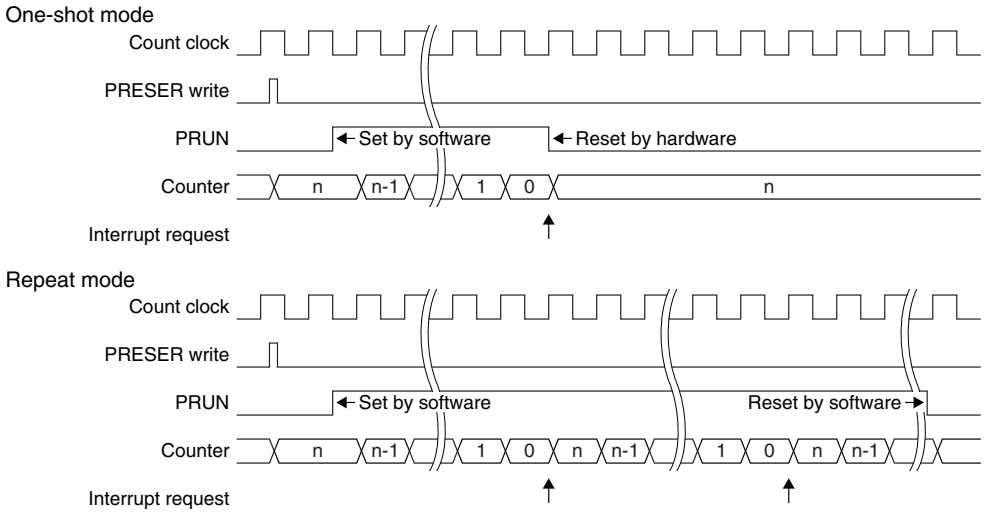


Figure 11.6.1 Count Operation

11.7 T8 Output Signals

The T8 module outputs underflow pulses when the counter underflows.
These pulses are used for timer interrupt requests.
These pulses are also used to generate the serial transfer clock for the internal serial interface.
The clock generated is sent to the internal peripheral module, as shown below.
T8 Ch.0 output clock → SPI
T8 Ch.1 output clock → I²C master

Use the following equations to calculate the reload data register value for obtaining the desired transfer rate:

SPI $$TR = \frac{ct\_clk}{bps \times 2} - 1$$

I²C master $$TR = \frac{ct\_clk}{bps \times 4} - 1$$

ct_clk: Count clock frequency (Hz)
TR: Reload data (0–255)
bps: Transfer rate (bits/s)

11.8 T8 Interrupts

Each channel of the T8 module outputs an interrupt request to the interrupt controller (ITC) when the counter underflows.

Underflow interrupt

When the counter underflows, the interrupt flag T8IF/T8_INT*x* register, which is provided for each channel in the T8 module, is set to 1. At the same time, an interrupt request is sent to the ITC if T8IE/T8_INT*x* register has been set to 1 (interrupt enabled). An interrupt is generated if the ITC and S1C17 Core interrupt conditions are satisfied.
If T8IE is set to 0 (interrupt disabled, default), no interrupt request will be sent to the ITC.
For specific information on interrupt processing, see the "Interrupt Controller (ITC)" chapter.

Notes: • The T8 module interrupt flag T8IF must be reset in the interrupt handler routine after a T8 interrupt has occurred to prevent recurring interrupts.
• Reset T8IF before enabling T8 interrupts with T8IE to prevent occurrence of unwanted interrupt. T8IF is reset by writing 1.

11.9 Control Register Details

Table 11.9.1 List of T8 Registers

Address	Register name		Function
0x4240	T8_CLK0	T8 Ch.0 Count Clock Select Register	Selects a count clock.
0x4242	T8_TR0	T8 Ch.0 Reload Data Register	Sets reload data.
0x4244	T8_TC0	T8 Ch.0 Counter Data Register	Counter data
0x4246	T8_CTL0	T8 Ch.0 Control Register	Sets the timer mode and starts/stops the timer.
0x4248	T8_INT0	T8 Ch.0 Interrupt Control Register	Controls the interrupt.
0x4260	T8_CLK1	T8 Ch.1 Count Clock Select Register	Selects a count clock.
0x4262	T8_TR1	T8 Ch.1 Reload Data Register	Sets reload data.
0x4264	T8_TC1	T8 Ch.1 Counter Data Register	Counter data
0x4266	T8_CTL1	T8 Ch.1 Control Register	Sets the timer mode and starts/stops the timer.
0x4268	T8_INT1	T8 Ch.1 Interrupt Control Register	Controls the interrupt.

The T8 registers are described in detail below.

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

T8 Ch.*x* Count Clock Select Registers (T8_CLK*x*)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
T8 Ch.*x* Count Clock Select Register (T8_CLK*x*)	**0x4240** **0x4260** (16 bits)	D15–4	**–**	reserved	–		–	–	0 when being read.
		D3–0	**DF[3:0]**	Count clock division ratio select	DF[3:0]	Division ratio	0x0	R/W	Source clock = PCLK
					0xf	reserved			
					0xe	1/16384			
					0xd	1/8192			
					0xc	1/4096			
					0xb	1/2048			
					0xa	1/1024			
					0x9	1/512			
					0x8	1/256			
					0x7	1/128			
					0x6	1/64			
					0x5	1/32			
					0x4	1/16			
					0x3	1/8			
					0x2	1/4			
					0x1	1/2			
					0x0	1/1			

D[15:4] Reserved

D[3:0] DF[3:0]: Count Clock Division Ratio Select Bits

Selects a PCLK division ratio to generate the count clock.

Table 11.9.2 PCLK Division Ratio Selection

DF[3:0]	Division ratio	DF[3:0]	Division ratio
0xf	Reserved	0x7	1/128
0xe	1/16384	0x6	1/64
0xd	1/8192	0x5	1/32
0xc	1/4096	0x4	1/16
0xb	1/2048	0x3	1/8
0xa	1/1024	0x2	1/4
0x9	1/512	0x1	1/2
0x8	1/256	0x0	1/1

(Default: 0x0)

Note: Make sure the counter is halted before setting the count clock.

T8 Ch.*x* Reload Data Registers (T8_TR*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T8 Ch.*x* Reload Data Register (T8_TR*x*)	**0x4242** **0x4262** (16 bits)	D15–8	**–**	reserved	–	–	–	0 when being read.
		D7–0	**TR[7:0]**	Reload data TR7 = MSB TR0 = LSB	0x0 to 0xff	0x0	R/W	

D[15:8] Reserved

D[7:0] TR[7:0]: Reload Data Bits

Sets the counter initial value. (Default: 0x0)

The reload data set in this register is preset to the counter when the timer is reset or the counter underflows. If the timer is started after resetting, it counts down from the reload value (initial value). This means that the reload value and the input clock frequency determine the time elapsed from the point at which the timer starts until the underflow occurs (or between underflows). The time determined is used to obtain the desired wait time, the intervals between periodic interrupts, and the programmable serial interface transfer clock.

T8 Ch.*x* Counter Data Registers (T8_TC*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T8 Ch.*x* Counter Data Register (T8_TC*x*)	**0x4244** **0x4264** (16 bits)	D15–8	**–**	reserved	–	–	–	0 when being read.
		D7–0	**TC[7:0]**	Counter data TC7 = MSB TC0 = LSB	0x0 to 0xff	0xff	R	

D[15:8] Reserved

D[7:0] TC[7:0]: Counter Data Bits

The counter data can be read out. (Default: 0xff)

This register is read-only and cannot be written to.

T8 Ch.*x* Control Registers (T8_CTL*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T8 Ch.*x* Control Register (T8_CTL*x*)	**0x4246** **0x4266** (16 bits)	D15–5	**–**	reserved	–	–	–	Do not write 1.
		D4	**TRMD**	Count mode select	1 One shot 0 Repeat	0	R/W	
		D3–2	**–**	reserved	–	–	–	0 when being read.
		D1	**PRESER**	Timer reset	1 Reset 0 Ignored	0	W	
		D0	**PRUN**	Timer run/stop control	1 Run 0 Stop	0	R/W	

D[15:5] Reserved (Do not write 1.)

D4 TRMD: Count Mode Select Bit

Selects the count mode.

1 (R/W): One-shot mode

0 (R/W): Repeat mode (default)

Setting TRMD to 0 sets the timer to repeat mode. In this mode, once the count starts, the timer continues to run until stopped by the application program. When the counter underflows, the timer presets the counter to the reload data register value and continues the count. Thus, the timer periodically outputs an underflow pulse. Set the timer to this mode to generate periodic interrupts or to generate a serial transfer clock.

Setting TRMD to 1 sets the timer to one-shot mode. In this mode, the 8-bit timer stops automatically as soon as the counter underflows. This means only one interrupt can be generated after the timer starts. Note that the timer presets the counter to the reload data register value, then stops when an underflow occurs. Set the timer to this mode to set a specific wait time.

D[3:2] Reserved

D1 PRESER: Timer Reset Bit

Resets the timer.

1 (W): Reset

0 (W): Ignored

0 (R): Always 0 when read (default)

Writing 1 to this bit presets the counter to the reload data value.

D0 PRUN: Timer Run/Stop Control Bit

Controls the timer RUN/STOP.
1 (R/W): Run
0 (R/W): Stop (default)

The timer starts counting when PRUN is written as 1 and stops when written as 0. When the timer is stopped, the counter data is retained until reset or until the next RUN state.

T8 Ch.*x* Interrupt Control Registers (T8_INT*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T8 Ch.*x* Interrupt Control Register (T8_INT*x*)	0x4248 0x4268 (16 bits)	D15–9	–	reserved	–	–	–	0 when being read.
		D8	T8IE	T8 interrupt enable	1 Enable 0 Disable	0	R/W	
		D7–1	–	reserved	–	–	–	0 when being read.
		D0	T8IF	T8 interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.

D[15:9] Reserved

D8 T8IE: T8 Interrupt Enable Bit

Enables or disables interrupts caused by counter underflows for each channel.
1 (R/W): Interrupt enabled
0 (R/W): Interrupt disabled (default)

Setting T8IE to 1 enables T8 interrupt requests to the ITC; setting to 0 disables interrupts.

D[7:1] Reserved

D0 T8IF: T8 Interrupt Flag Bit

Indicates whether the cause of counter underflow interrupt has occurred or not.
1 (R): Cause of interrupt has occurred
0 (R): No cause of interrupt has occurred (default)
1 (W): Flag is reset
0 (W): Ignored

T8IF is the T8 module interrupt flag that is set to 1 when the counter underflows.
T8IF is reset by writing 1.

12 16-bit PWM Timers (T16A2)

12.1 T16A2 Module Overview

The S1C17F57 includes a 16-bit PWM timer (T16A2) module that consists of counter blocks and comparator/capture blocks. This timer can be used as an interval timer, PWM waveform generator, external event counter and a count capture unit to measure external event periods.
The features of T16A2 are listed below.

- Two channels of 16-bit up counter blocks
- Two channels of comparator/capture blocks to which a counter block to be connected is selectable
- Allows selection of a count clock asynchronously with the CPU clock.
- Supports event counter function using an external clock.
- The comparator compares the counter value with two specified comparison values to generate interrupts and a PWM waveform.
- The capture unit captures counter values using two external trigger signals and generates interrupts.

Figure 12.1.1 shows the T16A2 configuration.

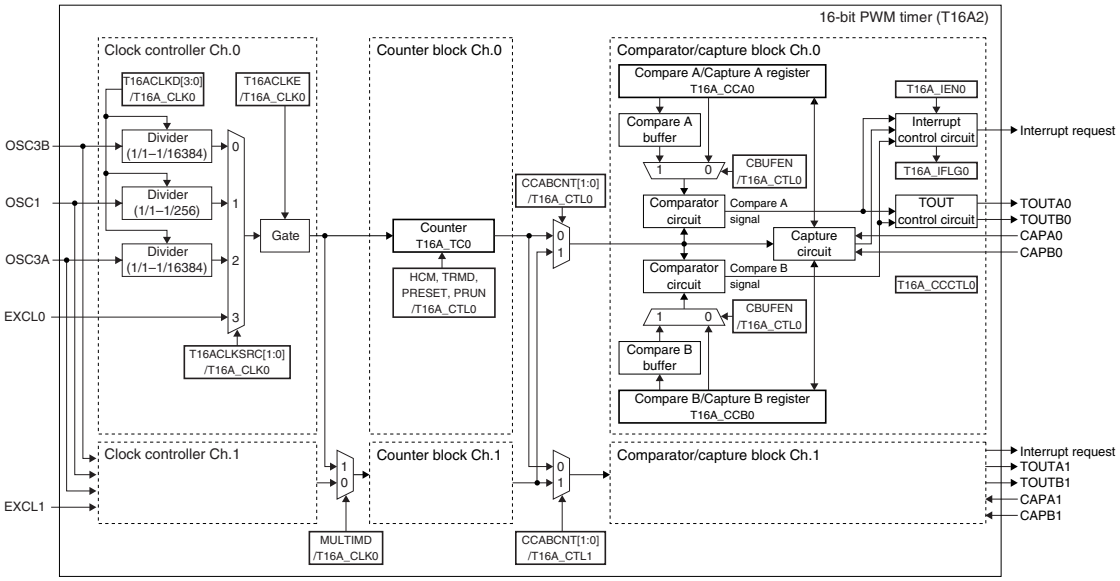


Figure 12.1.1 T16A2 Configuration

Clock controller

T16A2 includes two channels of clock controllers that generate the count clock for the counters. The clock source and division ratio can be selected with software.

Counter block

The counter block includes a 16-bit up-counter that operates with an OSC3B, OSC3A, or OSC1 division clock, or the external count clock input from outside the IC. The T16A2 module allows software to run and stop the counter of each channel, and to reset the counter value (cleared to 0) as well as selection of the count clock. The counter can also be reset by the compare B signal output from the comparator/capture block.

Comparator/capture block

The comparator/capture block provides a counter comparison function (comparator mode) and a count capture function (capture mode). When comparator mode is selected via software, the comparator/capture block can be used as a PWM waveform or clock generator. When capture mode is selected, this block can be used as a count capture unit for measuring external event periods/cycles. The comparator circuit generates the compare A and B signals that represent matching between compare A/B register values (set via software) and the counter value, and outputs the signals to the TOUT control circuit and the interrupt control circuit. The TOUT control circuit generates a PWM or other signal from the compare A and B signals and outputs it to the external TOUTA*x* and TOUTB*x* pins. The capture circuit loads the counter value to the capture A or B register using the CAPA*x* or CAPB*x* input signal that represents external events issued as a trigger. The interrupt control circuit outputs an interrupt signal to the interrupt controller (ITC) module according to the interrupt condition that has been set.
Comparator mode and capture mode cannot be used simultaneously in the same channel.

Combination of counter block channel and comparator/capture block channel

Generally, a counter block is connected to the comparator/capture block with the same channel number. The counter block and the comparator/capture block in different channels can also be connected. This allows a counter to use two or more comparator/capture blocks for expanding the comparison/capturing function from two systems to maximum four systems (details are described later).

Note: The letter '*x*' in register and pin names refers to a channel number (0 or 1).
Example: T16A_CTL*x* register, TOUTA*x* pin
Ch.0: T16A_CTL0 register, TOUTA0
Ch.1: T16A_CTL1 register, TOUTA1

12.2 T16A2 Input/Output Pins

Table 12.2.1 lists the input/output pins for the T16A2 module.

Table 12.2.1 List of T16A2 Pins

Pin name	I/O	Qty	Function
EXCL0 (for Ch.0) EXCL1 (for Ch.1)	I	2	External clock input pins Inputs an external clock for the event counter function.
CAPA0, CAPB0 (for Ch.0) CAPA1, CAPB1 (for Ch.1)	I	4	Counter-capture trigger signal input pins (effective in capture mode) The specified edge (falling edge, rising edge, or both) of the signal input to the CAPA*x* pin captures the counter data into the capture A register. The CAPB*x* pin input signal captures the counter data into the capture B register.
TOUTA0, TOUTB0 (for Ch.0) TOUTA1, TOUTB1 (for Ch.1)	O	4	Timer generating signal output pins (effective in comparator mode) Each channel has two output pins and the signals generated in different conditions can be output.

The T16A2 input/output pins (EXCL*x*, CAPA*x*, CAPB*x*, TOUTA*x*, and TOUTB*x*) are shared with I/O ports and are initially set as general purpose I/O port pins. The pin functions must be switched using the port function select bits to use the general purpose I/O port pins as T16A2 input/output pins.
For detailed information on pin function switching, see the "I/O Ports (P)" chapter.

12.3 Count Clock

The clock controller includes a clock source selector, dividers, and a gate circuit for controlling the count clock. The count clock can be controlled in each channel individually.

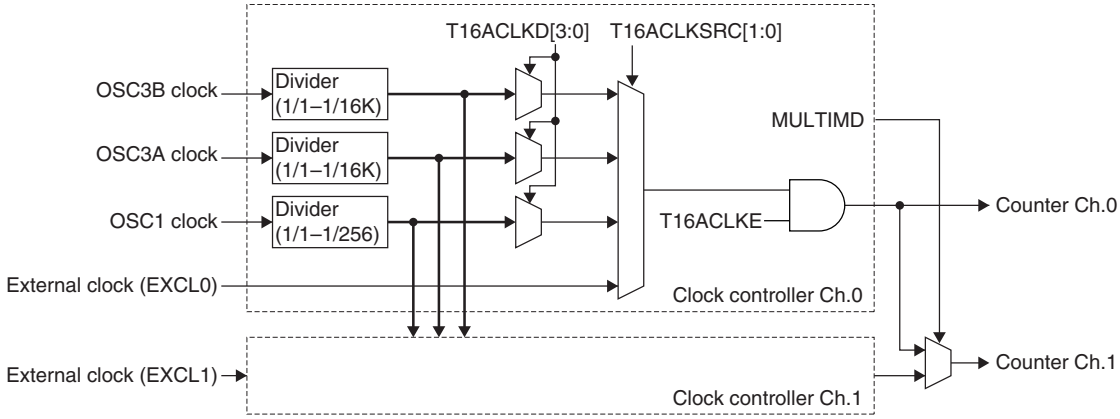


Figure 12.3.1 Clock Controller

Clock source selection

The clock source can be selected from OSC3B, OSC3A, OSC1, or external clock using T16ACLKSRC[1:0]/T16A_CLK*x* register.

Table 12.3.1 Clock Source Selection

T16ACLKSRC[1:0]	Clock source
0x3	External clock (EXCL*x*)
0x2	OSC3A
0x1	OSC1
0x0	OSC3B

(Default: 0x0)

When external clock is selected, the timer can be used as an event counter or for measuring pulse widths by inputting an external clock or pulses. The table below lists the external clock input pins. It is not necessary to switch their pin functions from general-purpose I/O port. However, do not set the I/O port to output mode.

Table 12.3.2 External Clock Input Pins

Channel	External clock input pin
T16A2 Ch.0	EXCL0
T16A2 Ch.1	EXCL1

Internal clock division ratio selection

When an internal clock (OSC3B, OSC3A, or OSC1) is selected, use T16ACLKD[3:0]/T16A_CLK*x* register to select the division ratio.

Table 12.3.3 Internal Clock Division Ratio Selection

T16ACLKD[3:0]	Division ratio	
	Clock source = OSC3B or OSC3A	Clock source = OSC1
0xf	Reserved	
0xe	1/16384	Reserved
0xd	1/8192	Reserved
0xc	1/4096	Reserved
0xb	1/2048	Reserved
0xa	1/1024	Reserved
0x9	1/512	F256 (Regulated 256 Hz clock)
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	

(Default: 0x0)

Clock enable

Clock supply to the counter is controlled using T16ACLKE/T16A_CLK*x* register. The T16ACLKE default setting is 0, which disables the clock supply. Setting T16ACLKE to 1 sends the clock generated as above to the counter. If T16A2 is not required, disable the clock supply to reduce current consumption.

Multi-comparator/capture mode

The T16A2 module allows a counter channel to be connected to multiple comparator/capture channels (multi-comparator/capture mode). In this case, all channels must be clocked with the Ch.0 clock. Use MULTIMD/T16A_CLK0 register to supply the Ch.0 clock to all channels. When using T16A2 in multi-comparator/capture mode, set MULTIMD to 1. When connecting the counter and comparator/capture block in the same channel (normal channel mode), set MULTIMD to 0 (default).

Note: Make sure the T16A2 count is stopped before setting the count clock.

12.4 T16A2 Operating Modes

The T16A2 module provides some operating modes to support various usages. This section describes the functions of each operating mode and how to enter the mode.

12.4.1 Comparator Mode and Capture Mode

The T16A_CCA*x* and T16A_CCB*x* registers that are embedded in the comparator/capture block can be set to comparator mode or capture mode, individually. The T16A_CCA*x* register mode is selected using CCAMD/T16A_CCCTL*x* register and the T16A_CCB*x* register mode is selected using CCBMD/T16A_CCCTL*x* register.

Comparator mode (CCAMD/CCBMD = 0, default)

The comparator mode compares the counter value and the comparison value set via software. It generates an interrupt and toggles the timer output signal level when the values are matched. The T16A_CCA*x* and T16A_CCB*x* registers function as the compare A and compare B registers that are used for loading compare values in this mode.

When the counter reaches the value set in the compare A register during counting, the comparator asserts the compare A signal. At the same time the compare A interrupt flag is set and the interrupt signal of the timer channel is output to the ITC if the interrupt has been enabled.

When the counter reaches the value set in the compare B register, the comparator asserts the compare B signal. At the same time the compare B interrupt flag is set and the interrupt signal of the timer channel is output to the ITC if the interrupt is enabled. Furthermore, the counter is reset to 0.

The compare A period (time from start of counting to occurrence of a compare A interrupt) and the compare B period (time from start of counting to occurrence of a compare B interrupt) can be calculated as follows:

Compare A period = (CCA + 1) / ct_clk [second]
Compare B period = (CCB + 1) / ct_clk [second]

CCA: Compare A register value set (0 to 65535)
CCB: Compare B register value set (0 to 65535)
ct_clk: Count clock frequency [Hz]

The compare A and compare B signals are also used to generate a timer output waveform (TOUT). See Section 12.6, "Timer Output Control," for more information.

To generate PWM waveform, the T16A_CCA*x* and T16A_CCB*x* registers must be both placed into comparator mode.

Compare buffers

The compare buffer is used to synchronize the comparison data update timings and the counter operation. Setting CBUFEN/T16A_CTL*x* register to 1 enables the compare buffer. The compare A and B signals will be generated by comparing the counter values with the compare A and B buffer values instead of the compare A and B register values. The compare A and B register values written via software are loaded to the compare A and B buffers when the compare B signal is generated.

Note: When writing data to the T16A_CCA*x* or T16A_CCB*x* register successively, data should be written at intervals of one or more T16A2 count clock cycles.

Capture mode (CCAMD/CCBMD = 1)

The capture mode captures the counter value when an external event such as a key entry occurs (at the specified edge of the external input signal). In this mode, the T16A_CCA*x* and/or T16A_CCB*x* registers function as the capture A and/or capture B registers.
The table below lists the input pins of the external trigger signals used for capturing counter values. The pin function of the corresponding ports must be switched for trigger input in advance. See the "I/O Ports (P)" chapter for switching the pin function.

Table 12.4.1.1 List of Counter Capture Trigger Signal Input Pins

Channel	Trigger input pins	
	Capture A	Capture B
T16A2 Ch.0	CAPA0	CAPB0
T16A2 Ch.1	CAPA1	CAPB1

The trigger edge of the signal can be selected using the CAPATRG[1:0]/T16A_CCCTL*x* register for capture A and CAPBTRG[1:0]/T16A_CCCTL*x* register for capture B.

Table 12.4.1.2 Capture Trigger Edge Selection

CAPATRG[1:0]/ CAPBTRG[1:0]	Trigger edge
0x3	Falling edge and rising edge
0x2	Falling edge
0x1	Rising edge
0x0	Not triggered

(Default: 0x0)

When a specified trigger edge is input during counting, the current counter value is loaded to the capture register. At the same time the capture A or capture B interrupt flag is set and the interrupt signal of the timer channel is output to the ITC if the interrupt has been enabled. This interrupt can be used to read the captured data from the T16A_CCA*x* or T16A_CCB*x* register. For example, external event cycles and pulse widths can be measured from the difference between two captured counter values read.
If the captured data is overwritten by the next trigger when the capture A or capture B interrupt flag has already been set, the overwrite interrupt flag will be set. This interrupt can be used to execute an overwrite error handling. To avoid occurrence of unnecessary overwrite interrupt, the capture A or capture B interrupt flag must be reset after the captured data has been read from the T16A_CCA*x* or T16A_CCB*x* register.

Notes:
- The correct captured data may not be obtained if the captured data is read at the same time the next value is being captured. Read the capture register twice to check if the read data is correct as necessary.
- To capture counter data properly, both the High and Low period of the CAP*x* trigger signal must be longer than the source clock cycle time.

The setting of CAPATRG[1:0] or CAPBTRG[1:0] is ineffective in comparator mode. No counter capturing operation will be performed even if a trigger edge is specified.
The capture mode cannot generate/output the TOUT signal as no compare signal is generated.

12.4.2 Repeat Mode and One-Shot Mode

Each counter features two count modes: repeat mode and one-shot mode. The count mode is selected using TRMD /T16A_CTL*x* register.

Repeat mode (TRMD = 0, default)

Setting TRMD to 0 sets the corresponding counter to repeat mode.
In this mode, once the count starts, the counter continues running until stopped by the application program. The counter continues the count even if the counter returns to 0 due to a counter overflow. The counter should be set to this mode to generate periodic interrupts at desired intervals or to generate a timer output waveform.

One-shot mode (TRMD = 1)

Setting TRMD to 1 sets the corresponding counter to one-shot mode.

In this mode, the counter stops automatically as soon as the compare B signal is generated. The counter should be set to this mode to set a specific wait time or for pulse width measurement.

12.4.3 Normal Channel Mode and Multi-Comparator/Capture Mode

One channel of the T16A2 module basically consists of a counter block and a comparator/capture block. The T16A2 module also allows the application to use expanded comparator/capture function by connecting two or more comparator/capture blocks to one counter block. To support this expansion, two operating modes are provided: normal channel mode and multi-comparator/capture mode. This operating mode can be selected using MULTIMD/ T16A_CLK0 register.

Normal channel mode (MULTIMD = 0, default)

Set the T16A2 module to this mode when using it as two channels of different timers by connecting a counter block with the comparator/capture block of the same channel. In this mode, the counters can use different count clocks.

Each timer channel provides CCABCNT[1:0]/T16A_CTL*x* register to select a counter channel to be connected to the comparator/capture block.

Table 12.4.3.1 Counter Selection

CCABCNT[1:0]	Counter channel
0x3, 0x2	Reserved
0x1	Ch.1 (Counter 1)
0x0	Ch.0 (Counter 0)

(Default: 0x0)

When using the T16A2 module in normal channel mode, be sure to connect the counter block to the comparator/capture block in the same channel.

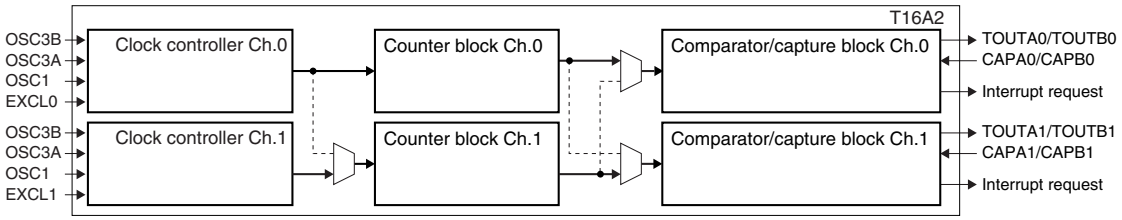


Figure 12.4.3.1 Timer Configuration in Normal Channel Mode

Note: Do not connect a counter block to a comparator/capture block in a different channel in normal channel mode (MULTIMD = 0), as normal operation cannot be guaranteed.

Multi-comparator/capture mode (MULTIMD = 1)

In order to set three or more comparison values for one counter or to capture the contents of one counter using three or more trigger signals, two or more comparator/capture blocks can be connected to one counter. Multi-comparator/capture mode is provided for this purpose. In this mode, any counter block can be combined with the comparator/capture blocks using CCABCNT[1:0] described above. Note, however, that the count clock is fixed at one type for counter Ch.0, regardless of the counter to be used. The clock settings for other channel are ineffective.

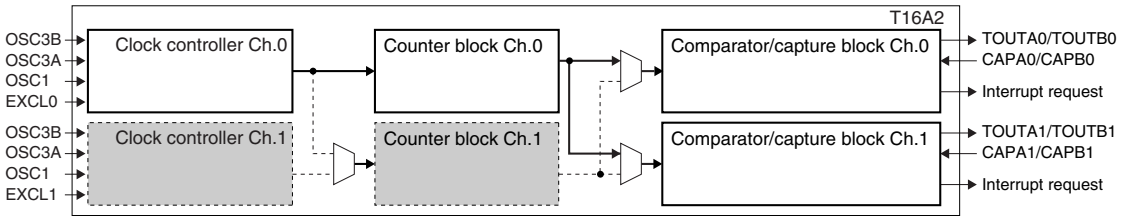


Figure 12.4.3.2 Timer Configuration in Multi-Comparator/Capture Mode

12.4.4 Normal Clock Mode and Half Clock Mode

T16A2 supports half clock mode to control the duty ratio of the PWM output waveform with high accuracy. In half clock mode, T16A2 uses the dual-edge counter, which counts at the rising and falling edges of the count clock, to compare with the compare A register. This makes it possible to control the duty ratio with double accuracy as compared to normal clock mode.
Use HCM/T16A_CTL*x* register to select half clock mode.

Normal clock mode (HCM = 0, default)

In normal clock mode, T16A2 generates a compare A signal when the T16A_TC*x* register value matches the T16A_CCA*x* register.

Half clock mode (HCM = 1)

In half clock mode, T16A2 generates a compare A signal when the dual-edge counter value matches the T16A_CCA*x* register.

Notes:
- T16A2 must be placed into comparator mode to set half clock mode, as it is effective only when PWM waveform is generated.
  Be sure to set T16A2 to normal clock mode (HCM = 0) under a condition shown below.
  (1) When T16A2 is placed into capture mode
  (2) When TOUTAMD/T16A_CCCTL*x* register is set to 0x2 or 0x3
  (3) When TOUTBMD/T16A_CCCTL*x* register is set to 0x2 or 0x3
- The dual-edge counter value cannot be read.
- Do not use the compare A interrupt in half clock mode.
- In half clock mode, the T16A_CCB*x* register setting value must be less than [T16A_CCA*x* setting value / 2 + 0x8000].

12.5 Counter Control

12.5.1 Counter Reset

The counter can be reset to 0 by writing 1 to PRESET/T16A_CTL*x* register.

Normally, the counter should be reset by writing 1 to this bit before starting the count.
The counter is reset by the hardware if the counter reaches the compare B register value after the count starts.

12.5.2 Counter RUN/STOP Control

Make the following settings before starting the count operation.
(1) Switch the input/output pin functions to be used for T16A2. Refer to the "I/O Port (P)" chapter.
(2) Select operating modes. See Section 12.4.
(3) Select the clock source. See Section 12.3.
(4) Configure the timer outputs (TOUT). See Section 12.6.
(5) If using interrupts, set the interrupt level and enable the T16A2 interrupts. See Section 12.7.
(6) Reset the counter to 0. See Section 12.5.1.
(7) Set comparison data (in comparator mode). See Section 12.4.1.

Each timer channel provides PRUN/T16A_CTL*x* register to control the counter operation.
The counter starts counting when 1 is written to PRUN. Writing 0 to PRUN disables clock input and stops the count.
This control does not affect the counter data. The counter data is retained even when the count is halted, allowing resumption of the count from that data.
If PRUN and PRESET are written as 1 simultaneously, the counter starts counting after reset.

12.5.3 Reading Counter Values

The counter value can be read from T16A2TC[15:0]/T16A_TC*x* register even if the counter is running. However, the counter value should be read at once using a 16-bit transfer instruction. If data is read twice using an 8-bit transfer instruction, the correct value may not be obtained due to occurrence of count up between readings.

12.5.4 Counter Operation and Interrupt Timing Charts

Comparator mode

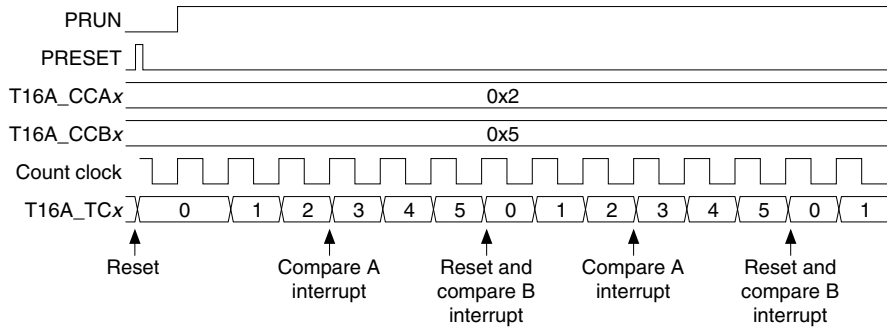


Figure 12.5.4.1 Operation Timing in Comparator Mode

Capture mode

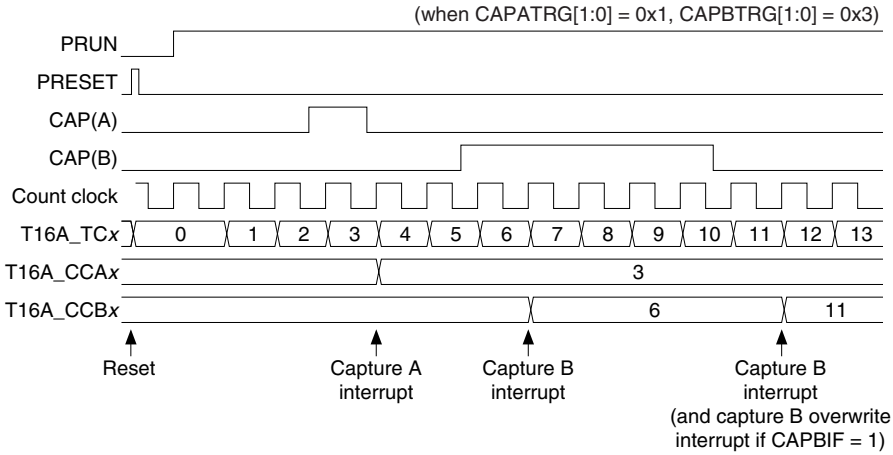


Figure 12.5.4.2 Operation Timing in Capture Mode

12.6 Timer Output Control

The timer that has been set in comparator mode can generate TOUT signals using the compare A and compare B signals and can output it to external devices. Each timer channel provides two TOUT outputs, thus the T16A2 module can output up to four TOUT signals. Figure 12.6.1 shows the TOUT output circuit (one timer channel).

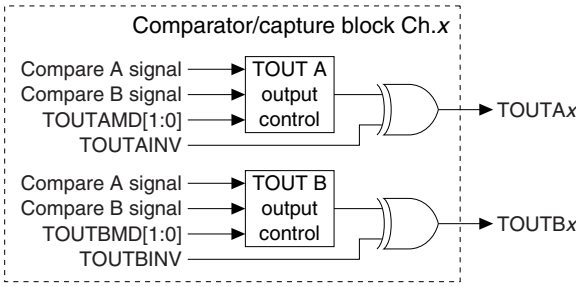


Figure 12.6.1 TOUT Output Circuit

Each timer channel includes two TOUT output circuits and their signal generation and output can be controlled individually. Although the output circuit and register names use letters 'A' and 'B' to distinguish two systems, it does not mean that they correspond to compare A and B signals.

TOUT output pins

Table 12.6.1 lists correspondence between the TOUT pins and the timer channels. The pin function of the corresponding ports must be switched for TOUT output in advance. See the "I/O Ports (P)" chapter for switching the pin function.

Table 12.6.1 List of TOUT Output Pins

Channel	TOUT output pin	
	System A	System B
T16A2 Ch.0	TOUTA0	TOUTB0
T16A2 Ch.1	TOUTA1	TOUTB1

TOUT generation mode

TOUTAMD[1:0]/T16A_CCCTL*x* register (for system A) or TOUTBMD[1:0]/T16A_CCCTL*x* register (for system B) is used to set how the TOUT signal is changed by the compare A and compare B signals.

Table 12.6.2 TOUT Generation Mode

TOUTAMD[1:0]/ TOUTBMD[1:0]	When compare A occurs	When compare B occurs
0x3	No change	Toggle
0x2	Toggle	No change
0x1	Rise	Fall
0x0	Disable output	

(Default: 0x0)

TOUTAMD[1:0] and TOUTBMD[1:0] are also used to turn the TOUT outputs On and Off.

TOUT signal polarity selection

By default, an active High output signal is generated. This logic can be inverted using TOUTAINV/T16A_CCCTL*x* register (for system A) or TOUTBINV/T16A_CCCTL*x* register (for system B). Writing 1 to TOUTAINV/TOUTBINV sets the timer to generate an active Low TOUT signal.

Resetting the counter sets the TOUT signal to the inactive level.

Figure 12.6.2 illustrates the TOUT output waveform.

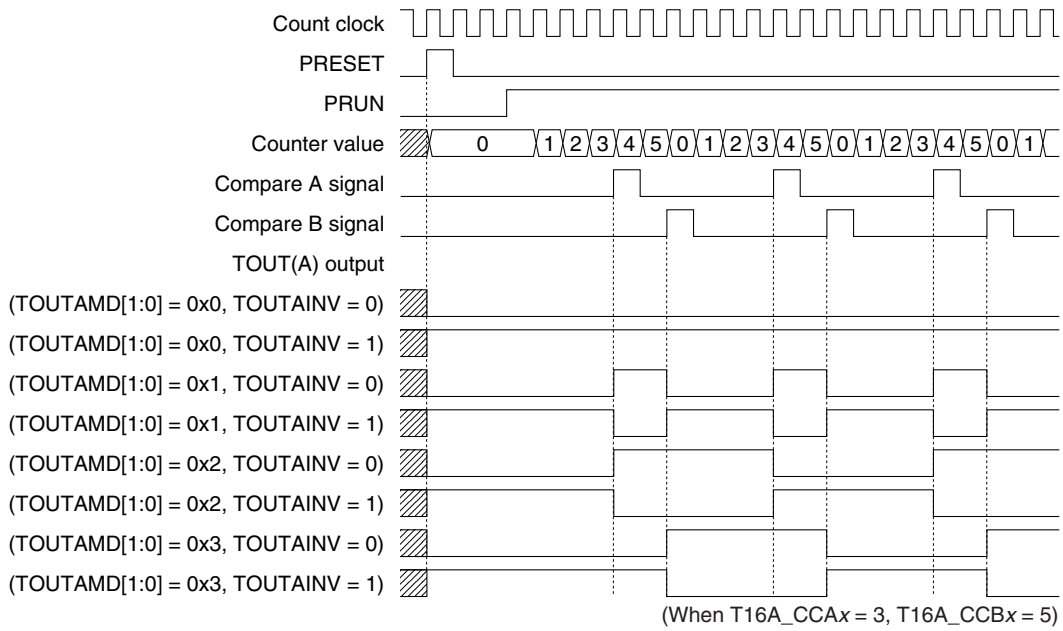
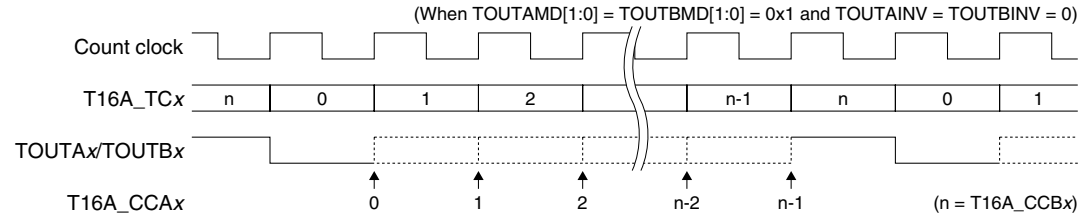


Figure 12.6.2 TOUT Output Waveform

PWM waveform output timings

Normal clock mode (HCM = 0)



Example: HCM = 0, T16A_CCA*x* = 1, and T16A_CCB*x* = 5

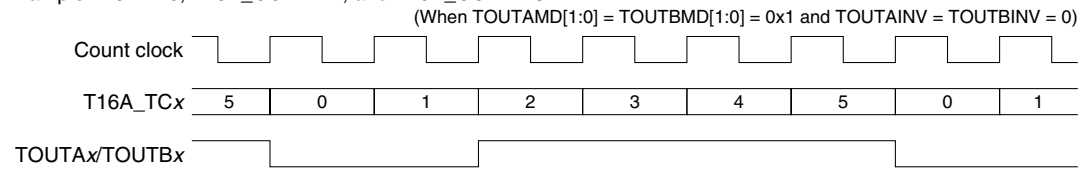
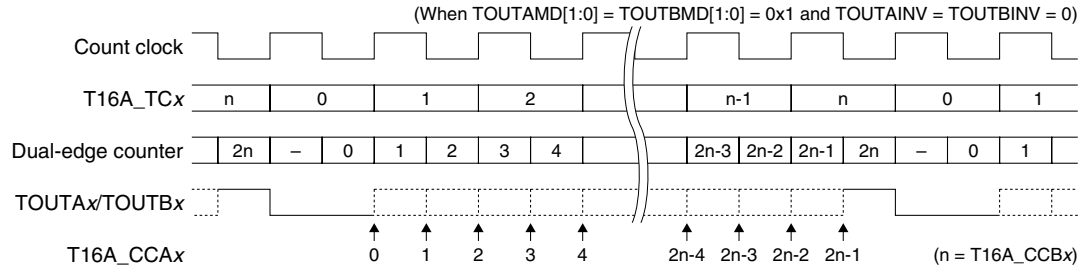


Figure 12.6.3 PWM Waveform Output Timings in Normal Clock Mode

Half clock mode (HCM = 1)



Example: HCM = 1, T16A_CCA*x* = 1, and T16A_CCB*x* = 5

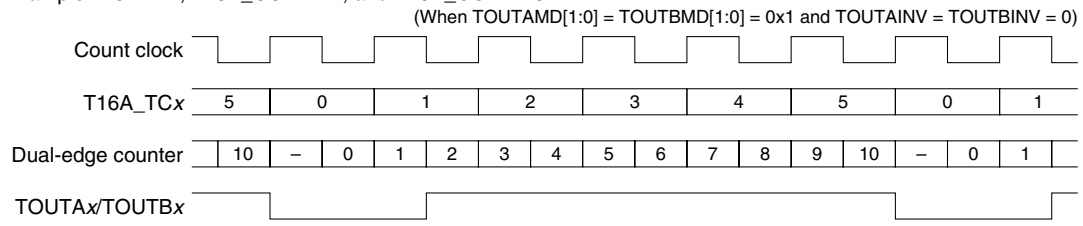


Figure 12.6.4 PWM Waveform Output Timings in Half Clock Mode

12.7 T16A2 Interrupts

The T16A2 module can generate the following six kinds of interrupts:

- Compare A interrupt (in comparator mode)
- Compare B interrupt (in comparator mode)
- Capture A interrupt (in capture mode)
- Capture B interrupt (in capture mode)
- Capture A overwrite interrupt (in capture mode)
- Capture B overwrite interrupt (in capture mode)

Each timer channel outputs a single interrupt signal shared by the above interrupt causes to the interrupt controller (ITC). Read the interrupt flags in the T16A2 module to identify the interrupt cause that has been occurred.

Interrupts in comparator mode

Compare A interrupt

This interrupt request is generated when the counter matches the compare A register value during counting in comparator mode. It sets the interrupt flag CAIF/T16A_IFLG*x* register in the T16A2 module to 1.
To use this interrupt, set CAIE/T16A_IEN*x* register to 1. If CAIE is set to 0 (default), interrupt requests for this cause is not sent to the ITC.

Compare B interrupt

This interrupt request is generated when the counter matches the compare B register value during counting in comparator mode. It sets the interrupt flag CBIF/T16A_IFLG*x* register in the T16A2 module to 1.
To use this interrupt, set CBIE/T16A_IEN*x* register to 1. If CBIE is set to 0 (default), interrupt requests for this cause is not sent to the ITC.

Interrupts in capture mode

Capture A interrupt

This interrupt request is generated when the counter value is captured in the capture A register by an external trigger during counting in capture mode. It sets the interrupt flag CAPAIF/T16A_IFLG*x* register in the T16A2 module to 1.
To use this interrupt, set CAPAIE/T16A_IEN*x* register to 1. If CAPAIE is set to 0 (default), interrupt requests for this cause is not sent to the ITC.

Capture B interrupt

This interrupt request is generated when the counter value is captured in the capture B register by an external trigger during counting in capture mode. It sets the interrupt flag CAPBIF/T16A_IFLG*x* register in the T16A2 module to 1.
To use this interrupt, set CAPBIE/T16A_IEN*x* register to 1. If CAPBIE is set to 0 (default), interrupt requests for this cause is not sent to the ITC.

Capture A overwrite interrupt

This interrupt request is generated if the capture A register is overwritten by a new external trigger when the capture A interrupt flag CAPAIF has been set (a counter value has already been loaded to the capture A register). It sets the interrupt flag CAPAOWIF/T16A_IFLG*x* register in the T16A2 module to 1.
To use this interrupt, set CAPAOWIE/T16A_IEN*x* register to 1. If CAPAOWIE is set to 0 (default), interrupt requests for this cause is not sent to the ITC.
CAPAOWIF will be set if the capture A register is overwritten when CAPAIF has been set regardless of whether the capture A register has been read or not. Therefore, be sure to reset CAPAIF immediately after the capture A register is read.

Capture B overwrite interrupt

This interrupt request is generated if the capture B register is overwritten by a new external trigger when the capture B interrupt flag CAPBIF has been set (a counter value has already been loaded to the capture B register). It sets the interrupt flag CAPBOWIF/T16A_IFLG*x* register in the T16A2 module to 1.
To use this interrupt, set CAPBOWIE/T16A_IEN*x* register to 1. If CAPBOWIE is set to 0 (default), interrupt requests for this cause is not sent to the ITC.
CAPBOWIF will be set if the capture B register is overwritten when CAPBIF has been set regardless of whether the capture B register has been read or not. Therefore, be sure to reset CAPBIF immediately after the capture B register is read.

If the interrupt flag is set to 1 when the interrupt has been enabled, the T16A2 module outputs an interrupt request to the ITC. An interrupt is generated if the ITC and S1C17 core interrupt conditions are satisfied.
For more information on interrupt control registers and the operation when an interrupt occurs, see the “Interrupt Controller (ITC)” chapter.

Notes: • Reset the interrupt flag before enabling interrupts with the interrupt enable bit to prevent occurrence of unwanted interrupt. The interrupt flag is reset by writing 1.

• After an interrupt occurs, the interrupt flag in the T16A2 module must be reset in the interrupt handler routine.

12.8 Control Register Details

Table 12.8.1 List of T16A2 Registers

Address	Register name		Function
0x5068	T16A_CLK0	T16A Clock Control Register Ch.0	Controls the T16A2 Ch.0 clock.
0x5069	T16A_CLK1	T16A Clock Control Register Ch.1	Controls the T16A2 Ch.1 clock.
0x5400	T16A_CTL0	T16A Counter Ch.0 Control Register	Controls the counter.
0x5402	T16A_TC0	T16A Counter Ch.0 Data Register	Counter data
0x5404	T16A_CCCTL0	T16A Comparator/Capture Ch.0 Control Register	Controls the comparator/capture block and TOUT.
0x5406	T16A_CCA0	T16A Compare/Capture Ch.0 A Data Register	Compare A/capture A data
0x5408	T16A_CCB0	T16A Compare/Capture Ch.0 B Data Register	Compare B/capture B data
0x540a	T16A_IEN0	T16A Compare/Capture Ch.0 Interrupt Enable Register	Enables/disables interrupts.
0x540c	T16A_IFLG0	T16A Compare/Capture Ch.0 Interrupt Flag Register	Displays/sets interrupt occurrence status.
0x5420	T16A_CTL1	T16A Counter Ch.1 Control Register	Controls the counter.
0x5422	T16A_TC1	T16A Counter Ch.1 Data Register	Counter data
0x5424	T16A_CCCTL1	T16A Comparator/Capture Ch.1 Control Register	Controls the comparator/capture block and TOUT.
0x5426	T16A_CCA1	T16A Compare/Capture Ch.1 A Data Register	Compare A/capture A data
0x5428	T16A_CCB1	T16A Compare/Capture Ch.1 B Data Register	Compare B/capture B data
0x542a	T16A_IEN1	T16A Compare/Capture Ch.1 Interrupt Enable Register	Enables/disables interrupts.
0x542c	T16A_IFLG1	T16A Compare/Capture Ch.1 Interrupt Flag Register	Displays/sets interrupt occurrence status.

The T16A2 registers are described in detail below.

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

T16A Clock Control Registers Ch.*x* (T16A_CLK*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16A Clock Control Register Ch.*x* (T16A_CLK*x*)	**0x5068 0x5069** (8 bits)	D7–4	**T16ACLKD [3:0]**	Clock division ratio select	T16ACLKD[3:0]: 0xf reserved / reserved; 0xe 1/16384 / reserved; 0xd 1/8192 / reserved; 0xc 1/4096 / reserved; 0xb 1/2048 / reserved; 0xa 1/1024 / reserved; 0x9 1/512 / F256; 0x8 1/256 / 1/256; 0x7 1/128 / 1/128; 0x6 1/64 / 1/64; 0x5 1/32 / 1/32; 0x4 1/16 / 1/16; 0x3 1/8 / 1/8; 0x2 1/4 / 1/4; 0x1 1/2 / 1/2; 0x0 1/1 / 1/1 (Division ratio: OSC3A or OSC3B / OSC1)	0x0	R/W	
		D3–2	**T16ACLK SRC[1:0]**	Clock source select	T16ACLKSRC[1:0]: 0x3 External clock; 0x2 OSC3A; 0x1 OSC1; 0x0 OSC3B (Clock source)	0x0	R/W	
		D1	**MULTIMD**	Multi-comparator/capture mode select	1 Multi / 0 Normal	0	R/W	T16A_CLK0
		–	–	reserved	–	–	–	T16A_CLK1 0 when being read.
		D0	**T16ACLKE**	Count clock enable	1 Enable / 0 Disable	0	R/W	

D[7:4] T16ACLKD[3:0]: Clock Division Ratio Select Bits

Selects the division ratio for generating the count clock when an internal clock (OSC3B, OSC3A, or OSC1) is used.

Table 12.8.2 Internal Clock Division Ratio Selection

T16ACLKD[3:0]	Division ratio	
	Clock source = OSC3B or OSC3A	Clock source = OSC1
0xf	Reserved	
0xe	1/16384	Reserved
0xd	1/8192	Reserved
0xc	1/4096	Reserved
0xb	1/2048	Reserved
0xa	1/1024	Reserved
0x9	1/512	F256 (Regulated 256 Hz clock)
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	

(Default: 0x0)

D[3:2] T16ACLKSRC[1:0]: Clock Source Select Bits

Selects the count clock source.

Table 12.8.3 Clock Source Selection

T16ACLKSRC[1:0]	Clock source
0x3	External clock (EXCL*x*)
0x2	OSC3A
0x1	OSC1
0x0	OSC3B

(Default: 0x0)

When using an external clock as the count clock, supply the clock to the EXCL*x* pin.

D1 MULTIMD: Multi-Comparator/Capture Mode Select Bit (T16A_CLK0 register)

Sets the T16A2 module to multi-comparator/capture mode.

1 (R/W): Multi-comparator/capture mode

0 (R/W): Normal channel mode (default)

In multi-comparator/capture mode, the clock for Ch.0 configured in the T16A_CLK0 register is supplied to all timer channels. In normal channel mode, different clock configured for each channel individually is supplied to the respective counter.

D1 Reserved (T16A_CLK1 register)

D0 T16ACLKE: Count Clock Enable Bit

Enables or disables the count clock supply to the counter.

1 (R/W): Enabled (on)

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

The T16ACLKE default setting is 0, which disables the clock supply. Setting T16ACLKE to 1 sends the clock selected as above to the counter. If timer operation is not required, disable the clock supply to reduce current consumption.

T16A Counter Ch.*x* Control Registers (T16A_CTL*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16A Counter Ch.*x* Control Register (T16A_CTL*x*)	0x5400 0x5420 (16 bits)	D15–7	–	reserved	–	–	–	0 when being read.
		D6	HCM	Half clock mode enable	1 Enable 0 Disable	0	R/W	
		D5–4	CCABCNT [1:0]	Counter select	CCABCNT[1:0] Counter Ch. 0x3, 0x2 reserved 0x1 Ch.1 0x0 Ch.0	0x0	R/W	
		D3	CBUFEN	Compare buffer enable	1 Enable 0 Disable	0	R/W	
		D2	TRMD	Count mode select	1 One-shot 0 Repeat	0	R/W	
		D1	PRESET	Counter reset	1 Reset 0 Ignored	0	W	0 when being read.
		D0	PRUN	Counter run/stop control	1 Run 0 Stop	0	R/W	

D[15:7] Reserved

D6 HCM: Half Clock Mode Enable Bit

Sets T16A2 to half clock mode.
1 (R/W): Enabled (half clock mode)
0 (R/W): Disabled (normal clock mode) (default)

Setting HCM to 1 places T16A2 into half clock mode. In half clock mode, T16A2 uses the dual-edge counter, which counts at the rising and falling edges of the count clock, to generate a compare A signal when the dual-edge counter value matches the T16A_CCA*x* register. This makes it possible to control the duty ratio with double accuracy as compared to normal clock mode.
Setting HCM to 0 places T16A2 into normal clock mode. In normal clock mode, T16A2 generates a compare A signal when the T16A_TC*x* register value matches the T16A_CCA*x* register.

Notes:
- T16A2 must be placed into comparator mode to set half clock mode, as it is effective only when PWM waveform is generated.
  Be sure to set T16A2 to normal clock mode under a condition shown below.
  (1) When T16A2 is placed into capture mode
  (2) When TOUTAMD/T16A_CCCTL*x* register is set to 0x2 or 0x3
  (3) When TOUTBMD/T16A_CCCTL*x* register is set to 0x2 or 0x3
- The dual-edge counter value cannot be read.
- Do not use the compare A interrupt in half clock mode.
- In half clock mode, the T16A_CCB*x* register setting value must be less than [T16A_CCA*x* setting value / 2 + 0x8000].

D[5:4] CCABCNT[1:0]: Counter Select Bits

Selects a counter to be connected to the comparator/capture block of each channel in multi-comparator/capture mode (MULTIMD/T16A_CLK0 register = 1).

Table 12.8.4 Counter Selection

CCABCNT[1:0]	Counter channel
0x3, 0x2	Reserved
0x1	Ch.1 (Counter 1)
0x0	Ch.0 (Counter 0)

(Default: 0x0)

When using the T16A2 module in normal channel mode (T16A2MULTIMD = 0), be sure to connect the counter of the same channel to each comparator/capture block.

D3 CBUFEN: Compare Buffer Enable Bit

Enables or disables writing to the compare buffer.
1 (R/W): Enabled
0 (R/W): Disabled (default)

Setting CBUFEN to 1 enables the compare buffer. The compare A and B signals will be generated by comparing the counter values with the compare A and B buffer values instead of the compare A and B register values. The compare A and B register values written via software are loaded to the compare A and B buffers when the compare B signal is generated.

Setting CBUFEN to 0 disables the compare buffer. The compare A and B signals will be generated by comparing the counter values with the compare A and B register values.

Note: Make sure the counter is halted (PRUN = 0) before setting CBUFEN.

D2 TRMD: Count Mode Select Bit

Selects the count mode.
1 (R/W): One-shot mode
0 (R/W): Repeat mode (default)

Setting TRMD to 0 sets the counter to repeat mode. In this mode, once the count starts, the counter continues counting until stopped by the application program.
Setting TRMD to 1 sets the counter to one-shot mode. In this mode, the counter stops counting automatically as soon as the compare B signal is generated.

D1 PRESET: Counter Reset Bit

Resets the counter.
1 (W): Reset
0 (W): Ignored
0 (R): Normally 0 when read out (default)

Writing 1 to this bit resets the counter to 0.

D0 PRUN: Counter Run/Stop Control Bit

Starts/stops the count.
1 (W): Run
0 (W): Stop
1 (R): Counting
0 (R): Stopped (default)

The counter starts counting when PRUN is written as 1 and stops when written as 0. The counter data is retained even if the counter is stopped.

T16A Counter Ch.*x* Data Registers (T16A_TC*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16A Counter Ch.*x* Data Register (T16A_TC*x*)	**0x5402 0x5422** (16 bits)	D15–0	**T16A2TC [15:0]**	Counter data T16A2TC15 = MSB T16A2TC0 = LSB	0x0 to 0xffff	0x0	R	

D[15:0] T16A2TC[15:0]: Counter Data Bits

Counter data can be read out. (Default: 0x0)
The counter value can be read out even if the counter is running. However, the counter value should be read at once using a 16-bit transfer instruction. If data is read twice using an 8-bit transfer instruction, the correct value may not be obtained due to occurrence of count up between readings.

T16A Comparator/Capture Ch.*x* Control Registers (T16A_CCCTL*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16A Comparator/ Capture Ch.*x* Control Register (T16A_CCCTL*x*)	**0x5404** **0x5424** (16 bits)	D15–14	**CAPBTRG [1:0]**	Capture B trigger select	CAPBTRG[1:0] Trigger edge 0x3 ↑ and ↓ 0x2 ↓ 0x1 ↑ 0x0 None	0x0	R/W	
		D13–12	**TOUTBMD [1:0]**	TOUT B mode select	TOUTBMD[1:0] Mode 0x3 cmp B: ↑ or ↓ 0x2 cmp A: ↑ or ↓ 0x1 cmp A: ↑, B: ↓ 0x0 Off	0x0	R/W	
		D11–10	**–**	reserved	–	–	–	0 when being read.
		D9	**TOUTBINV**	TOUT B invert	1 Invert 0 Normal	0	R/W	
		D8	**CCBMD**	T16A_CCB register mode select	1 Capture 0 Comparator	0	R/W	
		D7–6	**CAPATRG [1:0]**	Capture A trigger select	CAPATRG[1:0] Trigger edge 0x3 ↑ and ↓ 0x2 ↓ 0x1 ↑ 0x0 None	0x0	R/W	
		D5–4	**TOUTAMD [1:0]**	TOUT A mode select	TOUTAMD[1:0] Mode 0x3 cmp B: ↑ or ↓ 0x2 cmp A: ↑ or ↓ 0x1 cmp A: ↑, B: ↓ 0x0 Off	0x0	R/W	
		D3–2	**–**	reserved	–	–	–	0 when being read.
		D1	**TOUTAINV**	TOUT A invert	1 Invert 0 Normal	0	R/W	
		D0	**CCAMD**	T16A_CCA register mode select	1 Capture 0 Comparator	0	R/W	

D[15:14] CAPBTRG[1:0]: Capture B Trigger Select Bits

Selects the trigger edge(s) of the external signal (CAPB*x*) at which the counter value is captured in the capture B register.

Table 12.8.5 Capture B Trigger Edge Selection

CAPBTRG[1:0]	Trigger edge
0x3	Falling edge and rising edge
0x2	Falling edge
0x1	Rising edge
0x0	Not triggered

(Default: 0x0)

CAPBTRG[1:0] are control bits for capture mode and are ineffective in comparator mode.

D[13:12] TOUTBMD[1:0]: TOUT B Mode Select Bits

Configures how the TOUT B signal waveform (TOUTB*x* output) is changed by the compare A and compare B signals. These bits are also used to turn the TOUT B output On and Off.

Table 12.8.6 TOUT B Generation Mode

TOUTBMD[1:0]	When compare A occurs	When compare B occurs
0x3	No change	Toggle
0x2	Toggle	No change
0x1	Rise	Fall
0x0	Disable output	

(Default: 0x0)

TOUTBMD[1:0] are control bits for comparator mode and are ineffective in capture mode.

D[11:10] Reserved

D9 TOUTBINV: TOUT B Invert Bit

Selects the TOUT B signal (TOUTB*x* output) polarity.

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

0 (R/W): Normal (active High) (default)

Writing 1 to TOUTBINV generates an active Low signal (Off level = High) for the TOUT B output. When TOUTBINV is 0, an active High signal (Off level = Low) is generated.

TOUTBINV is a control bit for comparator mode and is ineffective in capture mode.

D8 CCBMD: T16A_CCB Register Mode Select Bit

Selects the T16A_CCB*x* register function (comparator mode or capture mode).
1 (R/W): Capture mode
0 (R/W): Comparator mode (default)

Writing 1 to CCBMD configures the T16A_CCB*x* register as the capture B register (capture mode) to which the counter data will be loaded by the external trigger signal. When CCBMD is 0, the T16A_CCB*x* register functions as the compare B register (comparator mode) for writing a comparison value to generate the compare B signal.

D[7:6] CAPATRG[1:0]: Capture A Trigger Select Bits

Selects the trigger edge(s) of the external signal (CAPA*x*) at which the counter value is captured in the capture A register.

Table 12.8.7 Capture A Trigger Edge Selection

CAPATRG[1:0]	Trigger edge
0x3	Falling edge and rising edge
0x2	Falling edge
0x1	Rising edge
0x0	Not triggered

(Default: 0x0)

CAPATRG[1:0] are control bits for capture mode and are ineffective in comparator mode.

D[5:4] TOUTAMD[1:0]: TOUT A Mode Select Bits

Configures how the TOUT A signal waveform (TOUTA*x* output) is changed by the compare A and compare B signals. These bits are also used to turn the TOUT A output On and Off.

Table 12.8.8 TOUT A Generation Mode

TOUTAMD[1:0]	When compare A occurs	When compare B occurs
0x3	No change	Toggle
0x2	Toggle	No change
0x1	Rise	Fall
0x0	Disable output	

(Default: 0x0)

TOUTAMD[1:0] are control bits for comparator mode and are ineffective in capture mode.

D[3:2] Reserved

D1 TOUTAINV: TOUT A Invert Bit

Selects the TOUT A signal (TOUTA*x* output) polarity.
1 (R/W): Inverted (active Low)
0 (R/W): Normal (active High) (default)

Writing 1 to TOUTAINV generates an active Low signal (Off level = High) for the TOUT A output. When TOUTAINV is 0, an active High signal (Off level = Low) is generated.
TOUTAINV is a control bit for comparator mode and is ineffective in capture mode.

D0 CCAMD: T16A_CCA Register Mode Select Bit

Selects the T16A_CCA*x* register function (comparator mode or capture mode).
1 (R/W): Capture mode
0 (R/W): Comparator mode (default)

Writing 1 to CCAMD configures the T16A_CCA*x* register as the capture A register (capture mode) to which the counter data will be loaded by the external trigger signal. When CCAMD is 0, the T16A_CCA*x* register functions as the compare A register (comparator mode) for writing a comparison value to generate the compare A signal.

T16A Comparator/Capture Ch.*x* A Data Registers (T16A_CCA*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16A Comparator/ Capture Ch.*x* A Data Register (T16A_CCA*x*)	**0x5406 0x5426** (16 bits)	D15–0	**CCA[15:0]**	Compare/capture A data CCA15 = MSB CCA0 = LSB	0x0 to 0xffff	0x0	R/W	

D[15:0] CCA[15:0]: Compare/Capture A Data Bits

In comparator mode (CCAMD/ T16A_CCCTL*x* register = 0)

Sets a compare A data, which will be compared with the counter value, through this register.

When CBUFEN/T16A_CTL*x* register is set to 0, compare A data will be set to the compare A register after a lapse of two T16A2 count clock cycles from the time when it is written to this register.

When CBUFEN is set to 1, the data written to this register is loaded to the compare A buffer. The buffer contents are loaded into the compare A register when the compare B signal is generated.

The compare A register is always directly accessed when being read regardless of the CBUFEN setting.

The data set is compared with the counter data. When the counter reaches the comparison value set, the compare A signal is asserted and a cause of compare A interrupt occurs. Furthermore, the TOUT output waveform changes when TOUTAMD[1:0]/T16A_CCCTL*x* register or TOUTBMD[1:0]/T16A_CCCTL*x* register is set to 0x2 or 0x1. These processes do not affect the counter data and the count up operation.

In capture mode (CCAMD = 1)

When the counter value is captured at the external trigger signal (CAPA*x*) edge selected using CAPATRG[1:0]/T16A_CCCTL*x* register, the captured value is loaded to this register. At the same time a capture A interrupt can be generated, thus the captured counter value can be read out in the interrupt handler.

T16A Comparator/Capture Ch.*x* B Data Registers (T16A_CCB*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16A Comparator/ Capture Ch.*x* B Data Register (T16A_CCB*x*)	**0x5408 0x5428** (16 bits)	D15–0	**CCB[15:0]**	Compare/capture B data CCB15 = MSB CCB0 = LSB	0x0 to 0xffff	0x0	R/W	

D[15:0] CCB[15:0]: Compare/Capture B Data Bits

In comparator mode (CCBMD/ T16A_CCCTL*x* register = 0)

Sets a compare B data, which will be compared with the counter value, through this register.

When CBUFEN/T16A_CTL*x* register is set to 0, compare B data will be set to the compare B register after a lapse of two T16A2 count clock cycles from the time when it is written to this register.

When CBUFEN is set to 1, the data written to this register is loaded to the compare B buffer. The buffer contents are loaded into the compare B register when the compare B signal is generated.

The compare B register is always directly accessed when being read regardless of the CBUFEN setting.

The data set is compared with the counter data. When the counter reaches the comparison value set, the compare B signal is asserted and a cause of compare B interrupt occurs. The counter is reset to 0. Furthermore, the TOUT output waveform changes when TOUTAMD[1:0]/T16A_CCCTL*x* register or TOUTBMD[1:0]/T16A_CCCTL*x* register is set to 0x3 or 0x1.

In capture mode (CCBMD = 1)

When the counter value is captured at the external trigger signal (CAPB*x*) edge selected using CAPBTRG[1:0]/T16A_CCCTL*x* register, the captured value is loaded to this register. At the same time a capture B interrupt can be generated, thus the captured counter value can be read out in the interrupt handler.

T16A Comparator/Capture Ch.*x* Interrupt Enable Registers (T16A_IEN*x*)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
T16A Comparator/ Capture Ch.*x* Interrupt Enable Register (T16A_IEN*x*)	0x540a 0x542a (16 bits)	D15–6	–	reserved	–				–	–	0 when being read.
		D5	CAPBOWIE	Capture B overwrite interrupt enable	1	Enable	0	Disable	0	R/W	
		D4	CAPAOWIE	Capture A overwrite interrupt enable	1	Enable	0	Disable	0	R/W	
		D3	CAPBIE	Capture B interrupt enable	1	Enable	0	Disable	0	R/W	
		D2	CAPAIE	Capture A interrupt enable	1	Enable	0	Disable	0	R/W	
		D1	CBIE	Compare B interrupt enable	1	Enable	0	Disable	0	R/W	
		D0	CAIE	Compare A interrupt enable	1	Enable	0	Disable	0	R/W	

D[15:6] Reserved

D5 CAPBOWIE: Capture B Overwrite Interrupt Enable Bit

Enables or disables capture B overwrite interrupts.
1 (R/W): Interrupt enabled
0 (R/W): Interrupt disabled (default)

Setting CAPBOWIE to 1 enables capture B overwrite interrupt requests to the ITC. Setting it to 0 disables interrupts.

D4 CAPAOWIE: Capture A Overwrite Interrupt Enable Bit

Enables or disables capture A overwrite interrupts.
1 (R/W): Interrupt enabled
0 (R/W): Interrupt disabled (default)

Setting CAPAOWIE to 1 enables capture A overwrite interrupt requests to the ITC. Setting it to 0 disables interrupts.

D3 CAPBIE: Capture B Interrupt Enable Bit

Enables or disables capture B interrupts.
1 (R/W): Interrupt enabled
0 (R/W): Interrupt disabled (default)

Setting CAPBIE to 1 enables capture B interrupt requests to the ITC. Setting it to 0 disables interrupts.

D2 CAPAIE: Capture A Interrupt Enable Bit

Enables or disables capture A interrupts.
1 (R/W): Interrupt enabled
0 (R/W): Interrupt disabled (default)

Setting CAPAIE to 1 enables capture A interrupt requests to the ITC. Setting it to 0 disables interrupts.

D1 CBIE: Compare B Interrupt Enable Bit

Enables or disables compare B interrupts.
1 (R/W): Interrupt enabled
0 (R/W): Interrupt disabled (default)

Setting CBIE to 1 enables compare B interrupt requests to the ITC. Setting it to 0 disables interrupts.

D0 CAIE: Compare A Interrupt Enable Bit

Enables or disables compare A interrupts.
1 (R/W): Interrupt enabled
0 (R/W): Interrupt disabled (default)

Setting CAIE to 1 enables compare A interrupt requests to the ITC. Setting it to 0 disables interrupts.

T16A Comparator/Capture Ch.*x* Interrupt Flag Registers (T16A_IFLG*x*)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
T16A Comparator/ Capture Ch.*x* Interrupt Flag Register (T16A_IFLG*x*)	0x540c	D15–6	–	reserved	–				–	–	0 when being read.
	0x542c (16 bits)	D5	CAPBOWIF	Capture B overwrite interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D4	CAPAOWIF	Capture A overwrite interrupt flag					0	R/W	
		D3	CAPBIF	Capture B interrupt flag					0	R/W	
		D2	CAPAIF	Capture A interrupt flag					0	R/W	
		D1	CBIF	Compare B interrupt flag					0	R/W	
		D0	CAIF	Compare A interrupt flag					0	R/W	

D[15:6] Reserved

D5 CAPBOWIF: Capture B Overwrite Interrupt Flag Bit

Indicates whether the cause of capture B overwrite interrupt has occurred or not.

1 (R): Cause of interrupt has occurred
0 (R): No cause of interrupt has occurred (default)
1 (W): Flag is reset
0 (W): Ignored

CAPBOWIF is a T16A2 interrupt flag that is set to 1 when the capture B register is overwritten.
CAPBOWIF is reset by writing 1.

D4 CAPAOWIF: Capture A Overwrite Interrupt Flag Bit

Indicates whether the cause of capture A overwrite interrupt has occurred or not.

1 (R): Cause of interrupt has occurred
0 (R): No cause of interrupt has occurred (default)
1 (W): Flag is reset
0 (W): Ignored

CAPAOWIF is a T16A2 interrupt flag that is set to 1 when the capture A register is overwritten.
CAPAOWIF is reset by writing 1.

D3 CAPBIF: Capture B Interrupt Flag Bit

Indicates whether the cause of capture B interrupt has occurred or not.

1 (R): Cause of interrupt has occurred
0 (R): No cause of interrupt has occurred (default)
1 (W): Flag is reset
0 (W): Ignored

CAPBIF is a T16A2 interrupt flag that is set to 1 when the counter value is captured in the capture B register.
CAPBIF is reset by writing 1.

D2 CAPAIF: Capture A Interrupt Flag Bit

Indicates whether the cause of capture A interrupt has occurred or not.

1 (R): Cause of interrupt has occurred
0 (R): No cause of interrupt has occurred (default)
1 (W): Flag is reset
0 (W): Ignored

CAPAIF is a T16A2 interrupt flag that is set to 1 when the counter value is captured in the capture A register.
CAPAIF is reset by writing 1.

D1 CBIF: Compare B Interrupt Flag Bit

Indicates whether the cause of compare B interrupt has occurred or not.

1 (R): Cause of interrupt has occurred
0 (R): No cause of interrupt has occurred (default)
1 (W): Flag is reset
0 (W): Ignored

CBIF is a T16A2 interrupt flag that is set to 1 when the counter reaches the value set in the compare B register.
CBIF is reset by writing 1.

D0 CAIF: Compare A Interrupt Flag Bit

Indicates whether the cause of compare A interrupt has occurred or not.

1 (R): Cause of interrupt has occurred
0 (R): No cause of interrupt has occurred (default)
1 (W): Flag is reset
0 (W): Ignored

CAIF is a T16A2 interrupt flag that is set to 1 when the counter reaches the value set in the compare A register.
CAIF is reset by writing 1.

13 Clock Timer (CT)

13.1 CT Module Overview

The S1C17F57 includes a clock timer module (CT) that uses the OSC1 oscillator as its clock source. This timer can be used for generating cyclic interrupts to implement a software clock function.
The features of the CT module are listed below.

- 8-bit binary counter (128 Hz to 1 Hz)
- 32 Hz, 8 Hz, 2 Hz, and 1 Hz interrupts can be generated.

Figure 13.1.1 shows the CT configuration.

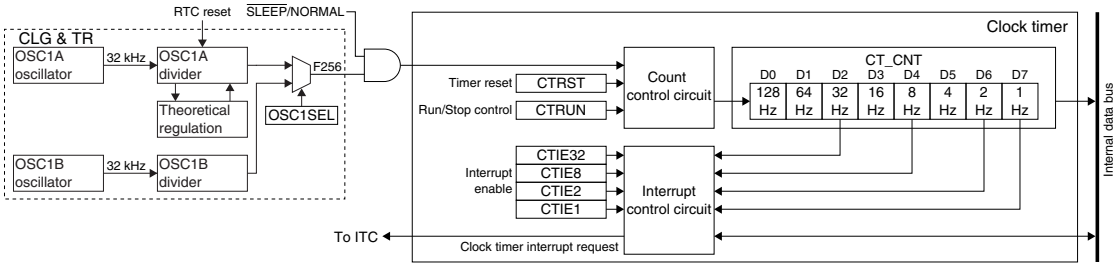


Figure 13.1.1 CT Configuration

The CT module consists of an 8-bit binary counter that uses the 256 Hz signal divided from the OSC1 clock as the input clock and allows data for each bit (128 Hz to 1 Hz) to be read out by software. The clock timer can also generate interrupts using the 32 Hz, 8 Hz, 2 Hz, and 1 Hz signals. This clock timer is normally used for various timing functions, such as a clock.

13.2 Operation Clock

The CT module uses the 256 Hz clock output by the CLG module as the operation clock (normally, the CT module is clocked by the F256 clock (regulated 256 Hz clock) derived from the OSC1A divider). Therefore, the OSC1 oscillator must be turned on before starting the CT module. However, the clock is not supplied to the CT module in SLEEP mode even if the OSC1 oscillator is on. For detailed information on clock control, see the "Clock Generator (CLG)" and "Theoretical Regulation (TR)" chapter.

Notes:
- The CT module input clock frequency is 256 Hz only when the OSC1 clock frequency is 32.768 kHz. The frequency described in this chapter will vary accordingly for other OSC1 clock frequencies.
- The CT module can also be operated with the OSC1B divider clock (about 256 Hz) even if OSC1B is selected as the OSC1 clock source in the CLG. However, the CT module cannot be used as an accurate clock.
- The OSC1A divider is reset when the RTC starts running (when 1 is written to RTCRUN/RTC_CTL register). This affects the count operations of the CT module, as new 256 Hz cycle begins from that point.

13.3 Timer Reset

Reset the timer by writing 1 to CTRST/CT_CTL register. This clears the counter to 0.
Apart from this operation, the counter is also cleared by an initial reset.

13.4 Timer RUN/STOP Control

Make the following settings before starting CT.
(1) If using interrupts, set the interrupt level and enable interrupts for the clock timer. See Section 13.5.
(2) Reset the timer. See Section 13.3.

The clock timer includes CTRUN/CT_CTL register for Run/Stop control.
The clock timer starts operating when 1 is written to CTRUN. Writing 0 to CTRUN disables clock input and stops the operation.
This control does not affect the counter (CT_CNT register) data. The counter data is retained even when the count is halted, allowing resumption of the count from that data.
If 1 is written to both CTRUN and CTRST simultaneously, the clock timer starts counting after resetting.
A cause of interrupt occurs during counting at the 32 Hz, 8 Hz, 2 Hz, and 1 Hz signal falling edges. If interrupts are enabled, an interrupt request is sent to the interrupt controller (ITC).

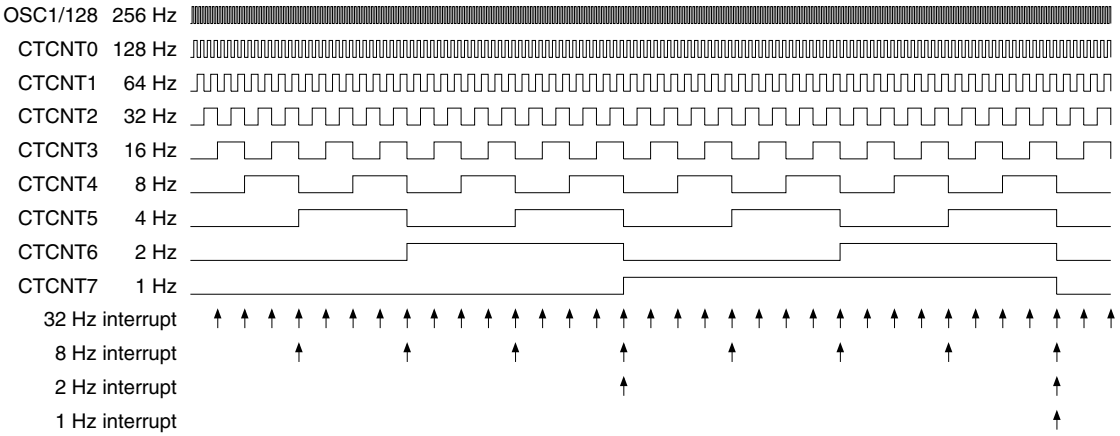


Figure 13.4.1 Clock Timer Timing Chart

Notes: • The clock timer switches to Run/Stop status synchronized with the 256 Hz signal falling edge after data is written to CTRUN. When 0 is written to CTRUN, the timer stops after counting an additional "+1." 1 is retained for CTRUN reading until the timer actually stops.
Figure 13.4.2 shows the Run/Stop control timing chart.

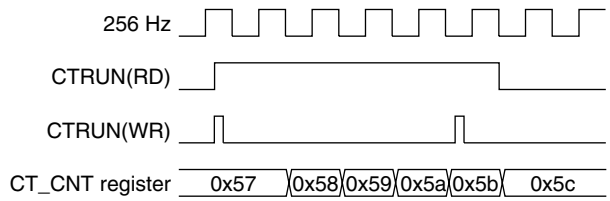


Figure 13.4.2 Run/Stop Control Timing Chart

• Executing the `slp` instruction while the timer is running (CTRUN = 1) will destabilize the timer operation during restarting from SLEEP status. When switching to SLEEP status, stop the timer (CTRUN = 0) before executing the `slp` instruction.

13.5 CT Interrupts

The CT module includes functions for generating the following four kinds of interrupts:
32 Hz, 8 Hz, 2 Hz, and 1 Hz interrupts

The CT module outputs a single interrupt signal shared by the above four interrupt causes to the interrupt controller (ITC). The interrupt flag in the CT module should be read to identify the cause of interrupt that occurred.

32 Hz, 8 Hz, 2 Hz, and 1 Hz interrupts

The 32 Hz, 8 Hz, 2 Hz, and 1 Hz signal falling edges set the corresponding interrupt flag in the CT module to 1. At the same time, an interrupt request is sent to the ITC if the corresponding interrupt enable bit has been set to 1 (interrupt enabled). An interrupt is generated if the ITC and S1C17 Core interrupt conditions are satisfied.
If the interrupt enable bit is set to 0 (interrupt disabled, default), no interrupt request will be sent to the ITC.

Table 13.5.1 CT Interrupt Flags and Interrupt Enable Bits

Cause of interrupt	Interrupt flag	Interrupt enable bit
32 Hz Interrupt	CTIF32/CT_IFLG register	CTIE32/CT_IMSK register
8 Hz Interrupt	CTIF8/CT_IFLG register	CTIE8/CT_IMSK register
2 Hz Interrupt	CTIF2/CT_IFLG register	CTIE2/CT_IMSK register
1 Hz Interrupt	CTIF1/CT_IFLG register	CTIE1/CT_IMSK register

For specific information on interrupt processing, see the "Interrupt Controller (ITC)" chapter.

Notes: • The CT module interrupt flag must be reset in the interrupt handler routine after a CT interrupt has occurred to prevent recurring interrupts.

• Reset the interrupt flag before enabling CT interrupts with the interrupt enable bit to prevent occurrence of unwanted interrupt. The interrupt flag is reset by writing 1.

13.6 Control Register Details

Table 13.6.1 List of CT Registers

Address	Register name		Function
0x5000	CT_CTL	Clock Timer Control Register	Resets and starts/stops the timer.
0x5001	CT_CNT	Clock Timer Counter Register	Counter data
0x5002	CT_IMSK	Clock Timer Interrupt Mask Register	Enables/disables interrupt.
0x5003	CT_IFLG	Clock Timer Interrupt Flag Register	Indicates/resets interrupt occurrence status.

The CT registers are described in detail below.

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

Clock Timer Control Register (CT_CTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Clock Timer Control Register (CT_CTL)	0x5000 (8 bits)	D7–5	–	reserved	–	–	–	0 when being read.
		D4	CTRST	Clock timer reset	1 Reset 0 Ignored	0	W	
		D3–1	–	reserved	–	–	–	
		D0	CTRUN	Clock timer run/stop control	1 Run 0 Stop	0	R/W	

D[7:5] Reserved

D4 CTRST: Clock Timer Reset Bit

Resets the clock timer.
1 (W): Reset
0 (W): Ignored
0 (R): Always 0 when read (default)

Writing 1 to this bit resets the counter to 0x0. When reset in Run state, the clock timer restarts immediately after resetting. The reset data 0x0 is retained when in Stop state.

D[3:1] Reserved

D0 CTRUN: Clock Timer Run/Stop Control Bit

Controls the clock timer Run/Stop.
1 (R/W): Run
0 (R/W): Stop (default)

The clock timer starts counting when CTRUN is written as 1 and stops when written as 0. The counter data is retained at Stop state until a reset or the next Run state.

Clock Timer Counter Register (CT_CNT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Clock Timer Counter Register (CT_CNT)	0x5001 (8 bits)	D7–0	CTCNT[7:0]	Clock timer counter value	0x0 to 0xff	0x0	R	

D[7:0] CTCNT[7:0]: Clock Timer Counter Value Bits

The counter data can be read out. (Default: 0x0)
This register is read-only and cannot be written to.
The bits correspond to various frequencies, as follows:
D7: 1 Hz, D6: 2 Hz, D5: 4 Hz, D4: 8 Hz, D3: 16 Hz, D2: 32 Hz, D1: 64 Hz, D0: 128 Hz

Note: The correct counter value may not be read out (reading is unstable) if the register is read while counting is underway. Read the counter register twice in succession and treat the value as valid if the values read are identical.

Clock Timer Interrupt Mask Register (CT_IMSK)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Clock Timer Interrupt Mask Register (CT_IMSK)	0x5002 (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.
		D3	CTIE32	32 Hz interrupt enable	1 Enable 0 Disable	0	R/W	
		D2	CTIE8	8 Hz interrupt enable	1 Enable 0 Disable	0	R/W	
		D1	CTIE2	2 Hz interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	CTIE1	1 Hz interrupt enable	1 Enable 0 Disable	0	R/W	

This register enables or disables interrupt requests individually for the 32 Hz, 8 Hz, 2 Hz, and 1 Hz signals. Setting CTIE* to 1 enables CT interrupts for the corresponding frequency signal falling edge, while setting to 0 disables interrupts.

D[7:4] Reserved

D3 CTIE32: 32 Hz Interrupt Enable Bit

Enables or disables 32 Hz interrupts.
1 (R/W): Interrupt enabled
0 (R/W): Interrupt disabled (default)

D2 CTIE8: 8 Hz Interrupt Enable Bit

Enables or disables 8 Hz interrupts.
1 (R/W): Interrupt enabled
0 (R/W): Interrupt disabled (default)

D1 CTIE2: 2 Hz Interrupt Enable Bit

Enables or disables 2 Hz interrupts.
1 (R/W): Interrupt enabled
0 (R/W): Interrupt disabled (default)

D0 CTIE1: 1 Hz Interrupt Enable Bit

Enables or disables 1 Hz interrupts.
1 (R/W): Interrupt enabled
0 (R/W): Interrupt disabled (default)

Clock Timer Interrupt Flag Register (CT_IFLG)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Clock Timer Interrupt Flag Register (CT_IFLG)	0x5003 (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.
		D3	CTIF32	32 Hz interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D2	CTIF8	8 Hz interrupt flag		0	R/W	
		D1	CTIF2	2 Hz interrupt flag		0	R/W	
		D0	CTIF1	1 Hz interrupt flag		0	R/W	

This register indicates the occurrence state of interrupt causes due to 32 Hz, 8 Hz, 2 Hz, and 1 Hz signals. If a CT interrupt occurs, identify the interrupt cause (frequency) by reading the interrupt flag in this register. CTIF* is a CT module interrupt flag that is set to 1 at the falling edge of the corresponding 32 Hz, 8 Hz, 2 Hz, or 1 Hz interrupt. CTIF* is reset by writing 1.

D[7:4] Reserved

D3 CTIF32: 32 Hz Interrupt Flag Bit

Indicates whether the cause of 32 Hz interrupt has occurred or not.

1 (R): Cause of interrupt has occurred
0 (R): No cause of interrupt has occurred (default)
1 (W): Flag is reset
0 (W): Ignored

D2 CTIF8: 8 Hz Interrupt Flag Bit

Indicates whether the cause of 8 Hz interrupt has occurred or not.

1 (R): Cause of interrupt has occurred
0 (R): No cause of interrupt has occurred (default)
1 (W): Flag is reset
0 (W): Ignored

D1 CTIF2: 2 Hz Interrupt Flag Bit

Indicates whether the cause of 2 Hz interrupt has occurred or not.

1 (R): Cause of interrupt has occurred
0 (R): No cause of interrupt has occurred (default)
1 (W): Flag is reset
0 (W): Ignored

D0 CTIF1: 1 Hz Interrupt Flag Bit

Indicates whether the cause of 1 Hz interrupt has occurred or not.

1 (R): Cause of interrupt has occurred
0 (R): No cause of interrupt has occurred (default)
1 (W): Flag is reset
0 (W): Ignored

14 Stopwatch Timer (SWT)

14.1 SWT Module Overview

The S1C17F57 includes a 1/100-second stopwatch timer module (SWT) that uses the OSC1 oscillator as its clock source. This timer can be used to implement a software stopwatch function.
The features of the SWT module are listed below.

- Two 4-bit BCD counters (approximately 1/100 and 1/10-second counters)
- Approximately 100 Hz, approximately 10 Hz, and 1 Hz interrupts can be generated.

Figure 14.1.1 shows the SWT configuration.

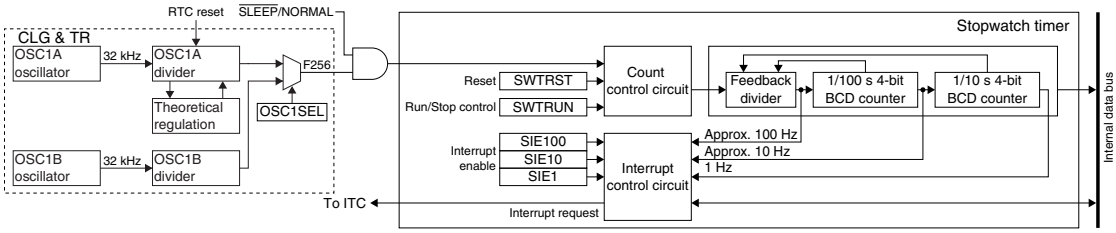


Figure 14.1.1 SWT Configuration

The SWT module consists of two 4-bit BCD counters (1/100 and 1/10 second) that use the 256 Hz signal divided from the OSC1 clock as the input clock and allows count data to be read out by software.
The SWT module can also generate interrupts using the 100 Hz (approximately 100 Hz), 10 Hz (approximately 10 Hz), and 1 Hz signals.

14.2 Operation Clock

The SWT module uses the 256 Hz clock output by the CLG module as the operation clock (normally, the SWT module is clocked by the F256 clock (regulated 256 Hz clock) derived from the OSC1A divider). Therefore, the OSC1 oscillator must be turned on before starting the SWT module. However, the clock is not supplied to the SWT module in SLEEP mode even if the OSC1 oscillator is on. For detailed information on clock control, see the "Clock Generator (CLG)" and "Theoretical Regulation (TR)" chapter.

Notes:
- The SWT module input clock frequency is 256 Hz only when the OSC1 clock frequency is 32.768 kHz. The frequency described in this chapter will vary accordingly for other OSC1 clock frequencies.
- The SWT module can also be operated with the OSC1B divider clock (about 256 Hz) even if OSC1B is selected as the OSC1 clock source in the CLG. However, the SWT module cannot be used as an accurate clock.
- The OSC1A divider is reset when the RTC starts running (when 1 is written to RTCRUN/RTC_CTL register). This affects the count operations of the SWT module, as new 256 Hz cycle begins from that point.

14.3 BCD Counters

The SWT module consists of 1/100-second and 1/10-second 4-bit BCD counters.
The 1/100-second and 1/10-second counter values can be read from BCD100[3:0]/SWT_BCNT register and BCD10[3:0]/SWT_BCNT register, respectively.

Count-up Pattern

A feedback divider is used to generate 100 Hz, 10 Hz, and 1 Hz signals from the 256 Hz clock. The counter count-up pattern varies as shown in Figure 14.3.1.

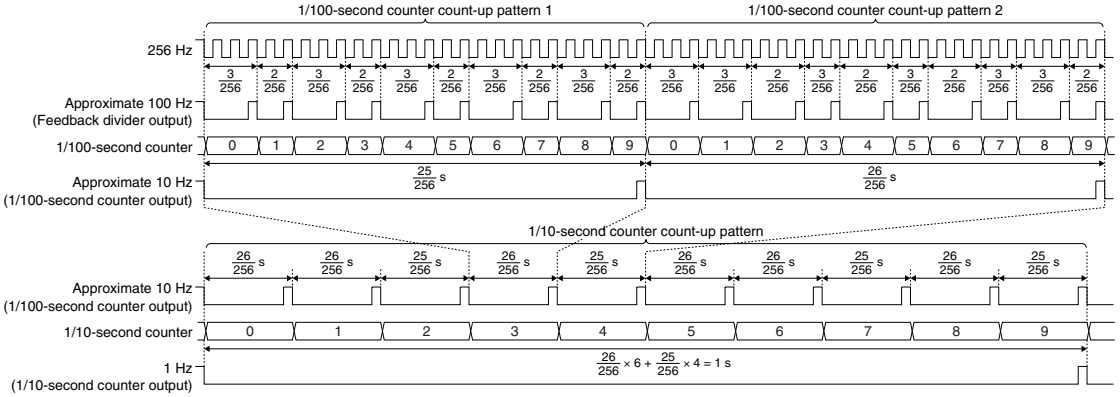


Figure 14.3.1 SWT Count-Up Patterns

The feedback divider generates an approximate 100 Hz signal at 2/256-second and 3/256-second intervals from the 256 Hz signal supplied from the CLG module.

The 1/100-second counter counts the approximate 100 Hz signal output by the feedback divider and generates an approximate 10 Hz signal at 25/256-second and 26/256-second intervals. Count-up will be pseudo 1/100-second counting at 2/256-second and 3/256-second intervals.

The 1/10-second counter counts the approximate 10 Hz signal generated by the 1/100-second counter at a ratio of 4:6, and generates a 1 Hz signal. Count-up will be pseudo 1/10-second counting at 25/256-second and 26/256-second intervals.

14.4 Timer Reset

Reset the SWT module by writing 1 to SWTRST/SWT_CTL register. This clears the counter to 0.
Apart from this operation, the counter is also cleared by initial reset.

14.5 Timer RUN/STOP Control

Make the following settings before starting SWT.

(1) If using interrupts, set the interrupt level and enable interrupts for the SWT module. See Section 14.6.
(2) Reset the timer. See Section 14.4.

The SWT module includes SWTRUN/SWT_CTL register for Run/Stop control.
The timer starts operating when 1 is written to SWTRUN. Writing 0 to SWTRUN disables clock input and stops the operation. This control does not affect the counter (SWT_BCNT register) data. The counter data is retained even when the count is halted, allowing resumption of the count from that data. If 1 is written to both SWTRUN and SWTRST simultaneously, the timer starts counting after resetting.
A cause of interrupt occurs during counting at the 100 Hz (approximate 100 Hz), 10 Hz (approximate 10 Hz), and 1 Hz signal falling edges. If interrupts are enabled, an interrupt request is sent to the interrupt controller (ITC).

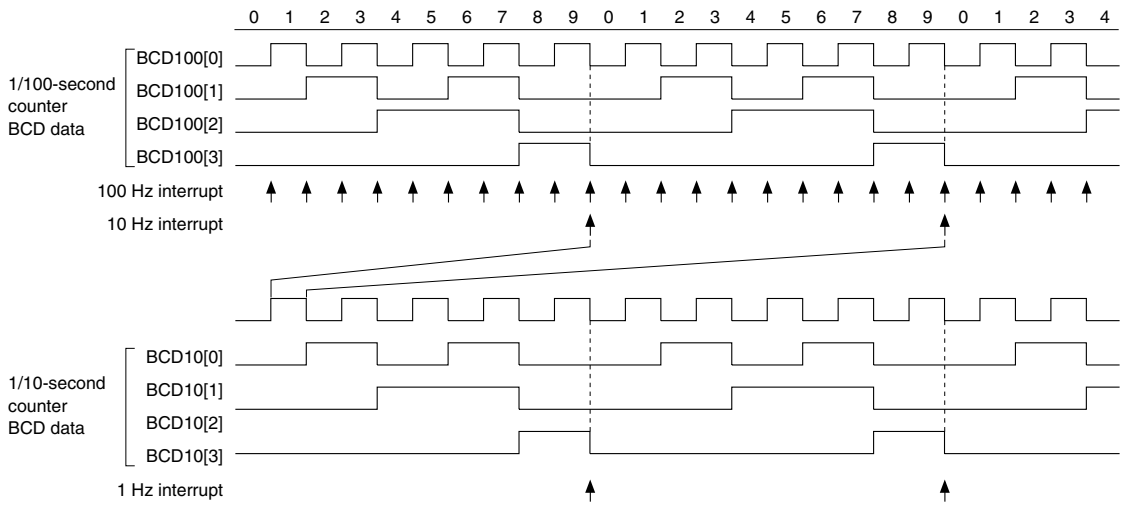


Figure 14.5.1 SWT Timing Chart

Notes: • The timer switches to Run/Stop status synchronized with the 256 Hz signal falling edge after data is written to SWTRUN. When 0 is written to SWTRUN, the timer stops after counting an additional "+1." 1 is retained for SWTRUN reading until the timer actually stops.
Figure 14.5.2 shows the Run/Stop control timing chart.

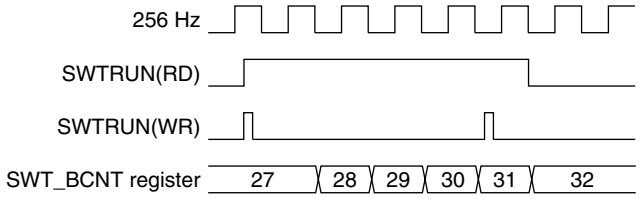


Figure 14.5.2 Run/Stop Control Timing Chart

• Executing the `slp` instruction while the timer is running (SWTRUN = 1) will destabilize the timer operation during restarting from SLEEP status. When switching to SLEEP status, stop the timer (SWTRUN = 0) before executing the `slp` instruction.

14.6 SWT Interrupts

The SWT module includes functions for generating the following three kinds of interrupts:
100 Hz, 10 Hz, and 1 Hz interrupts

The SWT module outputs a single interrupt signal shared by the above three interrupt causes to the interrupt controller (ITC). The interrupt flag in the SWT module should be read to identify the cause of interrupt that occurred.

100 Hz, 10 Hz, 1 Hz interrupts

The 100 Hz (approximate 100 Hz), 10 Hz (approximate 10 Hz), and 1 Hz signal falling edges set the corresponding interrupt flag in the SWT module to 1. At the same time, an interrupt request is sent to the ITC if the corresponding interrupt enable bit has been set to 1 (interrupt enabled). An interrupt is generated if the ITC and S1C17 Core interrupt conditions are satisfied.
If the interrupt enable bit is set to 0 (interrupt disabled, default), no interrupt request will be sent to the ITC.

Table 14.6.1 SWT Interrupt Flags and Interrupt Enable Bits

Cause of interrupt	Interrupt flag	Interrupt enable bit
100 Hz Interrupt	SIF100/SWT_IFLG register	SIE100/SWT_IMSK register
10 Hz Interrupt	SIF10/SWT_IFLG register	SIE10/SWT_IMSK register
1 Hz Interrupt	SIF1/SWT_IFLG register	SIE1/SWT_IMSK register

For specific information on interrupt processing, see the "Interrupt Controller (ITC)" chapter.

Notes: • The SWT module interrupt flag must be reset in the interrupt handler routine after a stopwatch timer interrupt has occurred to prevent recurring interrupts.

• Reset the interrupt flag before enabling SWT interrupts with the interrupt enable bit to prevent occurrence of unwanted interrupt. The interrupt flag is reset by writing 1.

14.7 Control Register Details

Table 14.7.1 List of SWT Registers

Address	Register name		Function
0x5020	SWT_CTL	Stopwatch Timer Control Register	Resets and starts/stops the timer.
0x5021	SWT_BCNT	Stopwatch Timer BCD Counter Register	BCD counter data
0x5022	SWT_IMSK	Stopwatch Timer Interrupt Mask Register	Enables/disables interrupt.
0x5023	SWT_IFLG	Stopwatch Timer Interrupt Flag Register	Indicates/resets interrupt occurrence status.

The SWT registers are described in detail below.

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

Stopwatch Timer Control Register (SWT_CTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Stopwatch Timer Control Register (SWT_CTL)	0x5020 (8 bits)	D7–5	–	reserved	–	–	–	0 when being read.
		D4	SWTRST	Stopwatch timer reset	1 Reset 0 Ignored	0	W	
		D3–1	–	reserved	–	–	–	
		D0	SWTRUN	Stopwatch timer run/stop control	1 Run 0 Stop	0	R/W	

D[7:5] Reserved

D4 SWTRST: Stopwatch Timer Reset Bit

Resets the SWT module.

1 (W): Reset

0 (W): Ignored

0 (R): Always 0 when read (default)

Writing 1 to this bit resets the counter to 0x0. When reset in Run state, the timer restarts immediately after resetting. The reset data 0x0 is retained when in Stop state.

D[3:1] Reserved

D0 SWTRUN: Stopwatch Timer Run/Stop Control Bit

Controls the timer Run/Stop.

1 (R/W): Run

0 (R/W): Stop (default)

The timer starts counting when SWTRUN is written as 1 and stops when written as 0. The counter data is retained at Stop state until a reset or the next Run state.

Stopwatch Timer BCD Counter Register (SWT_BCNT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Stopwatch Timer BCD Counter Register (SWT_BCNT)	0x5021 (8 bits)	D7–4	BCD10[3:0]	1/10 sec. BCD counter value	0 to 9	0	R	
		D3–0	BCD100[3:0]	1/100 sec. BCD counter value	0 to 9	0	R	

D[7:4] BCD10[3:0]: 1/10 Sec. BCD Counter Value Bit

The 1/10-second counter BCD data can be read out. (Default: 0)

This register is read-only and cannot be written to.

D[3:0] BCD100[3:0]: 1/100 Sec. BCD Counter Value Bit

The 1/100-second counter BCD data can be read out. (Default: 0)

This register is read-only and cannot be written to.

Note: The correct counter value may not be read out (reading is unstable) if the register is read while counting is underway. Read the counter register twice in succession and treat the value as valid if the values read are identical.

Stopwatch Timer Interrupt Mask Register (SWT_IMSK)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Stopwatch Timer Interrupt Mask Register (SWT_IMSK)	0x5022 (8 bits)	D7–3	–	reserved	–	–	–	0 when being read.
		D2	SIE1	1 Hz interrupt enable	1 Enable 0 Disable	0	R/W	
		D1	SIE10	10 Hz interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	SIE100	100 Hz interrupt enable	1 Enable 0 Disable	0	R/W	

This register enables or disables interrupt requests individually for the 100 Hz, 10 Hz, and 1 Hz signals. Setting SIE* to 1 enables SWT interrupts for the corresponding frequency signal falling edge, while setting to 0 disables interrupts.

D[7:3] Reserved

D2 SIE1: 1 Hz Interrupt Enable Bit

Enables or disables 1 Hz interrupts.

1 (R/W): Interrupt enabled

0 (R/W): Interrupt disabled (default)

D1 SIE10: 10 Hz Interrupt Enable Bit

Enables or disables 10 Hz interrupts.

1 (R/W): Interrupt enabled

0 (R/W): Interrupt disabled (default)

D0 SIE100: 100 Hz Interrupt Enable Bit

Enables or disables 100 Hz interrupts.

1 (R/W): Interrupt enabled

0 (R/W): Interrupt disabled (default)

Stopwatch Timer Interrupt Flag Register (SWT_IFLG)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Stopwatch Timer Interrupt Flag Register (SWT_IFLG)	0x5023 (8 bits)	D7–3	–	reserved	–	–	–	0 when being read.
		D2	SIF1	1 Hz interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D1	SIF10	10 Hz interrupt flag		0	R/W	
		D0	SIF100	100 Hz interrupt flag		0	R/W	

This register indicates the occurrence state of interrupt causes due to 100 Hz, 10 Hz, and 1 Hz signals. If an SWT interrupt occurs, identify the interrupt cause (frequency) by reading the interrupt flag in this register. SIF* is an SWT module interrupt flag that is set to 1 at the falling edge of the corresponding 100 Hz, 10 Hz, or 1 Hz interrupt. SIF* is reset by writing 1.

D[7:3] Reserved

D2 SIF1: 1 Hz Interrupt Flag Bit

Indicates whether the cause of 1 Hz interrupt has occurred or not.

1 (R): Cause of interrupt has occurred

0 (R): No cause of interrupt has occurred (default)

1 (W): Flag is reset

0 (W): Ignored

D1 SIF10: 10 Hz Interrupt Flag Bit

Indicates whether the cause of 10 Hz interrupt has occurred or not.

1 (R): Cause of interrupt has occurred

0 (R): No cause of interrupt has occurred (default)

1 (W): Flag is reset

0 (W): Ignored

D0 SIF100: 100 Hz Interrupt Flag Bit

Indicates whether the cause of 100 Hz interrupt has occurred or not.

1 (R): Cause of interrupt has occurred
0 (R): No cause of interrupt has occurred (default)
1 (W): Flag is reset
0 (W): Ignored

15 Watchdog Timer (WDT)

15.1 WDT Module Overview

The S1C17F57 includes a watchdog timer module (WDT) that uses the OSC1 oscillator as its clock source. This timer is used to detect CPU runaway.
The features of WDT are listed below.

- 10-bit up counter
- Either reset or NMI can be generated if the counter overflows.

Figure 15.1.1 shows the WDT configuration.

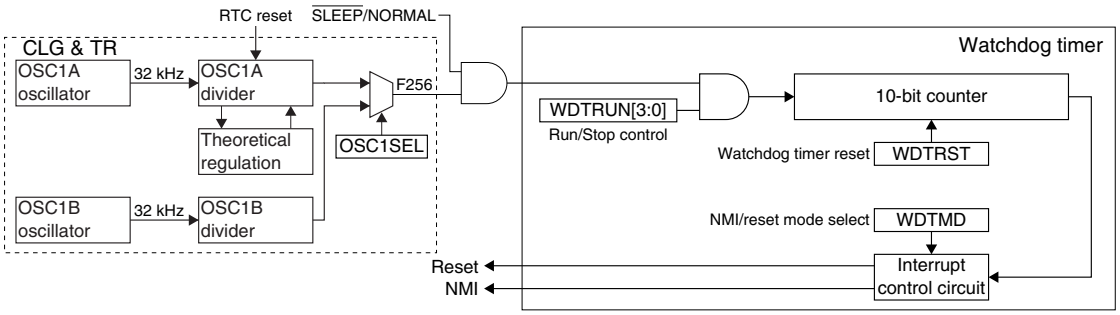


Figure 15.1.1 WDT Configuration

The WDT module generates an NMI or reset (selectable via software) to the CPU if not reset within 131,072/fOSC1 seconds (4 seconds when fOSC1 = 32.768 kHz).
Reset WDT via software within this cycle to prevent NMI/resets, which in turn enables runaway detection for programs that do not pass through the handler routine.

15.2 Operation Clock

The WDT module uses the 256 Hz clock output by the CLG module as the operation clock (normally, the WDT module is clocked by the F256 clock (regulated 256 Hz clock) derived from the OSC1A divider). Therefore, the OSC1 oscillator must be turned on before starting the WDT module. However, the clock is not supplied to the WDT module in SLEEP mode even if the OSC1 oscillator is on. For detailed information on clock control, see the "Clock Generator (CLG)" and "Theoretical Regulation (TR)" chapter.

Notes: • The WDT module input clock frequency is 256 Hz only when the OSC1 clock frequency is 32.768 kHz. The frequency and time described in this chapter will vary accordingly for other OSC1 clock frequencies.

• The WDT module can also be operated with the OSC1B divider clock (about 256 Hz) even if OSC1B is selected as the OSC1 clock source in the CLG.

• The OSC1A divider is reset when the RTC starts running (when 1 is written to RTCRUN/RTC_CTL register). This affects the count operations of the WDT module, as new 256 Hz cycle begins from that point.

15.3 WDT Control

15.3.1 NMI/Reset Mode Selection

WDTMD/WDT_ST register is used to select whether an NMI signal or a reset signal is output when WDT has not been reset within the NMI/reset generation cycle.
To generate an NMI, set WDTMD to 0 (default). Set to 1 to generate a reset.

15.3.2 WDT Run/Stop Control

WDT starts counting when a value other than 0b1010 is written to WDTRUN[3:0]/WDT_CTL register and stops when 0b1010 is written.
At initial reset, WDTRUN[3:0] is set to 0b1010 to stop the watchdog timer.
Since an NMI or reset may be generated immediately after running depending on the counter value, WDT should also be reset concurrently (before running the watchdog timer), as explained in the following section.

15.3.3 WDT Reset

To reset WDT, write 1 to WDTRST/WDT_CTL register.
A location should be provided for periodically processing the routine for resetting WDT before an NMI or reset is generated when using WDT. Process this routine within 131,072/f_{OSC1} second (4 seconds when f_{OSC1} = 32.768 kHz) cycle.
After resetting, WDT starts counting with a new NMI/Reset generation cycle.
If WDT is not reset within the NMI/Reset generation cycle for any reason, the CPU is switched to interrupt processing by NMI or reset, the interrupt vector is read out, and the interrupt handler routine is executed. The reset and NMI vector addresses are TTBR + 0x0 and TTBR + 0x08.

If the counter overflows and generates an NMI without WDT being reset, WDTST/WDT_ST register is set to 1. This bit is provided to confirm that WDT was the source of the NMI. The WDTST set to 1 is cleared to 0 by resetting WDT.

15.3.4 Operations in HALT and SLEEP Modes

HALT mode

The WDT module operates in HALT mode, as the clock is supplied. HALT mode is therefore cleared by an NMI or reset if it continues for more than the NMI/reset generation cycle. To disable WDT while in HALT mode, stop WDT by writing 0b1010 to WDTRUN[3:0]/WDT_CTL register before executing the `halt` instruction. Reset WDT before resuming operations after HALT mode is cleared.

SLEEP mode

The clock supplied from the CLG module is stopped in SLEEP mode, which also stops WDT. To prevent generation of an unnecessary NMI or reset after clearing SLEEP mode, reset WDT before executing the `slp` instruction. WDT should also be stopped as required using WDTRUN[3:0].

15.4 Control Register Details

Table 15.4.1 List of WDT Registers

Address	Register name		Function
0x5040	WDT_CTL	Watchdog Timer Control Register	Resets and starts/stops the timer.
0x5041	WDT_ST	Watchdog Timer Status Register	Sets the timer mode and indicates NMI status.

The WDT registers are described in detail below.

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

Watchdog Timer Control Register (WDT_CTL)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Watchdog Timer Control Register (WDT_CTL)	**0x5040** (8 bits)	D7–5	**–**	reserved	–				–	–	0 when being read.
		D4	**WDTRST**	Watchdog timer reset	1	Reset	0	Ignored	0	W	
		D3–0	**WDTRUN[3:0]**	Watchdog timer run/stop control	Other than 1010	Run	1010	Stop	1010	R/W	

D[7:5] Reserved

D4 WDTRST: Watchdog Timer Reset Bit

Resets WDT.
1 (W): Reset
0 (W): Ignored
0 (R): Always 0 when read (default)

Note: To use WDT, it must be reset by writing 1 to this bit within the NMI/reset generation cycle (4 seconds when f_{OSC1} = 32.768 kHz). This resets the up-counter to 0 and starts counting with a new NMI/reset generation cycle.

D[3:0] WDTRUN[3:0]: Watchdog Timer Run/Stop Control Bits

Controls WDT Run/Stop.
Values other than 0b1010 (R/W): Run
0b1010 (R/W): Stop (default)

Note: WDT must also be reset to prevent generation of an unnecessary NMI or Reset before starting WDT.

Watchdog Timer Status Register (WDT_ST)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Watchdog Timer Status Register (WDT_ST)	0x5041 (8 bits)	D7–2	–	reserved	–				–	–	0 when being read.
		D1	WDTMD	NMI/Reset mode select	1	Reset	0	NMI	0	R/W	
		D0	WDTST	NMI status	1	NMI occurred	0	Not occurred	0	R	

D[7:2] Reserved

D1 WDTMD: NMI/Reset Mode Select Bit

Selects NMI or reset generation on counter overflow.
1 (R/W): Reset
0 (R/W): NMI (default)

Setting this bit to 1 outputs a reset signal when the counter overflows. Setting to 0 outputs an NMI signal.

D0 WDTST: NMI Status Bit

Indicates a counter overflow and NMI occurrence.
1 (R): NMI occurred (counter overflow)
0 (R): NMI not occurred (default)

This bit confirms that WDT was the source of the NMI. The WDTST set to 1 is cleared to 0 by resetting WDT.
This is also set by a counter overflow if reset output is selected, but is cleared by initial reset and cannot be confirmed.

16 UART

16.1 UART Module Overview

The S1C17F57 includes a UART module for asynchronous communication. It includes a 2-byte receive data buffer and 1-byte transmit data buffer allowing successive data transfer. The UART module also includes an RZI modulator/demodulator circuit that enables IrDA 1.0-compatible infrared communications simply by adding basic external circuits.
The following shows the main features of the UART:

- Number of channels: 1 channel
- Transfer rate: 150 to 230,400 bps (150 to 115,200 bps in IrDA mode)
- Transfer clock: Internal clock (baud rate generator output) or an external clock (SCLK input) can be selected.
- Character length: 7 or 8 bits (LSB first)
- Parity mode: Even, odd, or no parity
- Stop bit: 1 or 2 bits
- Start bit: 1 bit fixed
- Supports full-duplex communications.
- Includes a 2-byte receive data buffer and a 1-byte transmit data buffer.
- Includes a baud rate generator with fine adjustment function.
- Includes an RZI modulator/demodulator circuit to support IrDA 1.0-compatible infrared communications.
- Can detect parity error, framing error, and overrun error during receiving.
- Can generate receive buffer full, transmit buffer empty, end of transmission and receive error interrupts.

Figure 16.1.1 shows the UART configuration.

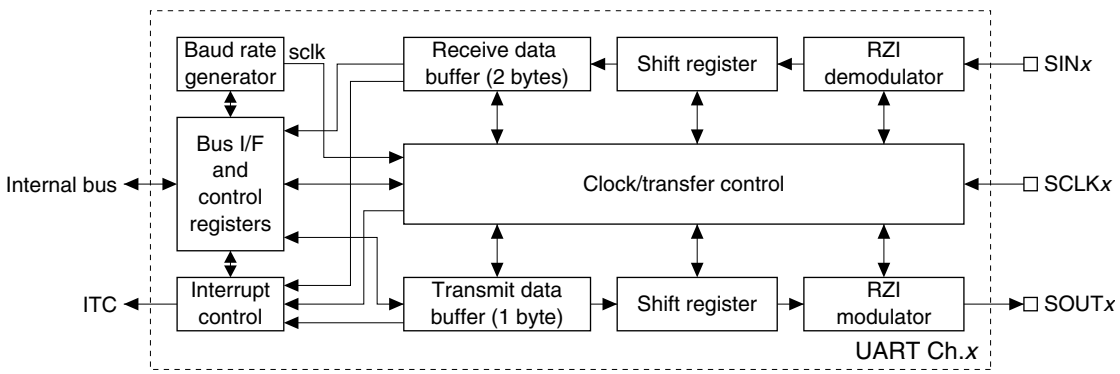


Figure 16.1.1 UART Configuration

Note: The letter '*x*' in register and pin names refers to a channel number (0).
Example: UART_CTL*x* register
Ch.0: UART_CTL0 register

16.2 UART Input/Output Pins

Table 16.2.1 lists the UART input/output pins.

Table 16.2.1 List of UART Pins

Pin name	I/O	Qty	Function
SIN0 (Ch.0)	I	1	UART data input pin Inputs serial data sent from an external serial device.
SOUT0 (Ch.0)	O	1	UART data output pin Outputs serial data sent to an external serial device.
SCLK0 (Ch.0)	I	1	UART clock input pin Inputs the transfer clock when an external clock is used.

The UART input/output pins (SIN*x*, SOUT*x*, SCLK*x*) are shared with I/O ports and are initially set as general purpose I/O port pins. The pin functions must be switched using the port function select bits to use the general purpose I/O port pins as UART input/output pins.
For detailed information on pin function switching, see the "I/O Ports (P)" chapter.

16.3 Baud Rate Generator

The UART module includes a baud rate generator to generate the transfer (sampling) clock. It consists of an 8-bit programmable timer with fine mode. The timer counts down from the initial value set via software and outputs an underflow signal when the counter underflows. The underflow signal is used to generate the transfer clock. The underflow cycle can be programmed by selecting the clock source and initial data, enabling the application program to obtain serial transfer rates as required. Fine mode provides a function that minimizes transfer rate errors.

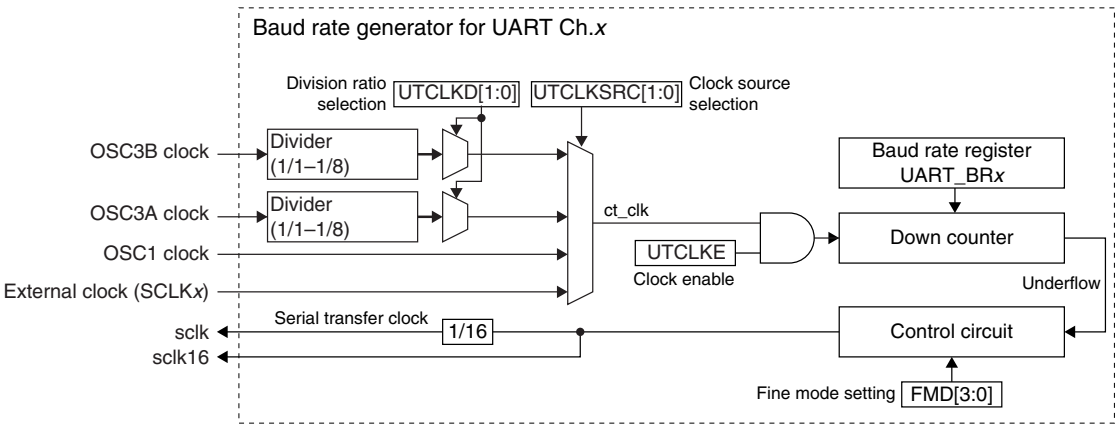


Figure 16.3.1 Baud Rate Generator

Clock source settings

The clock source can be selected from OSC3B, OSC3A, OSC1, or external clock using UTCLKSRC[1:0]/UART_CLK*x* register.

Table 16.3.1 Clock Source Selection

UTCLKSRC[1:0]	Clock source
0x3	External clock (SCLK*x*)
0x2	OSC3A
0x1	OSC1
0x0	OSC3B

(Default: 0x0)

Note: When inputting the external clock via the SCLK*x* pin, the clock duty ratio must be 50%.

When OSC3B or OSC3A is selected as the clock source, use UTCLKD[1:0]/UART_CLK*x* register to select the division ratio.

Table 16.3.2 OSC3B/OSC3A Division Ratio Selection

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

(Default: 0x0)

Clock supply to the counter is controlled using UTCLKE/UART_CLK*x* register. The UTCLKE default setting is 0, which disables the clock supply. Setting UTCLKE to 1 sends the clock selected to the counter.

Initial counter value setting

BR[7:0]/UART_BR*x* register is used to set the initial value for the down counter.
The initial counter value is preset to the down counter if the counter underflows. This means that the initial counter value and the count clock frequency determine the time elapsed between underflows.

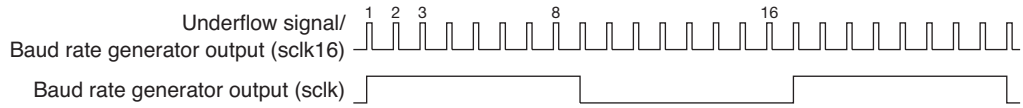


Figure 16.3.2 Counter Underflow and Clock Generated

Use the following equations to calculate the initial counter value for obtaining the desired transfer rate.

$$bps = \frac{ct\_clk}{\{(BR + 1) \times 16 + FMD\}}$$

$$BR = \left(\frac{ct\_clk}{bps} - FMD - 16\right) \div 16$$

ct_clk: Count clock frequency (Hz)
BR: BR[7:0] setting (0 to 255)
bps: Transfer rate (bit/s)
FMD: FMD[3:0] (fine mode) setting (0 to 15)

Note: The UART transfer rate is capped at 230,400 bps (115,200 bps in IrDA mode). Do not set faster transfer rates.

Fine Mode

Fine mode provides a function that minimizes transfer rate errors. The baud rate generator output clock can be set to the required frequency by selecting the appropriate clock source and initial counter data. Note that errors may occur, depending on the transfer rate. Fine mode extends the output clock cycle by delaying the underflow pulse from the counter. This delay can be specified with the FMD[3:0]/UART_FMD*x* register. FMD[3:0] specifies the delay pattern to be inserted into a 16 underflow period. Inserting one delay extends the output clock cycle by one count clock cycle.

Table 16.3.3 Delay Patterns Specified by FMD[3:0]

FMD[3:0]	Underflow number															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0x0	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
0x1	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	D
0x2	–	–	–	–	–	–	–	D	–	–	–	–	–	–	–	D
0x3	–	–	–	–	–	–	–	D	–	–	–	D	–	–	–	D
0x4	–	–	–	D	–	–	–	D	–	–	–	D	–	–	–	D
0x5	–	–	–	D	–	–	–	D	–	–	–	D	–	D	–	D
0x6	–	–	–	D	–	D	–	D	–	–	–	D	–	D	–	D
0x7	–	–	–	D	–	D	–	D	–	D	–	D	–	D	–	D
0x8	–	D	–	D	–	D	–	D	–	D	–	D	–	D	–	D
0x9	–	D	–	D	–	D	–	D	–	D	–	D	–	D	D	D
0xa	–	D	–	D	–	D	D	D	–	D	–	D	–	D	D	D
0xb	–	D	–	D	–	D	D	D	–	D	D	D	–	D	D	D
0xc	–	D	D	D	–	D	D	D	–	D	D	D	–	D	D	D
0xd	–	D	D	D	–	D	D	D	–	D	D	D	D	D	D	D
0xe	–	D	D	D	D	D	D	D	–	D	D	D	D	D	D	D
0xf	–	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D

D: Indicates the insertion of a delay cycle.

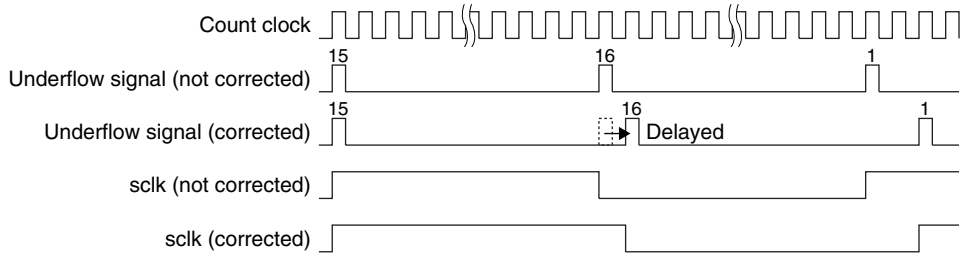


Figure 16.3.3 Delay Cycle Insertion in Fine Mode

At initial reset, FMD[3:0] is set to 0x0, preventing insertion of delay cycles.

Note: Make sure the UART is halted (RXEN/UART_CTL*x* register = 0) before setting the baud rate generator.

16.4 Transfer Data Settings

Set the following conditions to configure the transfer data format.
- Data length: 7 or 8 bits
- Start bit: Fixed at 1 bit
- Stop bit: 1 or 2 bits
- Parity bit: Even, odd, or no parity

Note: Make sure the UART is halted (RXEN/UART_CTL*x* register = 0) before changing transfer data format settings.

Data length

The data length is selected by CHLN/UART_MOD*x* register. Setting CHLN to 0 (default) configures the data length to 7 bits. Setting CHLN to 1 configures it to 8 bits.

Stop bit

The stop bit length is selected by STPB/UART_MOD*x* register. Setting STPB to 0 (default) configures the stop bit length to 1 bit. Setting STPB to 1 configures it to 2 bits.

Parity bit

Whether the parity function is enabled or disabled is selected by PREN/UART_MOD*x* register. Setting PREN to 0 (default) disables the parity function. In this case, no parity bit is added to the transfer data and the data is not checked for parity when received. Setting PREN to 1 enables the parity function. In this case, a parity bit is added to the transfer data and the data is checked for parity when received. When the parity function is enabled, the parity mode is selected by PMD/UART_MOD*x* register. Setting PMD to 0 (default) adds a parity bit and checks for even parity. Setting PMD to 1 adds a parity bit and checks for odd parity.

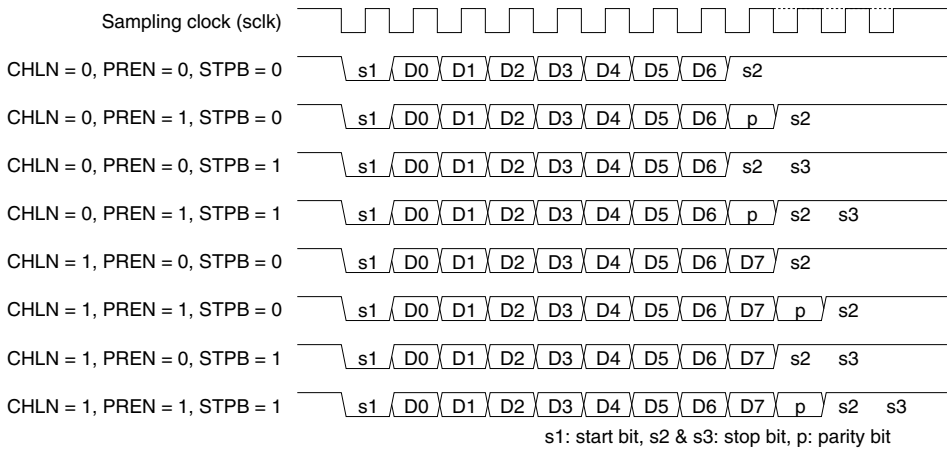


Figure 16.4.1 Transfer Data Format

16.5 Data Transfer Control

Make the following settings before starting data transfers.
(1) Select the input clock. (See Section 16.3.)
(2) Program the baud rate generator to output the transfer clock. (See Section 16.3.)
(3) Set the transfer data format. (See Section 16.4.)
(4) To use the IrDA interface, set IrDA mode. (See Section 16.8.)
(5) Set interrupt conditions to use UART interrupts. (See Section 16.7.)

Note: Make sure the UART is halted (RXEN/UART_CTL*x* register = 0) before changing the above settings.

Enabling data transfers

Set RXEN/UART_CTL*x* register to 1 to enable data transfers. This puts the transmitter/receiver circuit in ready-to-transmit/receive status.

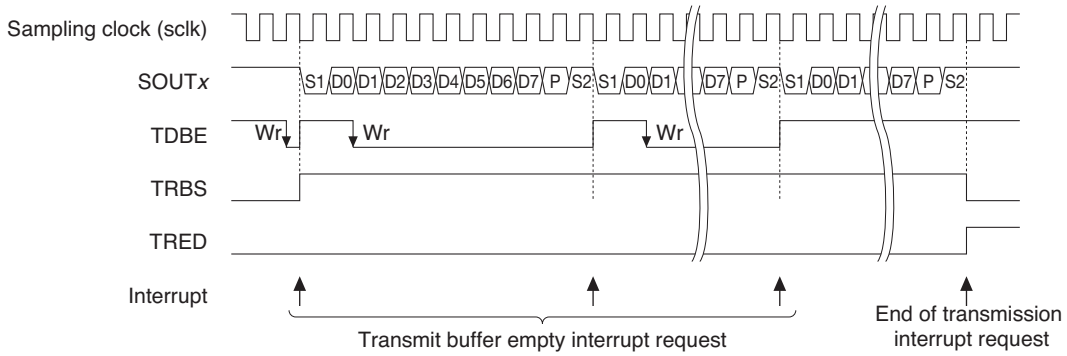
Note: Do not set RXEN to 0 while the UART is sending or receiving data.

Data transmission control

To start data transmission, write the transmit data to TXD[7:0]/UART_TXD*x* register.
The data is written to the transmit data buffer, and the transmitter circuit starts sending data.
The buffer data is sent to the transmit shift register, and the start bit is output from the SOUT*x* pin. The data in the shift register is then output from the LSB. The transfer data bit is shifted in sync with the sampling clock rising edge and output in sequence via the SOUT*x* pin. Following output of MSB, the parity bit (if parity is enabled) and the stop bit are output.

The transmitter circuit includes three status flags: TDBE/UART_ST*x* register, TRBS/UART_ST*x* register, and TRED/UART_ST*x* register.
The TDBE flag indicates the transmit data buffer status. This flag switches to 0 when the application program writes data to the transmit data buffer and reverts to 1 when the buffer data is sent to the transmit shift register. An interrupt can be generated when this flag is set to 1 (see Section 16.7). Subsequent data is sent after confirming that the transmit data buffer is empty either by using this interrupt or by reading the TDBE flag. The transmit data buffer size is 1 byte, but a shift register is provided separately to allow data to be written while the previous data is being sent. Always confirm that the transmit data buffer is empty before writing transmit data. Writing data while the TDBE flag is 0 will overwrite earlier transmit data inside the transmit data buffer.

The TRBS flag indicates the shift register status. This flag switches to 1 when transmit data is loaded from the transmit data buffer to the shift register and reverts to 0 once the data is sent. Read this flag to check whether the transmitter circuit is operating or at standby.
The TRED switches to 1 when the TRBS flag reverts to 0 from 1, indicating that transmit operation has completed. An interrupt can be generated when this flag is set to 1 (see Section 16.7). Use this interrupt for transmission end processing. The TRED flag is reset to 0 by writing 1.



S1: Start bit, S2: Stop bit, P: Parity bit, Wr: Data write to transmit data buffer

Figure 16.5.1 Data Transmission Timing Chart

Data reception control

The receiver circuit is activated by setting RXEN to 1, enabling data to be received from an external serial device.
When the external serial device sends a start bit, the receiver circuit detects its Low level and starts sampling the following data bits. The data bits are sampled at the sampling clock rising edge, and the lead bit is loaded into the receive shift register as LSB. Once the MSB has been received into the shift register, the received data is loaded into the receive data buffer. If parity checking is enabled, the receiver circuit checks the received data at the same time by checking the parity bit received immediately after the MSB.
The receive data buffer, a 2-byte FIFO, receives data until full.

Received data in the buffer can be read from RXD[7:0]/UART_RXD*x* register. The oldest data is read out first and data is cleared by reading.

The receiver circuit includes two buffer status flags: RDRY/UART_ST*x* register and RD2B/UART_ST*x* register.
The RDRY flag indicates that the receive data buffer still contains data. The RD2B flag indicates that the receive data buffer is full.

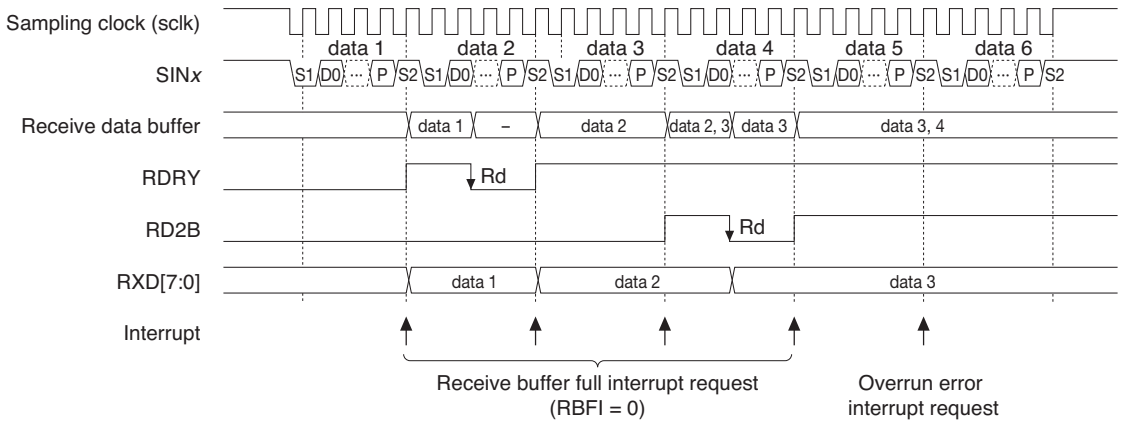
(1) RDRY = 0, RD2B = 0
The receive data buffer contents need not be read, since no data has been received.

(2) RDRY = 1, RD2B = 0
One 8-bit data has been received. Read the receive data buffer contents once. This resets the RDRY flag. The buffer reverts to state (1) above.
If the receive data buffer contents are read twice, the second data read will be invalid.

(3) RDRY = 1, RD2B = 1
Two 8-bit data have been received. Read the receive data buffer contents twice. The receive data buffer outputs the oldest data first. This resets the RD2B flag. The buffer then reverts to the state in (2) above. The second read outputs the most recent received data, after which the buffer reverts to the state in (1) above.
Even when the receive data buffer is full, the shift register can start receiving 8-bit data one more time. An overrun error will occur if receiving is finished before the receive data buffer has been read. In this case, the last received data cannot be read. The contents of the receive data buffer must be read out before an overrun error occurs. For detailed information on overrun errors, refer to Section 16.6.

The volume of data received can be checked by reading these flags.
The UART allows receive buffer full interrupts to be generated once data has been received in the receive data buffer. These interrupts can be used to read the receive data buffer. By default, a receive buffer full interrupt occurs when the receive data buffer receives one 8-bit data (status (2) above). This can be changed by setting RBFI/UART_CTL*x* register to 1 so that an interrupt occurs when the receive data buffer receives two 8-bit data.

Three error flags are also provided in addition to the flags previously mentioned. See Section 16.6 for detailed information on flags and receive errors.



S1: Start bit, S2: Stop bit, P: Parity bit, Rd: Data read from RXD[7:0]

Figure 16.5.2 Data Receiving Timing Chart

Disabling data transfers

Write 0 to RXEN to disable data transfers. The data being transferred cannot be guaranteed if RXEN is set to 0 while data is being sent or received. Before setting RXEN to 0, check the data transfer status with software in consideration of the communication procedure. The data transmit status can be checked using the TRBS flag.

Note: Setting RXEN to 0 empties the transmit data buffer, clearing any remaining data. The data being transferred cannot be guaranteed if RXEN is set to 0 while data is being sent or received.
Make sure that the TDBE flag is 1 and the TRBS and RDRY flags are both 0 before disabling data transfer.

16.6 Receive Errors

Three different receive errors may be detected while receiving data.
Since receive errors are interrupt causes, they can be processed by generating interrupts. For more information on UART interrupt control, see Section 16.7.

Parity error

If PREN/UART_MOD*x* register has been set to 1 (parity enabled), data received is checked for parity.
Data received in the shift register is checked for parity when sent to the receive data buffer. The matching is checked against the PMD/UART_MOD*x* register setting (odd or even parity). If the result is a non-match, a parity error is issued, and the parity error flag PER/UART_ST*x* register is set to 1. Even if this error occurs, the data received is sent to the receive data buffer, and the receiving operation continues. However, the received data cannot be guaranteed if a parity error occurs. The PER flag is reset to 0 by writing 1.

Framing error

A framing error occurs if the stop bit is received as 0 and the UART determines loss of sync. If the stop bit is set to two bits, only the first bit is checked.
The framing error flag FER/UART_ST*x* register is set to 1 if this error occurs. The received data is still transferred to the receive data buffer if this error occurs and the receiving operation continues, but the data cannot be guaranteed, even if no framing error occurs for subsequent data receiving. The FER flag is reset to 0 by writing 1.

Overrun error

Even if the receive data buffer is full (two 8-bit data already received), the third data can be received in the shift register. However, if the receive data buffer is not emptied (by reading out data received) by the time this data has been received, the third data received in the shift register will not be sent to the buffer and generate an overrun error. If an overrun error occurs, the overrun error flag OER/UART_ST*x* register is set to 1. The receiving operation continues even if this error occurs. The OER flag is reset to 0 by writing 1.

16.7 UART Interrupts

The UART includes a function for generating the following four different types of interrupts.
- Transmit buffer empty interrupt
- End of transmission interrupt
- Receive buffer full interrupt
- Receive error interrupt

The UART channel outputs one interrupt signal shared by the four above interrupt causes to the interrupt controller (ITC). Inspect the status flag and error flag to determine the interrupt cause occurred.

Transmit buffer empty interrupt

To use this interrupt, set TIEN/UART_CTL*x* register to 1. If TIEN is set to 1 while TDBE/UART_ST*x* register is 1 (transmit data buffer empty) or if TDBE is set to 1 (when the transmit data buffer becomes empty by loading the transmit data written to it to the shift register) while TIEN is 1, an interrupt request is sent to the ITC. An interrupt occurs if other interrupt conditions are met.

If TIEN is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

You can inspect the TDBE flag in the UART interrupt handler routine to determine whether the UART interrupt is attributable to a transmit buffer empty. If TDBE is 1, the next transmit data can be written to the transmit data buffer by the interrupt handler routine.

End of transmission interrupt

To use this interrupt, set TEIEN/UART_CTL*x* register to 1. If TEIEN is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

When the TRBS flag is reset to 0, the UART sets TRED/UART_ST*x* register to 1, indicating that the transmit operation has completed. If end of transmission interrupts are enabled (TEIEN = 1), an interrupt request is sent simultaneously to the ITC.

An interrupt occurs if other interrupt conditions are met. You can inspect the TRED flag in the UART interrupt handler routine to determine whether the UART interrupt is attributable to an end of transmission. If TRED is 1, the transmission processing can be terminated.

Receive buffer full interrupt

To use this interrupt, set RIEN/UART_CTL*x* register to 1. If RIEN is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

If the specified volume of received data is loaded into the receive data buffer when a receive buffer full interrupt is enabled (RIEN = 1), the UART outputs an interrupt request to the ITC. If RBFI/UART_CTL*x* register is 0, an interrupt request is output as soon as one received data is loaded into the receive data buffer (when RDRY/UART_ST*x* register is set to 1). If RBFI is 1, an interrupt request is output as soon as two received data are loaded into the receive data buffer (when RD2B/UART_ST*x* register is set to 1).

An interrupt occurs if other interrupt conditions are met. You can inspect the RDRY and RD2B flags in the UART interrupt handler routine to determine whether the UART interrupt is attributable to a receive buffer full. If RDRY or RD2B is 1, the received data can be read from the receive data buffer by the interrupt handler routine.

Receive error interrupt

To use this interrupt, set REIEN/UART_CTL*x* register to 1. If REIEN is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

The UART sets an error flag, PER, FER, or OER/UART_ST*x* register to 1 if a parity error, framing error, or overrun error is detected when receiving data. If receive error interrupts are enabled (REIEN = 1), an interrupt request is sent simultaneously to the ITC.

If other interrupt conditions are satisfied, an interrupt occurs. You can inspect the PER, FER, and OER flags in the UART interrupt handler routine to determine whether the UART interrupt was caused by a receive error. If any of the error flags has the value 1, the interrupt handler routine will proceed with error recovery.

For more information on interrupt processing, see the “Interrupt Controller (ITC)” chapter.

16.8 IrDA Interface

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

The transmit data output from the UART transmit shift register is input to the modulator circuit and output from the SOUT*x* pin after the Low pulse has been modulated to a 3 × sclk16 cycle.

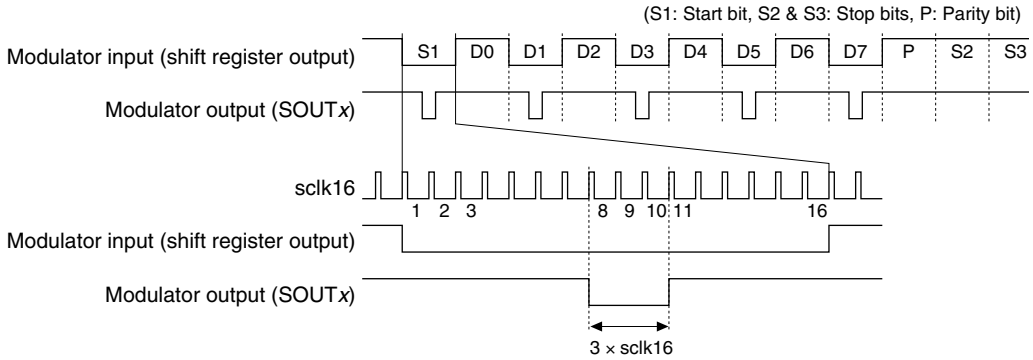


Figure 16.8.1 Transmission Signal Waveform

The received IrDA signal is input to the demodulator circuit and the Low pulse width is converted to 16 × sclk16 cycles before entry to the receive shift register. The demodulator circuit uses the pulse detection clock selected separately from the transfer clock to detect Low pulses input (when minimum pulse width = 1.41 µs/115,200 bps).

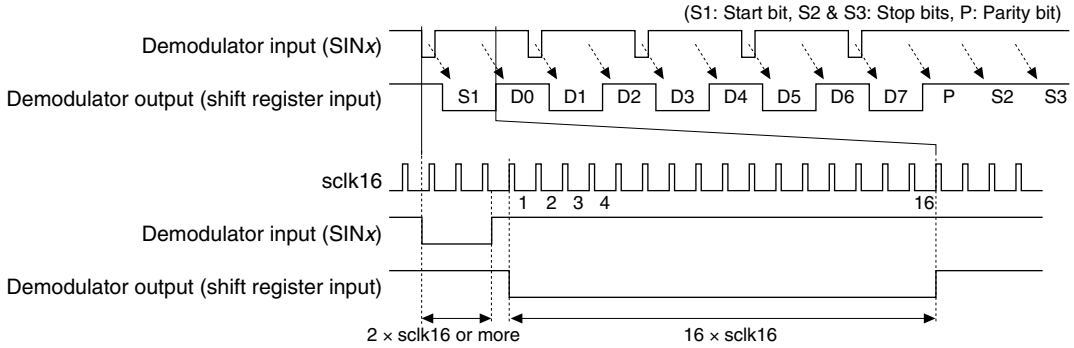


Figure 16.8.2 Receive Signal Waveform

IrDA enable

To use the IrDA interface function, set IRMD/UART_EXP*x* register to 1. This enables the RZI modulator/demodulator circuit.

Note: This setting must be performed before setting other UART conditions.

Serial data transfer control

Data transfer control in IrDA mode is identical to that for normal interfaces. For detailed information on data format settings and data transfer and interrupt control methods, refer to the preceding sections.

16.9 Control Register Details

Table 16.9.1 List of UART Registers

Address	Register name		Function
0x4100	UART_ST0	UART Ch.0 Status Register	Indicates transfer, buffer and error statuses.
0x4101	UART_TXD0	UART Ch.0 Transmit Data Register	Transmit data
0x4102	UART_RXD0	UART Ch.0 Receive Data Register	Receive data
0x4103	UART_MOD0	UART Ch.0 Mode Register	Sets transfer data format.
0x4104	UART_CTL0	UART Ch.0 Control Register	Controls data transfer.
0x4105	UART_EXP0	UART Ch.0 Expansion Register	Sets IrDA mode.
0x4106	UART_BR0	UART Ch.0 Baud Rate Register	Sets baud rate.
0x4107	UART_FMD0	UART Ch.0 Fine Mode Register	Sets fine mode.
0x506c	UART_CLK0	UART Ch.0 Clock Control Register	Selects the baud rate generator clock.

The UART registers are described in detail below.

Notes: • When data is written to the registers, the "Reserved" bits must always be written as 0 and not 1.

• The following UART bits should be set with transfers disabled (RXEN = 0).
- All UART_MOD*x* register bits (STPB, PMD, PREN, CHLN)
- RBFI bit in the UART_CTL*x* register
- All UART_EXP*x* register bits (IRMD)
- All UART_BR*x* register bits (BR[7:0])
- All UART_FMD*x* register bits (FMD[3:0])
- All UART_CLK*x* register bits (UTCLKD[1:0], UTCLKSRC[1:0], UTCLKE)

UART Ch.*x* Status Register (UART_ST*x*)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
UART Ch.*x* Status Register (UART_ST*x*)	**0x4100** (8 bits)	D7	**TRED**	End of transmission flag	1	Completed	0	Not completed	0	R/W	Reset by writing 1.
		D6	**FER**	Framing error flag	1	Error	0	Normal	0	R/W	
		D5	**PER**	Parity error flag	1	Error	0	Normal	0	R/W	
		D4	**OER**	Overrun error flag	1	Error	0	Normal	0	R/W	
		D3	**RD2B**	Second byte receive flag	1	Ready	0	Empty	0	R	
		D2	**TRBS**	Transmit busy flag	1	Busy	0	Idle	0	R	Shift register status
		D1	**RDRY**	Receive data ready flag	1	Ready	0	Empty	0	R	
		D0	**TDBE**	Transmit data buffer empty flag	1	Empty	0	Not empty	1	R	

D7 TRED: End of Transmission Flag Bit

Indicates whether the transmit operation has completed or not.

1 (R): Completed
0 (R): Not completed (default)
1 (W): Reset to 0
0 (W): Ignored

TRED is set to 1 when the TRBS flag is reset to 0 (when transmission has completed).
TRED is reset by writing 1.

D6 FER: Framing Error Flag Bit

Indicates whether a framing error has occurred or not.

1 (R): Error occurred
0 (R): No error (default)
1 (W): Reset to 0
0 (W): Ignored

FER is set to 1 when a framing error occurs. Framing errors occur when data is received with the stop bit set to 0. FER is reset by writing 1.

D5 PER: Parity Error Flag Bit

Indicates whether a parity error has occurred or not.

1 (R): Error occurred
0 (R): No error (default)
1 (W): Reset to 0
0 (W): Ignored

PER is set to 1 when a parity error occurs. Parity checking is enabled only when PREN/ UART_MOD*x* register is set to 1 and is performed when received data is transferred from the shift register to the receive data buffer. PER is reset by writing 1.

D4 OER: Overrun Error Flag Bit

Indicates whether an overrun error has occurred or not.

1 (R): Error occurred
0 (R): No error (default)
1 (W): Reset to 0
0 (W): Ignored

OER is set to 1 when an overrun error occurs. Overrun errors occur if the receive data buffer is full when data is received in the shift register. The receive data buffer is not overwritten even if this error occurs. The shift register is overwritten as soon as the error occurs.
OER is reset by writing 1.

D3 RD2B: Second Byte Receive Flag Bit

Indicates that the receive data buffer contains two received data.

1 (R): Second byte can be read
0 (R): Second byte not received (default)

RD2B is set to 1 when the second byte of data is loaded into the receive data buffer and is reset to 0 when the first data is read from the receive data buffer.

D2 TRBS: Transmit Busy Flag Bit

Indicates the transmit shift register status.

1 (R): Operating
0 (R): Standby (default)

TRBS is set to 1 when transmit data is loaded from the transmit data buffer into the shift register and is reset to 0 when the data transfer is completed. Inspect TRBS to determine whether the transmit circuit is operating or at standby.

D1 RDRY: Receive Data Ready Flag Bit

Indicates that the receive data buffer contains valid received data.

1 (R): Data can be read
0 (R): Buffer empty (default)

RDRY is set to 1 when received data is loaded into the receive data buffer and is reset to 0 when all data has been read from the receive data buffer.

D0 TDBE: Transmit Data Buffer Empty Flag Bit

Indicates the transmit data buffer status.

1 (R): Buffer empty (default)
0 (R): Data exists

TDBE is reset to 0 when transmit data is written to the transmit data buffer and is set to 1 when the data is transferred to the shift register.

UART Ch.*x* Transmit Data Register (UART_TXD*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
UART Ch.*x* Transmit Data Register (UART_TXD*x*)	**0x4101** (8 bits)	D7–0	**TXD[7:0]**	Transmit data TXD7(6) = MSB TXD0 = LSB	0x0 to 0xff (0x7f)	0x0	R/W	

D[7:0] TXD[7:0]: Transmit Data

Write transmit data to be set in the transmit data buffer. (Default: 0x0)

The UART starts transmitting when data is written to this register. Data written to TXD[7:0] is retained until sent to the transmit data buffer.

Transmitting data from within the transmit data buffer generates a cause of transmit buffer empty interrupt.

TXD7 (MSB) is invalid in 7-bit mode.

Serial converted data is output from the SOUT*x* pin beginning with the LSB, in which the bits set to 1 are output as High level and bits set to 0 as Low level signals.

This register can also be read.

UART Ch.*x* Receive Data Register (UART_RXD*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
UART Ch.*x* Receive Data Register (UART_RXD*x*)	**0x4102** (8 bits)	D7–0	**RXD[7:0]**	Receive data in the receive data buffer RXD7(6) = MSB RXD0 = LSB	0x0 to 0xff (0x7f)	0x0	R	Older data in the buffer is read out first.

D[7:0] RXD[7:0]: Receive Data

Data in the receive data buffer is read out in sequence, starting with the oldest. Received data is placed in the receive data buffer. The receive data buffer is a 2-byte FIFO that allows proper data reception until it fills, even if data is not read out. If the buffer is full and the shift register also contains received data, an overrun error will occur, unless the data is read out before reception of the subsequent data starts.

The receive circuit includes two receive buffer status flags: RDRY/UART_ST*x* register and RD2B/UART_ST*x* register. The RDRY flag indicates the presence of valid received data in the receive data buffer, while the RD2B flag indicates the presence of two received data in the receive data buffer.

A receive buffer full interrupt occurs when the received data in the receive data buffer reaches the number specified by RBFI/UART_CTL*x* register.

0 is loaded into RXD7 in 7-bit mode.

Serial data input via the SIN*x* pin is converted to parallel, with the initial bit as LSB, the High level bit as 1, and the Low level bit as 0. This data is then loaded into the receive data buffer.

This register is read-only. (Default: 0x0)

UART Ch.*x* Mode Register (UART_MOD*x*)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
UART Ch.*x* Mode Register (UART_MOD*x*)	**0x4103** (8 bits)	D7–5	**–**	reserved	–				–	–	0 when being read.
		D4	**CHLN**	Character length select	1	8 bits	0	7 bits	0	R/W	
		D3	**PREN**	Parity enable	1	With parity	0	No parity	0	R/W	
		D2	**PMD**	Parity mode select	1	Odd	0	Even	0	R/W	
		D1	**STPB**	Stop bit select	1	2 bits	0	1 bit	0	R/W	
		D0	**–**	reserved	–				–	–	0 when being read.

D[7:5] Reserved

D4 CHLN: Character Length Select Bit

Selects the serial transfer data length.

1 (R/W): 8 bits

0 (R/W): 7 bits (default)

D3 **PREN: Parity Enable Bit**

Enables the parity function.

1 (R/W): With parity

0 (R/W): No parity (default)

PREN is used to select whether received data parity checking is performed and whether a parity bit is added to transmit data. Setting PREN to 1 parity-checks the received data. A parity bit is automatically added to the transmit data. If PREN is set to 0, no parity bit is checked or added.

D2 **PMD: Parity Mode Select Bit**

Selects the parity mode.

1 (R/W): Odd parity

0 (R/W): Even parity (default)

Writing 1 to PMD selects odd parity; writing 0 to it selects even parity. Parity checking and parity bit addition are enabled only when PREN is set to 1. The PMD setting is disabled if PREN is 0.

D1 **STPB: Stop Bit Select Bit**

Selects the stop bit length.

1 (R/W): 2 bits

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

Writing 1 to STPB selects 2 stop bits; writing 0 to it selects 1 bit. The start bit is fixed at 1 bit.

D0 **Reserved**

UART Ch.*x* Control Register (UART_CTL*x*)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
UART Ch.*x* Control Register (UART_CTL*x*)	0x4104 (8 bits)	D7	TEIEN	End of transmission int. enable	1	Enable	0	Disable	0	R/W	
		D6	REIEN	Receive error int. enable	1	Enable	0	Disable	0	R/W	
		D5	RIEN	Receive buffer full int. enable	1	Enable	0	Disable	0	R/W	
		D4	TIEN	Transmit buffer empty int. enable	1	Enable	0	Disable	0	R/W	
		D3–2	–	reserved	–				–	–	0 when being read.
		D1	RBFI	Receive buffer full int. condition setup	1	2 bytes	0	1 byte	0	R/W	
		D0	RXEN	UART enable	1	Enable	0	Disable	0	R/W	

D7 **TEIEN: End of Transmission Interrupt Enable Bit**

Enables interrupt requests to the ITC when transmit operation has completed.

1 (R/W): Enabled

0 (R/W): Disabled (default)

Set this bit to 1 to terminate transmit processing using interrupts.

D6 **REIEN: Receive Error Interrupt Enable Bit**

Enables interrupt requests to the ITC when a receive error occurs.

1 (R/W): Enabled

0 (R/W): Disabled (default)

Set this bit to 1 to process receive errors using interrupts.

D5 **RIEN: Receive Buffer Full Interrupt Enable Bit**

Enables interrupt requests to the ITC caused when the received data quantity in the receive data buffer reaches the quantity specified in RBFI.

1 (R/W): Enabled

0 (R/W): Disabled (default)

Set this bit to 1 to read received data using interrupts.

D4 **TIEN: Transmit Buffer Empty Interrupt Enable Bit**

Enables interrupt requests to the ITC caused when transmission data in the transmit data buffer is sent to the shift register (i.e. when data transmission begins).

1 (R/W): Enabled

0 (R/W): Disabled (default)

Set this bit to 1 to write data to the transmit data buffer using interrupts.

D[3:2] Reserved

D1 RBFI: Receive Buffer Full Interrupt Condition Setup Bit

Sets the quantity of data in the receive data buffer to generate a receive buffer full interrupt.
1 (R/W): 2 bytes
0 (R/W): 1 byte (default)

If receive buffer full interrupts are enabled (RIEN = 1), the UART outputs an interrupt request to the ITC when the quantity of received data specified by RBFI is loaded into the receive data buffer.
If RBFI is 0, an interrupt request is output as soon as one received data is loaded into the receive data buffer (when RDRY/UART_ST*x* register is set to 1). If RBFI is 1, an interrupt request is output as soon as two received data are loaded into the receive data buffer (when RD2B/UART_ST*x* register is set to 1).

D0 RXEN: UART Enable Bit

Enables data transfer by the UART.
1 (R/W): Enabled
0 (R/W): Disabled (default)

Set RXEN to 1 before starting UART transfers. Setting RXEN to 0 disables data transfers. Set the transfer conditions while RXEN is 0.
Disabling transfers by writing 0 to RXEN also clears the transmit data buffer.

UART Ch.*x* Expansion Register (UART_EXP*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
UART Ch.*x* Expansion Register (UART_EXP*x*)	0x4105 (8 bits)	D7–1	–	reserved	–	–	–	0 when being read.
		D0	IRMD	IrDA mode select	1 On 0 Off	0	R/W	

D[7:1] Reserved

D0 IRMD: IrDA Mode Select Bit

Switches the IrDA interface function on and off.
1 (R/W): On
0 (R/W): Off (default)

Set IRMD to 1 to use the IrDA interface. When IRMD is set to 0, this module functions as a normal UART, with no IrDA functions.

UART Ch.*x* Baud Rate Register (UART_BR*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
UART Ch.*x* Baud Rate Register (UART_BR*x*)	0x4106 (8 bits)	D7–0	BR[7:0]	Baud rate setting	0x0 to 0xff	0x0	R/W	

D[7:0] BR[7:0]: Baud Rate Setting Bits

Sets the initial counter value of the baud rate generator. (Default: 0x0)
The counter in the baud rate generator repeats counting from the value set in this register to occurrence of counter underflow to generate the transfer (sampling) clock.
Use the following equations to calculate the initial counter value for obtaining the desired transfer rate.

$$bps = \frac{ct\_clk}{\{(BR + 1) \times 16 + FMD\}}$$

$$BR = \left(\frac{ct\_clk}{bps} - FMD - 16\right) \div 16$$

ct_clk: Count clock frequency (Hz)
BR: BR[7:0] setting (0 to 255)
bps: Transfer rate (bit/s)
FMD: FMD[3:0] (fine mode) setting (0 to 15)

UART Ch.*x* Fine Mode Register (UART_FMD*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
UART Ch.*x* Fine Mode Register (UART_FMD*x*)	**0x4107** (8 bits)	D7–4	**–**	reserved	–	–	–	0 when being read.
		D3–0	**FMD[3:0]**	Fine mode setup	0x0 to 0xf	0x0	R/W	Set a number of times to insert delay into a 16-underflow period.

D[7:4] Reserved

D[3:0] FMD[3:0]: Fine Mode Setup Bits

Corrects the transfer rate error. (Default: 0x0)

FMD[3:0] specifies the delay pattern to be inserted into a 16 underflow period of the baud rate generator output clock. Inserting one delay extends the output clock cycle by one count clock cycle.

Table 16.9.2 Delay Patterns Specified by FMD[3:0]

FMD[3:0]	Underflow number															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0x0	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
0x1	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	D
0x2	–	–	–	–	–	–	–	D	–	–	–	–	–	–	–	D
0x3	–	–	–	–	–	–	–	D	–	–	–	D	–	–	–	D
0x4	–	–	–	D	–	–	–	D	–	–	–	D	–	–	–	D
0x5	–	–	–	D	–	–	–	D	–	–	–	D	–	D	–	D
0x6	–	–	–	D	–	D	–	D	–	–	–	D	–	D	–	D
0x7	–	–	–	D	–	D	–	D	–	D	–	D	–	D	–	D
0x8	–	D	–	D	–	D	–	D	–	D	–	D	–	D	–	D
0x9	–	D	–	D	–	D	–	D	–	D	–	D	–	D	D	D
0xa	–	D	–	D	–	D	D	D	–	D	–	D	–	D	D	D
0xb	–	D	–	D	–	D	D	D	–	D	D	D	–	D	D	D
0xc	–	D	D	D	–	D	D	D	–	D	D	D	–	D	D	D
0xd	–	D	D	D	–	D	D	D	–	D	D	D	D	D	D	D
0xe	–	D	D	D	D	D	D	D	–	D	D	D	D	D	D	D
0xf	–	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D

D: Indicates the insertion of a delay cycle.

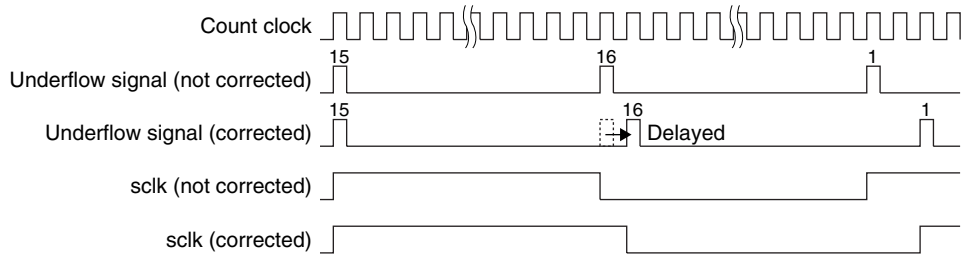


Figure 16.9.1 Delay Cycle Insertion in Fine Mode

UART Ch.*x* Clock Control Register (UART_CLK*x*)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
UART Ch.*x* Clock Control Register (UART_CLK*x*)	**0x506c** (8 bits)	D7–6	**–**	reserved	–		–	–	0 when being read.
		D5–4	**UTCLKD [1:0]**	Clock division ratio select	UTCLKD[1:0]	Division ratio	0x0	R/W	When the clock source is OSC3B or OSC3A
					0x3	1/8			
					0x2	1/4			
					0x1	1/2			
					0x0	1/1			
		D3–2	**UTCLKSRC [1:0]**	Clock source select	UTCLKSRC [1:0]	Clock source	0x0	R/W	
					0x3	External clock			
					0x2	OSC3A			
					0x1	OSC1			
					0x0	OSC3B			
		D1	**–**	reserved	–		–	–	0 when being read.
		D0	**UTCLKE**	Count clock enable	1 Enable	0 Disable	0	R/W	

D[7:6] Reserved

D[5:4] UTCLKD[1:0]: Clock Division Ratio Select Bits

Selects the division ratio for generating the count clock of the baud rate generator when OSC3B or OSC3A is used as the clock source.

Table 16.9.3 OSC3B/OSC3A Division Ratio Selection

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

(Default: 0x0)

D[3:2] UTCLKSRC[1:0]: Clock Source Select Bits

Selects the count clock source for the baud rate generator.

Table 16.9.4 Clock Source Selection

UTCLKSRC[1:0]	Clock source
0x3	External clock (SCLK*x*)
0x2	OSC3A
0x1	OSC1
0x0	OSC3B

(Default: 0x0)

D1 Reserved

D0 UTCLKE: Count Clock Enable Bit

Enables or disables the count clock supply to the counter of the baud rate generator.

1 (R/W): Enabled (on)

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

The UTCLKE default setting is 0, which disables the clock supply. Setting UTCLKE to 1 sends the clock selected to the counter.

17 SPI

17.1 SPI Module Overview

The S1C17F57 includes a synchronized serial interface module (SPI).
The following shows the main features of the SPI:

- Number of channels: 1 channel
- Supports both master and slave modes.
- Data length: 8 bits fixed
- Supports both MSB first and LSB first modes.
- Contains one-byte receive data buffer and one-byte transmit data buffer.
- Supports full-duplex communications.
- Data transfer timing (clock phase and polarity variations) is selectable from among 4 types.
- Can generate receive buffer full and transmit buffer empty interrupts.

Figure 17.1.1 shows the SPI module configuration.

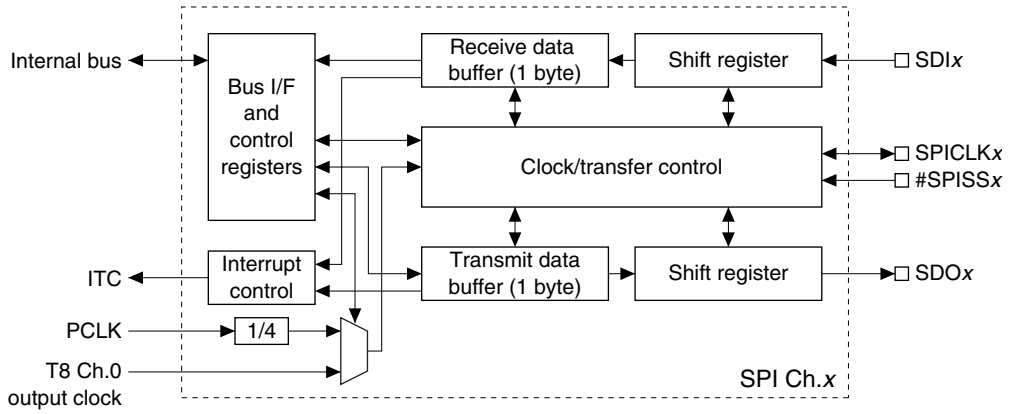


Figure 17.1.1 SPI Module Configuration

Note: The letter '*x*' in register and pin names refers to a channel number (0).
Example: SPI_CTL*x* register
Ch.0: SPI_CTL0 register

17.2 SPI Input/Output Pins

Table 17.2.1 lists the SPI pins.

Table 17.2.1 List of SPI Pins

Pin name	I/O	Qty	Function
SDI0 (Ch.0)	I	1	SPI data input pin Inputs serial data from SPI bus.
SDO0 (Ch.0)	O	1	SPI data output pin Outputs serial data to SPI bus.
SPICLK0 (Ch.0)	I/O	1	SPI external clock input/output pin Outputs SPI clock when SPI is in master mode. Inputs external clock when SPI is used in slave mode.
#SPISS0 (Ch.0)	I	1	SPI slave select signal (active Low) input pin SPI (Slave mode) is selected as a slave device by Low input to this pin.

Note: Use an I/O (P) port to output the slave select signal when the SPI module is configured to master mode.

The SPI input/output pins (SDI*x*, SDO*x*, SPICLK*x*, #SPISS*x*) are shared with I/O ports and are initially set as general purpose I/O port pins. The pin functions must be switched using the port function select bits to use the general purpose I/O port pins as SPI input/output pins.
For detailed information on pin function switching, see the "I/O Ports (P)" chapter.

17.3 SPI Clock

The master mode SPI uses the 8-bit timer (T8) Ch.0 output clock or a PCLK/4 clock to generate the SPI clock. This clock is output from the SPICLK*x* pin to the slave device while also driving the shift register.
Use MCLK/SPI_CTL*x* register to select whether the T8 Ch.0 output clock or PCLK/4 clock is used.
Setting MCLK to 1 selects the T8 Ch.0 output clock; setting to 0 selects the PCLK/4 clock.
Using the T8 Ch.0 output clock enables programmable transfer rates. For more information on T8 control, see the "8-bit timers (T8)" chapter.

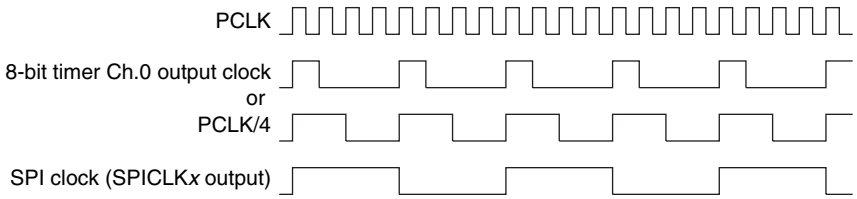


Figure 17.3.1 Master Mode SPI Clock

In slave mode, the SPI clock is input via the SPICLK*x* pin.

17.4 Data Transfer Condition Settings

The SPI module can be set to master or slave modes. The SPI clock polarity/phase and bit direction (MSB first/LSB first) can also be set via the SPI_CTL*x* register. The data length is fixed at 8 bits.

Note: Make sure the SPI module is halted (SPEN/SPI_CTL*x* register = 0) before master/slave mode selection and clock condition settings.

Master/slave mode selection

MSSL/SPI_CTL*x* register is used to set the SPI module to master mode or slave mode. Setting MSSL to 1 sets master mode; setting it to 0 (default) sets slave mode. In master mode, data is transferred using the internal clock. In slave mode, data is transferred by inputting the master device clock.

SPI clock polarity and phase settings

The SPI clock polarity is selected by CPOL/SPI_CTL*x* register. Setting CPOL to 1 treats the SPI clock as active Low; setting it to 0 (default) treats it as active High.
The SPI clock phase is selected by CPHA/SPI_CTL*x* register.

As shown below, these control bits set transfer timing.

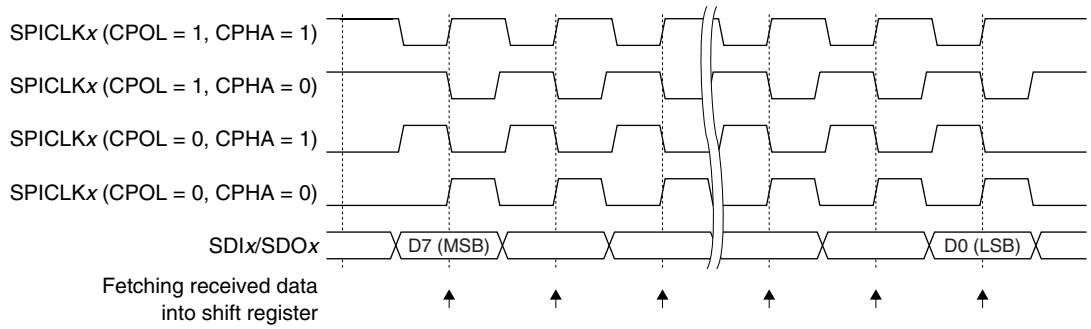


Figure 17.4.1 Clock and Data Transfer Timing

MSB first/LSB first settings

Use MLSB/SPI_CTL*x* register to select whether the data MSB or LSB is input/output first.
MSB first is selected when MLSB is 0 (default); LSB first is selected when MLSB is 1.

17.5 Data Transfer Control

Make the following settings before starting data transfers.
(1) Select the SPI clock source. (See Section 17.3.)
(2) Select master mode or slave mode. (See Section 17.4.)
(3) Set clock conditions. (See Section 17.4.)
(4) Set the interrupt conditions to use SPI interrupts. (See Section 17.6.)

Note: Make sure the SPI is halted (SPEN/SPI_CTL*x* register = 0) before setting the above conditions.

Enabling data transfers

Set SPEN/SPI_CTL*x* register to 1 to enable SPI operations. This enables SPI transfers and clock input/output.

Note: Do not set SPEN to 0 when the SPI module is transferring data.

Data transmission control

To start data transmission, write the transmit data to SPTDB[7:0]/SPI_TXD*x* register.
The data is written to the transmit data buffer, and the SPI module starts sending data. The buffer data is sent to the transmit shift register. In master mode, the module starts clock output from the SPICLK*x* pin. In slave mode, the module awaits clock input from the SPICLK*x* pin. The data in the shift register is shifted in sequence at the clock rising or falling edge, as determined by CPHA/SPI_CTL*x* register and CPOL/SPI_CTL*x* register (see Figure 17.4.1) and sent from the SDO*x* pin.

Note: Make sure that SPEN is set to 1 before writing data to the SPI_TXD*x* register.

The SPI module includes two status flags for transfer control: SPTBE/SPI_ST*x* register and SPBSY/SPI_ST*x* register.
The SPTBE flag indicates the transmit data buffer status. This flag switches to 0 when the application program writes data to the SPI_TXD*x* register (transmit data buffer) and reverts to 1 when the buffer data is sent to the transmit shift register. An interrupt can be generated when this flag is set to 1 (see Section 17.6). Subsequent data is sent after confirming that the transmit data buffer is empty either by using this interrupt or by inspecting the SPTBE flag. The transmit data buffer size is 1 byte, but a shift register is provided separately to allow data to be written while the previous data is being sent. Always confirm that the transmit data buffer is empty before writing transmit data. Writing data while the SPTBE flag is 0 will overwrite earlier transmit data inside the transmit data buffer.

In master mode, the SPBSY flag indicates the shift register status. This flag switches to 1 when transmit data is loaded from the transmit data buffer to the shift register and reverts to 0 once the data is sent. Read this flag to check whether the SPI module is operating or at standby.
In slave mode, SPBSY flag indicates the SPI slave selection signal (#SPISS*x* pin) status. The flag is set to 1 when the SPI module is selected as a slave module and is set to 0 when the module is not selected.

Note: When the SPI module is used in master mode with CPHA set to 0, the clock may change a minimum of one system clock (PCLK) cycle time from change of the first transmit data bit.

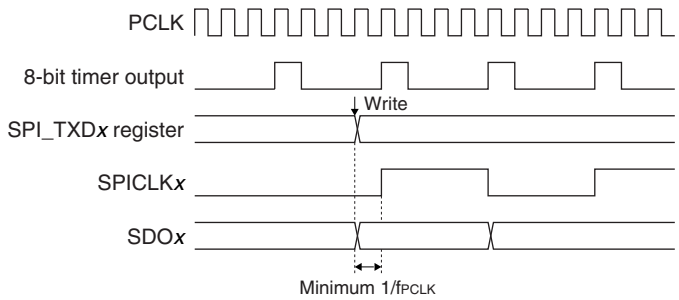


Figure 17.5.1 SDO*x* and SPICLK*x* Change Timings when CPHA = 0

The half SPICLK*x* cycle will be secured from change of data to change of the clock for the second and following transmit data bits and the second and following bytes during continuous transfer.

Data reception control

In master mode, write dummy data to SPTDB[7:0]/SPI_TXD*x* register. Writing to the SPI_TXD*x* register creates the trigger for reception as well as transmission start. Writing actual transmit data enables simultaneous transmission and reception.
This starts the SPI clock output from the SPICLK*x* pin.

Note: Make sure that SPEN is set to 1 before writing data to the SPI_TXD*x* register.

In slave mode, the module waits until the clock is input from the SPICLK*x* pin. There is no need to write to the SPI_TXD*x* register if no transmission is required. The receiving operation is started by the clock input from the master device. If data is transmitted simultaneously, write transmit data to the SPI_TXD*x* register before the clock is input.

The data is received in sequence in the shift register at the rising or falling edge of the clock determined by CPHA/SPI_CTL*x* register and CPOL/SPI_CTL*x* register. (See Figure 17.4.1.) The received data is loaded into the receive data buffer once the 8 bits of data are received in the shift register.
The received data in the buffer can be read from SPRDB[7:0]/SPI_RXD*x* register.

The SPI module includes SPRBF/SPI_ST*x* register for reception control.
The SPRBF flag indicates the receive data buffer status. This flag is set to 1 when the data received in the shift register is loaded into the receive data buffer, indicating that the received data can be read out. It reverts to 0 when the buffer data is read out from the SPI_RXD*x* register. An interrupt can be generated as soon as the flag is set to 1 (see Section 17.6). The received data should be read out either by using this interrupt or by inspecting the SPRBF flag to confirm that the receive data buffer contains valid received data. The receive data buffer is 1 byte in size, but a shift register is also provided, enabling received data to be retained in the buffer even while the subsequent data is being received. Note that the receive data buffer should be read out before receiving the subsequent data is complete. If receiving the subsequent data is complete before the receive data buffer contents are read out, the newly received data will overwrite the previous received data in the buffer.

In master mode, the SPBSY flag indicating the shift register status can be used in the same way while transferring data.

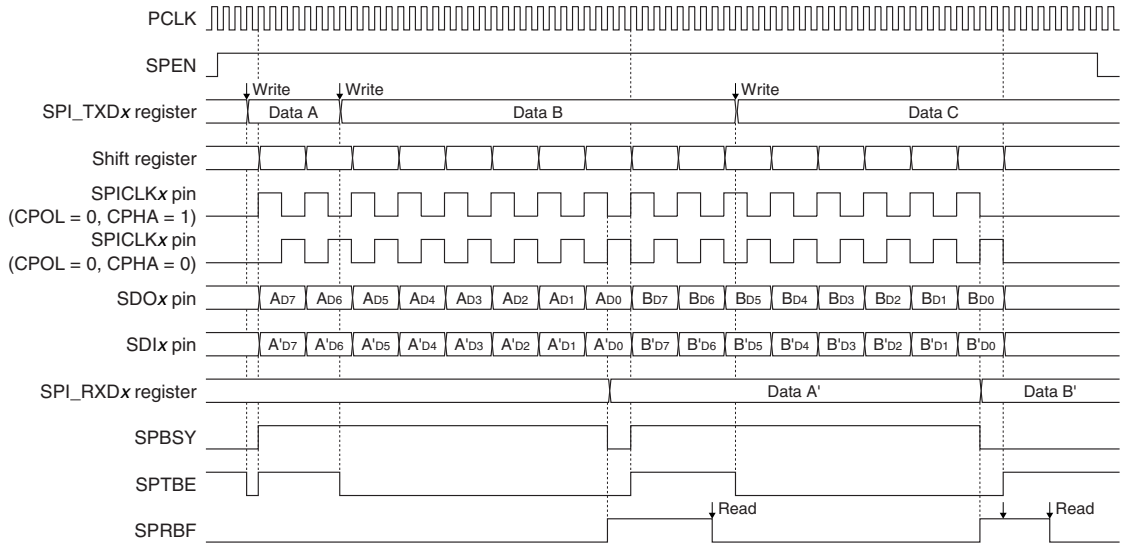


Figure 17.5.2 Data Transmission/Receiving Timing Chart (MSB first)

Disabling data transfers

After a data transfer is completed (both transmission and reception), write 0 to SPEN to disable data transfers. Confirm that the SPTBE flag is 1 and the SPBSY flag is 0 before disabling data transfer.

The data being transferred cannot be guaranteed if SPEN is set to 0 while data is being sent or received.

17.6 SPI Interrupts

Each channel of the SPI module includes a function for generating the following two different types of interrupts.

- Transmit buffer empty interrupt
- Receive buffer full interrupt

The SPI channel outputs one interrupt signal shared by the two above interrupt causes to the interrupt controller (ITC). Inspect the status flag to determine the interrupt cause occurred.

Transmit buffer empty interrupt

To use this interrupt, set SPTIE/SPI_CTL*x* register to 1. If SPTIE is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

When transmit data written to the transmit data buffer is transferred to the shift register, the SPI module sets SPTBE/SPI_ST*x* register to 1, indicating that the transmit data buffer is empty. If transmit buffer empty interrupts are enabled (SPTIE = 1), an interrupt request is sent simultaneously to the ITC.

An interrupt occurs if other interrupt conditions are met. You can inspect the SPTBE flag in the SPI interrupt handler routine to determine whether the SPI interrupt is attributable to a transmit buffer empty. If SPTBE is 1, the next transmit data can be written to the transmit data buffer by the interrupt handler routine.

Receive buffer full interrupt

To use this interrupt, set SPRIE/SPI_CTL*x* register to 1. If SPRIE is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

When data received in the shift register is loaded into the receive data buffer, the SPI module sets SPRBF/SPI_ST*x* register to 1, indicating that the receive data buffer contains readable received data. If receive buffer full interrupts are enabled (SPRIE = 1), an interrupt request is output to the ITC at the same time.

An interrupt occurs if other interrupt conditions are met. You can inspect the SPRBF flag in the SPI interrupt handler routine to determine whether the SPI interrupt is attributable to a receive buffer full. If SPRBF is 1, the received data can be read from the receive data buffer by the interrupt handler routine.

For more information on interrupt processing, see the “Interrupt Controller (ITC)” chapter.

17.7 Control Register Details

Table 17.7.1 List of SPI Registers

Address	Register name		Function
0x4320	SPI_ST0	SPI Ch.0 Status Register	Indicates transfer and buffer statuses.
0x4322	SPI_TXD0	SPI Ch.0 Transmit Data Register	Transmit data
0x4324	SPI_RXD0	SPI Ch.0 Receive Data Register	Receive data
0x4326	SPI_CTL0	SPI Ch.0 Control Register	Sets the SPI mode and enables data transfer.

The SPI registers are described in detail below.

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

SPI Ch.*x* Status Register (SPI_ST*x*)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
SPI Ch.*x* Status Register (SPI_ST*x*)	**0x4320** (16 bits)	D15–3	**–**	reserved	–				–	–	0 when being read.
		D2	**SPBSY**	Transfer busy flag (master)	1	Busy	0	Idle	0	R	
				ss signal low flag (slave)	1	ss = L	0	ss = H			
		D1	**SPRBF**	Receive data buffer full flag	1	Full	0	Not full	0	R	
		D0	**SPTBE**	Transmit data buffer empty flag	1	Empty	0	Not empty	1	R	

D[15:3] Reserved

D2 SPBSY: Transfer Busy Flag Bit (Master Mode)/ss Signal Low Flag Bit (Slave Mode)

Master mode

Indicates the SPI transfer status.

1 (R): Operating

0 (R): Standby (default)

SPBSY is set to 1 when the SPI starts data transfer in master mode and is maintained at 1 while transfer is underway. It is cleared to 0 once the transfer is complete.

Slave mode

Indicates the slave selection (#SPISS*x*) signal status.

1 (R): Low level (this SPI is selected)

0 (R): High level (this SPI is not selected) (default)

SPBSY is set to 1 when the master device asserts the #SPISS*x* signal to select this SPI module (slave device). It is returned to 0 when the master device clears the SPI module selection by negating the #SPISS*x* signal.

D1 SPRBF: Receive Data Buffer Full Flag Bit

Indicates the receive data buffer status.

1 (R): Data full

0 (R): No data (default)

SPRBF is set to 1 when data received in the shift register is sent to the receive data buffer (when receiving is completed), indicating that the data can be read. It reverts to 0 once the buffer data is read from the SPI_RXD*x* register.

D0 SPTBE: Transmit Data Buffer Empty Flag Bit

Indicates the transmit data buffer status.

1 (R): Empty (default)

0 (R): Data exists

SPTBE is set to 0 when transmit data is written to the SPI_TXD*x* register (transmit data buffer), and is set to 1 when the data is transferred to the shift register (when transmission starts).
Transmission data must be written to the SPI_TXD*x* register when this bit is 1.

SPI Ch.*x* Transmit Data Register (SPI_TXD*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SPI Ch.*x* Transmit Data Register (SPI_TXD*x*)	0x4322 (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	SPTDB[7:0]	SPI transmit data buffer SPTDB7 = MSB SPTDB0 = LSB	0x0 to 0xff	0x0	R/W	

D[15:8] Reserved

D[7:0] SPTDB[7:0]: SPI Transmit Data Buffer Bits

Sets transmit data to be written to the transmit data buffer. (Default: 0x0)

In master mode, transmission is started by writing data to this register. In slave mode, the contents of this register are sent to the shift register and transmission begins when the clock is input from the master.

SPTBE/SPI_ST*x* register is set to 1 (empty) as soon as data written to this register has been transferred to the shift register. A transmit buffer empty interrupt is generated at the same time. The subsequent transmit data can then be written, even while data is being transmitted.

Serial converted data is output from the SDO*x* pin, with the bit set to 1 as High level and the bit set to 0 as Low level.

Note: Make sure that SPEN is set to 1 before writing data to the SPI_TXD*x* register to start data transmission/reception.

SPI Ch.*x* Receive Data Register (SPI_RXD*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SPI Ch.*x* Receive Data Register (SPI_RXD*x*)	0x4324 (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	SPRDB[7:0]	SPI receive data buffer SPRDB7 = MSB SPRDB0 = LSB	0x0 to 0xff	0x0	R	

D[15:8] Reserved

D[7:0] SPRDB[7:0]: SPI Receive Data Buffer Bits

Contains the received data. (Default: 0x0)

SPRBF/SPI_ST*x* register is set to 1 (data full) as soon as data is received and the shift register data has been transferred to the receive data buffer. A receive buffer full interrupt is generated at the same time. Data can then be read until subsequent data is received. If receiving the subsequent data is completed before the register has been read out, the new received data overwrites the contents.

Serial data input from the SDI*x* pin is converted to parallel, with the High level bit set to 1 and the Low level bit set to 0. The data is the loaded into this register.

This register is read-only.

SPI Ch.*x* Control Register (SPI_CTL*x*)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
SPI Ch.*x* Control Register (SPI_CTL*x*)	0x4326 (16 bits)	D15–10	–	reserved	–				–	–	0 when being read.
		D9	MCLK	SPI clock source select	1	T8 Ch.0	0	PCLK/4	0	R/W	
		D8	MLSB	LSB/MSB first mode select	1	LSB	0	MSB	0	R/W	
		D7–6	–	reserved	–				–	–	0 when being read.
		D5	SPRIE	Receive data buffer full int. enable	1	Enable	0	Disable	0	R/W	
		D4	SPTIE	Transmit data buffer empty int. enable	1	Enable	0	Disable	0	R/W	
		D3	CPHA	Clock phase select	1	Data out	0	Data in	0	R/W	These bits must be set before setting SPEN to 1.
		D2	CPOL	Clock polarity select	1	Active L	0	Active H	0	R/W	
		D1	MSSL	Master/slave mode select	1	Master	0	Slave	0	R/W	
		D0	SPEN	SPI enable	1	Enable	0	Disable	0	R/W	

Note: Do not access to the SPI_CTL*x* register while SPBSY/SPI_ST*x* register is set to 1 or SPRBF/SPI_ST*x* register is set to 1 (while data is being transmitted/received).

D[15:10] Reserved

D9 MCLK: SPI Clock Source Select Bit

Selects the SPI clock source.
1 (R/W): 8-bit timer Ch.0
0 (R/W): PCLK/4 (default)

D8 MLSB: LSB/MSB First Mode Select Bit

Selects whether data is transferred with MSB first or LSB first.
1 (R/W): LSB first
0 (R/W): MSB first (default)

D[7:6] Reserved

D5 SPRIE: Receive Data Buffer Full Interrupt Enable Bit

Enables or disables SPI receive data buffer full interrupts.
1 (R/W): Enabled
0 (R/W): Disabled (default)

Setting SPRIE to 1 enables the output of SPI interrupt requests to the ITC due to a receive data buffer full. These interrupt requests are generated when the data received in the shift register is transferred to the receive data buffer (when reception is completed).
SPI interrupts are not generated by receive data buffer full if SPRIE is set to 0.

D4 SPTIE: Transmit Data Buffer Empty Interrupt Enable Bit

Enables or disables SPI transmit data buffer empty interrupts.
1 (R/W): Enabled
0 (R/W): Disabled (default)

Setting SPTIE to 1 enables the output of SPI interrupt requests to the ITC due to a transmit data buffer empty. These interrupt requests are generated when the data written to the transmit data buffer is transferred to the shift register (when transmission starts).
SPI interrupts are not generated by transmit data buffer empty if SPTIE is set to 0.

D3 CPHA: Clock Phase Select Bit

Selects the SPI clock phase. (Default: 0)
Set the data transfer timing together with CPOL. (See Figure 17.7.1.)

D2 CPOL: Clock Polarity Select Bit

Selects the SPI clock polarity.
1 (R/W): Active Low
0 (R/W): Active High (default)

Set the data transfer timing together with CPHA. (See Figure 17.7.1.)

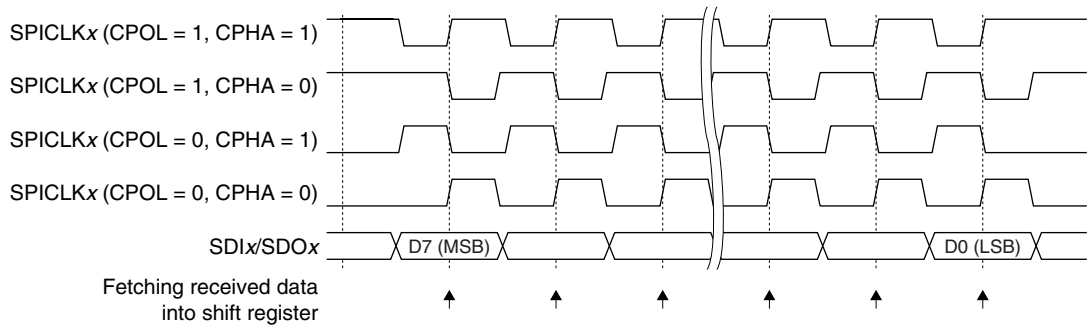


Figure 17.7.1 Clock and Data Transfer Timing

D1 MSSL: Master/Slave Mode Select Bit

Sets the SPI module to master or slave mode.
1 (R/W): Master mode
0 (R/W): Slave mode (default)

Setting MSSL to 1 selects master mode; setting it to 0 selects slave mode. Master mode performs data transfer with the internal clock. In slave mode, data is transferred by inputting the clock from the master device.

D0 SPEN: SPI Enable Bit

Enables or disables SPI module operation.
1 (R/W): Enabled
0 (R/W): Disabled (default)

Setting SPEN to 1 starts the SPI module operation, enabling data transfer.
Setting SPEN to 0 stops the SPI module operation.

Note: The SPEN bit should be set to 0 before setting the CPHA, CPOL, and MSSL bits.

18 I²C Master (I2CM)

18.1 I2CM Module Overview

The S1C17F57 includes an I²C master (I2CM) module that supports two-wire communications. The I2CM module operates as an I²C bus master device and can communicate with I²C-compliant slave devices.
The following shows the main features of I2CM:

- Operates as an I²C bus master device (as single master only).
- Supports standard (100 kbps) and fast (400 kbps) modes.
- Supports 8-bit data length only (MSB first).
- 7-bit addressing mode (10-bit addressing is possible by software control.)
- Includes one-byte receive data buffer and one-byte transmit data buffer.
- Can generate start, repeated start, and stop conditions.
- Supports half-duplex communications.
- Supports clock stretch function.
- Includes a noise filter function to help improve the reliability of data transfers.
- Can generate transmit buffer empty and receive buffer full interrupts.
- The input filter for the SDA and SCL inputs does not comply with the standard for removing noise spikes less than 50 ns.

Figure 18.1.1 shows the I2CM configuration.

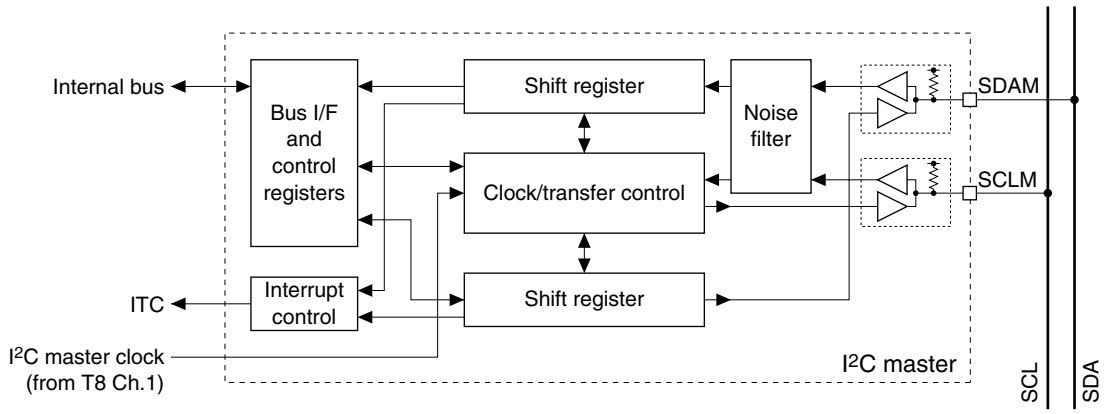


Figure 18.1.1 I2CM Module Configuration

18.2 I2CM Input/Output Pins

Table 18.2.1 lists the I2CM pins.

Table 18.2.1 List of I2CM Pins

Pin name	I/O	Qty	Function
SDAM	I/O	1	I2CM data input/output pin (see Note below) Inputs serial data from the I²C bus. Also outputs serial data to the I²C bus.
SCLM	I/O	1	I2CM clock input/output pin (see Note below) Inputs SCL line status. Also outputs a serial clock.

The I2CM input/output pins (SDAM, SCLM) are shared with I/O ports and are initially set as general purpose I/O port pins. The pin functions must be switched using the port function select bits to use the general purpose I/O port pins as I2CM input/output pins.
For detailed information on pin function switching, see the "I/O Ports (P)" chapter.

Note: The pins go to high impedance status when the port function is switched.
The SCLM and SDAM pins do not output a high level, so these lines should be pulled up to VDD with an external pull-up resistor. Be sure to avoid pulling these pins up to a voltage that exceeds the VDD level.

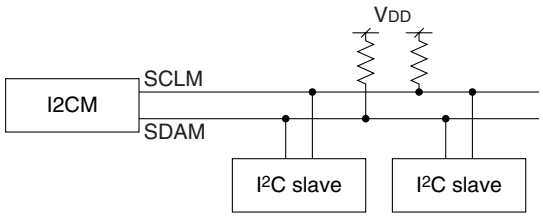


Figure 18.2.1 I²C Connection Example

18.3 Synchronization Clock

The I2CM module uses the internal clock (I2CM clock) output by the 8-bit timer (T8) Ch.1 as the synchronization clock. This clock is output from the SCLM pin to the slave device while also driving the shift register. The clock should be programmed to output a signal matching the transfer rate from T8 Ch.1. For more information on T8 control, see the "8-bit timers (T8)" chapter.

When the I2CM module is used to communicate with a slave device that performs clock stretching, the maximum transfer rate is limited to 50 kbps in standard mode or 200 kbps in fast mode.

The I2CM module does not function as a slave device. The SCLM input pin is used to check the I²C bus SCL signal status. It is not used for synchronization clock input.

18.4 Settings Before Data Transfer

The I2CM module includes an optional noise filter function that can be selected via the application program.

Noise filter function

The I2CM module includes a function for filtering noise from the SDAM and SCLM pin input signals. This function is enabled by setting NSERM/I2CM_CTL register to 1. Note that using this function requires setting the I2CM clock (T8 Ch.1 output clock) frequency to 1/6 or less of PCLK.

18.5 Data Transfer Control

Make the following settings before starting data transfers.
(1) Configure T8 Ch.1 to output the I2CM clock. (See the T8 module chapter.)
(2) Select the option function. (See Section 18.4.)
(3) Set the interrupt conditions to use I2CM interrupts. (See Section 18.6.)

Note: Make sure the I2CM module is halted (I2CMEN/I2CM_EN register = 0) before changing the above settings.

Enabling data transfers

Set I2CMEN/I2CM_EN register to 1 to enable I2CM operations. This enables I2CM transfers and clock input/output.

Note: Do not set I2CMEN to 0 when the I2CM module is transferring data.

Starting Data transfer

To start data transfers, the I²C master (this module) must generate a start condition. The slave address is then sent to establish communications.

(1) Generating start condition

The start condition applies when the SCL line is maintained at High and the SDA line is pulled down to Low.

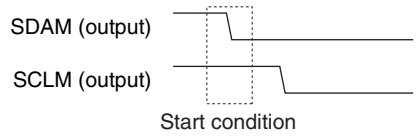


Figure 18.5.1 Start Condition

The start condition is generated by setting STRT/I2CM_CTL register to 1.
STRT is automatically reset to 0 once the start condition is generated. The I²C bus is busy from this point on.

(2) Slave address transmission

Once the start condition has been generated, the I²C master (this module) sends a bit indicating the slave address and transfer direction for communications. I²C slave addresses are either 7-bit or 10-bit. This module uses an 8-bit transfer data register to send the slave address and transfer direction bit, enabling single transfers in 7-bit address mode. In 10-bit mode, data is sent twice or three times under software control. Figure 18.5.2 shows the configuration of the address data.

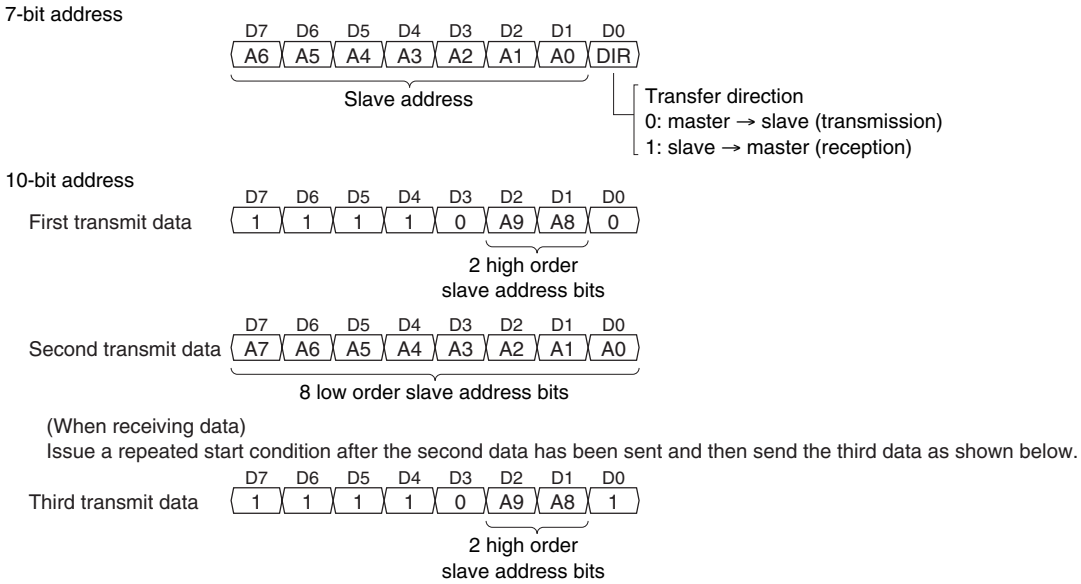


Figure 18.5.2 Transmit Data Specifying Slave Address and Transfer Direction

The transfer direction bit indicates the data transfer direction after the slave address has been sent. This is set to 0 when sending data from the master to the slave and to 1 when receiving data from the slave. To send a slave address, set the address with the transfer direction bit to RTDT[7:0]/I2CM_DAT register. At the same time, set TXE/I2CM_DAT register transmitting the address to 1.
After the slave address has been output, data can be sent and received as many times as required. Data must be sent or received according to the transfer direction set together with the slave address.

Data transmission control

The procedure for transmitting data is described below. Data transmission is performed by the same procedure as for slave address transmission.
To send byte data, set the transmit data to RTDT[7:0] and set TXE to 1 to transmit 1 byte.
When TXE is set to 1, the I2CM module begins data transmission in sync with the clock. If the previous data is currently being transmitted, data transmission starts after this has been completed. The I2CM module first transfers the data written to the shift register, then starts outputting the clock from the SCLM pin. TXE is reset to 0 at this point and a cause of interrupt occurs, enabling the subsequent transmission data and TXE to be set.

The data bits in the shift register are shifted in sequence at the clock falling edge and output via the SDAM pin with the MSB leading. The I2CM module outputs 9 clocks with each data transmission. In the 9th clock cycle, the I2CM module sets the SDA line into high impedance to receive an ACK or NAK sent from the slave device. The slave device returns ACK (0) to the master if the data is received. If the data is not received, the SDA line is not pulled down, which the I2CM module interprets to mean a NAK (1) (transmission failed).

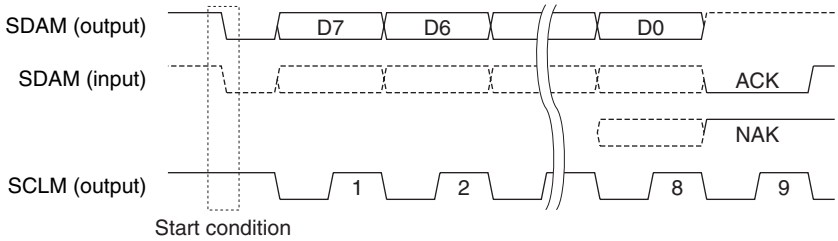


Figure 18.5.3 ACK and NAK

The I2CM module includes two status bits for transmission control: TBUSY/I2CM_CTL register and RTACK/I2CM_DAT register.
The TBUSY flag indicates the data transmission status. This flag becomes 1 when transmission starts (including slave address transmission) and reverts to 0 once data transmission ends. Inspect the flag to check whether the I2CM module is currently transmitting or at standby.
The RTACK bit indicates whether or not the slave device returned an ACK for the previous transmission. RTACK is 0 if an ACK was returned and 1 if ACK was not returned.

Data reception control

The procedure for receiving data is described below. When receiving data, the slave address must be sent with the transfer direction bit set to 1.
To receive data, set RXE/I2CM_DAT register to 1 for receiving 1 byte. When TXE/I2CM_DAT register is set to 1 for sending the slave address, RXE can also be set to 1 at the same time. If both TXE and RXE are set to 1, TXE takes priority.
When RXE is set to 1, allowing receiving to start, the I2CM module starts outputting the clock from the SCLM pin with the SDA line at high impedance. The data is shifted into the shift register from the MSB first in sync with the clock.
RXE is reset to 0 when D7 is loaded.
The received data is loaded to RTDT[7:0] once the 8-bit data has been received in the shift register.

The I2CM module includes two status bits for receive control: RBRDY/I2CM_DAT register and RBUSY/I2CM_CTL register.
The RBRDY flag indicates the received data status. This flag becomes 1 when the data received in the shift register is loaded to RTDT[7:0] and reverts to 0 when the received data is read out from RTDT[7:0]. Interrupts can also be generated once the flag value becomes 1.
The RBUSY flag indicates the receiving operation status. This flag is 1 when receiving starts and reverts to 0 when the data is received. Inspect the flag to determine whether the I2CM module is currently receiving or in standby.

The I2CM module outputs 9 clocks with each data reception. In the 9th clock cycle, an ACK or NAK is sent to the slave via the SDAM pin. The bit state sent can be set in RTACK/I2CM_DAT register. To send ACK, set RTACK to 0. To send NAK, set RTACK to 1.

End of data transfers (Generating stop condition)

To end data transfers after all data has been transferred, the I²C master (this module) must generate a stop condition. The stop condition applies when the SCL line is maintained at High and the SDA line is pulled up from Low to High.

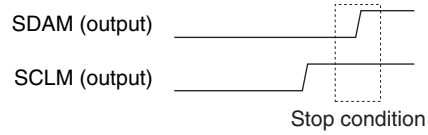


Figure 18.5.4 Stop Condition

The stop condition is generated by setting STP/I2CM_CTL register to 1.
When STP is set to 1, the I2CM module pulls up the I²C bus SDA line from Low to High with the SCL line maintained at High to generates a stop condition. The I²C bus subsequently switches to free state.
Before STP can be set to 1, confirm that TBUSY or RBUSY is reset to 0 from 1 (this indicates that the I2CM module has finished data transmit/receive operation) and then make the wait time longer than 1/4 of the I²C clock cycle set. The stop condition is generated as soon as data transfer (including ACK transfer) ends. STP is reset to 0 when the stop condition is generated.

Continuing data transfer (Generating Repeated start condition)

To make it possible to continue with a different data transfer after data transfer completion, the I²C master (this module) can generate a repeated start condition.

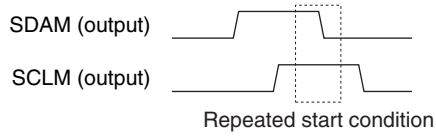


Figure 18.5.5 Repeated Start Condition

The repeated start condition is generated by setting STRT/I2CM_CTL register to 1 when the I²C bus is busy.
STRT is automatically reset to 0 once the repeated start condition is generated. Slave address transmission is subsequently possible with the I²C bus remaining in the busy state.

Disabling data transfer

After the stop condition has been generated, write 0 to I2CMEN to disable data transfers. To determine whether the stop condition has been generated, check to see if STP is automatically cleared to 0 after it is set to 1 by polling.
When I2CMEN is set to 0 while the I²C bus is in busy status, the SCLM and SDAM output levels and transfer data at that point cannot be guaranteed.

Timing chart

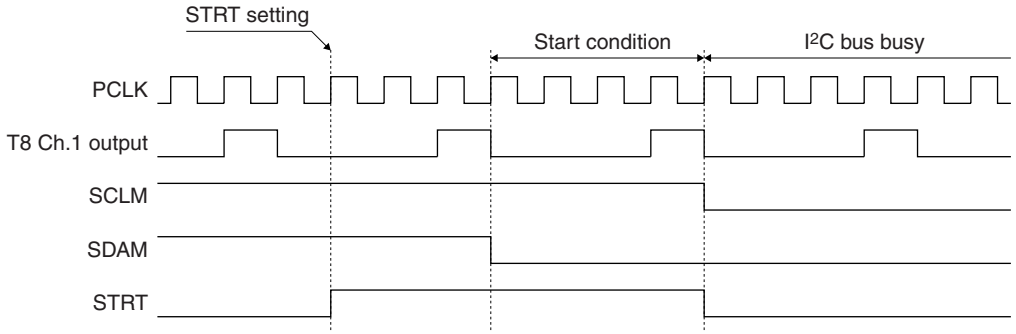


Figure 18.5.6 Start Condition Generation

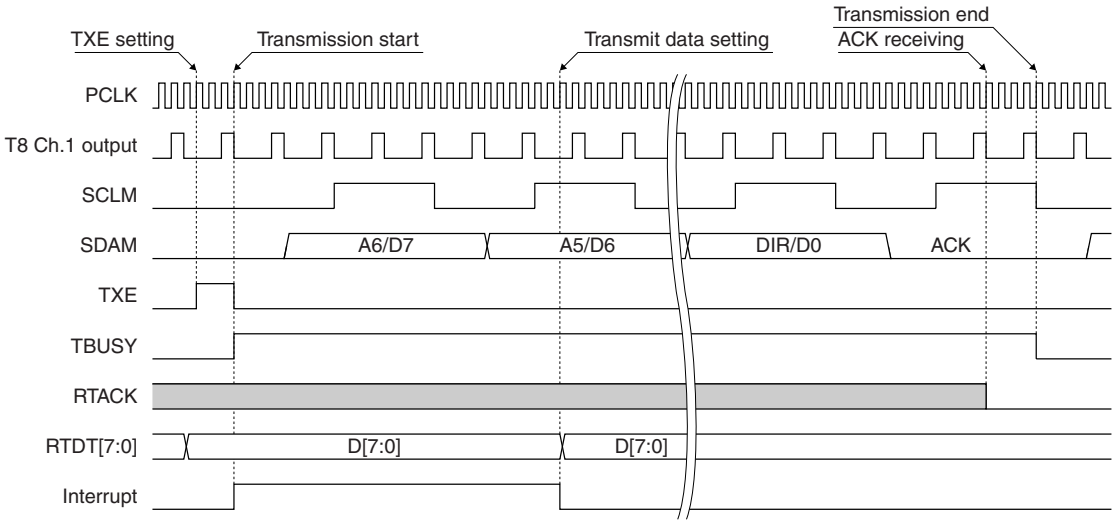


Figure 18.5.7 Slave Address Transmission/Data Transmission

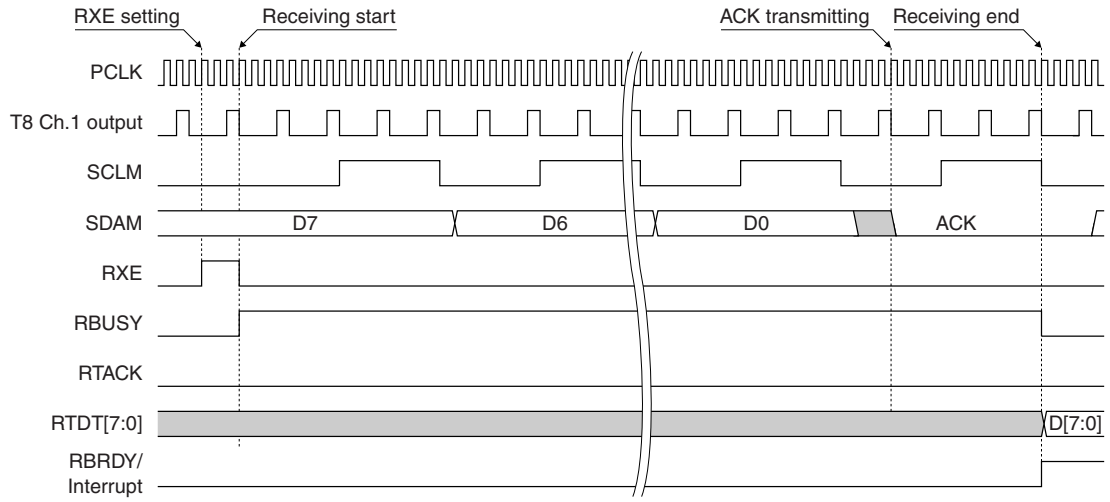


Figure 18.5.8 Data Receiving

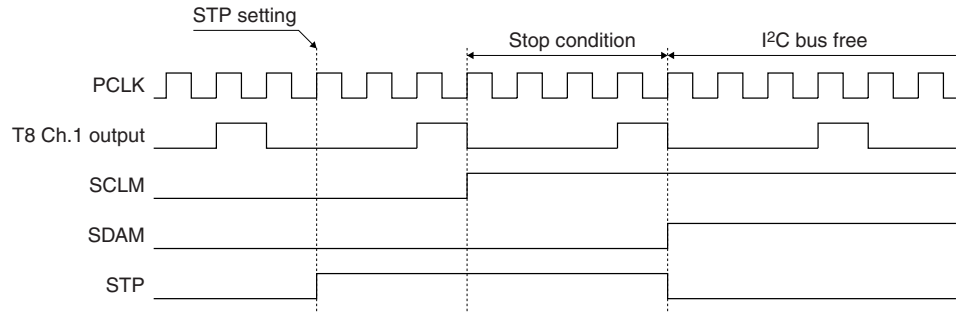


Figure 18.5.9 Stop Condition Generation

18.6 I2CM Interrupts

The I2CM module includes a function for generating the following two different types of interrupts.

- Transmit buffer empty interrupt
- Receive buffer full interrupt

The I2CM module outputs one interrupt signal shared by the two above interrupt causes to the interrupt controller (ITC).

Transmit buffer empty interrupt

To use this interrupt, set TINTE/I2CM_ICTL register to 1. If TINTE is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.
If transmit buffer empty interrupts are enabled (TINTE = 1), an interrupt request is output to the ITC as soon as the transmit data set in RTDT[7:0]/I2CM_DAT register is transferred to the shift register.
An interrupt occurs if other interrupt conditions are satisfied.

Checking whether a transmit buffer empty interrupt has occurred or not

A transmit buffer empty interrupt has occurred if TXE/I2CM_DAT register is read as 0 in the procedure shown below.

(1) Set TINTE/I2CM_ICTL register to 1.
(2) Write data to RTDT[7:0]/I2CM_DAT register.
(3) Set TXE/I2CM_DAT register to 1. (This can be performed simultaneously with Step 2 above.)
(4) An I2CM interrupt occurs.
(5) Read TXE/I2CM_DAT register.

Clearing the cause of transmit buffer empty interrupt

Write data to RTDT[7:0]/I2CM_DAT register.

Notes: • Data will not be sent if TXE/I2CM_DAT register is set to 0.

• If RTDT[7:0] contains data received from the I²C bus, it will be overwritten.

Receive buffer full interrupt

To use this interrupt, set RINTE/I2CM_ICTL register to 1. If RINTE is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.
If receive buffer full interrupts are enabled (RINTE = 1), an interrupt request is output to the ITC as soon as the data received in the shift register is loaded to RTDT[7:0].
An interrupt occurs if other interrupt conditions are met.

Checking whether a receive buffer full interrupt has occurred or not

A receive buffer full interrupt has occurred if RBRDY/I2CM_DAT register is read as 1 in the procedure shown below.

(1) Set RINTE/I2CM_ICTL register to 1.
(2) An I2CM interrupt occurs.
(3) Read RBRDY/I2CM_DAT register.

Clearing the cause of receive buffer full interrupt

Read data from RTDT[7:0]/I2CM_DAT register.

For more information on interrupt processing, see the “Interrupt Controller (ITC)” chapter.

18.7 Control Register Details

Table 18.7.1 List of I2CM Registers

Address	Register name		Function
0x4340	I2CM_EN	I²C Master Enable Register	Enables the I²C master module.
0x4342	I2CM_CTL	I²C Master Control Register	Controls the I²C master operation and indicates transfer status.
0x4344	I2CM_DAT	I²C Master Data Register	Transmit/receive data
0x4346	I2CM_ICTL	I²C Master Interrupt Control Register	Controls the I²C master interrupt.

The I2CM module registers are described in detail below.

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

I²C Master Enable Register (I2CM_EN)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
I²C Master Enable Register (I2CM_EN)	0x4340 (16 bits)	D15–1	–	reserved	–	–	–	0 when being read.
		D0	I2CMEN	I²C master enable	1 Enable 0 Disable	0	R/W	

D[15:1] Reserved

D0 I2CMEN: I²C Master Enable Bit

Enables or disables I2CM module operation.

1 (R/W): Enabled

0 (R/W): Disabled (default)

Setting I2CMEN to 1 starts the I2CM module operation, enabling data transfer. Setting I2CMEN to 0 stops the I2CM module operation.

I²C Master Control Register (I2CM_CTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
I²C Master Control Register (I2CM_CTL)	0x4342 (16 bits)	D15–10	–	reserved	–	–	–	0 when being read.
		D9	RBUSY	Receive busy flag	1 Busy 0 Idle	0	R	
		D8	TBUSY	Transmit busy flag	1 Busy 0 Idle	0	R	
		D7–5	–	reserved	–	–	–	0 when being read.
		D4	NSERM	Noise remove on/off	1 On 0 Off	0	R/W	
		D3–2	–	reserved	–	–	–	0 when being read.
		D1	STP	Stop control	1 Stop 0 Ignored	0	R/W	
		D0	STRT	Start control	1 Start 0 Ignored	0	R/W	

D[15:10] Reserved

D9 RBUSY: Receive Busy Flag Bit

Indicates the I2CM receiving status.

1 (R): Operating

0 (R): Standby (default)

RBUSY is set to 1 when the I2CM starts data receiving and is maintained at 1 while receiving is underway. It is cleared to 0 once reception is completed.

D8 TBUSY: Transmit Busy Flag Bit

Indicates the I2CM transmission status.

1 (R): Operating

0 (R): Standby (default)

TBUSY is set to 1 when the I2CM starts data transmission and is maintained at 1 while transmission is underway. It is cleared to 0 once transmission is completed.

D[7:5] Reserved

D4 NSERM: Noise Remove On/Off Bit

Turns the noise filter function on or off.

1 (R/W): On

0 (R/W): Off (default)

The I2CM module includes a function for filtering noise from the SDAM and SCLM pin input signals. This function is enabled by setting NSERM to 1. Note that using this function requires setting the I2CM clock (T8 Ch.1 output clock) frequency to 1/6 or less of PCLK.

D[3:2] Reserved

D1 STP: Stop Control Bit

Generates the stop condition.

1 (R/W): Stop condition generated

0 (R/W): Ineffective (default)

By setting STP to 1, the I2CM module generates the stop condition by pulling up the I²C bus SDA line from Low to High with the SCL line maintaining at High. The I²C bus subsequently becomes free. Note that the stop condition will be generated only if STP is 1 and TXE/I2CM_DAT register, RXE/I2CM_DAT register, and STRT are set to 0 when data transfer is completed (including ACK transfer). STP is automatically reset to 0 if the stop condition is generated.

D0 STRT: Start Control Bit

Generates the start condition.
1 (R/W): Start condition generated
0 (R/W): Ineffective (default)

By setting STRT to 1, the I2CM module generates the start condition by pulling down the I²C bus SDA line to Low with SCL line maintaining at High.
The repeated start condition can be generated by setting STRT to 1 when the I²C bus is busy.
STRT is automatically reset to 0 once the start condition or repeated start condition is generated. The I²C bus subsequently becomes busy.

I²C Master Data Register (I2CM_DAT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
I²C Master Data Register (I2CM_DAT)	0x4344 (16 bits)	D15–12	–	reserved	–	–	–	0 when being read.
		D11	**RBRDY**	Receive buffer ready flag	1 Ready 0 Empty	0	R	
		D10	**RXE**	Receive execution	1 Receive 0 Ignored	0	R/W	
		D9	**TXE**	Transmit execution	1 Transmit 0 Ignored	0	R/W	
		D8	**RTACK**	Receive/transmit ACK	1 Error 0 ACK	0	R/W	
		D7–0	**RTDT[7:0]**	Receive/transmit data RTDT7 = MSB RTDT0 = LSB	0x0 to 0xff	0x0	R/W	

D[15:12] Reserved

D11 RBRDY: Receive Buffer Ready Flag Bit

Indicates the receive buffer status.
1 (R): Receive data exists
0 (R): No receive data (default)

The RBRDY flag becomes 1 when the data received in the shift register is loaded to RTDT[7:0] and reverts to 0 when the receive data is read out from RTDT[7:0]. Interrupts can also be generated once the flag value becomes 1.

D10 RXE: Receive Execution Bit

Receives 1 byte of data.
1 (R/W): Data reception start
0 (R/W): Ineffective (default)

Setting RXE to 1 and TXE to 0 starts receiving for 1 byte of data. RXE can be set to 1 for subsequent reception, even if the slave address is being sent or data is being received. RXE is reset to 0 as soon as D7 is loaded to the shift register.

D9 TXE: Transmit Execution Bit

Transmits 1 byte of data.
1 (R/W): Data transmission start
0 (R/W): Ineffective (default)

Transmission is started by setting the transmit data to RTDT[7:0] and writing 1 to TXE. TXE can be set to 1 for subsequent transmission, even if the slave address or data is being sent. TXE is reset to 0 as soon as the data set in RTDT[7:0] is transferred to the shift register.

D8 RTACK: Receive/Transmit ACK Bit

When transmitting data

Indicates the response bit status.
1 (R/W): Error (NAK)
0 (R/W): ACK (default)

RTACK becomes 0 when ACK is returned from the slave after 1 byte of data is sent, indicating that the slave has received the data correctly. If RTACK is 1, the slave device is not operating or the data was not received correctly.

When receiving data

Sets the response bit sent to the slave.
1 (R/W): Error (NAK)
0 (R/W): ACK (default)

To return an ACK after data has been received, RTACK should be set to 0 before the I2CM module sends the response bit. To return a NAK, set RTACK to 1.

D[7:0] RTDT[7:0]: Receive/Transmit Data Bits

When transmitting data

Sets the transmit data. (Default: 0x0)
Data transmission is started by setting TXE to 1. If a slave address or data is currently being transmitted, transmission begins once the previous transmission is completed. Serial converted data is output from the SDAM pin with MSB leading and bits set to 0 as Low level. A cause of transmit buffer empty interrupt is generated as soon as the data written to this register is transferred to the shift register, after which the subsequent transmission data can be written.

When receiving data

The received data can be read out. (Default: 0x0)
Data reception is started by setting RXE to 1. If a slave address is currently being transmitted or data is currently being received, the new reception starts once the previous data has been transferred. The RBRDY flag is set and a cause of receive buffer full interrupt generated as soon as reception is completed and the shift register data is transferred to this register. Data can then be read until the subsequent data has been received. If the subsequent data is received before this register is read out, the contents are overwritten by the most recent received data. Serial data input from the SDAM pin with MSB leading is converted to parallel, with the High level bit set to 1 and the Low level bit set to 0, then loaded to this register.

I²C Master Interrupt Control Register (I2CM_ICTL)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
I²C Master Interrupt Control Register (I2CM_ICTL)	0x4346 (16 bits)	D15–2	–	reserved	–				–	–	0 when being read.
		D1	RINTE	Receive interrupt enable	1	Enable	0	Disable	0	R/W	
		D0	TINTE	Transmit interrupt enable	1	Enable	0	Disable	0	R/W	

D[15:2] Reserved

D1 RINTE: Receive Interrupt Enable Bit

Enables or disables I2CM receive buffer full interrupts.
1 (R/W): Enabled
0 (R/W): Disabled (default)

Setting RINTE to 1 enables the output of I2CM interrupt requests to the ITC due to a receive data buffer full. These interrupt requests are generated when the data received in the shift register is transferred to RTDT[7:0]/I2CM_DAT register (when reception is completed).
I2CM interrupts are not generated by receive data buffer full if RINTE is set to 0.

D0 TINTE: Transmit Interrupt Enable Bit

Enables or disables I2CM transmit buffer empty interrupts.
1 (R/W): Enabled
0 (R/W): Disabled (default)

Setting TINTE to 1 enables the output of I2CM interrupt requests to the ITC due to a transmit buffer empty. These interrupt requests are generated when the data written to RTDT[7:0] is transferred to the shift register.
I2CM interrupts are not generated by transmit buffer empty if TINTE is set to 0.

19 I²C Slave (I2CS)

19.1 I2CS Module Overview

The S1C17F57 includes an I²C slave (I2CS) module that supports two-wire communications. The I2CS module operates as an I²C bus slave device and can communicate with an I²C-compliant master device.
The following shows the main features of I2CS:

- Operates as an I²C bus slave device.
- Supports standard (100 kbps) and fast (400 kbps) modes.
- Supports 8-bit data length only (MSB first).
- Supports 7-bit addressing mode.
- Includes one-byte receive data buffer and one-byte transmit data buffer.
- Can detect start and stop conditions.
- Supports half-duplex communications.
- Supports clock stretch function.
- Supports forced bus release function.
- Includes a noise filter function to help improve the reliability of data transfers.
- Can generate transmit buffer empty, receive buffer full, and bus status interrupts.
- The input filter for the SDA and SCL inputs does not comply with the standard for removing noise spikes less than 50 ns.

Figure 19.1.1 shows the I2CS configuration.

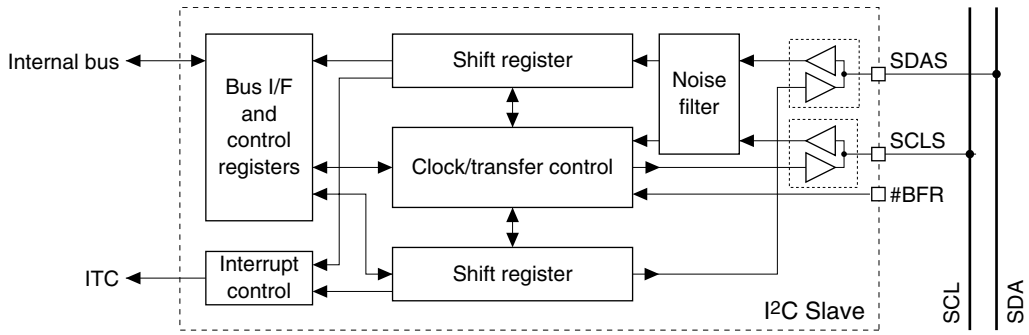


Figure 19.1.1 I2CS Module Configuration

Note: The I2CS module does not support general call address and 10-bit address mode.

19.2 I2CS Input/Output Pins

Table 19.2.1 lists the I2CS pins.

Table 19.2.1 List of I2CS Pins

Pin name	I/O	Qty	Function
SDAS	I/O	1	I2CS data input/output pin (see Note below) Inputs serial data from the I²C bus. Also outputs serial data to the I²C bus.
SCLS	I/O	1	I2CS clock input/output pin (see Note below) Inputs SCL line status from the I²C bus. Also outputs a low level to put the I²C bus into clock stretch status.
#BFR	I	1	I²C bus free request input pin A Low pulse input to this pin requests the I2CS to release the I²C bus. When the bus free request input has been enabled with software, a Low pulse initializes the communication process of the I2CS module and sets the SDAS and SCLS pins into high impedance.

The I2CS input/output pins (SDAS, SCLS, #BFR) are shared with I/O ports and are initially set as general purpose I/O port pins. The pin functions must be switched using the port function select bits to use the general purpose I/O port pins as I2CS input/output pins. For detailed information on pin function switching, see the "I/O Ports (P)" chapter.

Note: The pins go to high impedance status when the port function is switched.
The SCLS and SDAS pins do not output a high level, so these lines should be pulled up to VDD with an external pull-up resistor. Be sure to avoid pulling these pins up to a voltage that exceeds the VDD level.

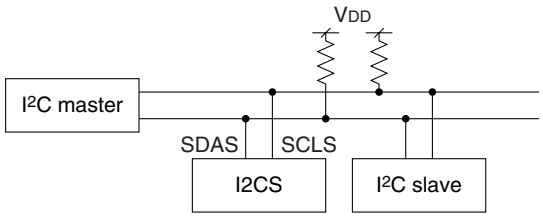


Figure 19.2.1 I2C Connection Example

19.3 Operation Clock

The I2CS module operates with the clock output from the external I2C master device by inputting it from the SCLS pin.
The I2CS module also uses the peripheral module clock (PCLK) for its operations. The PCLK frequency must be set eight-times or higher than the SCLS input clock frequency during data transfer. In standby status, use of the asynchronous address detection function allows the application to lower the PCLK clock frequency to reduce current consumption. For more information, see "Asynchronous address detection function" in Section 19.4.3.

19.4 Initializing I2CS

19.4.1 Reset

The I2CS module must be reset to initialize the communication process and to set the I2C bus into free status (high impedance). The following shows two methods for resetting the module:

(1) Software reset

The I2CS module can be reset using SOFTRESET/I2CS_CTL register.
To reset the I2CS module, write 1 to SOFTRESET to place the I2CS module into reset status, then write 0 to SOFTRESET to release it from reset status. It is not necessary to insert a waiting time between writing 1 and 0.
The I2CS module initializes the I2C communication process and put the SDAS and SCLS pins into high-impedance to be ready to detect a start condition. Furthermore, the I2CS control bits except for SOFTRESET are initialized. Perform the software reset in the initial setting process before staring communication.

(2) Bus free request with an input from the #BFR pin

The I2CS module can accept bus free requests via the #BFR pin. The bus free request support is disabled by default. To enable this function, set BFREQ_EN/I2CS_CTL register to 1.
When this function is enabled, a low pulse (One peripheral module clock (PCLK) cycles or more pulse width is required. Two PCLK clock cycles or more pulse width is recommended.) input to the #BFR pin sets BFREQ/I2CS_STAT register to 1. This initializes the I2C communication process and puts the SDAS and SCLS pins into high-impedance. The control registers will not be initialized as distinct from the software reset described above.

Note: When BFREQ is set to 1 (an interrupt can be used for checking this status), perform a software reset and set the registers again.

19.4.2 Setting Slave Address

I2C devices have a unique slave address to identify each device.
The I2CS module supports 7-bit address (does not support 10-bit address), and the address of this module must be set to SADRS[6:0]/I2CS_SADRS register.

19.4.3 Optional Functions

The I2CS module has a clock stretch, asynchronous address detection, and noise filter optional functions selectable in the application program.

Clock stretch function

After data and ACK are transmitted or received, the slave device may issue a wait request to the master device until it is ready to transmit/receive by pulling the I2C bus SCL line down to low. The I2CS module supports this clock stretch function.
The master device enters a standby state until the wait request is canceled (the SCL line goes high). The clock stretch function in this module is disabled by default. When using the clock stretch function, set CLKSTR_EN/I2CS_CTL register to 1 before starting data communication. Note that the data setup time (after the SDAS pin outputs the MSB of SDATA[7:0]/I2CS_TRNS register until I2CS turns the SCLS pin pull-down resistor off) while the I2CS module is operating with the clock stretch function enabled varies depending on the I2CS module operating clock (PCLK) frequency.

Asynchronous address detection function

The I2CS module operation clock (PCLK) frequency must be set eight-times or higher than the transfer rate during data transfer. However, the PCLK frequency can be lowered to reduce current consumption if no other processing is required during standby for data transfer. The asynchronous address detection function is provided to detect the I2C slave address sent from the master in this status.
The asynchronous address detection function in this module is disabled by default. When using the asynchronous address detection function, set ASDET_EN/I2CS_CTL register to 1.
If the slave address sent from the master has matched with one that has been set in this I2CS module when the asynchronous address detection function has been enabled, the I2CS module generates a bus status interrupt and returns NAK to the I2C master to request for resending the slave address.
Set the PCLK frequency to eight-times or higher than the transfer rate and reset ASDET_EN to 0 in the interrupt handler routine. Data transfer will be able to resume normally after the master retries transmission. After the master generates a stop condition to put the I2C bus into free status, the asynchronous address detection function can be enabled again to lower the operating speed.

Notes: • When the asynchronous address detection function is enabled, the I2C bus signals are input without passing through the noise filter. Therefore, the slave address may not be detected in a high-noise environment.

• When the asynchronous address detection function is enabled, data transfer cannot be performed even if the PCLK frequency is eight-times or higher than the transfer rate. Be sure to disable the asynchronous address detection function during normal operation.

Noise filter

The I2CS module includes a function to remove noise from the SDAS and SCLS input signals. This function is enabled by setting NF_EN/I2CS_CTL register to 1.

19.5 Data Transfer Control

Make the following settings before starting data transfers.
(1) Initialize the I2CS module. See Section 19.4.
(2) Set the interrupt conditions to use I2CS interrupt. See Section 19.6.

Note: Make sure that the I2CS module is disabled (I2CSEN/I2CS_CTL register = 0) before setting the conditions above.

Enabling data transfers

First, set I2CSEN/I2CS_CTL register to 1 to enable I2CS operation. This makes the I2CS in ready-to-transmit/receive status in which a start condition can be detected.

Note: Do not set the I2CSEN bit to 0 while the I2CS module is transmitting/receiving data.

Starting data transfer

To start data transmission/reception, set COM_MODE/I2CS_CTL register to 1 to enable data communications. When the slave address for this module that has been sent from the master is received after a start condition is detected, the I2CS module returns an ACK (SDAS = low) and starts operating for data reception or data transmission according to the transfer direction bit that has been received with the slave address.

When COM_MODE is 0 (default), the I2CS module does not send back a response if the master has sent the slave address of this module (it is regarded as that the I2CS module has returned a NAK to the master).

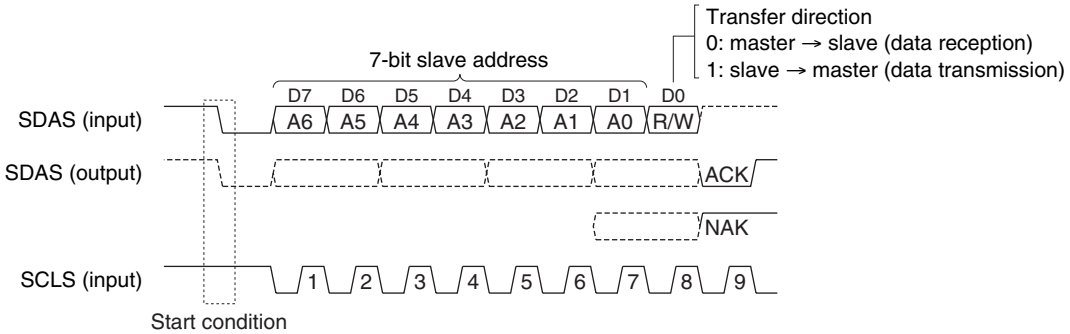


Figure 19.5.1 Receiving Slave Address and Data Direction Bit

When a start condition is detected, BUSY/I2CS_ASTAT register is set to 1 to indicate that the I²C bus is put into busy status. When the slave address of this module is received, SELECTED/I2CS_ASTAT register is set to 1 to indicate that this module has been selected as the I²C slave device. BUSY is maintained at 1 until a stop condition is detected. SELECTED is maintained at 1 until a stop condition or repeated start condition is detected.
The value of the transfer direction bit is set to R/W/I2CS_ASTAT register, so use R/W to select the transmit- or receive-handling.

If the slave address of this module is detected when the asynchronous address detection function has been enabled, ASDET/I2CS_STAT register is set to 1. The I2CS module generates a bus status interrupt and returns NAK to the I²C master to request for resending the slave address. Set the PCLK frequency to eight-times or higher than the transfer rate and disable the asynchronous address detection function in the interrupt handler routine. Data transfer will be able to resume normally after the master retries transmission. ASDET can be cleared by writing 1.

Data transmission

The following describes a data transmission procedure.
The I2CS module starts data transmission process when both SELECTED and R/W are set to 1. It sets TXEMP/I2CS_ASTAT register to 1 to issue a request to the application program to write transmit data. Write transmit data to SDATA[7:0]/I2CS_TRNS register.

When setting the first transmit data after this module has been selected as the slave device, follow the precautions described below.

When the clock stretch function is disabled (default)

Transmit data must be written to SDATA[7:0] within 1 cycle of the I²C clock (SCLS input clock) after TXEMP has been set to 1. This time is not enough for data preparation, so write transmit data before TXEMP has been set to 1. If the previous transmit data is still stored in SDATA[7:0], it is overwritten with the new data to be transferred. Therefore, the clear operation (see below) using TBUF_CLR is unnecessary.

When the asynchronous address detection function is used, the data written before ASDET_EN is reset to 0 becomes invalid. Therefore, transmission data must be written after TXEMP has been set to 1.

When the clock stretch function is enabled

The master device is placed into wait status by the clock stretch function, so transmit data can be written after TXEMP is set. However, if the previous transmit data is still stored in SDATA[7:0], it will be sent immediately after TXEMP has been set. In order to avoid this problem, clear the I2CS_TRNS register using TBUF_CLR/I2CS_CTL register before this module is selected as the slave device. The I2CS_TRNS register is cleared by writing 1 to TBUF_CLR then writing 0 to it.

It is not necessary to clear the I2CS_TRNS register if the first transmit data is written before TXEMP has been set.

When the asynchronous address detection function is used, the data written before ASDET_EN is reset to 0 becomes invalid. Therefore, transmission data must be written after TXEMP has been set to 1.

For writing transmit data other than the first time, use an interrupt that can be generated when TXEMP is set to 1. TXEMP is also set to 1 when the transmit data written to SDATA[7:0] is loaded to the sift register during transmission. TXEMP is cleared by writing transmit data to SDATA[7:0].

When the clock stretch function is disabled (default)

When the clock stretch function has been disabled, data must be written to the I2CS_TRNS register within 7 cycles of the I²C clock (SCLS input clock) from TXEMP being set to 1.

If data has not been written in this period, the current register value (previous transmit data) will be sent. In this case, TXUDF/I2CS_STAT register is set to 1 to indicate that invalid data has been sent. An interrupt can be generated when TXUDF is set to 1, so an error handling should be performed in the interrupt handler routine. TXUDF is cleared by writing 1.

When the clock stretch function is enabled

When the clock stretch function has been enabled, the I2CS module pulls down the SCLS pin to low to generate a clock stretch (wait) status until transmit data is written to the I2CS_TRNS register.

Transmit data bits are output from the SDAS pin in sync with the SCLS input clock sent from the master. The MSB is output first. After the eight bits has been output, the master sends back an ACK or NAK in the ninth clock cycle.

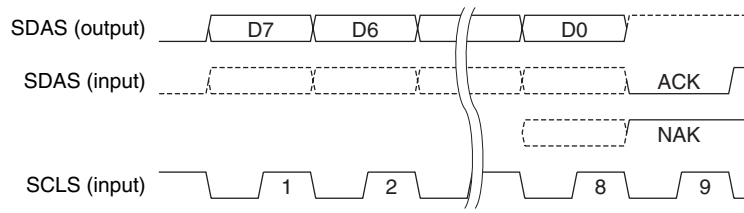


Figure 19.5.2 ACK and NAK

The ACK bit indicates that the master could receive data. It is also a transmit request bit, therefore, the next transmit data must be written in advance. Receiving an ACK generates a clock stretch status when the clock stretch function has been enabled, so data can be written after an ACK is received.

A NAK will be returned from the master if the master could not receive data or when the master terminates data reception. In this case a clock stretch status is not generated even if the clock stretch function has been enabled. Read DA_NAK/I2CS_STAT register to check if an ACK is returned or if a NAK is returned. DA_NAK is set to 0 when an ACK is returned or set to 1 when a NAK is returned. An interrupt can be generated when DA_NAK is set to 1, so an error or termination handling can be performed in the interrupt handler routine. DA_NAK is cleared by writing 1.

The SDA line status during data transmission is input in the module and is compare with the output data. The comparison results are set to DMS/I2CS_STAT register. DMS is set to 0 when data is output correctly. If the SDA line status is different from the output data, DMS is set to 1. This may be caused by a low pull-up resistor value or another device that is controlling the SDA line. An interrupt can be generated when DMS is set to 1, so an error handling can be performed in the interrupt handler routine. DMS is cleared by writing 1.

Note: If the I2CS module has sent back a NAK as the response to the address sent by the master when the conditions shown below are all met, the master must wait for 33 µs or more before it can send another slave address (except when the master sends the I2CS slave address again).
1. The transfer rate is set to 320 kbps or higher.
2. The asynchronous address detection function is enabled.
3. The I2CS module is placed into transfer standby state and OSC1 is used as the operating clock (PCLK).

Data reception

The following describes a data receive procedure.

The I2CS module starts data receiving process when SELECTED is set to 1 and R/W is set to 0. The received data bits are input from the SDAS pin in sync with the SCLS input clock sent from the master. When the eight-bit data (MSB first) is received in the shift register, the received data is loaded to RDATA[7:0]/I2CS_RECV register.

When the received data is loaded to RDATA[7:0], RXRDY/I2CS_ASTAT register is set to 1 to issue a request to the application program to read RDATA[7:0]. An interrupt can be generated when RXRDY is set to 1, so the received data should be read in the interrupt handler routine. RXRDY is cleared by reading the received data.

When the clock stretch function is disabled (default)

When the clock stretch function has been disabled, data must be read from the I2CS_RECV register within 7 cycles of the I²C clock (SCLS input clock) from RXRDY being set to 1.

When the clock stretch function is enabled

When the clock stretch function has been enabled, the I2CS module pulls down the SCLS pin to low to generate a clock stretch (wait) status until the received data is read from the I2CS_RECV register.

If the next data has been received without reading the received data, RDATA[7:0] will be overwritten. In this case, RXOVF/I2CS_STAT register is set to 1 to indicate that the received data has been overwritten. An interrupt can be generated when RXOVF is set to 1, so an error handling should be performed in the interrupt handler routine. RXOVF is cleared by writing 1.

To return NAK during data reception

During data reception (master transmission), the I2CS module sends back an ACK (SDAS = low) every time an 8-bit data has been received (by default setting). The response code can be changed to NAK (SDAS = Hi-Z) by setting NAK_ANS/I2CS_CTL register. An ACK will be sent when NAK_ANS is 0 or a NAK will be sent when NAK_ANS is set to 1.

NAK_ANS should be set within 7 cycles of the I²C clock (SCLS input clock) after RXRDY has been set to 1 by receiving data just prior to one required for returning NAK.

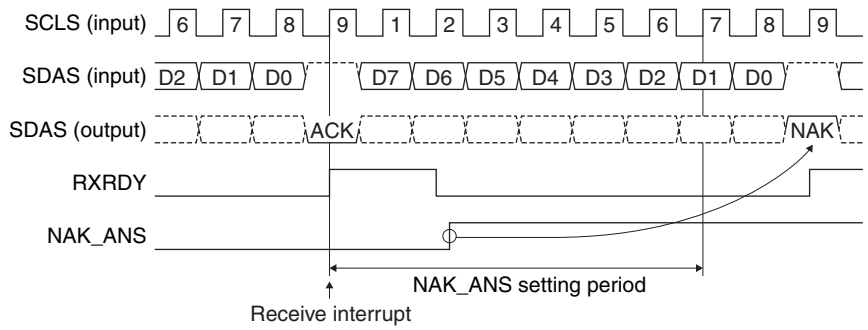


Figure 19.5.3 NAK_ANS Setting and NAK Response Timing

End of data transfer (detecting stop condition)

Data transfers will be terminated when the master generates a stop condition. The stop condition is a state in which the SDA line is pulled up from Low to High with the SCL line maintained at High.

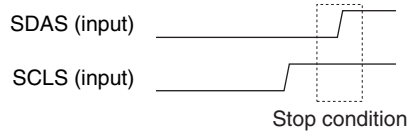


Figure 19.5.4 Stop Condition

If a stop condition is detected while the I2CS module is selected as the slave device (SELECTED = 1), the I2CS module sets DA_STOP/I2CS_STAT register to 1. At the same time, it sets the SDAS and SCLS pins into high-impedance and initializes the I2C communication process to enter standby state that is ready to detect the next start condition. Also SELECTED and BUSY are reset to 0.

An interrupt can be generated when DA_STOP is set to 1, so a communication terminating process should be performed in the interrupt handler routine. DA_STOP is cleared by writing 1.

Disabling data transfer

After data transfer has finished, write 0 to the COM_MODE/I2CS_CTL register to disable data transfer.

Always make sure that BUSY and SELECTED are 0 before disabling data transfer.

To deactivate the I2CS module, set I2CSEN/I2CS_CTL register to 0.

Timing charts

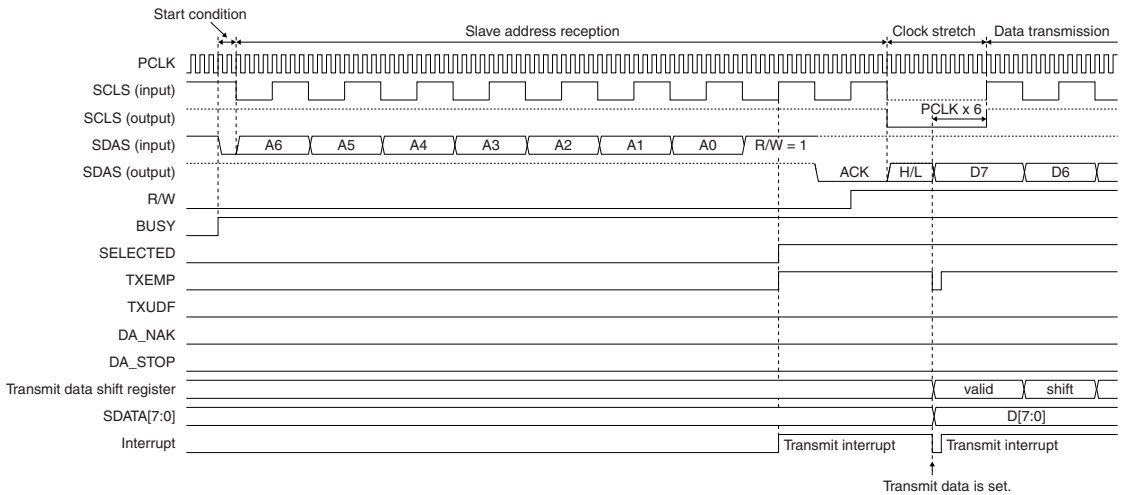


Figure 19.5.5 I2CS Timing Chart 1 (start condition → data transmission)

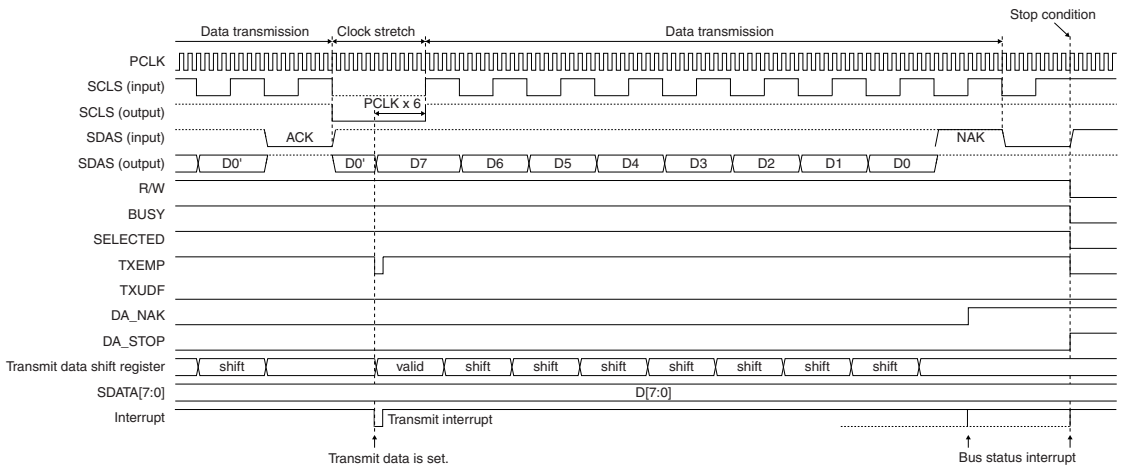


Figure 19.5.6 I2CS Timing Chart 2 (data transmission → stop condition)

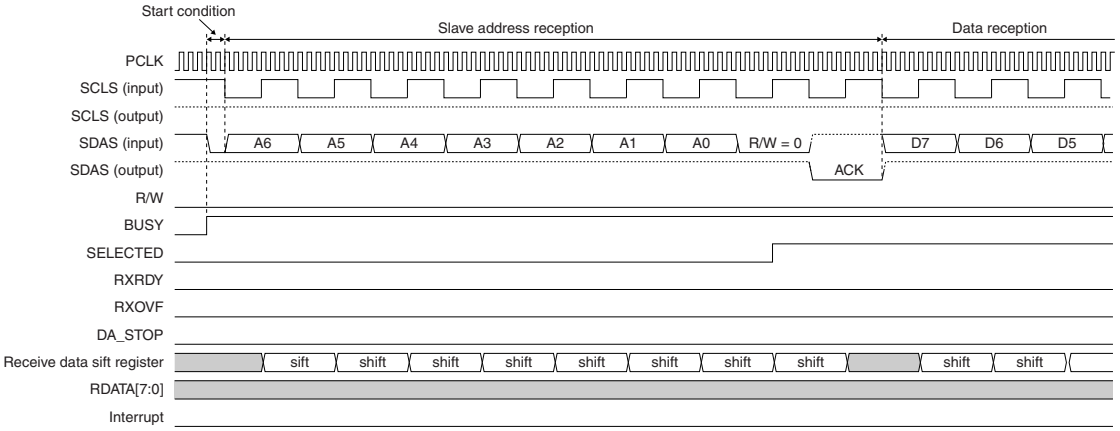


Figure 19.5.7 I2CS Timing Chart 3 (start condition → data reception)

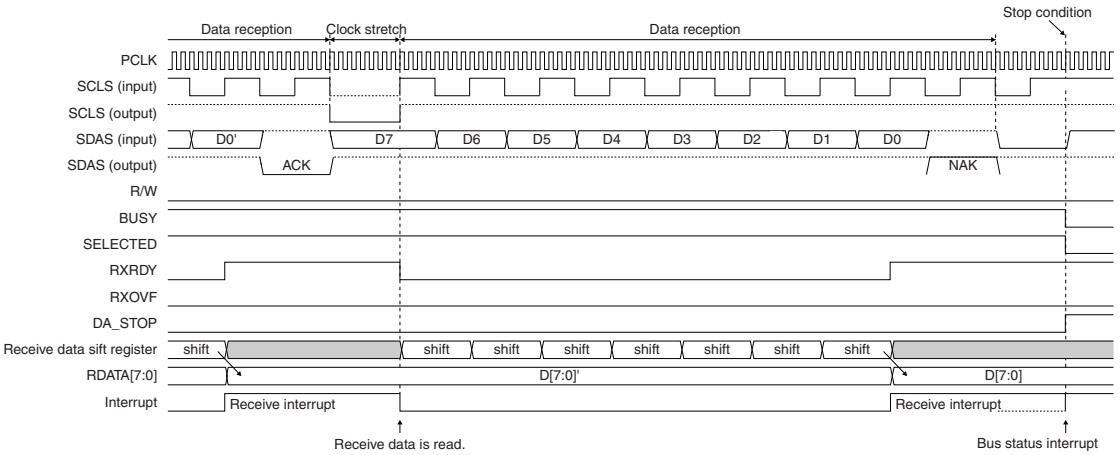


Figure 19.5.8 I2CS Timing Chart 4 (data reception → stop condition)

19.6 I2CS Interrupts

The I2CS module includes a function for generating the following three different types of interrupts.

- Transmit interrupt
- Receive interrupt
- Bus status interrupt

The I2CS module outputs one interrupt signal shared by the three above interrupt causes to the interrupt controller (ITC).

Transmit interrupt

When a read request (R/W bit = 1) issued by the I2C master is received and if transmit data has not be written to SDATA[7:0]/I2CS_TRANS register, an interrupt signal is output to the ITC. This interrupt can be used to write transmit data to SDATA[7:0]. Writing transmit data to SDATA[7:0] clears the interrupt signal. After that, an interrupt signal is also output to the ITC when transmit data written to SDATA[7:0] is sent to the transmit shift register (TXEMP/I2CS_ASTAT register = 1) during transmission. Set TXEMP_IEN/I2CS_ICTL register to 1 when using this interrupt. If TXEMP_IEN is set to 0 (default), interrupt requests by this cause will not be sent to the ITC.

This interrupt will not occur after a stop condition that terminates transmission is generated even if SDATA[7:0] is empty, as TXEMP will not be set to 1.

Receive interrupt

When the received data is loaded to RDATA[7:0]/I2CS_RECV register, RXRDY/I2CS_ASTAT register is set to 1 and an interrupt signal is output to the ITC. This interrupt can be used to read the received data from RDATA[7:0]. Set RXRDY_IEN/I2CS_ICTL register to 1 when using this interrupt. If RXRDY_IEN is set to 0 (default), interrupt requests by this cause will not be sent to the ITC.

Bus status interrupt

The I2CS module provides the status bits listed below to represent the transmit/receive and I²C bus statuses (see Section 19.5 for details of each function).

1. ASDET/I2CS_STAT register: This bit is set to 1 when the slave address is detected by the asynchronous address detection function.
2. TXUDF/I2CS_STAT register: This bit is set to 1 when a transmit operation has started before transmit data is written. (When the clock stretch function is disabled)
3. DA_NAK/I2CS_STAT register: This bit is set to 1 when a NAK is returned from the master during transmission.
4. DMS/I2CS_STAT register: This bit is set to 1 when the SDA line status is different from transfer data.
   DMS will also be set to 1 when another slave device issues ACK to this I²C slave address (when ASDET_EN/I2CS_CTL register = 0).

   Note: When the master device of the I²C bus, which has multiple slave devices connected including this IC, starts communication with another slave device, the I2CS module issues NAK in response to the sent slave address. On the other hand, the selected slave device issues ACK. Therefore, DMS may be set due to a difference between the output value of this IC and the SDA line status. When SELECTED/I2CS_ASTAT register is set to 0, you can ignore DMS without a problem even if it is set to 1 as there is a difference in the response code (ACK/NAK) from the selected slave device.
   When the I2CS module is placed into asynchronous address detection mode (ASDET_EN = 1), a DMS does not occur as in the condition above.
5. RXOVF/I2CS_STAT register: This bit is set to 1 when the next data has been received before the received data is read (the received data is overwritten). (When the clock stretch function is disabled)
6. BFREQ/I2CS_STAT register: This bit is set to 1 when a bus free request is accepted.
7. DA_STOP/I2CS_STAT register: This bit is set to 1 if a stop condition or a repeated start condition is detected while this module is selected as the slave device.

When one of the bits listed above is set to 1, BSTAT/I2CS_STAT register is set to 1 and an interrupt signal is output to the ITC. An interrupt occurs if other interrupt conditions are satisfied. This interrupt can be used to perform an error or terminate handling.
Set BSTAT_IEN/I2CS_ICTL register to 1 when using this interrupt. If BSTAT_IEN is set to 0 (default), interrupt requests by this cause will not be sent to the ITC.

For more information on interrupt processing, see the "Interrupt Controller (ITC)" chapter.

19.7 Control Register Details

Table 19.7.1 List of I2CS Registers

Address	Register name		Function
0x4360	I2CS_TRNS	I²C Slave Transmit Data Register	I²C slave transmit data
0x4362	I2CS_RECV	I²C Slave Receive Data Register	I²C slave receive data
0x4364	I2CS_SADRS	I²C Slave Address Setup Register	Sets the I²C slave address.
0x4366	I2CS_CTL	I²C Slave Control Register	Controls the I²C slave module.
0x4368	I2CS_STAT	I²C Slave Status Register	Indicates the I²C bus status.
0x436a	I2CS_ASTAT	I²C Slave Access Status Register	Indicates the I²C slave access status.
0x436c	I2CS_ICTL	I²C Slave Interrupt Control Register	Controls the I²C slave interrupt.

The I2CS module registers are described in detail below.

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

I2C Slave Transmit Data Register (I2CS_TRNS)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
I2C Slave Transmit Data Register (I2CS_TRNS)	0x4360 (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	SDATA[7:0]	I2C slave transmit data	0–0xff	0x0	R/W	

D[15:8] Reserved

D[7:0] SDATA[7:0]: I2C Slave Transmit Data Bits

Sets a transmit data in this register. (Default: 0x0)

The serial-converted data is output from the SDAS pin beginning with the MSB, in which the bits set to 0 are output as Low-level signals. When the data set in this register is sent to the shift register, a transmit interrupt occurs. The next transmit data can be written to the register after that.

If the clock stretch function has been disabled, data must be written to this register within 7 cycles of the I2C clock (SCLS input clock) after a transmit interrupt has been occurred.

However, when setting the first transmit data after this module has been selected as the slave device, follow the precautions described below.

When the clock stretch function is disabled (default)

Transmit data must be written to SDATA[7:0] within 1 cycle of the I2C clock (SCLS input clock) after TXEMP has been set to 1. This time is not enough for data preparation, so write transmit data before TXEMP has been set to 1. If the previous transmit data is still stored in SDATA[7:0], it is overwritten with the new data to be transferred. Therefore, the clear operation (see below) using TBUF_CLR is unnecessary.

When the asynchronous address detection function is used, the data written before ASDET_EN is reset to 0 becomes invalid. Therefore, transmission data must be written after TXEMP has been set to 1.

When the clock stretch function is enabled

The master device is placed into wait status by the clock stretch function, so transmit data can be written after TXEMP is set. However, if the previous transmit data is still stored in SDATA[7:0], it will be sent immediately after TXEMP has been set. In order to avoid this problem, clear the I2CS_TRNS register using TBUF_CLR/I2CS_CTL register before this module is selected as the slave device. The I2CS_TRNS register is cleared by writing 1 to TBUF_CLR then writing 0 to it.

It is not necessary to clear the I2CS_TRNS register if the first transmit data is written before TXEMP has been set.

When the asynchronous address detection function is used, the data written before ASDET_EN is reset to 0 becomes invalid. Therefore, transmission data must be written after TXEMP has been set to 1.

I2C Slave Receive Data Register (I2CS_RECV)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
I2C Slave Receive Data Register (I2CS_RECV)	0x4362 (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	RDATA[7:0]	I2C slave receive data	0–0xff	0x0	R	

D[15:8] Reserved

D[7:0] RDATA[7:0]: I2C Slave Receive Data Bits

The received data can be read from this register. (Default: 0x0)

The serial data input from the SDAS pin beginning with the MSB is converted into parallel data, with the high-level signals changed to 1 and the low-level signals changed to 0. The resulting data is stored in this register.

When a receive operation is completed and the data received in the shift register is loaded to this register, RXRDY/I2CS_ASTAT register is set and a receive interrupt occurs. Thereafter, the data can be read out.

When the clock stretch function has been disabled, data must be read from this register within 7 cycles of the I2C clock (SCLS input clock) after RXRDY is set to 1. If the next data has been received without reading the received data, this register will be overwritten with the newly received data.

I2C Slave Address Setup Register (I2CS_SADRS)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
I2C Slave Address Setup Register (I2CS_SADRS)	0x4364 (16 bits)	D15–7	–	reserved	–	–	–	0 when being read.
		D6–0	SADRS[6:0]	I2C slave address	0–0x7f	0x0	R/W	

D[15:7] Reserved

D[6:0] SADRS[6:0]: I2CS Address Bits

Sets the slave address of the I2CS module to this register. (Default: 0x0)

I2C Slave Control Register (I2CS_CTL)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
I2C Slave Control Register (I2CS_CTL)	0x4366 (16 bits)	D15–9	–	reserved	–				–	–	0 when being read.
		D8	TBUF_CLR	I2CS_TRNS register clear	1	Clear state	0	Normal	0	R/W	
		D7	I2CSEN	I2C slave enable	1	Enable	0	Disable	0	R/W	
		D6	SOFTRESET	Software reset	1	Reset	0	Cancel	0	R/W	
		D5	NAK_ANS	NAK answer	1	NAK	0	ACK	0	R/W	
		D4	BFREQ_EN	Bus free request enable	1	Enable	0	Disable	0	R/W	
		D3	CLKSTR_EN	Clock stretch On/Off	1	On	0	Off	0	R/W	
		D2	NF_EN	Noise filter On/Off	1	On	0	Off	0	R/W	
		D1	ASDET_EN	Async.address detection On/Off	1	On	0	Off	0	R/W	
		D0	COM_MODE	I2C slave communication mode	1	Active	0	Standby	0	R/W	

D[15:9] Reserved

D8 TBUF_CLR: I2CS_TRNS Register Clear Bit

Clears the I2CS_TRNS register.

1 (R/W): Clear state

0 (R/W): Normal state (clear state cancellation) (default)

When TBUF_CLR is set to 1, the I2CS_TRNS register enters clear state. After that writing 0 to TBUF_CLR returns the I2CS_TRNS register to normal state. It is not necessary to insert a waiting time between writing 1 and 0.

If a new transmission is started when the I2CS_TRNS register still stores data for the previous transmission that has already finished, the data will be sent when TXEMP/I2CS_ASTAT register is set. In order to avoid this problem, clear the I2CS_TRNS register using TBUF_CLR before starting transmission (before slave selection). The clear operation is not required if transmit data is written to the I2CS_TRNS register before TXEMP is set to 1.

Data can be written to the I2CS_TRNS register even if it is placed into clear state (TBUF_CLR = 1). However, this writing does not reset TXEMP to 0. Note that TXEMP is not reset to 0 when TBUF_CLR is set back to 0. Therefore, data must be written to the I2CS_TRNS register when TBUF_CLR = 0.

D7 I2CSEN: I2C Slave Enable Bit

Enables or disables operations of the I2CS module.

1 (R/W): Enabled

0 (R/W): Disabled (default)

When I2CSEN is set to 1, the I2CS module is activated and data transfer is enabled.

When I2CSEN is set to 0, the I2CS module goes off.

D6 SOFTRESET: Software Reset Bit

Resets the I2CS module.

1 (R/W): Reset

0 (R/W): Cancel reset state (default)

To reset the I2CS module, write 1 to SOFTRESET to place the I2CS module into reset status, then write 0 to SOFTRESET to release it from reset status. It is not necessary to insert a waiting time between writing 1 and 0. The I2CS module initializes the I2C communication process and put the SDAS and SCLS pins into high-impedance to be ready to detect a start condition. Furthermore, the I2CS control bits except for SOFTRESET are initialized. Perform the software reset in the initial setting process before staring communication.

D5 **NAK_ANS: NAK Answer Bit**

Specifies the acknowledge bit to be sent after data reception.
1 (R/W): NAK
0 (R/W): ACK (default)

When an eight-bit data is received, the I2CS module sends back an ACK (SDAS = low) or a NAK (SDAS = Hi-Z). Either ACK or NAK should be specified using NAK_ANS within 7 cycles of the I2C clock (SCLS input clock) after RXRDY has been set to 1 by receiving the previous data.

D4 **BFREQ_EN: Bus Free Request Enable Bit**

Enables or disables I2C bus free requests by inputting a low pulse to the #BFR pin.
1 (R/W): Enabled
0 (R/W): Disabled (default)

To accept I2C bus free requests, set BFREQ_EN to 1. When a bus free request is accepted, BFREQ/I2CS_STAT register is set to 1. This initializes the I2C communication process and puts the SDAS and SCLS pins into high-impedance. The control registers will not be initialized in this process.
When BFREQ_EN is set to 0, low pulse inputs to the #BFR pin are ignored and BFREQ is not set to 1.

D3 **CLKSTR_EN: Clock Stretch On/Off Bit**

Turns the clock stretch function on or off.
1 (R/W): On
0 (R/W): Off (default)

After data and ACK are transmitted or received, the slave device may issue a wait request to the master device until it is ready to transmit/receive by pulling the I2C bus SCL line down to Low. The I2CS module supports this clock stretch function. The master device enters a standby state until the wait request is canceled (the SCL line goes high). When using the clock stretch function, set CLKSTR_EN to 1 before starting data communication.

D2 **NF_EN: Noise Filter On/Off Bit**

Turns the noise filter on or off.
1 (R/W): On
0 (R/W): Off (default)

The I2CS module contains a function to remove noise from the SDAS and SCLS input signals. This function is enabled by setting NF_EN to 1.

D1 **ASDET_EN: Async. Address Detection On/Off Bit**

Turns the asynchronous address detection function on or off.
1 (R/W): On
0 (R/W): Off (default)

The I2CS module operation clock (PCLK) frequency must be set eight-times or higher than the transfer rate during data transfer. However, the PCLK frequency can be lowered to reduce current consumption if no other processing is required during standby for data transfer.
The asynchronous address detection function is provided to detect the I2C slave address sent from the master in this status. This function is enabled by setting ASDET_EN to 1. If the slave address sent from the master has matched with one that has been set in this I2CS module when the asynchronous address detection function has been enabled, the I2CS module generates a bus status interrupt and returns NAK to the I2C master to request for resending the slave address. Set the PCLK frequency to eight-times or higher than the transfer rate and reset ASDET_EN to 0 in the interrupt handler routine. Data transfer will be able to resume normally after the master retries transmission. After the master generates a stop condition to put the I2C bus into free status, the asynchronous address detection function can be enabled again to lower the operating speed.

Notes: • When the asynchronous address detection function is enabled, the I2C bus signals are input without passing through the noise filter. Therefore, the slave address may not be detected in a high-noise environment.

- When the asynchronous address detection function is enabled, data transfer cannot be performed even if the PCLK frequency is eight-times or higher than the transfer rate. Be sure to disable the asynchronous address detection function during normal operation.

D0 COM_MODE: I²C Slave Communication Mode Bit

Enables or disables data communication.
1 (R/W): Enabled
0 (R/W): Disabled (default)

Set COM_MODE to 1 to enable data communication after setting I2CSEN to 1 to enable I2CS operation. When COM_MODE is 0 (default), the I2CS module does not send back a response if the master has sent the slave address of this module (it is regarded as that the I2CS module has returned a NAK to the master).

I²C Slave Status Register (I2CS_STAT)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
I²C Slave Status Register (I2CS_STAT)	0x4368 (16 bits)	D15–8	–	reserved	–				–	–	0 when being read.
		D7	BSTAT	Bus status transition	1	Changed	0	Unchanged	0	R	
		D6	–	reserved	–				–	–	0 when being read.
		D5	TXUDF	Transmit data underflow	1	Occurred	0	Not occurred	0	R/W	Reset by writing 1.
			RXOVF	Receive data overflow							
		D4	BFREQ	Bus free request	1	Occurred	0	Not occurred	0	R/W	
		D3	DMS	Output data mismatch	1	Error	0	Normal	0	R/W	
		D2	ASDET	Async. address detection status	1	Detected	0	Not detected	0	R/W	
		D1	DA_NAK	NAK receive status	1	NAK	0	ACK	0	R/W	
		D0	DA_STOP	STOP condition detect	1	Detected	0	Not detected	0	R/W	

D[15:8] Reserved

D7 BSTAT: Bus Status Transition Bit

Indicates transition of the bus status.
1 (R): Changed
0 (R): Unchanged (default)

When one of the TXUDF/RXOVF, BFREQ, DMS, ASDET, DA_NAK, and DA_STOP bits is set to 1, BSTAT is also set to 1 and an interrupt signal is output to the ITC if the interrupt is enabled with BSTAT_IEN/I2CS_ICTL register. This interrupt can be used to perform an error or terminate handling. BSTAT will be reset to 0 when the TXUDF/RXOVF, BFREQ, DMS, ASDET, DA_NAK, and DA_STOP bits are all reset to 0.

D6 Reserved

D5 TXUDF: Transmit Data Underflow Bit (for transmission)
RXOVF: Receive Data Overflow Bit (for reception)

Indicates the transmit/receive data register status.
1 (R/W): Data underflow/overflow has been occurred
0 (R/W): Data underflow/overflow has not been occurred (default)

This bit is effective during transmission/reception when the clock stretch function is disabled. If a data transmission begins before transmit data is written to the I2CS_TRNS register, it is regarded as a transmit data underflow and TXUDF is set to 1. If the next data reception has completed before the received data is read from the I2CS_RECV register and the I2CS_RECV register value is overwritten with the newly received data, it is regarded as a data overflow and RXOVF is set to 1.
At the same time, an interrupt signal is output to the ITC if the interrupt is enabled with BSTAT_IEN/I2CS_ICTL register. This interrupt can be used to perform an error handling.
After TXUDF/RXOVF is set to 1, it is reset to 0 by writing 1.

D4 BFREQ: Bus Free Request Bit

Indicates the I²C bus free request input status.
1 (R/W): Request has been issued
0 (R/W): Request has not been issued (default)

If BFREQ_EN/I2CS_CTL register has been set to 1 (bus free request enabled), a low pulse longer than five peripheral module clock (PCLK) cycles input to the #BFR pin sets BFREQ to 1 and the bus free request is accepted. When a bus free request is accepted, the I2CS module initializes the I²C communication process and puts the SDAS and SCLS pins into high-impedance. The control registers will not be initialized in this process.
At the same time, an interrupt signal is output to the ITC if the interrupt is enabled with BSTAT_IEN/I2CS_ICTL register. This interrupt can be used to perform an error handling.
After BFREQ is set to 1, it is reset to 0 by writing 1.
If BFREQ_EN is set to 0, low pulse inputs to the #BFR pin are ignored and BFREQ is not set to 1.

D3 DMS: Output Data Mismatch Bit

Represents the results of comparison between output data and SDA line status.
1 (R/W): Error has been occurred
0 (R/W): Error has not been occurred (default)

The I²C bus SDA line status during data transmission is input in the module and is compared with the output data. The comparison results are set to DMS. DMS is set to 0 when data is output correctly. If the SDA line status is different from the output data, DMS is set to 1. This may be caused by a low pull-up resistor value or another device that is controlling the SDA line. At the same time, an interrupt signal is output to the ITC if the interrupt is enabled with BSTAT_IEN/I2CS_ICTL register. This interrupt can be used to perform an error handling. After DMS is set to 1, it is reset to 0 by writing 1.

Note: When the master device of the I²C bus, which has multiple slave devices connected including this IC, starts communication with another slave device, the I2CS module of this IC issues NAK in response to the sent slave address. On the other hand, the selected slave device issues ACK. Therefore, DMS may be set due to a difference between the output value of this IC and the SDA line status. When SELECTED/I2CS_ASTAT register is set to 0, you can ignore DMS without a problem even if it is set to 1 as there is a difference in the response code (ACK/NAK) from the selected slave device. When the I2CS module is placed into asynchronous address detection mode (ASDET_EN = 1), a DMS does not occur as in the condition above.

D2 ASDET: Async. Address Detection Status Bit

Indicates the asynchronous address detection status.
1 (R/W): Detected
0 (R/W): Not detected (default)

The I2CS module operation clock (PCLK) frequency must be set eight-times or higher than the transfer rate during data transfer. However, the PCLK frequency can be lowered to reduce current consumption if no other processing is required during standby for data transfer. The asynchronous address detection function is provided to detect the I²C slave address sent from the master in this status. ASDET is set to 1 if the slave address of the I2CS module is detected when the asynchronous address detection function has been enabled by setting ASDET_EN/I2CS_CTL register.
The I2CS module returns a NAK to the I²C master to request for resending the slave address. At the same time, an interrupt signal is output to the ITC if the interrupt is enabled with BSTAT_IEN/I2CS_ICTL register. Set the PCLK frequency to eight-times or higher than the transfer rate and reset ASDET_EN to 0 in the interrupt handler routine. Data transfer will be able to resume normally after the master retries transmission. After ASDET is set to 1, it is reset to 0 by writing 1.

D1 DA_NAK: NAK Receive Status Bit

Indicates the acknowledge bit returned from the master.
1 (R/W): NAK
0 (R/W): ACK (default)

DA_NAK is set to 0 when an ACK is returned from the master after an eight-bit data has been sent. This indicates that the master could receive data. If DA_NAK is 1, it indicates that the master could not receive data or the master terminates data reception. At the same time DA_NAK is set to 1, an interrupt signal is output to the ITC if the interrupt is enabled with BSTAT_IEN/I2CS_ICTL register. This interrupt can be used to perform an error handling. After DA_NAK is set to 1, it is reset to 0 by writing 1.

D0 DA_STOP: Stop Condition Detect Bit

Indicates that a stop condition or a repeated start condition is detected.
1 (R/W): Detected
0 (R/W): Not detected (default)

If a stop condition or a repeated start condition is detected while the I2CS module is selected as the slave device (SELECTED/I2CS_ASTAT register = 1), the I2CS module sets DA_STOP to 1. At the same time, it initializes the I²C communication process.
When DA_STOP is set to 1, an interrupt signal is output to the ITC if the interrupt is enabled with BSTAT_IEN/I2CS_ICTL register. This interrupt can be used to perform a terminate handling. After DA_STOP is set to 1, it is reset to 0 by writing 1.

I²C Slave Access Status Register (I2CS_ASTAT)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
I²C Slave Access Status Register (I2CS_ASTAT)	0x436a (16 bits)	D15–5	–	reserved	–				–	–	0 when being read.
		D4	RXRDY	Receive data ready	1	Ready	0	Not ready	0	R	
		D3	TXEMP	Transmit data empty	1	Empty	0	Not empty	0	R	
		D2	BUSY	I²C bus status	1	Busy	0	Free	0	R	
		D1	SELECTED	I²C slave select status	1	Selected	0	Not selected	0	R	
		D0	R/W	Read/write direction	1	Output	0	Input	0	R	

D[15:5] Reserved

D4 RXRDY: Receive Data Ready Bit

Indicates that the received data is ready to read.
1 (R): Received data ready
0 (R): No received data (default)

When the received data is loaded to the I2CS_RECV register, RXRDY is set to 1. At the same time, an interrupt signal is output to the ITC if the interrupt is enabled with RXRDY_IEN/I2CS_ICTL register. This interrupt can be used to read the received data from the I2CS_RECV register.
After RXRDY is set to 1, it is reset to 0 when the I2CS_RECV register is read.

D3 TXEMP: Transmit Data Empty Bit

Indicates that transmit data can be written.
1 (R): Transmit data empty (data can be written)
0 (R): Transmit data still stored (data cannot be written) (default)

TXEMP is set to 1 if the I2CS_TRNS register is empty when a read request (R/W bit = 1) issued by the I²C master is received or if the I2CS_TRNS register becomes empty until a stop condition that terminates transmission is generated by the I²C master. At the same time TXEMP set to 1, an interrupt signal is output to the ITC if the interrupt is enabled with TXEMP_IEN/I2CS_ICTL register. This interrupt can be used to write the next transmit data to the I2CS_TRNS register.
After TXEMP is set to 1, it is reset to 0 when data is written to the I2CS_TRNS register.

D2 BUSY: I²C Bus Status Bit

Indicates the I²C bus status.
1 (R): Bus busy status
0 (R): Bus free status (default)

When the I2CS module detects a start condition or detects that the SCLS or SDAS signal goes low, BUSY is set to 1 to indicate that the I²C bus enters busy status. The slave select status whether this module is selected as the slave device or not does not affect the BUSY status. After BUSY is set to 1, it is reset to 0 when a STOP condition is detected.

D1 SELECTED: I²C Slave Select Status Bit

Indicates that this module is selected as the I²C slave device.
1 (R): Selected
0 (R): Not selected (default)

When the slave address that is set in this module is received, SELECTED is set to 1 to indicate that this module is selected as the I²C slave device. After SELECTED is set to 1, it is reset to 0 when a stop condition or a repeated start condition is detected.

D0 R/W: Read/Write Direction Bit

Represents the transfer direction bit value.

1 (R): Output (master read operation)

0 (R): Input (master write operation) (default)

The transfer direction bit value that has been received with the slave address is set to R/W. Use R/W to select the transmit- or receive-handling.

I2C Slave Interrupt Control Register (I2CS_ICTL)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
I2C Slave Interrupt Control Register (I2CS_ICTL)	0x436c (16 bits)	D15–3	–	reserved	–				–	–	0 when being read.
		D2	BSTAT_IEN	Bus status interrupt enable	1	Enable	0	Disable	0	R/W	
		D1	RXRDY_IEN	Receive interrupt enable	1	Enable	0	Disable	0	R/W	
		D0	TXEMP_IEN	Transmit interrupt enable	1	Enable	0	Disable	0	R/W	

D[15:3] Reserved

D2 BSTAT_IEN: Bus Status Interrupt Enable Bit

Enables or disables the bus status interrupt.

1 (R/W): Enabled

0 (R/W): Disabled (default)

When BSTAT_IEN is set to 1, I2C bus status interrupt requests to the ITC are enabled. A bus status interrupt request occurs when BSTAT/I2CS_STAT register is set to 1. (See description of BSTAT.) When BSTAT_IEN is set to 0, a bus status interrupt will not be generated.

D1 RXRDY_IEN: Receive Interrupt Enable Bit

Enables or disables the I2CS receive interrupt.

1 (R/W): Enabled

0 (R/W): Disabled (default)

When RXRDY_IEN is set to 1, I2CS receive interrupt requests to the ITC are enabled. A receive interrupt request occurs when the data received in the shift register is loaded to the I2CS_RECV register (receive operation completed). When RXRDY_IEN is set to 0, a receive interrupt will not be generated.

D0 TXEMP_IEN: Transmit Interrupt Enable Bit

Enables or disables the I2CS transmit interrupt.

1 (R/W): Enabled

0 (R/W): Disabled (default)

When TXEMP_IEN is set to 1, I2CS transmit interrupt requests to the ITC are enabled. A transmit interrupt request occurs when the data written to the I2CS_TRNS register is transferred to the shift register. When TXEMP_IEN is set to 0, a transmit interrupt will not be generated.

20 EPD Controller/Driver (EPD)

20.1 EPD Module Overview

The S1C17F57 includes an EPD controller/driver (EPD) module to implement EPD display control functions. The main features of the EPD module are listed below.

- Number of driver outputs: 64 segments + 2 back plane + 2 top plane outputs
- EPD drive voltages: Two voltage values, VEPD and VSS
- EPD drive power control function
- Includes an EPD display waveform memory (programmable drive waveform).
- Includes a display data memory.
- Supports reverse, all white, and all black display functions.
- Can generate an interrupt at the end of the drive waveform output.
- Allows the software to direct control the outputs from the segment, back plane, and top plane pins.
- A trigger signal and clock can be output to an external Seiko Epson EPD driver.

Figure 20.1.1 shows the configuration of the EPD module.

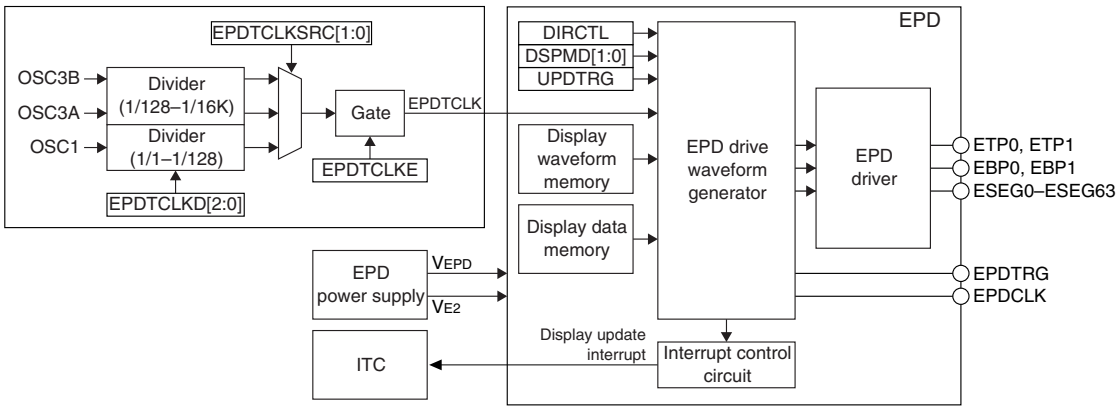


Figure 20.1.1 EPD Module Configuration

20.2 EPD Output Pins

Table 20.2.1 lists the EPD pins.

Table 20.2.1 List of EPD Pins

Pin name	I/O	Qty	Function
ESEG0–ESEG63	O	64	EPD segment output pins These pins output the segment drive waveforms.
ETP0 ETP1	O	2	EPD top plane output pins These two pins output the same top plane drive waveforms.
EBP0 EBP1	O	2	EPD back plane output pins These two pins output the same back plane drive waveforms.
EPDTRG	O	1	External EPD driver trigger output pin Outputs a trigger signal to update a screen when an external EPD driver is used.
EPDCLK	O	1	External EPD driver clock output pin Outputs the operating clock when an external EPD driver is used.

The EPD module output pins (EPDTRG, EPDCLK) are shared with I/O ports and are initially set as general purpose I/O port pins. The pin functions must be switched using the port function select bits to use the general purpose I/O port pins as EPD module output pins. For detailed information on pin function switching, see the "I/O Ports (P)" chapter.

20.3 Power Supply for EPD Driver

The chip generates the EPD drive voltage VEPD using the on-chip EPD power supply circuit, so no external power supply is required. For more information on the power supply, see the "Power Supply" chapter.

Note: An external power supply can be used if the on-chip EPD power supply circuit has not enough drive capability. For more information on use of an external power supply, see the "Power Supply" chapter.

20.4 EPD Clock

The EPD controller includes a clock source selector, dividers, and a gate circuit for controlling the EPD timing clock (reference clock for generating drive waveforms).
When using the EPD controller in wave mode or when supplying a clock to the external EPD driver, make the settings shown below to supply the clock to the EPD drive waveform generator. These settings are not required when the EPD controller is used in direct mode.

Clock source selection

Select the clock source from OSC3B, OSC3A, and OSC1 using EPDTCLKSRC[1:0]/EPD_TCLK register.

Table 20.4.1 Clock Source Selection

EPDTCLKSRC[1:0]	Clock source
0x3	Reserved
0x2	OSC3A
0x1	OSC1
0x0	OSC3B

(Default: 0x0)

Clock division ratio selection

Select the division ratio using EPDTCLKD[2:0]/EPD_TCLK register.

Table 20.4.2 Division Ratio Selection

EPDTCLKD[2:0]	Division ratio	
	Clock source = OSC3B or OSC3A	Clock source = OSC1
0x7	1/16384	1/128
0x6	1/8192	1/64
0x5	1/4096	1/32
0x4	1/2048	1/16
0x3	1/1024	1/8
0x2	1/512	1/4
0x1	1/256	1/2
0x0	1/128	1/1

(Default: 0x0)

Clock enable

The clock supply is enabled with EPDTCLKE/EPD_TCLK register. The EPDTCLKE default setting is 0, which stops the clock. Setting EPDTCLKE to 1 feeds the clock generated as above to the EPD drive waveform generator. If no EPD controller operation is required or when the EPD controller is used in direct mode, stop the clock to reduce current consumption.

20.5 Operating Mode

The EPD controller supports two operating modes (wave mode and direct mode) that can be selected using DIRCTL/EPD_CTL register.

Wave mode (DIRCTL = 0, default)

When a screen update trigger is issued, the EPD controller outputs the drive waveforms programmed in the display waveform memory. In this mode, no CPU power is required for generating drive waveforms. For more information on the waveform programming, see Section 20.6.

Direct mode (DIRCTL = 1)

This mode is provided to control the segment, top plane, and back plane pin outputs directly via software. The program must control the waveform generation in real time.

20.6 Display Waveform Memory

The display waveform memory stores the settings for the EPD controller to generate drive waveforms in wave mode.
The display waveform memory is capable of storing up to 32 timing sets (Timing set 0 to Timing set 31) that consist of 15 bits. Timing set 0 represents the initial state and its period generated immediately after a screen update trigger is issued. The changes in the waveforms should be programmed one by one in the subsequent timing sets (1 to n = Max. 31).
A timing set consists of the bits shown below.

Table 20.6.1 Contents of a Timing Set

Bit name*	Contents
EOW (End Of Wave, D15)	Specifies the end of waveform. The waveform generation is completed at the timing set in which EOW is set to 1. The pins are set into high-impedance state until the subsequent display update trigger is issued. The timing sets other than the last must be programmed with EOW set to 0.
HIZ (High Impedance, D13)	Sets the segment and back plane pins into high-impedance state. When this bit is set to 1, the segment and back plane pins are placed into high-impedance state for the period specified in the timing set (BB/BW/WB/WW settings are ignored). When this bit is set to 0, the pins go to the level specified by BB/BW/WB/WW.
TP (Top Plane, D12)	Sets the waveform output from the top plane pins. When this bit is set to 1, the top plane pins go to the V_{EPD} level for the period specified in the timing set. When this bit is set to 0, the pins go to the V_{SS} level.
BB (Black to Black, D11)	Sets the waveform output from the segment and back plane pins to maintain black display during update (black to black). When this bit is set to 1, the segment and back plane pins go to the V_{EPD} level for the period specified in the timing set. When this bit is set to 0, the pins go to the V_{SS} level.
BW (Black to White, D10)	Sets the waveform output from the segment and back plane pins to switch black display to white during update. When this bit is set to 1, the segment and back plane pins go to the V_{EPD} level for the period specified in the timing set. When this bit is set to 0, the pins go to the V_{SS} level.
WB (White to Black, D9)	Sets the waveform output from the segment and back plane pins to switch white display to black during update. When this bit is set to 1, the segment and back plane pins go to the V_{EPD} level for the period specified in the timing set. When this bit is set to 0, the pins go to the V_{SS} level.
WW (White to White, D8)	Sets the waveform output from the segment and back plane pins to maintain white display during update (white to white). When this bit is set to 1, the segment and back plane pins go to the V_{EPD} level for the period specified in the timing set. When this bit is set to 0, the pins go to the V_{SS} level.
INTV[7:0] (Interval, D[7:0])	Specifies the timing set period in a number of EPDTCLK clocks. Time [s] = (INTV[7:0] + 1)/EPDTCLK frequency

* D*x* indicates a bit number in the EPD_WAVE*x* registers.

The conditions above can be set using the EPD_WAVE0 to EPD_WAVE31 registers provided for the timing sets individually. The registers consist of the bits with the names shown above.

The following simple example shows the correspondence between the settings in the timing sets and the waveforms generated.

Table 20.6.2 Setting Example of Timing Sets

Timing set number (register)	EOW (D15)	HIZ (D13)	TP (D12)	BB (D11)	BW (D10)	WB (D9)	WW (D8)	INTV[7:0] (D[7:0])
0 (EPD_WAVE0)	0	0	0	0	0	0	0	0x1
1 (EPD_WAVE1)	0	0	1	0	1	0	1	0x2
2 (EPD_WAVE2)	0	0	0	0	0	1	1	0x0
3 (EPD_WAVE3)	0	1	1	*	*	*	*	0x1
4 (EPD_WAVE4)	1	0	0	0	0	0	0	0x3

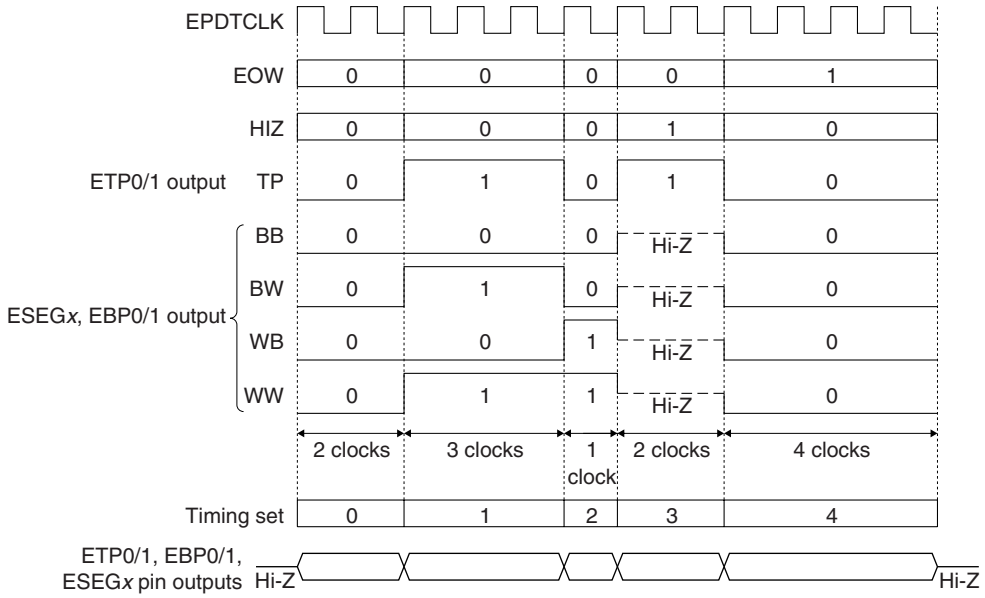


Figure 20.6.1 Example of display waveforms (by settings in Table 20.6.2)

Note: Table 20.6.2 and Figure 20.6.1 show merely an example to explain relationships between the settings and waveforms, not those that can be used for actual EPD driving.

20.7 Display Data Memory

The EPD module includes a display data memory that consists of 64 bits (for segment outputs) + 2 bits (for top plane and back plane outputs).
Segment output data should be set to the SEG0 to SEG63 bits in the EPD_SEGDAT0 to EPD_SEGDAT3 registers. Back plane output data should be set to BP/EPD_PLNDAT register. Writing 1 to the bit specifies black display in wave mode or high (V_{EPD}) level output in direct mode; writing 0 specifies white display in wave mode or low (V_{SS}) level output in direct mode.
Top plane output data should be set to TP/EPD_PLNDAT register. This bit is effective only in direct mode. Writing 1 specifies high (V_{EPD}) level output; writing 0 specifies low (V_{SS}) level output.
In wave mode, the top plane output is controlled according to the contents of the display waveform memory.

The display data memory settings take effect when a display update trigger (described later) is issued. Writing to the display data memory does not update the screen.

After an initial reset, the display data memory is cleared to 0.

20.8 Display Control in Wave Mode

This section describes the display control method in wave mode.
Make the following all settings regardless of the sequence before starting display:

(1) Set up the EPD timing clock to feed it to the EPD drive waveform generator. See Section 20.4.

(2) Configures the EPD power supply circuit to supply the EPD drive voltage to the EPD controller. See the "Power Supply" chapter.

Note: Be sure to supply an external power voltage when the on-chip EPD power supply circuit is not used (see the "Power Supply" chapter).

(3) Program the drive waveforms in the display waveform memory. See Section 20.6.

(4) Enable EPD interrupts. See Section 20.12.

Control the display as the flow chart shown in Figure 20.8.1.

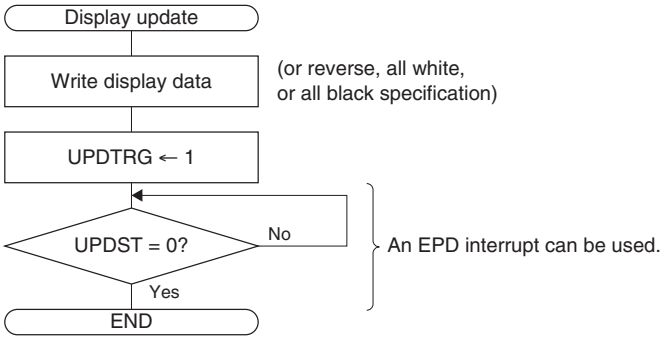


Figure 20.8.1 Display Control in Wave Mode

Normal display control

Display can be updated in the procedure shown below.

(1) Write segment and back plane display data for update to the display data memory.

(2) Write 1 to UPDTRG/EPD_CTL register to issue a display update trigger.
The EPD drive waveform generator outputs the drive waveforms programmed in the display waveform memory to the segment, top plane and back plane pins according to the contents of the display data memory and the current display data. The waveform generation status can be checked using UPDST/EPD_CTL register. UPDST goes 1 by a display update trigger and reverts to 0 upon completion of the programmed drive waveform output. The last display data before updating are stored in the EPD controller.

(3) When the display update has completed (at the same time UPDST reverts to 0), an EPD interrupt occurs.

Notes: • Be sure to avoid rewriting the display data memory while UPDST is set to 1 (display update is underway).

- Writing 1 to UPDTRG is ineffective while UPDST is set to 1.
- The EPD controller generates the drive waveforms with the current display data regarded as 0 in the first display update after an initial reset.
- When direct mode is switched to wave mode, fluctuations on screen may occur only in the first display update.

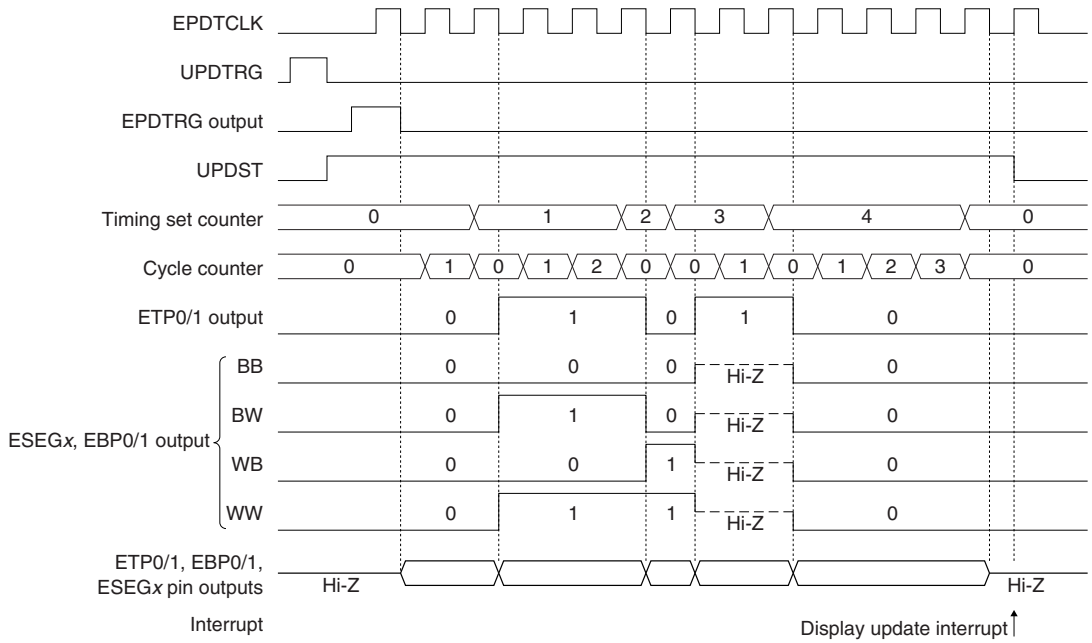


Figure 20.8.2 Timing Chart of Waveform Output Example (by settings in Table 20.6.2)

Reverse, all white, and all black display control

In wave mode, the screen can be reversed or turned to all white or black without rewriting the display data memory via software. Use DSPMD[1:0]/EPD_CTL register for this control.

Table 20.8.1 Display Control

DSPMD[1:0]	Display mode
0x3	All black display
0x2	All white display
0x1	Reverse display
0x0	Normal display

(Default: 0x0)

By writing 1 to UPDTRG after altering DSPMD[1:0], drive waveforms are output to change the display.
However, do not write 1 to UPDTRG while UPDST is set to 1.

All white display mode: In this mode, writing 1 to UPDTRG updates the display so that the all segments will turn to white regardless of the display memory contents.

All black display mode: In this mode, writing 1 to UPDTRG updates the display so that the all segments will turn to black regardless of the display memory contents.

Reverse display mode: In this mode, meaning of the display memory bits are swapped between 1 and 0. In other words, writing 1 to UPDTRG in this mode updates the display so that the segments/back plane corresponding to the bits set to 1 will turn to white, and those corresponding to the bits set to 0 will turn to black.

Note: Do not alter DSPMD[1:0] while UPDST is set to 1.

20.9 Display Control in Direct Mode

This section describes the display control method in direct mode.
Before starting display, configures the EPD power supply circuit to supply the EPD drive voltage to the EPD controller. See the "Power Supply" chapter.

Update the display as the flow chart shown in Figure 20.9.1.

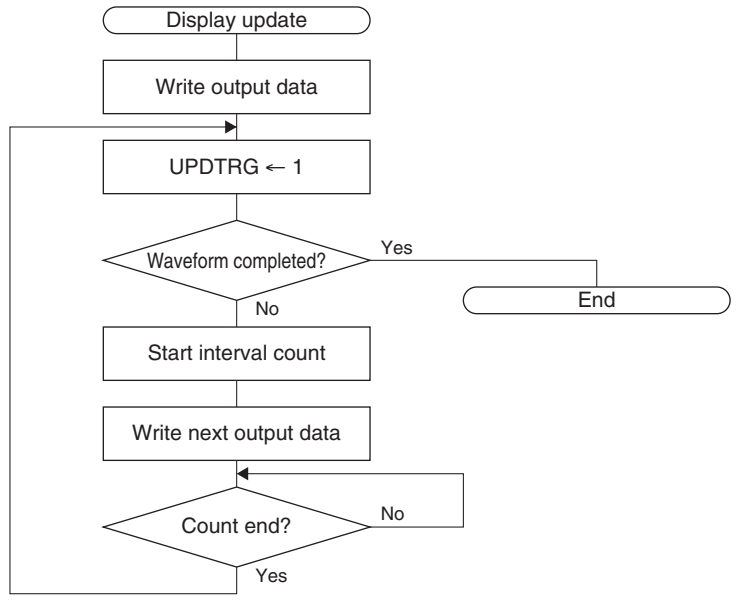


Figure 20.9.1 Display Control in Direct Mode

(1) Write the initial output level (high/low/Hi-Z) of the segment, top plane, and back plane pins to the display data memory.
To set the segment and back plane pins into high-impedance, write 1 to SEGHZ/EPD_PLNDAT register. To set the top plane pins into high-impedance, write 1 to TPHZ/EPD_PLNDAT register. Settings in the display data memory are ineffective if SEGHZ and TPHZ is 1.

(2) Write 1 to UPDTRG/EPD_CTL register to issue a display update trigger.
The EPD drive waveform generator outputs the contents of the display data memory to the segment, top plane and back plane pins.

(3) Start counting the current level output period using a timer.

(4) Write the subsequent output data to the display data memory during counting.

(5) Wait for the end of counting such as a timer interrupt.

(6) Terminate the processing when the display update waveforms have been output.
Return to Step (2) if the display update waveforms have not been output completely.

20.10 Outputs for External Driver

The EPD controller is able to output the timing clock (EPDCLK) and the display update trigger signal allowing use of an external EPD driver. The clock is output from the EPDCLK pin and the trigger signal is output from the EPDTRG pin. No output control is required, note, however, that the output pins must be switched for EPD outputs using the port function select bits, as the pins are configured for I/O ports by default. For the trigger signal output timing, see Figure 20.8.2.

20.11 EPD Interrupt

The EPD module includes a function for generating interrupts at the end of display update waveform output in wave mode.

Display update interrupt

This interrupt request occurs when the drive waveform output by a display update trigger (output of the timing set in which EOW is set) has finished and sets the interrupt flag DUPDIF/EPD_INT register in the EPD module to 1. See Figure 20.8.2 for the interrupt timing.
To use this interrupt, set DUPDIE/EPD_INT register to 1. When DUPDIE is set to 0 (default), interrupt requests for this interrupt cause are not sent to the interrupt controller (ITC). If DUPDIF is set to 1 while DUPDIE is set to 1 (interrupt enabled), the EPD module outputs an interrupt request to the ITC. An interrupt is generated if the ITC and S1C17 Core interrupt conditions are satisfied.
For more information on interrupt processing, see the "Interrupt Controller (ITC)" chapter.

Notes: • To prevent interrupt recurrences, the EPD module interrupt flag DUPDIF must be reset in the interrupt handler routine after an EPD interrupt has occurred.

• To prevent unwanted interrupts, DUPDIF should be reset before enabling EPD interrupts with DUPDIE.

20.12 Control Register Details

Table 20.12.1 List of EPD Registers

Address	Register name		Function
0x5070	EPD_TCLK	EPD Timing Clock Control Register	Selects the EPD timing clock.
0x5604	EPD_CTL	EPD Display Control Register	Controls display on EPD.
0x5606	EPD_INT	EPD Interrupt Control Register	Controls interrupts.
0x5620	EPD_PLNDAT	EPD Top/Back Plane Data Register	Top plane/back plane output data
0x5622	EPD_SEGDAT0	EPD Segment Data Register 0	Segment output data (ESEG0–ESEG15)
0x5624	EPD_SEGDAT1	EPD Segment Data Register 1	Segment output data (ESEG16–ESEG31)
0x5626	EPD_SEGDAT2	EPD Segment Data Register 2	Segment output data (ESEG32–ESEG47)

Address	Register name		Function
0x5628	EPD_SEGDAT3	EPD Segment Data Register 3	Segment output data (ESEG48–ESEG63)
0x5640	EPD_WAVE0	EPD Wave Timing Set 0 Register	Display waveform data (Timing set 0)
0x5642	EPD_WAVE1	EPD Wave Timing Set 1 Register	Display waveform data (Timing set 1)
:	:	:	:
0x567e	EPD_WAVE31	EPD Wave Timing Set 31 Register	Display waveform data (Timing set 31)

The EPD module registers are described in detail below.

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

EPD Timing Clock Control Register (EPD_TCLK)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
EPD Timing Clock Control Register (EPD_TCLK)	**0x5070** (8 bits)	D7	**–**	reserved	–	–	–	0 when being read.
		D6–4	**EPDTCLKD [2:0]**	EPD timing clock division ratio select	EPDTCLKD [2:0]: Division ratio (OSC3B/OSC3A, OSC1) 0x7: 1/16384, 1/128 0x6: 1/8192, 1/64 0x5: 1/4096, 1/32 0x4: 1/2048, 1/16 0x3: 1/1024, 1/8 0x2: 1/512, 1/4 0x1: 1/256, 1/2 0x0: 1/128, 1/1	0x0	R/W	
		D3–2	**EPDTCLK SRC[1:0]**	EPD timing clock source select	EPDTCLK SRC[1:0]: Clock source 0x3: reserved 0x2: OSC3A 0x1: OSC1 0x0: OSC3B	0x0	R/W	
		D1	**–**	reserved	–	–	–	0 when being read.
		D0	**EPDTCLKE**	EPD timing clock enable	1: Enable, 0: Disable	0	R/W	

Note: The EPD timing clock must be set up when the EPD controller is used in wave mode or when supplying the clock to an external EPD driver. The settings are not required when the EPD controller is used in direct mode.

D7 Reserved

D[6:4] EPDTCLKD[2:0]: EPD Timing Clock Division Ratio Select Bits

Selects the division ratio for generating the EPD timing clock.

Table 20.12.2 Clock Division Ratio Selection

EPDTCLKD[2:0]	Division ratio	
	Clock source = OSC3B or OSC3A	Clock source = OSC1
0x7	1/16384	1/128
0x6	1/8192	1/64
0x5	1/4096	1/32
0x4	1/2048	1/16
0x3	1/1024	1/8
0x2	1/512	1/4
0x1	1/256	1/2
0x0	1/128	1/1

(Default: 0x0)

D[3:2] EPDTCLKSRC[1:0]: EPD Timing Clock Source Select Bits

Selects the EPD timing clock source.

Table 20.12.3 Clock Source Selection

EPDTCLKSRC[1:0]	Clock source
0x3	Reserved
0x2	OSC3A
0x1	OSC1
0x0	OSC3B

(Default: 0x0)

D1 **Reserved**

D0 **EPDTCLKE: EPD Timing Clock Enable Bit**

Enables or disables the EPD timing clock supply to the drive waveform generator.
1 (R/W): Enabled (on)
0 (R/W): Disabled (off) (default)

The EPDTCLKE default setting is 0, which disables the clock supply. Setting EPDTCLKE to 1 sends the clock selected as above to the drive waveform generator. If EPD controller operation is not required or is used in direct mode, disable the clock supply to reduce current consumption.

EPD Display Control Register (EPD_CTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
EPD Display Control Register (EPD_CTL)	0x5604 (16 bits)	D15–5	–	reserved	–	–	–	0 when being read.
		D4	DIRCTL	Wave/direct mode select	1 Direct 0 Wave	0	R/W	
		D3–2	DSPMD [1:0]	Display mode select	DSPMD[1:0] Display mode 0x3 All black 0x2 All white 0x1 Reverse 0x0 Normal	0x0	R/W	Effective only in wave mode
		D1	UPDST	Display update status	1 Busy 0 Idle	0	R	Always set to 0 in direct control mode.
		D0	UPDTRG	Display update trigger	1 Trigger 0 Ignored	0	W	0 when being read.

D[15:5] **Reserved**

D4 **DIRCTL: Wave/Direct Mode Select Bit**

Sets the EPD controller into wave or direct mode.
1 (R/W): Direct mode
0 (R/W): Wave mode (default)

Writing 0 to DIRCTL sets the EPD controller into wave mode. When a screen update trigger is issued in wave mode, the EPD controller outputs the drive waveforms programmed in the display waveform memory. In this mode, no CPU power is required for generating drive waveforms.
Writing 1 to DIRCTL sets the EPD controller into direct mode. This mode is provided to control the segment, top plane, and back plane pin outputs directly via software. The program must control the waveform generation in real time.

D[3:2] **DSPMD[1:0]: Display Mode Select Bits (Wave mode)**

Selects a display mode.

Table 20.12.4 Display Control

DSPMD[1:0]	Display mode
0x3	All black display
0x2	All white display
0x1	Reverse display
0x0	Normal display

(Default: 0x0)

To turn the display to reverse, all white, or all black, write 1 to UPDTRG after setting DSPMD[1:0]. However, do not write 1 to UPDTRG while UPDST is set to 1. This display control does not affect the display data memory.
To perform normal display, set DSPMD[1:0] to 0x0 and perform display update operations.
The display mode selection is effective only in wave mode. DSPMD[1:0] settings are ineffective in direct mode.

D1 **UPDST: Display Update Status Bit (Wave mode)**

Indicates the display update status in wave mode.
1 (R): Busy
0 (R): Idle (default)

UPDST goes 1 when a display update operation is started by writing 1 to UPDTRG, and reverts to 0 upon completion of the programmed drive waveform output (output of the timing set in which EOW is set).
UPDST is ineffective in direct mode and is always read as 0.

Note: Be sure to avoid rewriting the display data memory and altering DSPMD[1:0] while UPDST is set to 1.

D0 UPDTRG: Display Update Trigger Bit

Starts display update sequence.
1 (W): Display update trigger
0 (W): Ignored (default)

In wave mode, the EPD drive waveform generator outputs the drive waveforms programmed in the display waveform memory to the segment, top plane and back plane pins according to the contents of the display data memory. Writing 1 to UPDTRG is ineffective while UPDST is set to 1 (display update is underway).
In direct mode, the EPD drive waveform generator switches the segment, top plane and back plane pin output levels to the contents of the display data memory.

EPD Interrupt Control Register (EPD_INT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
EPD Interrupt Control Register (EPD_INT)	0x5606 (16 bits)	D15–9	–	reserved	–	–	–	0 when being read.
		D8	DUPDIF	Display update interrupt flag	1 Occurred 0 Not occurred	0	R/W	Reset by writing 1.
		D7–1	–	reserved	–	–	–	0 when being read.
		D0	DUPDIE	Display update interrupt enable	1 Enable 0 Disable	0	R/W	

D[15:9] Reserved

D8 DUPDIF: Display Update Interrupt Flag Bit (Wave mode)

Indicates whether the cause of display update interrupt has occurred or not.
1 (R): Cause of interrupt has occurred
0 (R): No cause of interrupt has occurred (default)
1 (W): Flag is reset
0 (W): Ignored

DUPDIF is set when the drive waveform output by a display update trigger (output of the timing set in which EOW is set) has finished. DUPDIF is reset by writing 1.

D[7:1] Reserved

D0 DUPDIE: Display Update Interrupt Enable Bit (Wave mode)

Enables or disables display update interrupts.
1 (R/W): Interrupt enabled
0 (R/W): Interrupt disabled (default)

Setting DUPDIE to 1 enables EPD interrupt requests to the ITC. Setting it to 0 disables interrupts.

EPD Top/Back Plane Data Register (EPD_PLNDAT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
EPD Top/Back Plane Data Register (EPD_PLNDAT)	0x5620 (16 bits)	D15–10	–	reserved	–	–	–	0 when being read.
		D9	SEGHZ	Segment/back plane Hi-Z control	1 Hi-Z 0 Normal	0	R/W	Effective only in direct mode
		D8	TPHZ	Top plane Hi-Z control	1 Hi-Z 0 Normal	0	R/W	
		D7–5	–	reserved	–	–	–	0 when being read.
		D4	TP	Top plane control data	1 High 0 Low	0	R/W	Effective only in direct mode
		D3–1	–	reserved	–	–	–	0 when being read.
		D0	BP	Back plane display data	1 Black 0 White	0	R/W	Wave mode
					1 High 0 Low			Direct mode

D[15:10] Reserved

D9 SEGHZ: Segment/Back Plane Hi-Z Control Bit (Direct mode)

Sets the segment/back plane outputs into high-impedance state.
1 (R/W): High-impedance
0 (R/W): Normal output (default)

Issuing a display update trigger (UPDTRG = 1) after setting SEGHZ to 1 sets the segment and back plane pins into high-impedance state. In this case, pin output level settings using the BP and SEG*xx* bits are ineffective.
Pin output level settings using the BP and SEG*xx* bits take effect when SEGHZ is set to 0.
Note that SEGHZ is ineffective in wave mode.

D8 TPHZ: Top Plane Hi-Z Control Bit (Direct mode)

Sets the top plane outputs into high-impedance state.
1 (R/W): High-impedance
0 (R/W): Normal output (default)

Issuing a display update trigger (UPDTRG = 1) after setting TPHZ to 1 sets the top plane pins into high-impedance state. In this case, pin output level settings using the TP bit is ineffective.
Pin output level settings using the TP bit take effect when TPHZ is set to 0.
Note that TPHZ is ineffective in wave mode.

D[7:5] Reserved

D4 TP: Top Plane Control Data Bit (Direct mode)

Sets the top plane output level.
1 (R/W): High level
0 (R/W): Low level (default)

Issuing a display update trigger (UPDTRG = 1) after setting TP to 1 sets the top plane pins to high. The pins go to low if TP is 0.
Note that TP is ineffective in wave mode.

D[3:1] Reserved

D0 BP: Back Plane Display Data Bit

Sets the back plane display data/output level.
1 (R/W): Black (wave mode)/High level (direct mode)
0 (R/W): White (wave mode)/Low level (direct mode) (default)

In wave mode, issuing a display update trigger (UPDTRG = 1) after setting BP to 1 outputs the drive waveform that turns the back plane display to black from the back plane pins. The pins output the drive waveform that turns the display to white if BP is 0.
In direct mode, issuing a display update trigger (UPDTRG = 1) after setting BP to 1 sets the back plane pins to high. The pins go to low if BP is 0.

Note: Settings to this register do not update the screen. A display update trigger using UPDTRG is required.

EPD Display Data Registers 0–3 (EPD_SEGDAT0–EPD_SEGDAT3)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
EPD Segment Data Register 0 (EPD_SEGDAT0)	**0x5622** (16 bits)	D15–0	**SEG[15:0]**	ESEG[15:0] display data	1	Black	0	White	0	R/W	Wave mode
					1	High	0	Low			Direct mode
EPD Segment Data Register 1 (EPD_SEGDAT1)	**0x5624** (16 bits)	D15–0	**SEG[31:16]**	ESEG[31:16] display data	1	Black	0	White	0	R/W	Wave mode
					1	High	0	Low			Direct mode
EPD Segment Data Register 2 (EPD_SEGDAT2)	**0x5626** (16 bits)	D15–0	**SEG[47:32]**	ESEG[47:32] display data	1	Black	0	White	0	R/W	Wave mode
					1	High	0	Low			Direct mode
EPD Segment Data Register 3 (EPD_SEGDAT3)	**0x5628** (16 bits)	D15–0	**SEG[63:48]**	ESEG[63:48] display data	1	Black	0	White	0	R/W	Wave mode
					1	High	0	Low			Direct mode

D[15:0] SEG*xx*: ESEG*xx* Display Data Bits

Sets the segment display data/output level.
1 (R/W): Black (wave mode)/High level (direct mode)
0 (R/W): White (wave mode)/Low level (direct mode) (default)

In wave mode, issuing a display update trigger (UPDTRG = 1) after setting SEG*xx* to 1 outputs the drive waveform that turns the segment to black from the ESEG*xx* pin. The pin outputs the drive waveform that turns the display to white if SEG*xx* is 0.
In direct mode, issuing a display update trigger (UPDTRG = 1) after setting SEG*xx* to 1 sets the segment pin to high. The pin goes low if SEG*xx* is 0.

Note: Settings to the data registers do not update the screen. A display update trigger using UPDTRG is required.

EPD Waveform Timing Set *x* Registers (EPD_WAVE*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
EPD Waveform Timing Set 0 Register (EPD_WAVE0) | EPD Waveform Timing Set 31 Register (EPD_WAVE31)	0x5640 | 0x567e (16 bits)	D15	EOW	End of wave	1 End 0 Continue	0	R/W	
		D14	–	reserved	–	–	–	0 when being read.
		D13	HIZ	High impedance	1 Hi-Z 0 Output	0	R/W	
		D12	TP	Top plane	1 High 0 Low	0	R/W	
		D11	BB	Black to black	1 High 0 Low	0	R/W	
		D10	BW	Black to white	1 High 0 Low	0	R/W	
		D9	WB	White to black	1 High 0 Low	0	R/W	
		D8	WW	White to white	1 High 0 Low	0	R/W	
		D7–0	INTV[7:0]	Interval	0 to 255	0x0	R/W	

These registers are used to set the display waveform memory for wave mode.
The display waveform memory is capable of storing up to 32 timing sets (Timing set 0 to Timing set 31) that consist of 15 bits. The EPD_WAVE*x* register corresponds to Timing set *x*. Timing set 0 represents the initial state and its period generated immediately after a screen update trigger is issued. The changes in the waveforms should be programmed one by one in the subsequent timing sets (1 to n = Max. 31).

D15 EOW: End of Wave Bit

Specifies the end of waveform.
1 (R/W): End
0 (R/W): Continued (default)

The waveform generation is completed at the timing set in which EOW is set to 1. The pins are set into high-impedance state until the subsequent display update trigger is issued. The timing sets other than the last must be programmed with EOW set to 0.

D14 Reserved

D13 HIZ: High Impedance Bit

Sets the segment and back plane pins into high-impedance state.
1 (R/W): High-impedance
0 (R/W): High/low output (default)

When this bit is set to 1, the segment and back plane pins are placed into high-impedance state for the period specified in the timing set (BB/BW/WB/WW settings are ignored). When this bit is set to 0, the pins go to the level specified by BB/BW/WB/WW.

D12 TP: Top Plane Bit

Sets the waveform output from the top plane pins.
1 (R/W): High
0 (R/W): Low (default)

When this bit is set to 1, the top plane pins go to the high (V_{EPD}) level for the period specified in the timing set. When this bit is set to 0, the pins go to the low (V_{SS}) level.

D11 BB: Black to Black Bit

Sets the waveform output from the segment and back plane pins to maintain black display during update (black to black).
1 (R/W): High
0 (R/W): Low (default)

When this bit is set to 1, the segment and back plane pins go to the high (VEPD) level for the period specified in the timing set. When this bit is set to 0, the pins go to the low (VSS) level.

D10 BW: Black to White Bit

Sets the waveform output from the segment and back plane pins to switch black display to white during update.
1 (R/W): High
0 (R/W): Low (default)

When this bit is set to 1, the segment and back plane pins go to the high (VEPD) level for the period specified in the timing set. When this bit is set to 0, the pins go to the low (VSS) level.

D9 WB: White to Black Bit

Sets the waveform output from the segment and back plane pins to switch white display to black during update.
1 (R/W): High
0 (R/W): Low (default)

When this bit is set to 1, the segment and back plane pins go to the high (VEPD) level for the period specified in the timing set. When this bit is set to 0, the pins go to the low (VSS) level.

D8 WW: White to White Bit

Sets the waveform output from the segment and back plane pins to maintain white display during update (white to white).
1 (R/W): High
0 (R/W): Low (default)

When this bit is set to 1, the segment and back plane pins go to the high (VEPD) level for the period specified in the timing set. When this bit is set to 0, the pins go to the low (VSS) level.

D[7:0] INTV[7:0]: Interval Bits

Specifies the timing set period in a number of EPDTCLK clocks. (Default: 0x0)

Time [s] = (INTV[7:0] + 1)/EPDTCLK frequency

21 Sound Generator (SND)

21.1 SND Module Overview

The S1C17F57 includes a sound generator (SND) for generating a buzzer signal.
The main features of the SND module are outlined below.

- Provides buzzer inverted and non-inverted output pins to directly drive a piezoelectric buzzer.
- Programmable buzzer signal frequency (eight frequencies) and volume level (eight levels)
- Duty ratio controlled digital envelope function (attenuation time is selectable from four types.)
- One-shot output function (output time is selectable from four types.)

Figure 21.1.1 shows the SND configuration.

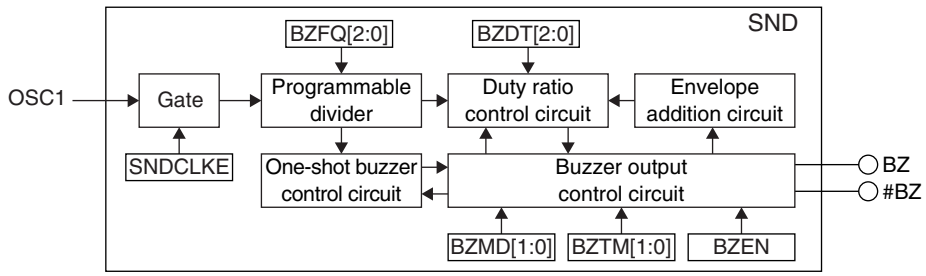


Figure 21.1.1 SND Module Configuration

21.2 SND Output Pins

Table 21.2.1 lists the SND pins.

Table 21.2.1 List of SND Pins

Pin name	I/O	Qty	Function
BZ	O	1	Buzzer non-inverted output pin Outputs the buzzer signal generated by the sound generator.
#BZ	O	1	Buzzer inverted output pin Outputs the inverted buzzer signal generated by the sound generator.

The SND module output pins (BZ, #BZ) are shared with I/O ports and are initially set as general purpose I/O port pins. The pin functions must be switched using the port function select bits to use the general purpose I/O port pins as SND module output pins. For detailed information on pin function switching, see the "I/O Ports (P)" chapter.

21.3 SND Operating Clock

The SND module uses the OSC1 clock (32.768 kHz Typ.) output from the CLG as its operating clock.
The OSC1 clock supply to the SND module is enabled with SNDCLKE/SND_CLK register. The SNDCLKE default setting is 0, which stops the clock. Setting SNDCLKE to 1 feeds the OSC1 clock to the SND module. Set SNDCLKE to 1 before performing buzzer output. If no buzzer output is required, stop the clock to reduce current consumption.
For more information on OSC1 oscillator control, see the "Clock Generator (CLG)" chapter.

Note: This chapter describes buzzer frequencies and one-shot output times assuming that the OSC1 clock frequency is 32.768 kHz. The frequencies and times vary depending on the OSC1 clock frequency.

21.4 Buzzer Frequency and Volume Settings

21.4.1 Buzzer Frequency

The SND module generates the buzzer signal by dividing the OSC1 clock (32.768 kHz). The buzzer frequency can be selected from among the eight types with different division ratios. BZFQ[2:0]/SND_BZFQ is used for this selection.

Table 21.4.1.1 Buzzer Frequency Selections

BZFQ[2:0]	Buzzer frequency (Hz)
0x7	1170.3
0x6	1365.3
0x5	1638.4
0x4	2048.0
0x3	2340.6
0x2	2730.7
0x1	3276.8
0x0	4096.0

(Default: 0x0)

21.4.2 Volume level

The buzzer volume level is controlled by changing the duty ratio of the buzzer signal.
The volume level can be selected from among eight types using BZDT[2:0]/SND_BZDT register.

Table 21.4.2.1 Volume Level Settings

Volume level	BZDT[2:0]	Duty ratio by buzzer frequency (Hz)			
		4096.0	3276.8	2730.7	2340.6
		2048.0	1638.4	1365.3	1170.3
Level 1 (Max.)	0x0	8/16	8/20	12/24	12/28
Level 2	0x1	7/16	7/20	11/24	11/28
Level 3	0x2	6/16	6/20	10/24	10/28
Level 4	0x3	5/16	5/20	9/24	9/28
Level 5	0x4	4/16	4/20	8/24	8/28
Level 6	0x5	3/16	3/20	7/24	7/28
Level 7	0x6	2/16	2/20	6/24	6/28
Level 8 (Min.)	0x7	1/16	1/20	5/24	5/28

(Default: 0x0)

Setting BZDT[2:0] to 0x0 turns the volume up to maximum level; setting it to 0x7 turns the volume down to minimum level.

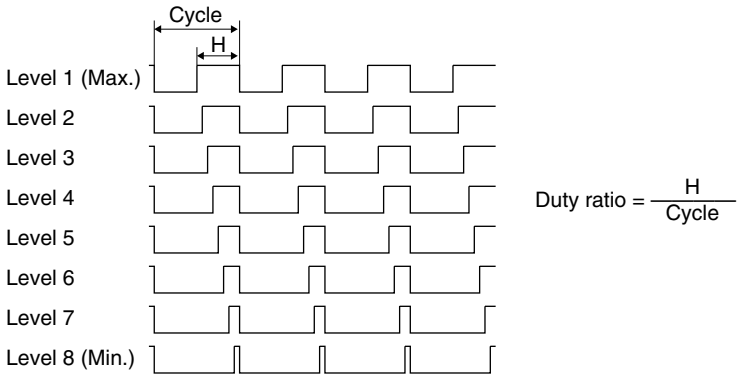


Figure 21.4.2.1 Buzzer Signal Waveforms by Different Duty Ratios

Note: BZDT[2:0] is ineffective in envelope mode, as the duty ratio is automatically controlled by the hardware.

21.5 Buzzer Mode and Output Control

21.5.1 Buzzer Mode Selection

The SND module supports three buzzer modes that allow different types of buzzer outputs.
BZMD[1:0]/SND_CTL register is used to select a buzzer mode.

Table 21.5.1.1 Buzzer Mode

BZMD[1:0]	Buzzer mode
0x3	Reserved
0x2	Envelope mode A software trigger starts buzzer output. The SND module automatically turns down the volume from Level 1 (maximum) and stops output when the volume reaches Level 8 (minimum).
0x1	One-shot mode This mode is provided for generating short buzzer sounds such as key operation sounds. The buzzer output starts by a software trigger and stops automatically after the specified time has elapsed.
0x0	Normal mode Buzzer output is turned on and off via software.

(Default: 0x0)

21.5.2 Output Control in Normal Mode

In normal mode, setting BZEN/SND_CTL register to 1 starts buzzer output and setting it to 0 stops the output. The buzzer frequency setting with BZFQ[2:0] and volume setting with BZDT[2:0] are both effective.

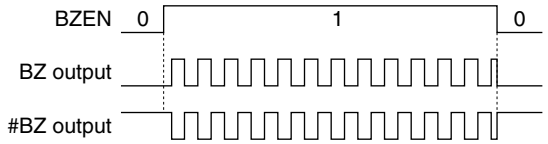


Figure 21.5.2.1 Buzzer Output in Normal Mode

Note: The buzzer signal is generated asynchronously to BZEN, so a hazard may occur when the signal is turned on or off by setting BZEN.

21.5.3 Output Control in One-shot Mode

The SND module has a one-shot output function for generating short buzzer sounds such as key operation sounds.

Output time selection

The one-shot buzzer output time can be selected from among four types shown below using BZTM[1:0]/SND_CTL register.

Table 21.5.3.1 One-shot Buzzer Output Time Selections

BZTM[1:0]	Output time
0x3	125 ms
0x2	62.5 ms
0x1	31.25 ms
0x0	15.63 ms

(Default: 0x0)

Output control

Writing 1 to BZEN/SND_CTL register starts one-shot buzzer output. When this trigger is issued, a buzzer signal is output from the buzzer output pin. When the set time has elapsed, the buzzer output stops.
BZEN functions as a status bit. It retains 1 while a one-shot buzzer signal is being output and reverts to 0 upon completion of the output.

Writing 0 to BZEN while a one-shot buzzer signal is being output stops the output immediately.
Writing 1 to BZEN again before a one-shot buzzer output is finished, a new one-shot output begins from that point.
The buzzer frequency setting with BZFQ[2:0] and volume setting with BZDT[2:0] are both effective in one-shot mode.
Figure 21.5.3.1 shows a timing chart in one-shot mode.

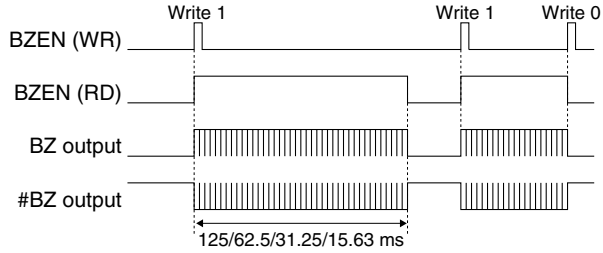


Figure 21.5.3.1 Buzzer Output in One-shot Mode

21.5.4 Output Control in Envelope Mode

In envelope mode, a digital envelope by duty control can be added to the buzzer signal. The SND module controls envelope by changing the duty ratio from Level 1 (maximum) to Level 8 (minimum) listed in Table 21.4.2.1.

Attenuation time selection

The envelope attenuation time (time to change the duty ratio) can be selected from among four types using BZTM[1:0]/SND_CTL register.

Table 21.5.4.1 Envelope Attenuation Time Selections

BZTM[1:0]	Attenuation time
0x3	125 ms
0x2	62.5 ms
0x1	31.25 ms
0x0	15.63 ms

(Default: 0x0)

Output control

Writing 1 to BZEN/SND_CTL register starts buzzer output in envelope mode. The duty ratio is set to Level 1 (maximum) at the beginning of the output and is stepped down every attenuation time selected. When attenuated down to Level 8 (minimum), the buzzer output stops.
BZEN functions as a status bit. It retains 1 while a buzzer signal is being output and reverts to 0 upon completion of the output.
Writing 0 to BZEN while a buzzer signal is being output stops the output immediately.
Writing 1 to BZEN again before a buzzer output is finished, the duty ratio returns to the maximum level and a new envelope output begins from that point.
Figure 21.5.4.1 shows a timing chart in envelope mode.

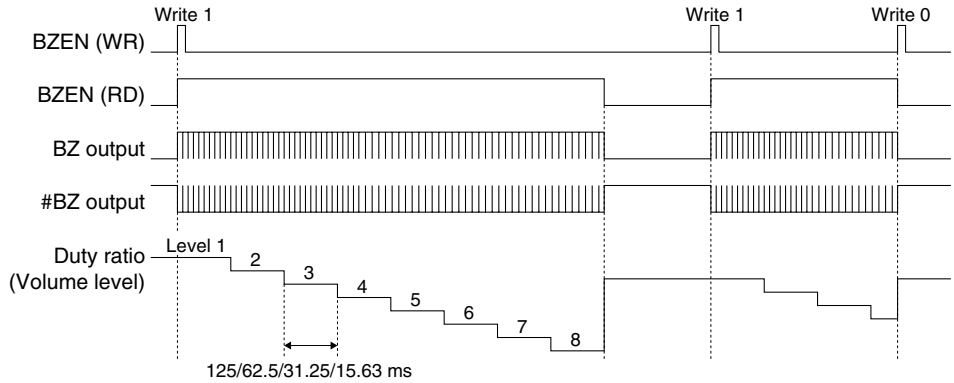


Figure 21.5.4.1 Buzzer Output in Envelope Mode

21.6 Control Register Details

Table 21.6.1 List of SND Registers

Address	Register name		Function
0x506e	SND_CLK	SND Clock Control Register	Controls the SND clock.
0x5180	SND_CTL	SND Control Register	Controls buzzer outputs.
0x5181	SND_BZFQ	Buzzer Frequency Control Register	Sets the buzzer frequency.
0x5182	SND_BZDT	Buzzer Duty Ratio Control Register	Sets the buzzer signal duty ratio.

The SND module registers are described in detail below.

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

SND Clock Control Register (SND_CLK)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SND Clock Control Register (SND_CLK)	**0x506e** (8 bits)	D7–1	**–**	reserved	–	–	–	0 when being read.
		D0	**SNDCLKE**	SND clock enable	1 Enable 0 Disable	0	R/W	

D[7:1] Reserved

D0 SNDCLKE: SND Clock Enable Bit

Enables or disables the OSC1 clock supply to the SND module.

1 (R/W): Enabled (on)

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

The SNDCLKE default setting is 0, which disables the clock supply. Setting SNDCLKE to 1 sends the OSC1 clock to the SND module to enable buzzer outputs. If no buzzer output is required, stop the clock to reduce current consumption.

SND Control Register (SND_CTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SND Control Register (SND_CTL)	**0x5180** (8 bits)	D7–6	**–**	reserved	–	–	–	0 when being read.
		D5–4	**BZTM[1:0]**	Buzzer envelope time/one-shot output time select	BZTM[1:0] Time 0x3 125 ms 0x2 62.5 ms 0x1 31.25 ms 0x0 15.63 ms	0x0	R/W	
		D3–2	**BZMD[1:0]**	Buzzer mode select	BZMD[1:0] Mode 0x3 reserved 0x2 Envelope 0x1 One-shot 0x0 Normal	0x0	R/W	
		D1	**–**	reserved	–	–	–	0 when being read.
		D0	**BZEN**	Buzzer output control	1 On/Trigger 0 Off	0	R/W	

D[7:6] Reserved

D[5:4] BZTM[1:0]: Buzzer Envelope Time/One-shot Output Time Select Bits

Selects an envelope attenuation time or a one-shot output time.

Table 21.6.2 Envelope Attenuation Time/One-shot Buzzer Output Time Selections

BZTM[1:0]	Attenuation time/One-shot output time
0x3	125 ms
0x2	62.5 ms
0x1	31.25 ms
0x0	15.63 ms

(Default: 0x0)

In envelope mode, an attenuation time (time to change the duty ratio) can be selected (see Figure 21.5.4.1).
In one-shot mode, a one-shot buzzer output time can be selected (see Figure 21.5.3.1).
BZTM[1:0] does not affect buzzer outputs in normal mode.

D[3:2] BZMD[1:0]: Buzzer Mode Select Bits

Selects a buzzer mode.

Table 21.6.3 Buzzer Mode

BZMD[1:0]	Buzzer mode
0x3	Reserved
0x2	Envelope mode A software trigger starts buzzer output. The SND module automatically turns down the volume from Level 1 (maximum) and stops output when the volume reaches Level 8 (minimum).
0x1	One-shot mode This mode is provided for generating short buzzer sounds such as key operation sounds. The buzzer output starts by a software trigger and stops automatically after the specified time has elapsed.
0x0	Normal mode Buzzer output is turned on and off via software.

(Default: 0x0)

D1 Reserved

D0 BZEN: Buzzer Output Control Bit

Controls buzzer output.
1 (R/W): On/Trigger
0 (R/W): Off (default)

Normal mode

Setting BZEN to 1 starts buzzer output and setting it to 0 stops the output.

One-shot mode

Writing 1 to BZEN starts one-shot buzzer output. When the time set with BZTM[1:0] has elapsed, the buzzer output stops. BZEN functions as a status bit. It retains 1 while a one-shot buzzer signal is being output and reverts to 0 upon completion of the output. Writing 0 to BZEN while a one-shot buzzer signal is being output stops the output immediately. Writing 1 to BZEN again before a one-shot buzzer output is finished, a new one-shot output begins from that point.

Envelope mode

Writing 1 to BZEN starts buzzer output in envelope mode. The duty ratio is set to Level 1 (maximum) at the beginning of the output and is stepped down every attenuation time selected. When attenuated down to Level 8 (minimum), the buzzer output stops. BZEN functions as a status bit. It retains 1 while a buzzer signal is being output and reverts to 0 upon completion of the output. Writing 0 to BZEN while a buzzer signal is being output stops the output immediately. Writing 1 to BZEN again before a buzzer output is finished, the duty ratio returns to the maximum level and a new envelope output begins from that point.

Buzzer Frequency Control Register (SND_BZFQ)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
Buzzer Frequency Control Register (SND_BZFQ)	0x5181 (8 bits)	D7–3	–	reserved	–		–	–	0 when being read.
		D2–0	BZFQ[2:0]	Buzzer frequency select	BZFQ[2:0]	Frequency	0x0	R/W	
					0x7	1170.3 Hz			
					0x6	1365.3 Hz			
					0x5	1638.4 Hz			
					0x4	2048.0 Hz			
					0x3	2340.6 Hz			
					0x2	2730.7 Hz			
					0x1	3276.8 Hz			
					0x0	4096.0 Hz			

D[7:3] Reserved

D[2:0] BZFQ[2:0]: Buzzer Frequency Select Bits

Selects a buzzer signal frequency.

Table 21.6.4 Buzzer Frequency Selections

BZFQ[2:0]	Buzzer frequency (Hz)
0x7	1170.3
0x6	1365.3
0x5	1638.4
0x4	2048.0
0x3	2340.6
0x2	2730.7
0x1	3276.8
0x0	4096.0

(Default: 0x0)

Buzzer Duty Ratio Control Register (SND_BZDT)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
Buzzer Duty Ratio Control Register (SND_BZDT)	0x5182 (8 bits)	D7–3	–	reserved	–		–	–	0 when being read.
		D2–0	BZDT[2:0]	Buzzer duty ratio select	BZDT[2:0]	Duty (volume)	0x0	R/W	
					0x7	Level 8 (Min.)			
					:	:			
					0x0	Level 1 (Max.)			

D[7:3] Reserved

D[2:0] BZDT[2:0]: Buzzer Duty Ratio Select Bits

Selects a duty ratio that determines the buzzer volume level.

Table 21.6.5 Volume Level Settings

Volume level	BZDT[2:0]	Duty ratio by buzzer frequency (Hz)			
		4096.0	3276.8	2730.7	2340.6
		2048.0	1638.4	1365.3	1170.3
Level 1 (Max.)	0x0	8/16	8/20	12/24	12/28
Level 2	0x1	7/16	7/20	11/24	11/28
Level 3	0x2	6/16	6/20	10/24	10/28
Level 4	0x3	5/16	5/20	9/24	9/28
Level 5	0x4	4/16	4/20	8/24	8/28
Level 6	0x5	3/16	3/20	7/24	7/28
Level 7	0x6	2/16	2/20	6/24	6/28
Level 8 (Min.)	0x7	1/16	1/20	5/24	5/28

(Default: 0x0)

Setting BZDT[2:0] to 0x0 turns the volume up to maximum level; setting it to 0x7 turns the volume down to minimum level.

Note: BZDT[2:0] is ineffective in envelope mode, as the duty ratio is automatically controlled by the hardware.

22 R/F Converter (RFC)

22.1 RFC Module Overview

The S1C17F57 includes an R/F converter (RFC) module with two conversion channels. It is capable of being used as a CR oscillation type A/D converter. A thermo-hygrometer can easily be implemented by connecting only resistive sensors (e.g., thermistor and humidity sensor) and a few passive elements (resistors and capacitors) to the R/F converter.
The following shows the features of the RFC module:

- Conversion method: Resistance to frequency conversion type
- Number of conversion channels: Max. 2 channels
- Oscillation mode: DC oscillation mode (for resistive sensors)
  AC oscillation mode (for resistive sensors)
- Counter length: 24 bits
- Five types of interrupts can be generated: Reference oscillation completion interrupt
  Sensor A oscillation completion interrupt
  Sensor B oscillation completion interrupt
  Measurement counter overflow error interrupt
  Time base counter overflow error interrupt

Figure 22.1.1 shows the RFC configuration.

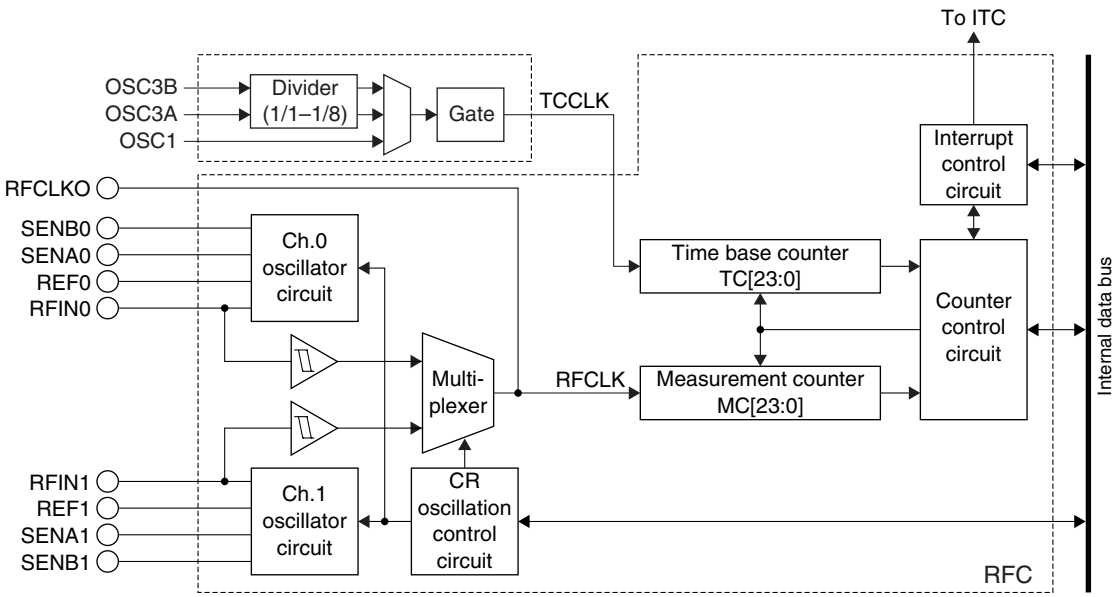


Figure 22.1.1 R/F Converter Configuration

The R/F converter converts the resistance of the sensor connected into frequency (RFCLK) using the embedded CR oscillator circuit, and counts this frequency using the measurement counter for a set period of time to provide the digital value equivalent to the sensor value. The time base counter is also included for generating the measurement time by counting an internal clock (TCCLK). In addition to CR oscillation using a sensor (sensor oscillation), the R/F converter performs CR oscillation using a reference element with less variation in the characteristics due to external factors (reference oscillation). This removes error factors such as voltage fluctuations and unevenness in quality to realize precise measurements. The CR oscillator circuit supports AC driving and external clock input as well as general DC driving, allowing use of various sensors.

22.2 RFC Input/Output Pins

Table 22.2.1 lists the RFC input/output pins.

Table 22.2.1 List of R/F Converter Input/Output Pins

Pin name	I/O	Qty	Function
SENB0/SENB1	I/O	2	Sensor B oscillation control pin (see Note 1 below)
SENA0/SENA1	I/O	2	Sensor A oscillation control pin (see Note 1 below)
REF0/REF1	I/O	2	Reference oscillation control pin (see Note 1 below)
RFIN0/RFIN1	I/O	2	RFCLK input and oscillation control pin (see Note 2 below)
RFCLKO	O	1	RFCLK monitoring output pin Outputs RFCLK to monitor the oscillation frequency.

Notes: 1. The pins go to high impedance status when the port function is switched for the R/F converter.
2. The RFIN*x* pin goes to Vss level when the port function is switched for the R/F converter. A large current may flow through the RFIN*x* pin if the pin is externally biased.

The R/F converter input/output pins are shared with I/O ports and are initially set as general-purpose I/O port pins. The pin functions must be switched using the port function select bits to use the general-purpose I/O port pins as R/F converter input/output pins.
For detailed information on pin function switching, see the "I/O Ports (P)" chapter.

22.3 Operation Clock

The RFC module includes a clock source selector, dividers, and a gate circuit for controlling the operation clock.

Note: The operation clock (TCCLK) must be enabled before setting the R/F converter. Otherwise, the R/F converter cannot operate normally.

Clock source selection

Use RFCCLKSRC[1:0]/RFC_CLK register to select the clock source from OSC3B, OSC3A, and OSC1.

Table 22.3.1 Clock Source Selection

RFCCLKSRC[1:0]	Clock source
0x3	Reserved
0x2	OSC3A
0x1	OSC1
0x0	OSC3B

(Default: 0x0)

Clock division ratio selection

When the clock source is OSC1

No division ratio needs to be selected when OSC1 is selected for the clock source. The OSC1 clock (typ. 32.768 kHz) is directly used as TCCLK.

When the clock source is OSC3B or OSC3A

When OSC3B or OSC3A is selected for the clock source, use RFCCLKD[1:0]/RFC_CLK register to select the division ratio.

Table 22.3.2 OSC3B/OSC3A Division Ratio Selection

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

(Default: 0x0)

The time base counter uses the clock selected here for counting. Selecting a high-speed clock improves the conversion accuracy. However the clock must be selected so that the time base counter will not overflow in the reference oscillation phase.

Clock enable

The clock supply is enabled with RFCCLKE/RFC_CLK register. The RFCCLKE default setting is 0, which stops the clock. Setting RFCCLKE to 1 feeds the clock generated as above to the RFC circuit. If no RFC operation is required, stop the clock to reduce current consumption.

22.4 Operating Modes

The R/F converter features two oscillation modes that use the RFC internal oscillator circuit and a mode for measuring an external input clock. Also it includes a CR oscillation clock (RFCLK) monitoring function and continuous oscillation function for measuring the oscillation clock frequency. Each channel can be set to a different mode.

22.4.1 Oscillation Mode

In measurements using the RFC internal oscillator circuit, operate the oscillator with the reference element and then the sensor for the same duration in time to count each oscillation frequency. The sensor value can be determined from the difference between the two count values by software. The R/F converter supports DC bias resistive sensors and AC bias resistive sensors. The RFC internal oscillator circuit can operate in two oscillation modes corresponding to the sensor to be used that is specified by SMODE[1:0]/RFC_CTL register.

Table 22.4.1.1 Oscillation Mode Selection

SMODE[1:0]	Oscillation mode
0x3–0x2	Reserved
0x1	AC oscillation mode for measuring resistive sensors
0x0	DC oscillation mode for measuring resistive sensors

(Default: 0x0)

DC oscillation mode for measuring resistive sensors (SMODE[1:0] = 0x0, default)

This mode drives the oscillator with the reference resistor and resistive sensor by applying DC bias voltage. Select this mode when a DC bias resistive sensor is connected. This mode enables two resistive sensors to be connected to a channel. One reference resistor and one reference capacitor is also required.

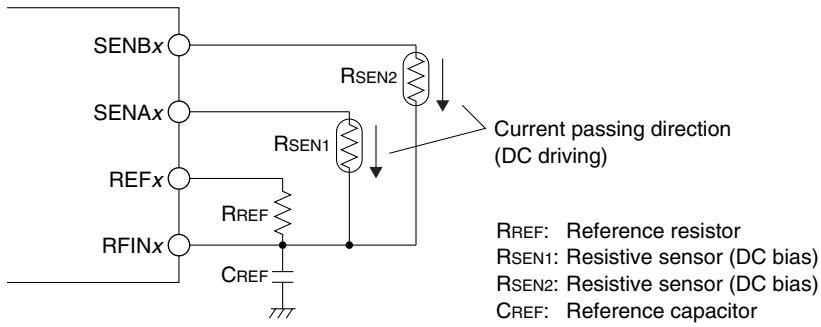


Figure 22.4.1.1 Connection Example in DC Oscillation Mode for Measuring Resistive Sensors

When one resistive sensor only is used, leave the unused pin open.

AC oscillation mode for measuring resistive sensors (SMODE[1:0] = 0x1)

This mode drives the oscillator with the reference resistor and resistive sensor by applying AC bias voltage. Select this mode when an AC bias resistive sensor is connected. This mode enables only one resistive sensor to be connected to a channel. One reference resistor and one reference capacitor is also required.

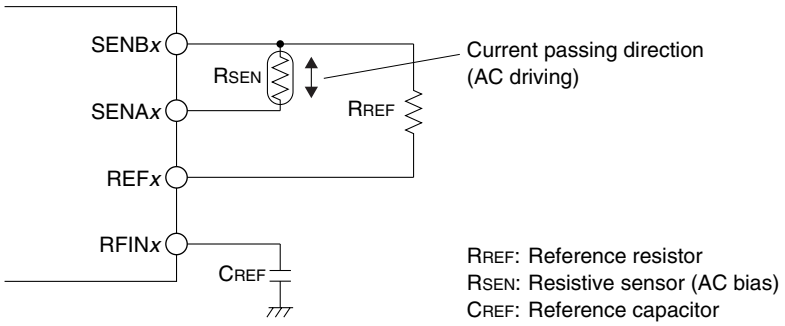


Figure 22.4.1.2 Connection Example in AC Oscillation Mode for Measuring Resistive Sensors

22.4.2 External Clock Input Mode (Event Counter Mode)

This mode enables to input clocks/pulses from an external circuit such as an oscillator and count them same as those of internal oscillation clocks. It supports rectangular waves, triangular waves, and sign waves to be input. (For the threshold voltage of the Schmitt input buffer, see "Electrical Characteristics.")
Setting EVTEN/RFC_CTL register to 1 enables this function. The measurement control procedure is the same as that when the internal oscillator circuit is used.

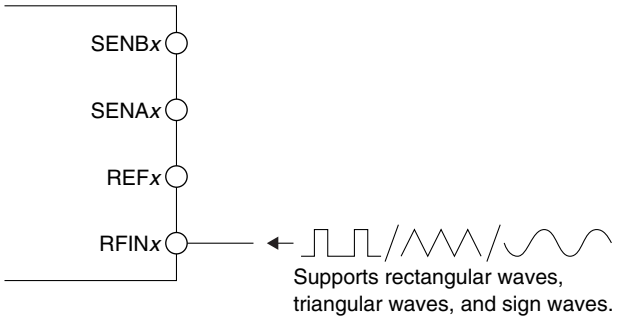


Figure 22.4.2.1 External Clock Input

The unused pins should be left open.

22.4.3 Functions for Measuring CR Oscillation Clock Frequency

CR Oscillation Clock (RFCLK) Monitoring Function

The CR oscillation clock (RFCLK) during converting can be output from the RFCLKO pin for monitoring. Use this output to measure the oscillation frequency.

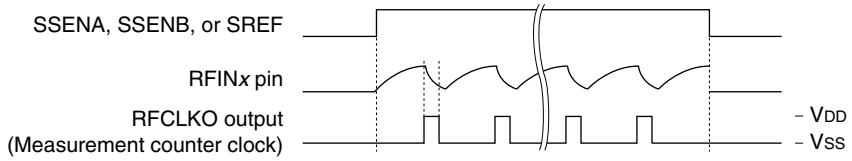


Figure 22.4.3.1 CR Oscillation Clock (RFCLK) Waveform

Continuous oscillation function

The CR oscillations by the sensor and reference element will automatically stop due to stop conditions. Setting 1 to CONEN/RFC_CTL register enables the continuous oscillation function and CR oscillation will continue until stopped by software. Using this function with the CR oscillation monitoring function helps easily measure the CR oscillation clock frequency.

22.5 RFC Counters

The R/F converter includes two kinds of counters: measurement counter (MC) counting the reference element and sensor oscillation clocks, and time base counter (TC) counting the TCCLK clock.

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 in 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 oscillation 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 by 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 0x0. 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 the 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 counters before starting reference oscillation and sensor oscillation.
Connecting the reference element and sensor with the same resistance will result <Initial value> = <Counter value at the end of sensor oscillation> (if no error introduced). Setting a small initial value to the measurement counter improves measurement accuracy. 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 cleared to 0x0 before starting reference oscillation.

22.6 Conversion Operations

The conversion operations by the R/F converter should be controlled in the following procedure regardless of the operating mode: initial settings, reference oscillation control, and sensor oscillation control. The R/F converter channels are controlled individually and both channels cannot operate simultaneously. This section describes these control procedure.
Although the following explanations assume that the internal oscillator circuit is used, the control procedures are the same even in external clock input mode. When the R/F converter is used in external clock input mode, select which oscillation is performed either reference or sensor and determine the counter initial values depending on the purpose for using.

22.6.1 Initial Settings

Clock and pin configurations

(1) Select the R/F converter operating clock (TCCLK) and enable the clock supply. (See Section 22.3.)

(2) Configure the pins to be used for the R/F converter by switching from general-purpose input/output ports. See the “I/O Ports (P)” chapter.

R/F converter channel and mode settings

(1) Set RFCEN/RFC_CTL register to 1 to enable the R/F converter.

(2) Select the channel to perform conversion using CHSEL/RFC_CTL register. Setting CHSEL to 0 (default) selects Ch. 0 and setting 1 selects Ch.1.

(3) Set the oscillation mode using SMODE[1:0]/RFC_CTL register. (See Section 22.4.1.)

22.6.2 Reference Oscillation Control

First, perform oscillation with the reference resistor/capacitor and obtain the time base counter value to perform sensor oscillation for the same period of time.

(1) Set the initial value (0x0 - n) to MC[23:0]/RFC_MC(H/L) registers (measurement counter). (See Section 22.5.)

(2) Set 0x0 to TC[23:0]/RFC_TC(H/L) registers (time base counter).

(3) Reset the cause-of-interrupt flags OVTCIF and EREFIF in the RFC_IFLG register by writing 1.

(4) Set SREF/RFC_TRG register to 1 to start reference oscillation.
The CR oscillator circuit starts oscillating with the reference resistor/capacitor and outputs the clock to the measurement counter. 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 0x0.

Note: For restrictions and precautions on control bit settings, see the descriptions in the "Control Register Details" section. The control bits may not be set to the desired values depending on conditions.

(5) When the measurement counter or the time base counter overflows (0xffffff → 0x0), SREF is reset to 0 and the reference oscillation stops automatically.

(6-1) The measurement counter overflow sets EREFIF to 1 indicating that the reference oscillation has been terminated normally. An interrupt can be generated at this point. Read the time base counter value (TC[23:0] = X) and store it to the memory by the interrupt handler routine. When this interrupt is not used, perform the same processing after checking if EREFIF has been set.

(6-2) The time base counter overflow sets OVTCIF to 1 indicating that the reference oscillation has been terminated abnormally. An interrupt can be generated at this point. Handle this error in the interrupt handler routine. When this interrupt is not used, perform the same processing after checking if OVTCIF has been set.

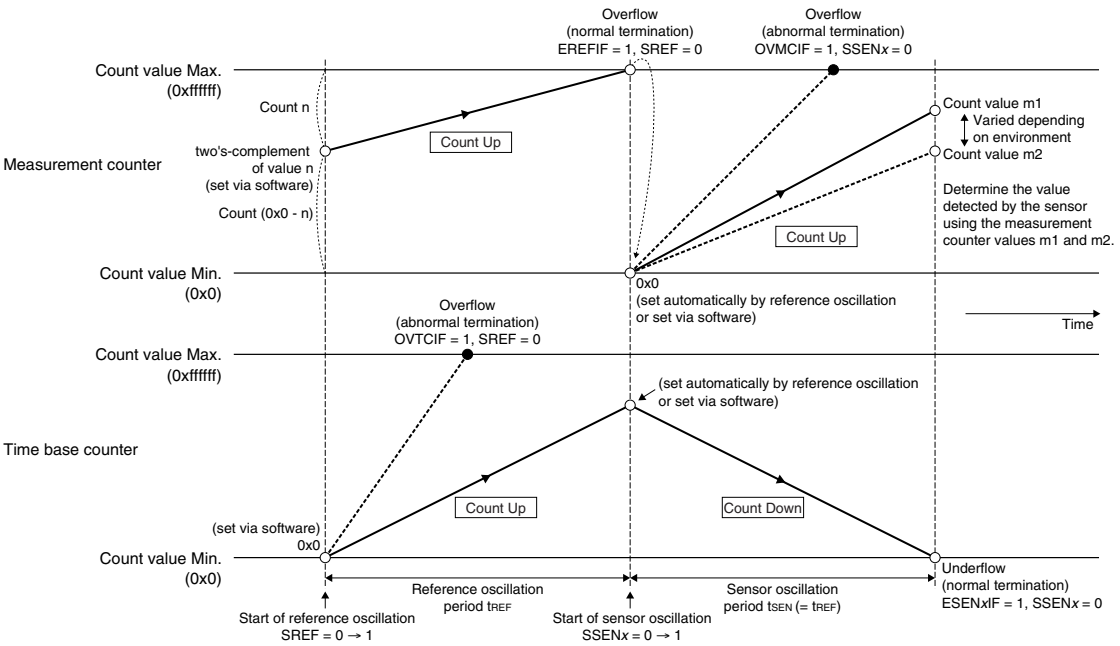


Figure 22.6.2.1 Counter Operations During Reference/Sensor Oscillation

22.6.3 Sensor Oscillation Control

Perform oscillation with the sensor for the period of time obtained by the time base counter in reference oscillation and count the oscillation clock by the measurement counter.

(1) Initialize MC[23:0] (measurement counter) by writing 0x0. This can be omitted after a reference oscillation has completed.

(2) Initialize TC[23:0] (time base counter) by writing the value (X) that has been counted in the time base counter during reference oscillation. This can be omitted after a reference oscillation has completed.

(3) Reset the cause-of-interrupt flags OVMCIF, ESENBIF, and ESENAIF in the RFC_IFLG register by writing 1.

(4) Set SSENA/RFC_TRG register (for sensor A) or SSENB/RFC_TRG register (for sensor B) to 1 to start sensor oscillation.
The CR oscillator circuit starts oscillating with the sensor and outputs the clock to the measurement counter. The measurement counter starts counting up using the CR oscillation clock from 0x0. The time base counter starts counting down using TCCLK from the initial value (X) that has been set.

Note: For restrictions and precautions on control bit settings, see the descriptions in the "Control Register Details" section. The control bits may not be set to the desired values depending on conditions.

(5) When the time base counter reaches 0x0 or the measurement counter overflows (0xffffff → 0x0), SSENA or SSENB is reset to 0 and the sensor oscillation stops automatically.

(6-1) The time base counter reached 0x0 sets ESENAIF (for sensor A) or ESENBIF (for sensor B) to 1 indicating that the sensor oscillation has been terminated normally. An interrupt can be generated at this point. Read the measurement counter value (MC[23:0] = m) and process the detection results by the interrupt handler routine. When this interrupt is not used, perform the same processing after checking if ESENAIF or ESENBIF has been set.

(6-2) The measurement counter overflow sets OVMCIF to 1 indicating that the sensor oscillation has been terminated abnormally. An interrupt can be generated at this point. Handle this error in the interrupt handler routine. When this interrupt is not used, perform the same processing after checking if OVMCIF has been set.

22.6.4 Forced Termination

To abort reference oscillation or sensor oscillation, write 0 to SREF (reference oscillation), SSENA (sensor A oscillation), or SSENB (sensor B oscillation) in the RFC_TRG register used to start the oscillation. The counters maintain the value at they stopped, note, however, that the conversion results cannot be guaranteed if the oscillation is resumed. When resuming oscillation, initialize the counters.

22.6.5 Conversion Error

Performing reference oscillation and sensor oscillation with the same resistor and capacitor results $n \approx m$. The difference between n and m is a conversion error. The conversion error may be introduced caused by temperature, voltage, and unevenness of IC quality, as well as external parts and on-board parasitic elements. For sample errors, see "Electrical Characteristics."

22.7 RFC Interrupts

The RFC module includes a function for generating the following five different types of interrupts.
- Reference oscillation completion interrupt
- Sensor A oscillation completion interrupt
- Sensor B oscillation completion interrupt
- Measurement counter overflow error interrupt
- Time base counter overflow error interrupt

The RFC module outputs one interrupt signal shared by the five above interrupt causes to the interrupt controller (ITC). Inspect the interrupt flag to determine the interrupt cause occurred.

Reference oscillation completion interrupt

To use this interrupt, set EREFIE/RFC_IMSK register to 1. If EREFIE is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.
When the measurement counter overflows and a reference oscillation is completed normally, the R/F converter sets EREFIF/RFC_IFLG register to 1. If reference oscillation completion interrupts are enabled (EREFIE = 1), an interrupt request is sent simultaneously to the ITC.

Sensor A oscillation completion interrupt

To use this interrupt, set ESENAIE/RFC_IMSK register to 1. If ESENAIE is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.
When the time base counter reaches 0x0 and a sensor A oscillation is completed normally, the R/F converter sets ESENAIF/RFC_IFLG register to 1. If sensor A oscillation completion interrupts are enabled (ESENAIE = 1), an interrupt request is sent simultaneously to the ITC.

Sensor B oscillation completion interrupt

To use this interrupt, set ESENBIE/RFC_IMSK register to 1. If ESENBIE is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.
When the time base counter reaches 0x0 and a sensor B oscillation is completed normally, the R/F converter sets ESENBIF/RFC_IFLG register to 1. If sensor B oscillation completion interrupts are enabled (ESENBIE = 1), an interrupt request is sent simultaneously to the ITC.

Measurement counter overflow error interrupt

To use this interrupt, set OVMCIE/RFC_IMSK register to 1. If OVMCIE is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.
When the measurement counter overflows and a sensor oscillation is terminated abnormally, the R/F converter sets OVMCIF/RFC_IFLG register to 1. If measurement counter overflow error interrupts are enabled (OVMCIE = 1), an interrupt request is sent simultaneously to the ITC.

Time base counter overflow error interrupt

To use this interrupt, set OVTCIE/RFC_IMSK register to 1. If OVTCIE is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.
When the time base counter overflows and a reference oscillation is terminated abnormally, the R/F converter sets OVTCIF/RFC_IFLG register to 1. If time base counter overflow error interrupts are enabled (OVTCIE = 1), an interrupt request is sent simultaneously to the ITC.

For more information on interrupt processing, see the "Interrupt Controller (ITC)" chapter.

Notes: • To prevent interrupt recurrences, the interrupt flag must be reset in the interrupt handler routine after an RFC interrupt has occurred. The interrupt flag is reset by writing 1.

• To prevent unwanted interrupts, reset the interrupt flags before enabling interrupts with the interrupt enable bits.

22.8 Control Register Details

Table 22.8.1 List of RFC Registers

Address	Register name		Function
0x5067	RFC_CLK	RFC Clock Control Register	Selects the operating clock.
0x53a0	RFC_CTL	RFC Control Register	Controls R/F converter.
0x53a2	RFC_TRG	RFC Oscillation Trigger Register	Controls oscillations.
0x53a4	RFC_MCL	RFC Measurement Counter Low Register	Measurement counter data
0x53a6	RFC_MCH	RFC Measurement Counter High Register	
0x53a8	RFC_TCL	RFC Time Base Counter Low Register	Time base counter data
0x53aa	RFC_TCH	RFC Time Base Counter High Register	
0x53ac	RFC_IMSK	RFC Interrupt Mask Register	Enables/disables interrupts.
0x53ae	RFC_IFLG	RFC Interrupt Flag Register	Indicates/resets interrupt occurrence status.

The R/F converter registers are described in detail below.

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

RFC Clock Control Registers (RFC_CLK)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
RFC Clock Control Register (RFC_CLK)	**0x5067** (8 bits)	D7–6	**–**	reserved	–		–	–	0 when being read.
		D5–4	**RFCCLKD [1:0]**	RFC clock division ratio select	RFCCLKD[1:0]	Division ratio	0x0	R/W	When the clock source is OSC3B or OSC3A
					0x3	1/8			
					0x2	1/4			
					0x1	1/2			
					0x0	1/1			
		D3–2	**RFCCLK SRC[1:0]**	RFC clock source select	RFCCLK SRC[1:0]	Clock source	0x0	R/W	
					0x3	reserved			
					0x2	OSC3A			
					0x1	OSC1			
					0x0	OSC3B			
		D1	**–**	reserved	–		–	–	0 when being read.
		D0	**RFCCLKE**	RFC clock enable	1 Enable	0 Disable	0	R/W	

D[7:6] Reserved

D[5:4] RFCCLKD[1:0]: RFC Clock Division Ratio Select Bits

Selects the division ratio for generating the TCCLK clock when OSC3B or OSC3A is used as the clock source.

Table 22.8.2 OSC3B/OSC3A Division Ratio Selection

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

(Default: 0x0)

D[3:2] RFCCLKSRC[1:0]: RFC Clock Source Select Bits

Selects the count clock source.

Table 22.8.3 RFC Clock Source Selection

RFCCLKSRC[1:0]	Clock source
0x3	Reserved
0x2	OSC3A
0x1	OSC1
0x0	OSC3B

(Default: 0x0)

D1 Reserved

D0 RFCCLKE: RFC Clock Enable Bit

Enables or disables the TCCLK clock supply.

1 (R/W): Enabled (on)

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

The RFCCLKE default setting is 0, which disables the clock supply. Setting RFCCLKE to 1 sends the clock selected to the R/F converter.

RFC Control Register (RFC_CTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RFC Control Register (RFC_CTL)	0x53a0 (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7	CONEN	Continuous oscillation enable	1 Enable 0 Disable	0	R/W	
		D6	EVTEN	Event counter mode enable	1 Enable 0 Disable	0	R/W	
		D5–4	SMODE[1:0]	Sensor oscillation mode select	SMODE[1:0] Sensor 0x3–0x2 reserved 0x1 AC resistive 0x0 DC resistive	0x0	R/W	
		D3–2	–	reserved	–	–	–	0 when being read.
		D1	CHSEL	Conversion channel select	1 Ch.1 0 Ch.0	0	R/W	
		D0	RFCEN	RFC enable	1 Enable 0 Disable	0	R/W	

D[15:8] Reserved

D7 CONEN: Continuous Oscillation Enable Bit

Enables continuous oscillation by disabling the automatic CR oscillation stop function.
1 (R/W): Continuous oscillation enabled
0 (R/W): Continuous oscillation disabled (default)

Setting 1 to CONEN disables the reference oscillation/sensor oscillation stop conditions so that the CR oscillator will continue oscillating. Set SREF (reference oscillation), SSENA (sensor A oscillation), or SSENB (sensor B oscillation) in the RFC_TRG register to 1 to start oscillation even in this mode, and set to 0 to stop oscillation.
Using this function with the CR oscillation monitoring function helps easily measure the CR oscillation clock frequency.

D6 EVTEN: Event Counter Mode Enable Bit

Enables external clock input mode (event counter mode).
1 (R/W): External clock input mode
0 (R/W): Normal mode (default)

Setting EVTEN to 1 enables the external clock input to the RFIN*x* pin. SREF (reference oscillation), SSENA (sensor A oscillation), or SSENB (sensor B oscillation) should be used to control starting oscillation (starting conversion) to perform converting operation even in this mode.

Note: Do not input an external clock before setting EVTEN to 1. The RFIN*x* pin is pulled down to Vss when the pin function is switched for the R/F converter.

D[5:4] SMODE[1:0]: Sensor Oscillation Mode Select Bits

Selects an oscillation mode.

Table 22.8.4 Oscillation Mode Selection

SMODE[1:0]	Oscillation mode
0x3–0x2	Reserved
0x1	AC oscillation mode for measuring resistive sensors
0x0	DC oscillation mode for measuring resistive sensors

(Default: 0x0)

For more information on the oscillation mode, see Section 22.4.1.

D[3:2] Reserved

D1 CHSEL: Conversion Channel Select Bit

Selects the channel to perform conversion.
1 (R/W): Ch.1
0 (R/W): Ch.0 (default)

The D[7:4] settings in this register and oscillation control using the RFC_TRG register are effective only for the channel specified by CHSEL.

D0 RFCEN: RFC Enable Bit

Enables or disables the R/F converter.

1 (R/W): Enabled

0 (R/W): Disabled (default)

Setting RFCEN to 1 enables the R/F converter to start converting operations. When RFCEN is 0, manipulations of the RFC_TRG register for oscillation control are ineffective.

RFC Oscillation Trigger Register (RFC_TRG)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
RFC Oscillation Trigger Register (RFC_TRG)	0x53a2 (16 bits)	D15–3	–	reserved	–				–	–	0 when being read.
		D2	SSENB	Sensor B oscillation control/status	1	Start/Run	0	Stop	0	R/W	
		D1	SSENA	Sensor A oscillation control/status	1	Start/Run	0	Stop	0	R/W	
		D0	SREF	Reference oscillation control/status	1	Start/Run	0	Stop	0	R/W	

D[15:3] Reserved

D2 SSENB: Sensor B Oscillation Control/Status 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 (default)

Sensor B cannot be used in AC oscillation mode for resistive sensors.

D1 SSENA: Sensor A Oscillation Control/Status 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 (default)

D0 SREF: Reference Oscillation Control/Status Bit

Controls CR oscillation for the reference element. This bit also indicates the CR oscillation status.

1 (W): Start oscillation

0 (W): Stop oscillation

1 (R): Being oscillated

0 (R): Stopped (default)

Notes:
- SREF, SSENA, and SSENB are all ineffective when RFCEN/RFC_CTL register is 0 (converting operation disabled).
- Writing 1 to SSENB does not start oscillation when SMODE[1:0]/RFC_CTL register is 0x1 (AC oscillation mode for resistive sensors).
- When writing 1 to SREF, SSENA, or SSENB to start oscillation, be sure to avoid that more than one bit are set to 1.
- Be sure to reset the interrupt flags in the RFC_IFLG register (EREFIF, ESENAIF, ESENBIF, OVMCIF, and OVTCIF) before starting oscillation using SREF, SSENA, and SSENB.

RFC Measurement Counter Low and High Registers (RFC_MCL, RFC_MCH)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RFC Measurement Counter Low Register (RFC_MCL)	**0x53a4** (16 bits)	D15–0	**MC[15:0]**	Measurement counter low-order 16-bit data	0x0–0xffff	0x0	R/W	
RFC Measurement Counter High Register (RFC_MCH)	**0x53a6** (16 bits)	D15–8	**–**	reserved	–	–	–	0 when being read.
		D7–0	**MC[23:16]**	Measurement counter high-order 8-bit data	0x0–0xff	0x0	R/W	

D[7:0]/RFC_MCH, D[15:0]/RFC_MCL

MC[23:0]: Measurement Counter Bits

Measurement counter data can be read and written to. (Default: 0x0)

Note: The measurement counter must be set from the low-order value (MC[15:0]/RFC_MCL register) first. The counter may not be set to the correct value if the high-order value (MC[23:16]/RFC_MCH register) is written first.

RFC Time Base Counter Low and High Registers (RFC_TCL, RFC_TCH)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RFC Time Base Counter Low Register (RFC_TCL)	**0x53a8** (16 bits)	D15–0	**TC[15:0]**	Time base counter low-order 16-bit data	0x0–0xffff	0x0	R/W	
RFC Time Base Counter High Register (RFC_TCH)	**0x53aa** (16 bits)	D15–8	**–**	reserved	–	–	–	0 when being read.
		D7–0	**TC[23:16]**	Time base counter high-order 8-bit data	0x0–0xff	0x0	R/W	

D[7:0]/RFC_TCH, D[15:0]/RFC_TCL

TC[23:0]: Time Base Counter Bits

Time base counter data can be read and written to. (Default: 0x0)

Note: The time base counter must be set from the low-order value (TC[15:0]/RFC_TCL register) first. The counter may not be set to the correct value if the high-order value (TC[23:16]/RFC_TCH register) is written first.

RFC Interrupt Mask Register (RFC_IMSK)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
RFC Interrupt Mask Register (RFC_IMSK)	**0x53ac** (16 bits)	D15–5	**–**	reserved	–				–	–	0 when being read.
		D4	**OVTCIE**	TC overflow error interrupt enable	1	Enable	0	Disable	0	R/W	
		D3	**OVMCIE**	MC overflow error interrupt enable	1	Enable	0	Disable	0	R/W	
		D2	**ESENBIE**	Sensor B oscillation completion interrupt enable	1	Enable	0	Disable	0	R/W	
		D1	**ESENAIE**	Sensor A oscillation completion interrupt enable	1	Enable	0	Disable	0	R/W	
		D0	**EREFIE**	Reference oscillation completion interrupt enable	1	Enable	0	Disable	0	R/W	

D[15:5] Reserved

D4 OVTCIE: TC Overflow Error Interrupt Enable Bit

Enables or disables time base counter overflow error interrupts.

1 (R/W): Interrupt enabled

0 (R/W): Interrupt disabled (default)

D3 OVMCIE: MC Overflow Error Interrupt Enable Bit

Enables or disables measurement counter overflow error interrupts.

1 (R/W): Interrupt enabled

0 (R/W): Interrupt disabled (default)

D2 ESENBIE: Sensor B Oscillation Completion Interrupt Enable Bit

Enables or disables sensor B oscillation completion interrupts.

1 (R/W): Interrupt enabled

0 (R/W): Interrupt disabled (default)

D1 ESENAIE: Sensor A Oscillation Completion Interrupt Enable Bit

Enables or disables sensor A oscillation completion interrupts.

1 (R/W): Interrupt enabled

0 (R/W): Interrupt disabled (default)

D0 EREFIE: Reference Oscillation Completion Interrupt Enable Bit

Enables or disables reference oscillation completion interrupts.

1 (R/W): Interrupt enabled

0 (R/W): Interrupt disabled (default)

RFC Interrupt Flag Register (RFC_IFLG)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
RFC Interrupt Flag Register (RFC_IFLG)	0x53ae (16 bits)	D15–5	–	reserved	–				–	–	0 when being read.
		D4	OVTCIF	TC overflow error interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D3	OVMCIF	MC overflow error interrupt flag					0	R/W	
		D2	ESENBIF	Sensor B oscillation completion interrupt flag					0	R/W	
		D1	ESENAIF	Sensor A oscillation completion interrupt flag					0	R/W	
		D0	EREFIF	Reference oscillation completion interrupt flag					0	R/W	

D[15:5] Reserved

D4 OVTCIF: TC Overflow Error Interrupt Flag Bit

Indicates the time base counter overflow error interrupt cause occurrence status.

1 (R): Cause of interrupt has occurred

0 (R): No cause of interrupt has occurred (default)

1 (W): Flag is reset

0 (W): Ignored

OVTCIF is set to 1 when a reference oscillation is terminated abnormally due to time base counter overflow. OVTCIF is reset to 0 by writing 1.

D3 OVMCIF: MC Overflow Error Interrupt Flag Bit

Indicates the measurement counter overflow error interrupt cause occurrence status.

1 (R): Cause of interrupt has occurred

0 (R): No cause of interrupt has occurred (default)

1 (W): Flag is reset

0 (W): Ignored

OVMCIF is set to 1 when a sensor oscillation is terminated abnormally due to measurement counter overflow. OVMCIF is reset to 0 by writing 1.

D2 ESENBIF: Sensor B Oscillation Completion Interrupt Flag Bit

Indicates the sensor B oscillation completion interrupt cause occurrence status.

1 (R): Cause of interrupt has occurred

0 (R): No cause of interrupt has occurred (default)

1 (W): Flag is reset

0 (W): Ignored

ESENBIF is set to 1 when the time base counter reaches 0x0 and a sensor B oscillation is completed normally. ESENBIF is reset to 0 by writing 1.

D1 ESENAIF: Sensor A Oscillation Completion Interrupt Flag Bit

Indicates the sensor A oscillation completion interrupt cause occurrence status.

1 (R): Cause of interrupt has occurred
0 (R): No cause of interrupt has occurred (default)
1 (W): Flag is reset
0 (W): Ignored

ESENAIF is set to 1 when the time base counter reaches 0x0 and a sensor A oscillation is completed normally. ESENAIF is reset to 0 by writing 1.

D0 EREFIF: Reference Oscillation Completion Interrupt Flag Bit

Indicates the reference oscillation completion interrupt cause occurrence status.

1 (R): Cause of interrupt has occurred
0 (R): No cause of interrupt has occurred (default)
1 (W): Flag is reset
0 (W): Ignored

EREFIF is set to 1 when the measurement counter overflows and a reference oscillation is completed normally. EREFIF is reset to 0 by writing 1.

23 Temperature Detection Circuit (TEM)

23.1 TEM Module Overview

The S1C17F57 includes a temperature detection circuit to detect chip temperature. The detected temperature can be read via software.
The following shows the features of the TEM module:

- Temperature sensor detectable temperature range: -40°C to 85°C
- Sensor output voltage to digital value (8 bits) conversion circuit using a linear search method
- Conversion time (comparison time) adjustment function
  * TEM compares the sensor detection voltage with the reference voltage.
- Can generate conversion completion interrupts.

Figure 23.1.1 shows the TEM module configuration.

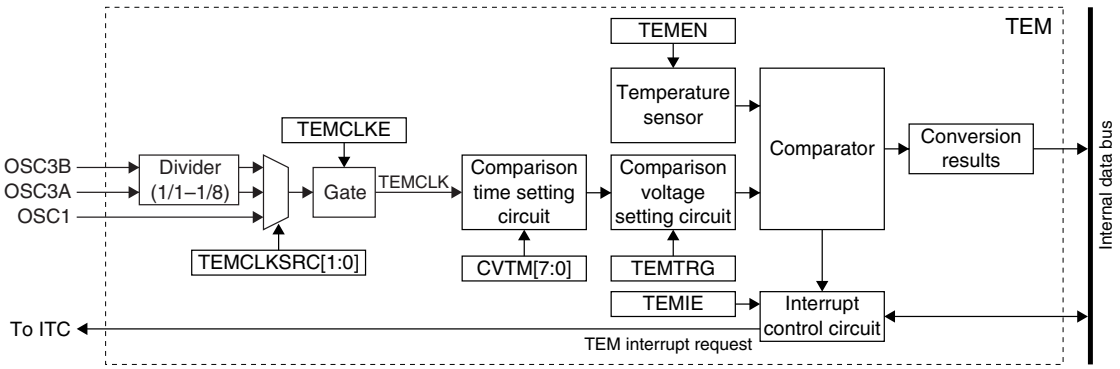


Figure 23.1.1 TEM Module Configuration

23.2 Operation Clock

The TEM module includes a clock source selector, dividers, and a gate circuit for controlling the operation clock (TEMCLK).

Clock source selection

Use TEMCLKSRC[1:0]/TEM_CLK register to select the clock source from OSC3B, OSC3A, and OSC1.

Table 23.2.1 Clock Source Selection

TEMCLKSRC[1:0]	Clock source
0x3	Reserved
0x2	OSC3A
0x1	OSC1
0x0	OSC3B

(Default: 0x0)

Clock division ratio selection

When the clock source is OSC1

No division ratio needs to be selected when OSC1 is selected for the clock source. The OSC1 clock (typ. 32.768 kHz) is directly used as TEMCLK.

When the clock source is OSC3B or OSC3A

When OSC3B or OSC3A is selected for the clock source, use TEMCLKD[1:0]/TEM_CLK register to select the division ratio.

Table 23.2.2 OSC3B/OSC3A Division Ratio Selection

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

(Default: 0x0)

Clock enable

The clock supply is enabled with TEMCLKE/TEM_CLK register. The TEMCLKE default setting is 0, which stops the clock. Setting TEMCLKE to 1 feeds the clock generated as above to the TEM circuit. If no TEM operation is required, stop the clock to reduce current consumption.

23.3 TEM Control

The TEM module detects temperature by comparing the temperature sensor output (that varies according to temperature) and a comparison voltage generated from an 8-bit digital value (0x0 to 0xff). The comparison is started from the comparison voltage equivalent to an 8-bit value 0, then it is incremented in the specified comparison cycles (maximum 256 steps). When the sensor output and a comparison voltage are matched, the 8-bit comparison voltage value is loaded to a register as the conversion result and the TEM module stops temperature conversion. An interrupt can be generated at this time, so the detection results can be read in the interrupt handler routine.

Comparison time setting

Set the time for comparing each comparison voltage by the comparator to CVTM[7:0]/TEM_TIME register. Be sure to set a 150 µs or more comparison time including clock frequency dispersion.

$$\text{Comparison time} = \frac{\text{CVTM} + 1}{f_{TEMCLK}} \geq 150\ \mu s$$

CVTM: CVTM[7:0] setting value (0 to 255)
f_{TEMCLK}: TEMCLK frequency

Temperature conversion control

Follow the procedure shown below to convert temperature using the TEM module.

(1) Configure the operating clock TEMCLK and supply it to the TEM module. (See Section 23.2.)

(2) Set the comparison time to CVTM[7:0].

(3) Enable TEM interrupts. (See Section 23.4.)

(4) Write 1 to TEMEN/TEM_CTL register to turn the temperature sensor on.
The temperature sensor starts detecting temperature.

(5) Wait at least 10 ms for the temperature sensor to stabilize.

(6) Write 1 to TEMTRG/TEM_CTL register to start temperature conversion.
TEMST/TEM_STAT register goes 1 during conversion and reverts to 0 upon completion of the conversion. Note, however, that maximum one TEMCLK cycle is required to set TEMST to 1 after writing 1 to TEMTRG.

The conversion time varies depending on the comparison time set and the current temperature.

The temperature conversion being executed can be stopped by writing 0 to TEMTRG. The conversion results become invalid in this case.

(7) A TEM interrupt occurs when the conversion has completed.
Read the conversion results from TEMP[7:0]/TEM_RSLT register after an interrupt has occurred.

(8) To perform subsequent temperature conversion, write 1 to TEMTRG again.

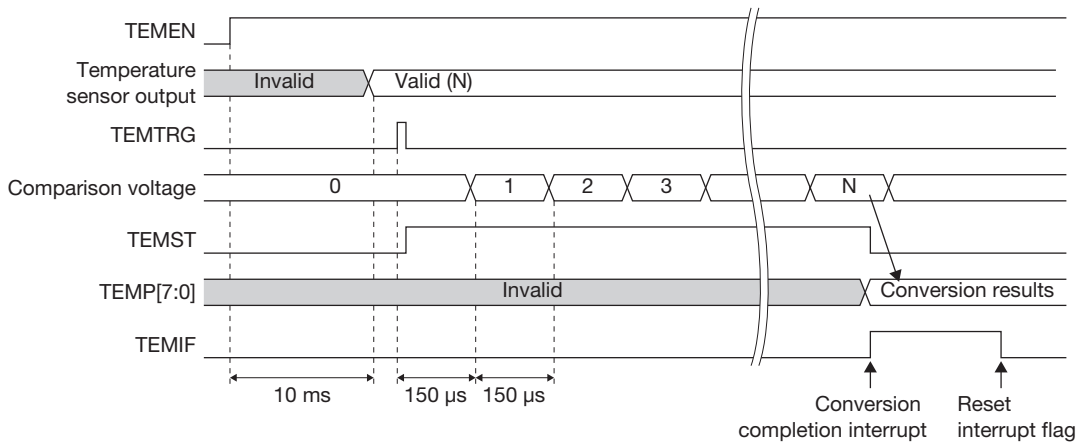


Figure 23.3.1 Temperature Conversion Operation

Correspondence between detection results and temperature

The table below lists the temperature (within the detection range) corresponding to the 8-bit value read from TEMP[7:0].

Table 23.3.1 Detected Temperature

Out of the guaranteed range		Guaranteed range		Out of the guaranteed range	
TEMP[7:0]	Temperature (°C)	TEMP[7:0]	Temperature (°C)	TEMP[7:0]	Temperature (°C)
0xff–0xdf	Invalid	0xb5	0.1	0x80	50.5
		0xb4	1.1	0x7f	51.5
		0xb3	2.0	0x7e	52.4
		0xb2	3.0	0x7d	53.3
		0xb1	3.9	0x7c	54.3
		0xb0	4.9	0x7b	55.2
		0xaf	5.9	0x7a	56.2
		0xae	6.8	0x79	57.1
		0xad	7.8	0x78	58.0
		0xac	8.7	0x77	59.0
		0xab	9.7	0x76	59.9
(0xdf)	(-40.8)	0xaa	10.7	0x75	60.8
0xde	-39.9	0xa9	11.6	0x74	61.8
0xdd	-38.9	0xa8	12.6	0x73	62.7
0xdc	-37.9	0xa7	13.5	0x72	63.6
0xdb	-36.9	0xa6	14.5	0x71	64.6
0xda	-35.9	0xa5	15.5	0x70	65.5
0xd9	-34.9	0xa4	16.4	0x6f	66.4
0xd8	-34.0	0xa3	17.4	0x6e	67.4
0xd7	-33.0	0xa2	18.3	0x6d	68.3
0xd6	-32.0	0xa1	19.3	0x6c	69.2
0xd5	-31.0	0xa0	20.2	0x6b	70.1
0xd4	-30.0	0x9f	21.2	0x6a	71.1
0xd3	-29.1	0x9e	22.1	0x69	72.0
0xd2	-28.1	0x9d	23.1	0x68	72.9
0xd1	-27.1	0x9c	24.0	0x67	73.9
0xd0	-26.1	0x9b	25.0	0x66	74.8
0xcf	-25.1	0x9a	26.0	0x65	75.7
0xce	-24.2	0x99	26.9	0x64	76.6
0xcd	-23.2	0x98	27.9	0x63	77.6
0xcc	-22.2	0x97	28.8	0x62	78.5
0xcb	-21.2	0x96	29.8	0x61	79.4
0xca	-20.3	0x95	30.7	0x60	80.3
0xc9	-19.3	0x94	31.7	0x5f	81.3
0xc8	-18.3	0x93	32.6	0x5e	82.2
0xc7	-17.3	0x92	33.5	0x5d	83.1
0xc6	-16.4	0x91	34.5	0x5c	84.0
0xc5	-15.4	0x90	35.4	0x5b	85.0
0xc4	-14.4	0x8f	36.4	(0x5a)	(85.9)
0xc3	-13.5	0x8e	37.3	0x5a–0x0	Invalid
0xc2	-12.5	0x8d	38.3		
0xc1	-11.5	0x8c	39.2		
0xc0	-10.5	0x8b	40.2		
0xbf	-9.6	0x8a	41.1		
0xbe	-8.6	0x89	42.1		
0xbd	-7.6	0x88	43.0		
0xbc	-6.7	0x87	43.9		
0xbb	-5.7	0x86	44.9		
0xba	-4.7	0x85	45.8		
0xb9	-3.8	0x84	46.8		
0xb8	-2.8	0x83	47.7		
0xb7	-1.8	0x82	48.7		
0xb6	-0.9	0x81	49.6		

Error within the guaranteed temperature range (0°C to 50°C): ±5°C

The effective range of the converted values is 0x5b to 0xde. If the TEMP[7:0] read value is out of the range, it should be handled as a conversion error.

Note: The detection results are the temperature inside the device detected by the sensor embedded in the device.

23.4 TEM Interrupt

The TEM module includes a function for generating interrupts when a temperature conversion is completed.
When a temperature conversion has completed, the interrupt flag TEMIF/TEM_STAT register is set to 1. To use this interrupt, set TEMIE/TEM_CTL register to 1. When TEMIE is set to 0 (default), interrupt requests for this cause will not be sent to the interrupt controller (ITC). If TEMIF is set to 1 while TEMIE is set to 1 (interrupt enabled), the TEM module outputs an interrupt request to the ITC. An interrupt is generated if the ITC and S1C17 Core interrupt conditions are satisfied.
TEMIF is reset by writing 1.

For more information on interrupt processing, see the "Interrupt Controller (ITC)" chapter.

Notes: • To prevent interrupt recurrences, the interrupt flag TEMIF must be reset in the interrupt handler routine after a TEM interrupt has occurred.

• To prevent unwanted interrupts, reset the interrupt flag TEMIF before enabling interrupts with the interrupt enable bit TEMIE.

23.5 Control Register Details

Table 23.5.1 List of TEM Registers

Address	Register name		Function
0x506f	TEM_CLK	TEM Clock Control Register	Controls the TEM clock.
0x51a0	TEM_TIME	TEM Comparison Time Setting Register	Sets the sensor output comparison time.
0x51a1	TEM_CTL	TEM Control Register	Controls the temperature detection circuit.
0x51a2	TEM_STAT	TEM Status Register	Indicates the conversion status.
0x51a3	TEM_RSLT	TEM Conversion Result Register	Temperature conversion results

The TEM module registers are described in detail below.

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

TEM Clock Control Register (TEM_CLK)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
TEM Clock Control Register (TEM_CLK)	**0x506f** (8 bits)	D7–6	**–**	reserved	–		–	–	0 when being read.
		D5–4	**TEMCLKD [1:0]**	TEM clock division ratio select	TEMCLKD[1:0]	Division ratio	0x0	R/W	When the clock source is OSC3B or OSC3A
					0x3	1/8			
					0x2	1/4			
					0x1	1/2			
					0x0	1/1			
		D3–2	**TEMCLK SRC[1:0]**	TEM clock source select	TEMCLK SRC[1:0]	Clock source	0x0	R/W	
					0x3	reserved			
					0x2	OSC3A			
					0x1	OSC1			
					0x0	OSC3B			
		D1	**–**	reserved	–		–	–	0 when being read.
		D0	**TEMCLKE**	TEM clock enable	1 Enable	0 Disable	0	R/W	

D[7:6] Reserved

D[5:4] TEMCLKD[1:0]: TEM Clock Division Ratio Select Bits

Selects the division ratio for generating the TEMCLK clock when OSC3B or OSC3A is used as the clock source.

Table 23.5.2 OSC3B/OSC3A Division Ratio Selection

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

(Default: 0x0)

D[3:2] TEMCLKSRC[1:0]: TEM Clock Source Select Bits

Selects the clock source.

Table 23.5.3 Clock Source Selection

TEMCLKSRC[1:0]	Clock source
0x3	Reserved
0x2	OSC3A
0x1	OSC1
0x0	OSC3B

(Default: 0x0)

D1 Reserved

D0 TEMCLKE: TEM Clock Enable Bit

Enables or disables the TEMCLK clock supply.

1 (R/W): Enabled (on)

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

The TEMCLKE default setting is 0, which disables the clock supply. Setting TEMCLKE to 1 sends the clock selected to the temperature detection circuit.

TEM Comparison Time Setting Register (TEM_TIME)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
TEM Comparison Time Setting Register (TEM_TIME)	0x51a0 (8 bits)	D7–0	CVTM[7:0]	Comparison time select $\frac{CVTM[7:0]+1}{f_{TEMCLK}} \geq 150\ \mu s$	0 to 255 clocks	0	R/W	

D[7:0] CVTM[7:0]: Comparison Time Select Bits

Sets the time for comparing the sensor output with the comparison voltage using the comparator. (Default: 0x0)

The comparison value must be set to 150 µs or more including clock frequency dispersion.

$$\text{Comparison time} = \frac{CVTM + 1}{f_{TEMCLK}} \geq 150\ \mu s$$

CVTM: CVTM[7:0] setting value (0 to 255)

f_{TEMCLK}: TEMCLK frequency

TEM Control Register (TEM_CTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
TEM Control Register (TEM_CTL)	0x51a1 (8 bits)	D7–5	–	reserved	–	–	–	0 when being read.
		D4	TEMIE	Conversion completion int. enable	1 Enable 0 Disable	0	R/W	
		D3–2	–	reserved	–	–	–	0 when being read.
		D1	TEMTRG	Conversion trigger	1 Start 0 Stop	0	W	
		D0	TEMEN	TEM enable	1 Enable 0 Disable	0	R/W	

D[7:5] Reserved

D4 TEMIE: Conversion Completion Interrupt Enable Bit

Enables or disables conversion completion interrupts.

1 (R/W): Interrupt enabled

0 (R/W): Interrupt disabled (default)

D[3:2] Reserved

D1 TEMTRG: Conversion Trigger Bit

Starts temperature conversion.

1 (W): Start conversion

0 (W): Stop conversion (default)

Writing 1 to TEMTRG starts conversion operation. When 1 is written to TEMEN to activate the temperature sensor, wait at least 10 ms for sensor to stabilize before starting conversion.

Writing 0 to TEMTRG during converting can abort the temperature conversion operation. The conversion results in this case become invalid.

D0 TEMEN: TEM Enable Bit

Enables or disables the temperature sensor to operate.

1 (R/W): Enabled

0 (R/W): Disabled (default)

Setting TEMEN to 1 enables the temperature sensor to generate the output voltage according to the detected temperature. The temperature sensor takes 10 ms until the output voltage is stabilized.

TEM Status Register (TEM_STAT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
TEM Status Register (TEM_STAT)	0x51a2 (8 bits)	D7–5	–	reserved	–	–	–	0 when being read.
		D4	TEMIF	Conversion completion interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D3–1	–	reserved	–	–	–	0 when being read.
		D0	TEMST	Conversion status	1 Busy 0 Idle	0	R	

D[7:5] Reserved

D4 TEMIF: Conversion Completion Interrupt Flag Bit

Indicates the conversion completion interrupt cause occurrence status.

1 (R): Cause of interrupt has occurred

0 (R): No cause of interrupt has occurred (default)

1 (W): Flag is reset

0 (W): Ignored

TEMIF is the interrupt flag of the TEM module and is set to 1 when a temperature conversion operation is completed. TEMIF is reset by writing 1.

D[3:1] Reserved

D0 TEMST: Conversion Status Bit

Indicates the temperature conversion operating status.

1 (R): Busy

0 (R): Idle (default)

TEMST goes 1 when a conversion operation is started by setting TEMTRG/TEM_CTL register and it reverts to 0 upon completion of conversion. Note, however, that maximum one TEMCLK cycle is required to set TEMST to 1 after writing 1 to TEMTRG.

TEM Conversion Result Register (TEM_RSLT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
TEM Conversion Result Register (TEM_RSLT)	0x51a3 (8 bits)	D7–0	TEMP[7:0]	Conversion result	0 to 255	0	R	

D[7:0] TEMP[7:0]: Conversion Result Bits

The temperature conversion results can be read out. (Default: 0x0)

Read values before conversion is completed become invalid.

For correspondence between the read value and temperature, see Table 23.3.1.

24 Supply Voltage Detection Circuit (SVD)

24.1 SVD Module Overview

The S1C17F57 includes an SVD (supply voltage detection) circuit to monitor the power voltage supplied to the VDD pin. It can be used to check whether the power supply voltage drops below the detection level set with software or not. The following shows the features of the SVD module:

- Power supply voltage to be detected: VDD
- Detection voltage levels: 13 levels (2.0 V to 3.2 V)

Figure 24.1.1 shows the SVD configuration.

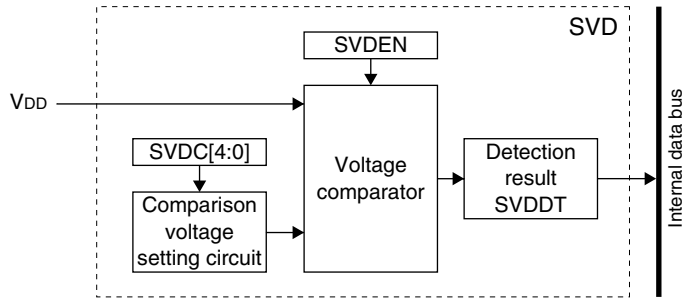


Figure 24.1.1 SVD Configuration

24.2 Comparison Voltage Setting

The SVD circuit compares the power supply voltage (VDD) against the comparison voltage set by software and outputs results indicating whether the power supply voltage exceeds this comparison voltage. The comparison voltage can be selected from among the 13 levels listed in Table 24.2.1 with the SVDC[4:0]/SVD_CMP register.

Table 24.2.1 Comparison Voltage Settings

SVDC[4:0]	Comparison Voltage	SVDC[4:0]	Comparison Voltage
0x1f	Reserved	0xf	2.10 V
0x1e		0xe	2.00 V
0x1d		0xd	Reserved
0x1c		0xc	
0x1b		0xb	
0x1a	3.20 V	0xa	
0x19	3.10 V	0x9	
0x18	3.00 V	0x8	
0x17	2.90 V	0x7	
0x16	2.80 V	0x6	
0x15	2.70 V	0x5	
0x14	2.60 V	0x4	
0x13	2.50 V	0x3	
0x12	2.40 V	0x2	
0x11	2.30 V	0x1	
0x10	2.20 V	0x0	

(Default: 0x0)

Note: The comparison voltage is effective only when it is set within the operating voltage range. If the comparison voltage set is out of the operating voltage range, no correct detection results will be obtained.

24.3 SVD Control

Power supply voltage detection using the SVD circuit is initiated by writing 1 to SVDEN/SVD_EN register. After that, the supply voltage detection results can be read out from SVDDT/SVD_RSLT register. By writing 0 to SVDEN, the SVD circuit sets the detection result at that point to SVDDT and stops detection.
The detection results and SVDDT readings are as follows.

- When power supply voltage (V_{DD}) ≥ comparison voltage: SVDDT = 0
- When power supply voltage (V_{DD}) < comparison voltage: SVDDT = 1

Notes: • An SVD circuit-enable response time is required to obtain stable detection results after SVDEN is altered from 0 to 1. Also when SVDC[4:0] is altered, an SVD circuit response time is required to obtain stable detection results. Wait until the response time has elapsed before reading SVDDT. Also when reading the detection results after stopping the SVD circuit, SVDEN should be set to 0 after the response time has elapsed. For these response times, see "Electrical Characteristics."

• Operating the SVD circuit increases current consumption. If power supply voltage detection is not required, stop SVD operations by setting SVDEN to 0.

24.4 Control Register Details

Table 24.4.1 List of SVD Registers

Address	Register name		Function
0x5100	SVD_EN	SVD Enable Register	Enables/disables the SVD operation.
0x5101	SVD_CMP	SVD Comparison Voltage Register	Sets the comparison voltage.
0x5102	SVD_RSLT	SVD Detection Result Register	Voltage detection results

The SVD module registers are described in detail below.

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

SVD Enable Register (SVD_EN)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SVD Enable Register (SVD_EN)	**0x5100** (8 bits)	D7–1	**–**	reserved	–	–	–	0 when being read.
		D0	**SVDEN**	SVD enable	1 Enable 0 Disable	0	R/W	

D[7:1] Reserved

D0 SVDEN: SVD Enable Bit

Enables or disables SVD operations.
1 (R/W): Enabled
0 (R/W): Disabled (default)

Setting SVDEN to 1 initiates power supply voltage detection; setting to 0 stops detection after loading the detection results to SVDDT/SVD_RSLT register.

Notes: • An SVD circuit-enable response time is required to obtain stable detection results after SVDEN is altered from 0 to 1. Also when SVDC[4:0] is altered, an SVD circuit response time is required to obtain stable detection results. Wait until the response time has elapsed before reading SVDDT. Also when reading the detection results after stopping the SVD circuit, SVDEN should be set to 0 after the response time has elapsed. For these response times, see "Electrical Characteristics."

• Operating the SVD circuit increases current consumption. If power supply voltage detection is not required, stop SVD operations by setting SVDEN to 0.

SVD Comparison Voltage Register (SVD_CMP)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
SVD Comparison Voltage Register (SVD_CMP)	0x5101 (8 bits)	D7–5	–	reserved	–		–	–	0 when being read.
		D4–0	SVDC[4:0]	SVD comparison voltage select	SVDC[4:0]	Voltage	0x0	R/W	
					0x1f–0x1b	reserved			
					0x1a	3.20 V			
					0x19	3.10 V			
					0x18	3.00 V			
					0x17	2.90 V			
					0x16	2.80 V			
					0x15	2.70 V			
					0x14	2.60 V			
					0x13	2.50 V			
					0x12	2.40 V			
					0x11	2.30 V			
					0x10	2.20 V			
					0xf	2.10 V			
					0xe	2.00 V			
					0xd–0x0	reserved			

D[7:5] Reserved

D[4:0] SVDC[4:0]: SVD Comparison Voltage Select Bits

Selects one of 13 comparison voltages for detecting voltage drops.

Table 24.4.2 Comparison Voltage Settings

SVDC[4:0]	Comparison Voltage	SVDC[4:0]	Comparison Voltage
0x1f	Reserved	0xf	2.10 V
0x1e		0xe	2.00 V
0x1d		0xd	Reserved
0x1c		0xc	
0x1b		0xb	
0x1a	3.20 V	0xa	
0x19	3.10 V	0x9	
0x18	3.00 V	0x8	
0x17	2.90 V	0x7	
0x16	2.80 V	0x6	
0x15	2.70 V	0x5	
0x14	2.60 V	0x4	
0x13	2.50 V	0x3	
0x12	2.40 V	0x2	
0x11	2.30 V	0x1	
0x10	2.20 V	0x0	

(Default: 0x0)

The SVD circuit compares the power supply voltage (V_{DD}) against the comparison voltage set by SVDC[4:0], and outputs results indicating whether the power supply voltage exceeds this comparison voltage.

Note: The comparison voltage is effective only when it is set within the operating voltage range. If the comparison voltage set is out of the operating voltage range, no correct detection results will be obtained.

SVD Detection Result Register (SVD_RSLT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SVD Detection Result Register (SVD_RSLT)	0x5102 (8 bits)	D7–1	–	reserved	–	–	–	0 when being read.
		D0	SVDDT	SVD detection result	1 Low 0 Normal	×	R	

D[7:1] Reserved

D0 SVDDT: SVD Detection Result Bit

Indicates the power supply voltage detection results.

1 (R): Power supply voltage (V_{DD}) < comparison voltage

0 (R): Power supply voltage (V_{DD}) ≥ comparison voltage

The SVD circuit compares the power supply voltage (V_{DD}) against the voltage set in SVDC[4:0]/SVD_CMP register while SVDEN/SVD_EN register = 1. The current power supply voltage status can be monitored by reading SVDDT. Also the detection result is set to SVDDT by writing 0 to SVDEN, so the power supply voltage status can be checked by reading SVDDT after that.

25 On-chip Debugger (DBG)

25.1 Resource Requirements and Debugging Tools

Debugging work area

Debugging requires a 64-byte debugging work area. For more information on the work area location, see the "Memory Map, Bus Control" chapter.
The start address for this debugging work area can be read from the DBRAM register (0xffff90).

Debugging tools

Debugging involves connecting ICDmini (S5U1C17001H) to the S1C17F57 debug pins and inputting the debug instruction from the debugger on the personal computer.
The following tools are required:
• S1C17 Family In-Circuit Debugger ICDmini (S5U1C17001H)
• S1C17 Family C compiler package (e.g., S5U1C17001C)

Debug pins

The following debug pins are used to connect ICDmini (S5U1C17001H).

Table 25.1.1 List of Debug Pins

Pin name	I/O	Qty	Function
DCLK	O	1	On-chip debugger clock output pin Outputs a clock to the ICDmini (S5U1C17001H).
DSIO	I/O	1	On-chip debugger data input/output pin Used to input/output debugging data and input the break signal.
DST2	O	1	On-chip debugger status signal output pin Outputs the processor status during debugging.

The on-chip debugger input/output pins (DCLK, DST2, DSIO) are shared with I/O ports and are initially set as the debug pins. If the debugging function is not used, these pins can be switched using the port function select bits to enable use as general-purpose I/O port pins.
For detailed information on pin function switching, see the "I/O Ports (P)" chapter.

25.2 Debug Break Operation Status

The S1C17 Core enters debug mode when the brk instruction is executed or a debug interrupt is generated by a break signal (Low) input to the DSIO pin. This state persists until the retd instruction is executed. During this time, hardware interrupts and NMIs are disabled.
The default setting halts peripheral circuit operations. This setting can be modified even when debugging is underway. The peripheral circuits that operate with PCLK will continue running in debug mode by setting DBRUN1/MISC_DMODE1 register to 1. Setting DBRUN1 to 0 (default) will stop these peripheral circuits in debug mode.
The peripheral circuits that operate with a clock other than PCLK will continue running in debug mode by setting DBRUN2/MISC_DMODE2 register to 1. Setting DBRUN2 to 0 (default) will stop these peripheral circuits in debug mode.
Some peripheral circuits, such as SPI, I2CS, and T16A2, that run with an external input clock will not stop operating even if the S1C17 Core enters debug mode.
The EPD controller/driver and RTC continue the operating status at occurrence of the debug interrupt.

25.3 Additional Debugging Function

The S1C17F57 expands the following on-chip debugging functions of the S1C17 Core.

Branching destination in debug mode

When a debug interrupt is generated, the S1C17 Core enters debug mode and branches to the debug processing routine. In this process, the S1C17 Core is designed to branch to address 0xfffc00. In addition to this branching destination, the S1C17F57 also allows designation of address 0x0 (beginning address of the internal RAM) as the branching destination when debug mode is activated. The branching destination address is selected using DBADR/MISC_IRAMSZ register. When the DBADR is set to 0 (default), the branching destination is set to 0xfffc00. When it is set to 1, the branching destination is set to 0x0.

Adding instruction breaks

The S1C17 Core supports two instruction breaks (hardware PC breaks). The S1C17F57 increased this number to five, adding the control bits and registers given below.

- IBE2/DCR register: Enables instruction breaks #2.
- IBE3/DCR register: Enables instruction breaks #3.
- IBE4/DCR register: Enables instruction breaks #4.
- IBAR2[23:0]/IBAR2 register: Set instruction break address #2.
- IBAR3[23:0]/IBAR3 register: Set instruction break address #3.
- IBAR4[23:0]/IBAR4 register: Set instruction break address #4.

Note that the debugger included in the S5U1C17001C (Ver. 1.2.1) or later is required to use five hardware PC breaks.

25.4 Control Register Details

Table 25.4.1 List of Debug Registers

Address	Register name		Function
0x4020	MISC_DMODE1	Debug Mode Control Register 1	Enables peripheral operations in debug mode (PCLK).
0x5322	MISC_DMODE2	Debug Mode Control Register 2	Enables peripheral operations in debug mode (except PCLK).
0x5326	MISC_IRAMSZ	IRAM Size Select Register	Selects the IRAM size.
0xffff90	DBRAM	Debug RAM Base Register	Indicates the debug RAM base address.
0xffffa0	DCR	Debug Control Register	Controls debugging.
0xffffb8	IBAR2	Instruction Break Address Register 2	Sets Instruction break address #2.
0xffffbc	IBAR3	Instruction Break Address Register 3	Sets Instruction break address #3.
0xffffd0	IBAR4	Instruction Break Address Register 4	Sets Instruction break address #4.

The debug registers are described in detail below.

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

• For debug registers not described here, refer to the S1C17 Core Manual.

Debug Mode Control Register 1 (MISC_DMODE1)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Debug Mode Control Register 1 (MISC_DMODE1)	**0x4020** (8 bits)	D7–2	**–**	reserved	–	–	–	0 when being read.
		D1	**DBRUN1**	Run/stop select in debug mode	1 Run 0 Stop	0	R/W	
		D0	**–**	reserved	–	–	–	0 when being read.

D[7:2] Reserved

D1 DBRUN1: Run/Stop Select Bit in Debug Mode

Selects the operating status of the peripheral circuits that operate with PCLK in debug mode.

1 (R/W): Run

0 (R/W): Stop (default)

Setting DBRUN1 to 1 enables the peripheral circuits that operate with PCLK to run even in debug mode. Setting it to 0 will stop them when the S1C17 Core enters debug mode. Set DBRUN1 to 1 to maintain running status for these peripheral circuits in debug mode.

D0 Reserved

Debug Mode Control Register 2 (MISC_DMODE2)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Debug Mode Control Register 2 (MISC_DMODE2)	0x5322 (16 bits)	D15–1	–	reserved	–	–	–	0 when being read.
		D0	DBRUN2	Run/stop select in debug mode (except PCLK peripheral circuits)	1 Run 0 Stop	0	R/W	

D[15:1] Reserved

D0 DBRUN2: Run/Stop Select Bit in Debug Mode (except PCLK peripheral circuits)

Selects the operating status of the peripheral circuits that operate with a clock other than PCLK in debug mode.

1 (R/W): Run

0 (R/W): Stop (default)

Setting DBRUN2 to 1 enables the peripheral circuits that operate with a clock other than PCLK to run even in debug mode. Setting it to 0 will stop them when the S1C17 Core enters debug mode. Set DBRUN2 to 1 to maintain running status for these peripheral circuits in debug mode.

Some peripheral circuits, such as SPI, I2CS, and T16A2, that run with an external input clock will not stop operating even if the S1C17 Core enters debug mode.

The EPD controller/driver and RTC continue the operating status at occurrence of the debug interrupt.

IRAM Size Select Register (MISC_IRAMSZ)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
IRAM Size Register (MISC_IRAMSZ)	0x5326 (16 bits)	D15–9	–	reserved	–	–	–	0 when being read.
		D8	DBADR	Debug base address select	1 0x0 0 0xfffc00	0	R/W	
		D7	–	reserved	–	–	–	0 when being read.
		D6–4	IRAMACTSZ [2:0]	IRAM actual size	0x3 (= 2KB)	0x3	R	
		D3	–	reserved	–	–	–	0 when being read.
		D2–0	IRAMSZ[2:0]	IRAM size select	IRAMSZ[2:0] / Size 0x5 / 512B 0x4 / 1KB 0x3 / 2KB Other / reserved	0x3	R/W	

D[15:9] Reserved

D8 DBADR: Debug Base Address Select Bit

Selects the branching destination address when a debug interrupt occurs.

1(R/W): 0x0

0(R/W): 0xfffc00 (default)

D7 Reserved

D[6:4] IRAMACTSZ[2:0]: IRAM Actual Size Bits

Indicates the actual internal RAM size embedded. (Default: 0x3)

D3 Reserved

D[2:0] IRAMSZ[2:0]: IRAM Size Select Bits

Selects the size of the internal RAM to be used.

Table 25.4.2 Internal RAM Size Selection

IRAMSZ[2:0]	Internal RAM size
0x5	512B
0x4	1KB
0x3	2KB
Other	Reserved

(Default: 0x3)

Note: The MISC_IRAMSZ register is write-protected. To alter this register settings, you must override this write-protection by writing 0x96 to the MISC_PROT register. Normally, the MISC_PROT register should be set to a value other than 0x96, except when altering the MISC_IRAMSZ register. Unnecessary rewriting of the MISC_IRAMSZ register may result in system malfunctions.

Debug RAM Base Register (DBRAM)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Debug RAM Base Register (DBRAM)	0xffff90 (32 bits)	D31–24	–	Unused (fixed at 0)	0x0	0x0	R	
		D23–0	DBRAM[23:0]	Debug RAM base address	0x7c0	0x7c0	R	

D[31:24] Not used (Fixed at 0)

D[23:0] DBRAM[23:0]: Debug RAM Base Address Bits

Read-only register containing the beginning address of the debugging work area (64 bytes).

Debug Control Register (DCR)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Debug Control Register (DCR)	0xffffa0 (8 bits)	D7	IBE4	Instruction break #4 enable	1	Enable	0	Disable	0	R/W	
		D6	IBE3	Instruction break #3 enable	1	Enable	0	Disable	0	R/W	
		D5	IBE2	Instruction break #2 enable	1	Enable	0	Disable	0	R/W	
		D4	DR	Debug request flag	1	Occurred	0	Not occurred	0	R/W	Reset by writing 1.
		D3	IBE1	Instruction break #1 enable	1	Enable	0	Disable	0	R/W	
		D2	IBE0	Instruction break #0 enable	1	Enable	0	Disable	0	R/W	
		D1	SE	Single step enable	1	Enable	0	Disable	0	R/W	
		D0	DM	Debug mode	1	Debug mode	0	User mode	0	R	

D7 IBE4: Instruction Break #4 Enable Bit

Enables or disables instruction break #4.
1 (R/W): Enabled
0 (R/W): Disabled (default)

If this bit is set to 1, the instruction fetch address and the value set in the IBAR4 register are compared. If they match, an instruction break is generated. If this bit is set to 0, no comparison is performed.

D6 IBE3: Instruction Break #3 Enable Bit

Enables or disables instruction break #3.
1 (R/W): Enabled
0 (R/W): Disabled (default)

If this bit is set to 1, the instruction fetch address and the value set in the IBAR3 register are compared. If they match, an instruction break is generated. If this bit is set to 0, no comparison is performed.

D5 IBE2: Instruction Break #2 Enable Bit

Enables or disables instruction break #2.
1 (R/W): Enabled
0 (R/W): Disabled (default)

If this bit is set to 1, the instruction fetch address and the value set in the IBAR2 register are compared. If they match, an instruction break is generated. If this bit is set to 0, no comparison is performed.

D4 DR: Debug Request Flag Bit

Indicates the presence or absence of an external debug request.

1 (R): Request generated
0 (R): Request not generated (default)
1 (W): Flag is reset
0 (W): Ignored

This flag is cleared (reset to 0) when 1 is written. It must be cleared before the debug processing routine is terminated by the retd instruction.

D3 IBE1: Instruction Break #1 Enable Bit

Enables or disables instruction break #1.

1 (R/W): Enabled
0 (R/W): Disabled (default)

If this bit is set to 1, the instruction fetch address and the value set in the IBAR1 register are compared. If they match, an instruction break is generated. If this bit is set to 0, no comparison is performed.

D2 IBE0: Instruction Break #0 Enable Bit

Enables or disables instruction break #0.

1 (R/W): Enabled
0 (R/W): Disabled (default)

If this bit is set to 1, the instruction fetch address and the value set in the IBAR0 register are compared. If they match, an instruction break is generated. If this bit is set to 0, no comparison is performed.

D1 SE: Single Step Enable Bit

Enables or disables single-step operations.

1 (R/W): Enabled
0 (R/W): Disabled (default)

D0 DM: Debug Mode Bit

Indicates the processor operating mode (debug mode or user mode).

1 (R): Debug mode
0 (R): User mode (default)

Instruction Break Address Register 2 (IBAR2)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Instruction Break Address Register 2 (IBAR2)	**0xffffb8** (32 bits)	D31–24	**–**	reserved	–	–	–	0 when being read.
		D23–0	**IBAR2[23:0]**	Instruction break address #2 IBAR223 = MSB IBAR20 = LSB	0x0 to 0xffffff	0x0	R/W	

D[31:24] Reserved

D[23:0] IBAR2[23:0]: Instruction Break Address #2 Bits

Sets instruction break address #2. (default: 0x000000)

Instruction Break Address Register 3 (IBAR3)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Instruction Break Address Register 3 (IBAR3)	**0xffffbc** (32 bits)	D31–24	**–**	reserved	–	–	–	0 when being read.
		D23–0	**IBAR3[23:0]**	Instruction break address #3 IBAR323 = MSB IBAR30 = LSB	0x0 to 0xffffff	0x0	R/W	

D[31:24] Reserved

D[23:0] IBAR3[23:0]: Instruction Break Address #3 Bits

Sets instruction break address #3. (default: 0x000000)

Instruction Break Address Register 4 (IBAR4)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Instruction Break Address Register 4 (IBAR4)	**0xffffd0** (32 bits)	D31–24	**–**	reserved	–	–	–	0 when being read.
		D23–0	**IBAR4[23:0]**	Instruction break address #4 IBAR423 = MSB IBAR40 = LSB	0x0 to 0xffffff	0x0	R/W	

D[31:24] Reserved

D[23:0] IBAR4[23:0]: Instruction Break Address #4 Bits

Sets instruction break address #4. (default: 0x000000)

26 Multiplier/Divider (COPRO)

26.1 Overview

The S1C17F57 has an embedded coprocessor that provides multiplier/divider functions.
The following shows the features of the multiplier/divider:

- Multiplication: Supports signed/unsigned multiplications.
  (16 bits × 16 bits = 32 bits)
  Can be executed in 1 cycle.
- Multiplication and accumulation (MAC): Supports signed MAC operations with overflow detection function.
  (16 bits × 16 bits + 32 bits = 32 bits)
  Can be executed in 1 cycle.
- Division: Supports signed/unsigned divisions.
  (16 bits ÷ 16 bits = 16 bits with 16-bit residue)
  Can be executed in 17 to 20 cycles.

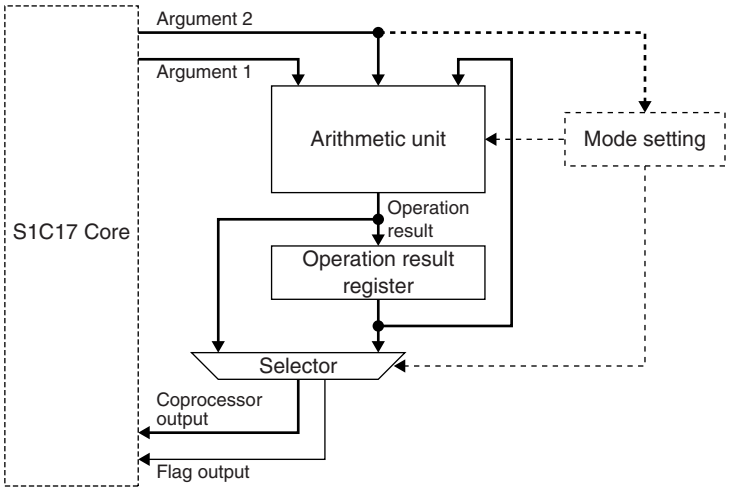


Figure 26.1.1 Multiplier/Divider Block Diagram

26.2 Operation Mode and Output Mode

The Multiplier/divider operates according to the operation mode specified by the application program. As listed in Table 26.2.1, the multiplier/divider supports nine operations.
The multiplication, division and MAC results are 32-bit data, therefore, the S1C17 Core cannot read them in one access cycle. The output mode is provided to specify the high-order 16 bits or low-order 16 bits of the operation results to be read from the multiplier/divider.
The operation and output modes can be specified with a 7-bit data by writing it to the mode setting register in the multiplier/divider. Use a "`ld.cw`" instruction for this writing.

`ld.cw %rd,%rs` %rs[6:0] is written to the mode setting register. (%rd: not used)
`ld.cw %rd,`*`imm7`* *imm7*[6:0] is written to the mode setting register. (%rd: not used)

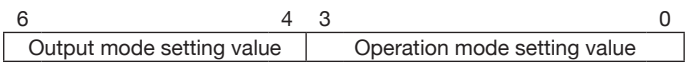


Figure 26.2.1 Mode Setting Register

Table 26.2.1 Mode Settings

Setting value (D[6:4])	Output mode	Setting value (D[3:0])	Operation mode
0x0	**16 low-order bits output mode** The low-order 16 bits of operation results can be read as the coprocessor output.	0x0	**Initialize mode 0** Clears the operation result register to 0x0.
0x1	**16 high-order bits output mode** The high-order 16 bits of operation results can be read as the coprocessor output.	0x1	**Initialize mode 1** Loads the 16-bit augend into the low-order 16 bits of the operation result register.
0x2–0x7	Reserved	0x2	**Initialize mode 2** Loads the 32-bit augend into the operation result register.
		0x3	**Operation result read mode** Outputs the data in the operation result register without computation.
		0x4	**Unsigned multiplication mode** Performs unsigned multiplication.
		0x5	**Signed multiplication mode** Performs signed multiplication.
		0x6	Reserved
		0x7	**Signed MAC mode** Performs signed MAC operation.
		0x8	**Unsigned division mode** Performs unsigned division.
		0x9	**Signed division mode** Performs signed division.
		0xa–0xf	Reserved

26.3 Multiplication

The multiplication function performs "A (32 bits) = B (16 bits) × C (16 bits)."
To perform a multiplication, set the operation mode to 0x4 (unsigned multiplication) or 0x5 (signed multiplication). Then send the 16-bit multiplicand (B) and 16-bit multiplier (C) to the multiplier/divider using a "`ld.ca`" instruction. The one-half (16 bits according to the output mode) result (A[15:0] or A[31:16]) and the flag status will be returned to the CPU registers. Another one-half should be read by setting the multiplier/divider into operation result read mode.

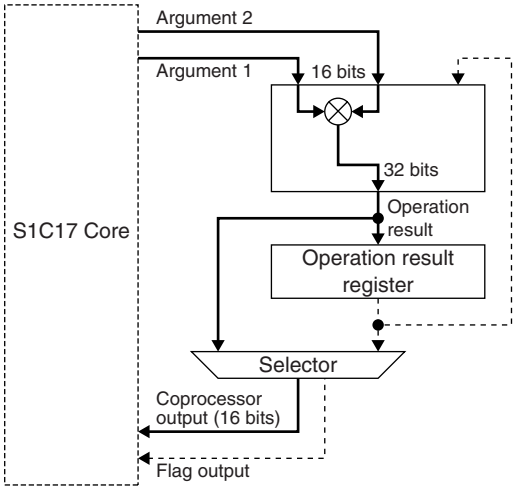


Figure 26.3.1 Data Path in Multiplication Mode

Table 26.3.1 Operation in Multiplication Mode

Mode setting value	Instruction	Operations	Flags	Remarks
0x04 or 0x05	`ld.ca %rd,%rs`	res[31:0] ← %rd × %rs %rd ← res[15:0]	psr (CVZN) ← 0b0000	The operation result register keeps the operation result until it is rewritten by other operation.
	(`ext` *imm9*) `ld.ca %rd,`*imm7*	res[31:0] ← %rd × *imm7/16* %rd ← res[15:0]		
0x14 or 0x15	`ld.ca %rd,%rs`	res[31:0] ← %rd × %rs %rd ← res[31:16]		
	(`ext` *imm9*) `ld.ca %rd,`*imm7*	res[31:0] ← %rd × *imm7/16* %rd ← res[31:16]		

res: operation result register

Example:

```
ld.cw %r0,0x4   ; Sets the modes (unsigned multiplication mode and 16 low-order bits output mode).
ld.ca %r0,%r1   ; Performs "res = %r0 × %r1" and loads the 16 low-order bits of the result to %r0.
ld.cw %r0,0x13  ; Sets the modes (operation result read mode and 16 high-order bits output mode).
ld.ca %r1,%r0   ; Loads the 16 high-order bits of the result to %r1.
```

26.4 Division

The division function performs "B (16 bits) ÷ C (16 bits) = A (16 bits), residue D (16 bits)."
To perform a division, set the operation mode to 0x8 (unsigned division) or 0x9 (signed division). Then send the 16-bit dividend (B) and 16-bit divisor (C) to the multiplier/divider using a "`ld.ca`" instruction. The quotient and the residue will be stored in the low-order 16 bits and the high-order 16 bits of the operation result register, respectively. The 16-bit quotient or residue according to the output mode specification and the flag status will be returned to the CPU registers. Another 16-bit result should be read by setting the multiplier/divider into operation result read mode.

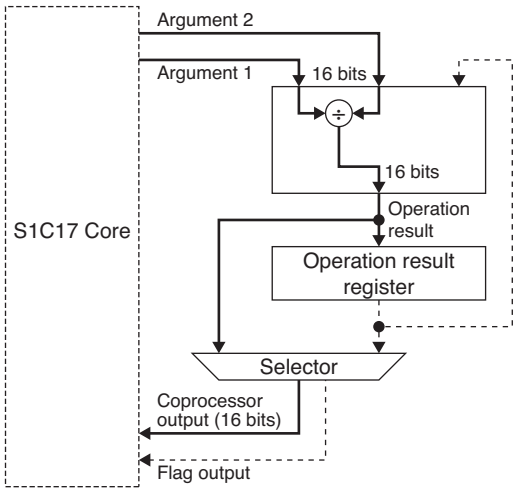


Figure 26.4.1 Data Path in Division Mode

Table 26.4.1 Operation in Division Mode

Mode setting value	Instruction	Operations	Flags	Remarks
0x08 or 0x09	`ld.ca %rd,%rs`	res[31:0] ← %rd ÷ %rs %rd ← res[15:0] (quotient)	psr (CVZN) ← 0b0000	The operation result register keeps the operation result until it is rewritten by other operation.
	(`ext` *imm9*) `ld.ca %rd,`*imm7*	res[31:0] ← %rd ÷ *imm7/16* %rd ← res[15:0] (quotient)		
0x018 or 0x19	`ld.ca %rd,%rs`	res[31:0] ← %rd ÷ %rs %rd ← res[31:16] (residue)		
	(`ext` *imm9*) `ld.ca %rd,`*imm7*	res[31:0] ← %rd ÷ *imm7/16* %rd ← res[31:16] (residue)		

res: operation result register

Example:

```
ld.cw %r0,0x8  ; Sets the modes (unsigned division mode and 16 low-order bits output mode).
ld.ca %r0,%r1  ; Performs "res = %r0 ÷ %r1" and loads the 16 low-order bits of the result (quotient) to %r0.
ld.cw %r0,0x13 ; Sets the modes (operation result read mode and 16 high-order bits output mode).
ld.ca %r1,%r0  ; Loads the 16 high-order bits of the result (residue) to %r1.
```

26.5 MAC

The MAC (multiplication and accumulation) function performs "A (32 bits) = B (16 bits) × C (16 bits) + A (32 bits)."
Before performing a MAC operation, the initial value (A) must be set to the operation result register.
To clear the operation result register (A = 0), just set the operation mode to 0x0. It is not necessary to send 0x0 to the multiplier/divider with another instruction.
To load a 16-bit value or a 32-bit value to the operation result register, set the operation mode to 0x1 (16 bits) or 0x2 (32 bits), respectively. Then send the initial value to the multiplier/divider using a "`ld.cf`" instruction.

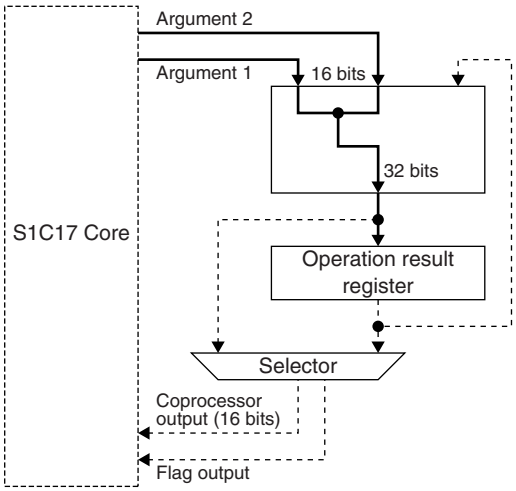


Figure 26.5.1 Data Path in Initialize Mode

Table 26.5.1 Initializing the Operation Result Register

Mode setting value	Instruction	Operations	Remarks
0x0	–	res[31:0] ← 0x0	Setting the operating mode executes the initialization without sending data.
0x1	`ld.cf %rd,%rs`	res[31:16] ← 0x0 res[15:0] ← %rs	
	(`ext` *imm9*) `ld.cf %rd,`*imm7*	res[31:16] ← 0x0 res[15:0] ← *imm7/16*	
0x2	`ld.cf %rd,%rs`	res[31:16] ← %rd res[15:0] ← %rs	
	(`ext` *imm9*) `ld.cf %rd,`*imm7*	res[31:16] ← %rd res[15:0] ← *imm7/16*	

res: operation result register

To perform a MAC operation, set the operation mode to 0x7 (signed MAC). Then send the 16-bit multiplicand (B) and 16-bit multiplier (C) to the multiplier/divider using a "`ld.ca`" instruction. The one-half (16 bits according to the output mode) result (A[15:0] or A[31:16]) and the flag status will be returned to the CPU registers. Another one-half should be read by setting the multiplier/divider into operation result read mode.
The overflow (V) flag in the PSR may be set to 1 according to the result. Other flags are set to 0.
When repeating the MAC operation without operation result read mode inserted, send multiplicand and multiplier data for number of required times. In this case it is not necessary to set the MAC mode every time.

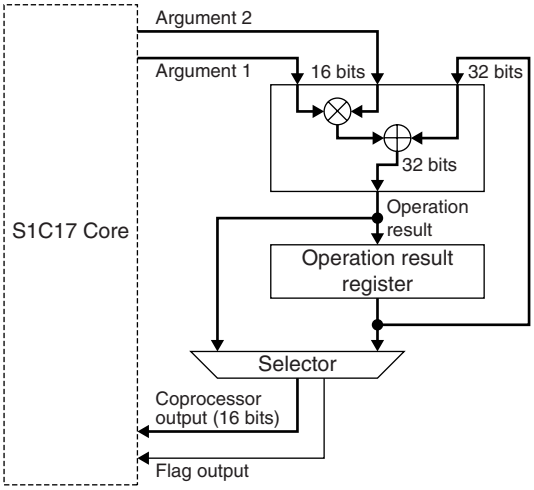


Figure 26.5.2 Data Path in MAC Mode

Table 26.5.2 Operation in MAC Mode

Mode setting value	Instruction	Operations	Flags	Remarks
0x07	`ld.ca %rd,%rs`	res[31:0] ← %rd × %rs + res[31:0] %rd ← res[15:0]	psr (CVZN) ← 0b0100 if an overflow has occurred Otherwise psr (CVZN) ← 0b0000	The operation result register keeps the operation result until it is rewritten by other operation.
	(`ext` *imm9*) `ld.ca %rd,`*imm7*	res[31:0] ← %rd × *imm7/16* + res[31:0] %rd ← res[15:0]		
0x17	`ld.ca %rd,%rs`	res[31:0] ← %rd × %rs + res[31:0] %rd ← res[31:16]		
	(`ext` *imm9*) `ld.ca %rd,`*imm7*	res[31:0] ← %rd × *imm7/16* + res[31:0] %rd ← res[31:16]		

res: operation result register

Example:

```
ld.cw %r0,0x7   ; Sets the modes (signed MAC mode and 16 low-order bits output mode).
ld.ca %r0,%r1   ; Performs "res = %r0 × %r1 + res" and loads the 16 low-order bits of the result to %r0.
ld.cw %r0,0x13  ; Sets the modes (operation result read mode and 16 high-order bits output mode).
ld.ca %r1,%r0   ; Loads the 16 high-order bits of the result to %r1.
```

Conditions to set the overflow (V) flag

An overflow occurs in a MAC operation and the overflow (V) flag is set to 1 when the signs of the multiplication result, operation result register value, and multiplication & accumulation result match the following conditions:

Table 26.5.3 Conditions to Set the Overflow (V) Flag

Mode setting value	Sign of multiplication result	Sign of operation result register value	Sign of multiplication & accumulation result
0x07	0 (positive)	0 (positive)	1 (negative)
0x07	1 (negative)	1 (negative)	0 (positive)

An overflow occurs when a MAC operation performs addition of positive values and a negative value results, or it performs addition of negative values and a positive value results. The coprocessor holds the operation result when the overflow (V) flag is cleared.

Conditions to clear the overflow (V) flag

The overflow (V) flag that has been set will be cleared when an overflow has not been occurred during execution of the "`ld.ca`" instruction for MAC operation or when the "`ld.ca`" or "`ld.cf`" instruction is executed in an operation mode other than operation result read mode.

26.6 Reading Operation Results

The "ld.ca" instruction cannot load a 32-bit operation result to a CPU register, so a multiplication or MAC operation returns the one-half (16 bits according to the output mode) result (A[15:0] or A[31:16]) and the flag status to the CPU registers. Another one-half should be read by setting the multiplier/divider into operation result read mode. The operation result register keeps the loaded operation result until it is rewritten by other operation.

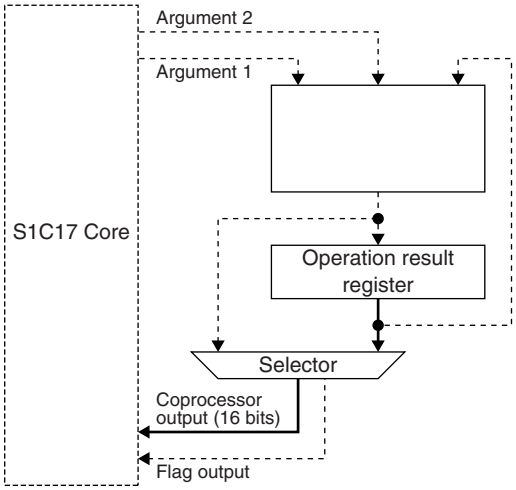


Figure 26.6.1 Data Path in Operation Result Read Mode

Table 26.6.1 Operation in Operation Result Read Mode

Mode setting value	Instruction	Operations	Flags	Remarks
0x03	ld.ca %rd,%rs	%rd ← res[15:0]	psr (CVZN) ← 0b0000	This operation mode does not affect the operation result register.
	ld.ca %rd,*imm7*	%rd ← res[15:0]		
0x13	ld.ca %rd,%rs	%rd ← res[31:16]		
	ld.ca %rd,*imm7*	%rd ← res[31:16]		

res: operation result register

27 Electrical Characteristics

27.1 Absolute Maximum Ratings

(V_{SS} = 0V)

Item	Symbol	Condition	Rated value	Unit
Power supply voltage	V_{DD}		-0.3 to 4.0	V
Flash programming voltage	V_{PP}		8	V
EPD power supply voltage	V_{EPD}		23	V
EPD power supply voltage	V_{E2}		3.8	V
Input voltage	V_I		-0.3 to V_{DD} + 0.5	V
Output voltage	V_O		-0.3 to V_{DD} + 0.5	V
High level output current	I_{OH}	1 pin	-10	mA
		Total of all pins	-20	mA
Low level output current	I_{OL}	1 pin	10	mA
		Total of all pins	20	mA
Storage temperature	Tstg		-65 to 125	°C
Soldering temperature/time	Tsol		260°C, 10 seconds (lead section)	–
COF mounting temperature/time	Tcofm		*1	–

*1 See the “Mounting Precautions” section in Appendix.

27.2 Recommended Operating Conditions

(V_{SS} = 0V) *1

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Operating power supply voltage	V_{DD}	Normal operation mode	2.0		3.6	V
Flash programming voltage	$V_{PP}P$		6.8	7.0	7.2	V
Flash programming temperature	T_{PP}		10		40	°C
Flash erasing voltage	$V_{PP}E$		7.3	7.5	7.7	V
Operating frequency	f_{OSC3A}	Crystal/ceramic oscillation	0.2		8.2	MHz
	f_{OSC1A}	Crystal oscillation		32.768		kHz
Operating temperature	Ta	During normal operation (Flash read only)	-40		85	°C
		During Flash programming and erasing	10		40	°C
Capacitor between V_{SS} and V_{D1}	C_1			0.1		µF
Capacitor between V_{SS} and V_{OSC}	C_2			0.1		µF
Capacitor between V_{SS} and V_{E1} *2	C_3			0.1		µF
Capacitor between V_{SS} and V_{E2} *2	C_4			0.1		µF
Capacitor between V_{SS} and V_{E3} *2	C_5			0.1		µF
Capacitor between V_{SS} and V_{E4} *2	C_6			0.1		µF
Capacitor between V_{SS} and V_{E5} *2	C_7			0.1		µF
Capacitor between C_{D1} and C_{D2} *2	C_8			0.1		µF
Capacitor between C_{B1} and C_{B2} *2	C_9			0.1		µF
Capacitor between C_{B3} and C_{B4} *2	C_{10}			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 capacitors are not required when EPD controller/driver is not used. In this case, leave the V_{E1} to V_{E5}, C_{D1} to C_{D2}, and C_{B1} to C_{B4} pins open.

*3 The component values should be determined after evaluating operations using an actual mounting board.

27.3 Current Consumption

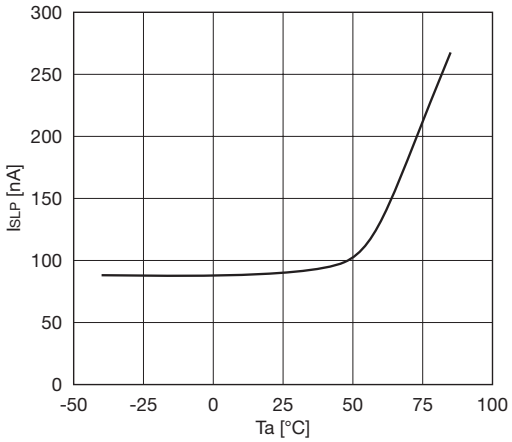
Unless otherwise specified: V_{DD} = 2.0 to 3.6V, V_{SS} = 0V, Ta = 25°C, PCKEN[1:0] = 0x3 (ON), RDWAIT[1:0] = 0x0 (no wait), OSC1A = no theoretical regulation correction, CCLKGR[1:0] = 0x0 (gear ratio 1/1), RTCRUN = 0 (OFF), HVLD = 0

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Current consumption in SLEEP mode	I_{SLP}	OSC1A = OFF, OSC1B = OFF, OSC3B = OFF		100	200	nA
		OSC1A = 32kHz, OSC3B = OFF, OSC3A = OFF, RTCRUN = 1 (ON)		210	260	nA
		OSC1B = 32kHz, OSC3B = OFF, OSC3A = OFF, RTCRUN = 1 (ON)		820	1040	nA
Current consumption in HALT mode	I_{HALT1}	OSC1A = 32kHz, OSC3B = OFF, OSC3A = OFF, PCKEN[1:0] = 0x0 (OFF)		0.55	0.69	µA
		OSC1A = 32kHz, OSC3B = OFF, OSC3A = OFF, PCKEN[1:0] = 0x0 (OFF), RTCRUN = 1 (ON)		0.5	0.63	µA
		OSC1A = 32kHz, OSC3B = OFF, OSC3A = OFF		1.5	1.7	µA
		OSC1B = 32kHz, OSC3B = OFF, OSC3A = OFF, PCKEN[1:0] = 0x0 (OFF)		1.1	1.3	µA
	I_{HALT2}	OSC1A = 32kHz, OSC3B = OFF, OSC3A = ON (1MHz ceramic)		90	120	µA
		OSC1A = 32kHz, OSC3B = OFF, OSC3A = ON (4MHz ceramic)		200	250	µA
	I_{HALT3}	OSC1A = 32kHz, OSC3B = ON (500kHz), OSC3A = OFF		80	120	µA
		OSC1A = 32kHz, OSC3B = ON (1MHz), OSC3A = OFF		110	160	µA
		OSC1A = 32kHz, OSC3B = ON (2MHz), OSC3A = OFF		160	220	µA
Current consumption during execution *1	I_{EXE1}	OSC1A = 32kHz, OSC3B = OFF, OSC3A = OFF, CPU = OSC1A		12	16	µA
		OSC1A = 32kHz, OSC3B = OFF, OSC3A = OFF, CPU = OSC1A, CCLKGR[1:0] = 0x2 (gear ratio 1/4)		5.4	6.6	µA
	I_{EXE2}	OSC1A = 32kHz, OSC3B = OFF, OSC3A = ON (1MHz ceramic), CPU = OSC3A		410	530	µA
		OSC1A = 32kHz, OSC3B = OFF, OSC3A = ON (1MHz ceramic), CPU = OSC3A, CCLKGR[1:0] = 0x2 (gear ratio 1/4)		210	270	µA
		OSC1A = 32kHz, OSC3B = OFF, OSC3A = ON (4MHz ceramic), CPU = OSC3A		1440	1880	µA
		OSC1A = 32kHz, OSC3B = OFF, OSC3A = ON (4MHz ceramic), CPU = OSC3A, CCLKGR[1:0] = 0x2 (gear ratio 1/4)		690	850	µA
	I_{EXE3}	OSC1A = 32kHz, OSC3B = ON (500kHz), OSC3A = OFF, CPU = OSC3B		250	330	µA
		OSC1A = 32kHz, OSC3B = ON (1MHz), OSC3A = OFF, CPU = OSC3B		430	570	µA
		OSC1A = 32kHz, OSC3B = ON (2MHz), OSC3A = OFF, CPU = OSC3B		770	1030	µA
		OSC1A = 32kHz, OSC3B = ON (2MHz), OSC3A = OFF, CPU = OSC3B, CCLKGR[1:0] = 0x2 (gear ratio 1/4)		400	520	µA
Current consumption during execution in heavy load protection mode *1	I_{EXE1H}	OSC1A = 32kHz, OSC3B = OFF, OSC3A = OFF, CPU = OSC1A, HVLD = 1		24	30	µA

*1 The values of current consumption during execution were measured when a test program consisting of 60.5% ALU instructions, 17% branch instructions, 12% memory read instructions, and 10.5% memory write instructions was executed continuously in the Flash memory.

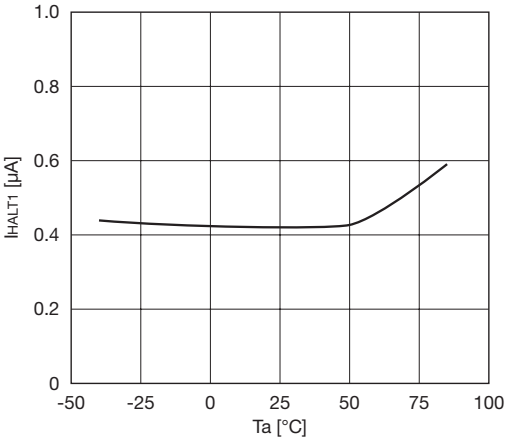
Current consumption-temperature characteristic in SLEEP mode

OSC1A = OFF, OSC1B = OFF, OSC3B = OFF, OSC3A = OFF, Typ. value



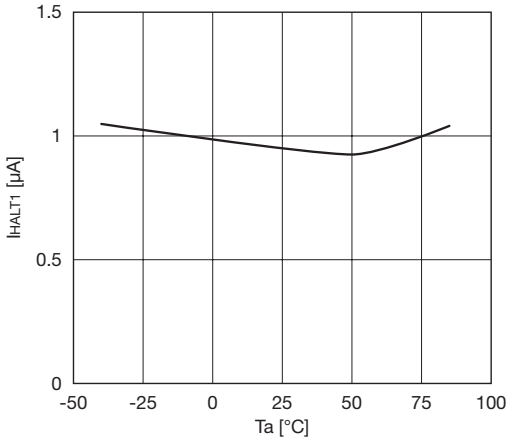
Current consumption-temperature characteristic in HALT mode (OSC1A operation)

OSC1A = 32.768kHz crystal, OSC3B = OFF, OSC3A = OFF, PCKEN[1:0] = 0x3, CCLKGR[1:0] = 0x0, Typ. value



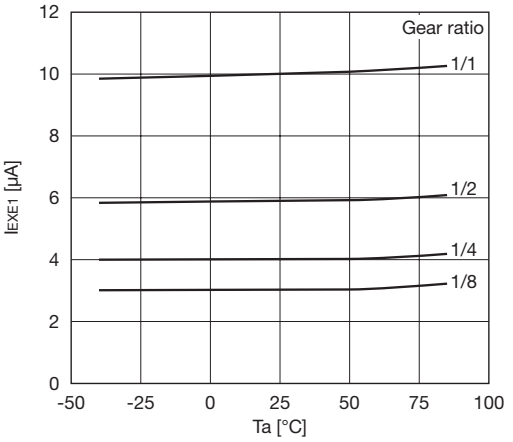
Current consumption-temperature characteristic in HALT mode (OSC1B operation)

OSC1B = 32kHz, OSC3B = OFF, OSC3A = OFF, PCKEN[1:0] = 0x0, CCLKGR[1:0] = 0x0, Typ. value



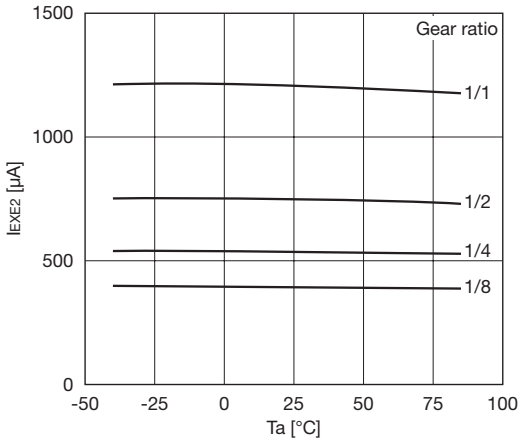
Current consumption-temperature characteristic during execution with OSC1A + clock gear

OSC1A = 32.768kHz crystal, OSC3B = OFF, OSC3A = OFF, PCKEN[1:0] = 0x3, Typ. value



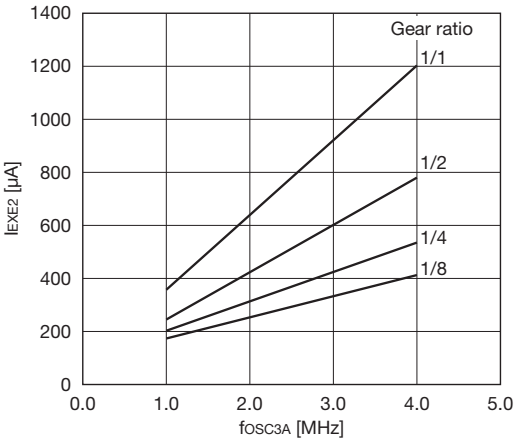
Current consumption-temperature characteristic during execution with OSC3A + clock gear

OSC3A = ON (4MHz ceramic), OSC3B = OFF, OSC1A = 32.768kHz crystal, PCKEN[1:0] = 0x3, Typ. value



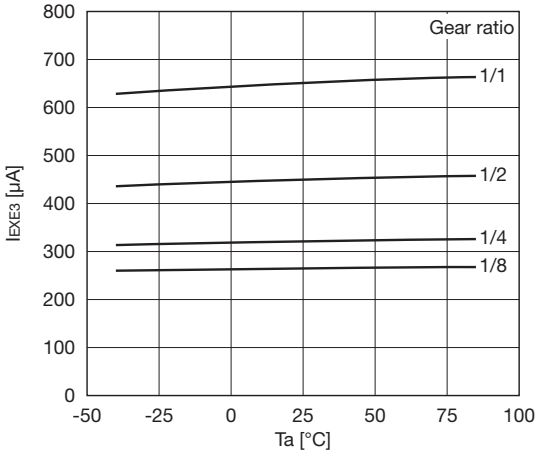
Current consumption-frequency characteristic during execution with OSC3A

OSC3A = ON, OSC3B = OFF, OSC1A = 32.768kHz crystal, PCKEN[1:0] = 0x3, Ta = 25°C, Typ. value



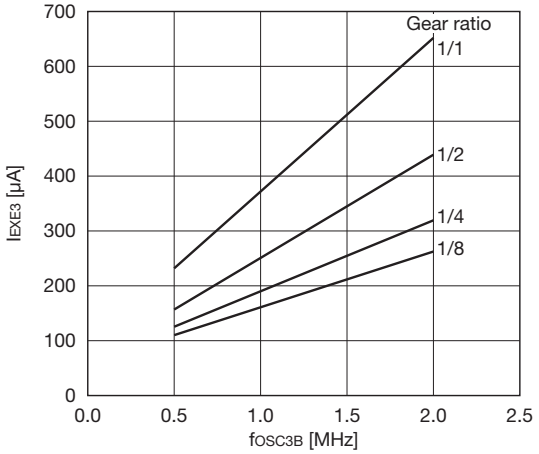
Current consumption-temperature characteristic during execution with OSC3B + clock gear

OSC3B = ON (2MHz), OSC3A = OFF,
OSC1A = 32.768kHz crystal, PCKEN[1:0] = 0x3, Typ. value



Current consumption-frequency characteristic during execution with OSC3B

OSC3B = ON, OSC3A = OFF, OSC1A = 32.768kHz crystal,
PCKEN[1:0] = 0x3, CCLKGR[1:0] = 0x2, Ta = 25°C, Typ. value



27.4 Oscillation Characteristics

Oscillation characteristics change depending on conditions (board pattern, components used, etc.). Use the following characteristics as reference values.

OSC1A crystal oscillation

Unless otherwise specified: V_{DD} = 2.0 to 3.6V, V_{SS} = 0V, Ta = 25°C, C_G = built-in, C_D = built-in, R_f = built-in, R_D = built-in, C_{G1} = 3pF, C_{D1} = 3pF

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Oscillation start time *1, *2, *3	t_{sta}				3	s
Built-in gate capacitance *1, *2, *3	C_G	In case of the chip		7		pF
Built-in drain capacitance	C_D	In case of the chip *1, *2		5		pF
		In case of the chip *3		4		pF

*1 Crystal resonator = C-002RX: manufactured by SEIKO EPSON (R_1 = 50kΩ Max., C_L = 7pF)
*2 Crystal resonator = MC-146: manufactured by SEIKO EPSON (R_1 = 65kΩ Max., C_L = 7pF)
*3 Crystal resonator = FC-12D: manufactured by SEIKO EPSON (R_1 = 75kΩ Max., C_L = 7pF)

OSC1B oscillation

Unless otherwise specified: V_{DD} = 2.0 to 3.6V, V_{SS} = 0V, Ta = 25°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Oscillation start time	t_{sta}				200	µs
Oscillation frequency *1 *2	f_{OSC1BC}	In case of the chip	Typ. × 0.95	32.768	Typ. × 1.05	kHz
	f_{OSC1BP}	In case of the package	Typ. × 0.94	32.768	Typ. × 1.06	kHz
Dependence of oscillation frequency on temperature *2	Tf_{OSC1B}	Frequency accuracy per ±1°C change in temperature (with reference to 25°C)		±0.12	±0.3	%/°C

*1 In chip mounting, the value may exceed the range shown above according to the mount condition on the board.
*2 Reference value

OSC3A crystal oscillation

Unless otherwise specified: V_{DD} = 2.0 to 3.6V, V_{SS} = 0V, Ta = 25°C, R_f = built-in, R_D = built-in, C_{G3} = 15pF, C_{D3} = 15pF

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Oscillation start time *1 *2	t_{sta}				20.0	ms

*1 Crystal resonator = MA-406: manufactured by SEIKO EPSON
*2 The oscillation start time varies according to the crystal resonator used and the C_{G3} and C_{D3} values.

OSC3A ceramic oscillation

Unless otherwise specified: V_{DD} = 2.0 to 3.6V, V_{SS} = 0V, Ta = 25°C, R_f = built-in, R_D = built-in

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Oscillation start time *1 *2	t_{sta}				1.0	ms

*1 Ceramic resonator = CSTCC2M00G56-R0: manufactured by Murata Manufacturing Co., Ltd. (SMD, C_{G3} = C_{D3} = 47pF built-in)
CSTCR4M00G53-R0: manufactured by Murata Manufacturing Co., Ltd. (SMD, C_{G3} = C_{D3} = 15pF built-in)
CSTLS4M00G53-B0: manufactured by Murata Manufacturing Co., Ltd. (leaded, C_{G3} = C_{D3} = 15pF built-in)
*2 The oscillation start time varies according to the ceramic resonator used and the C_{G3} and C_{D3} values.

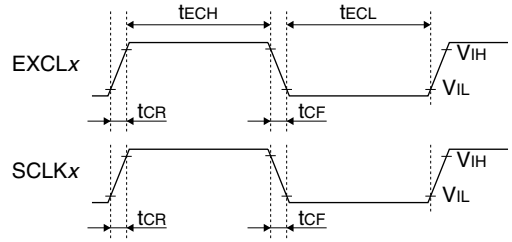
OSC3B oscillation

Unless otherwise specified: V_{DD} = 2.0 to 3.6V, V_{SS} = 0V, Ta = 25°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Oscillation start time	t_{sta}				5.0	µs
Oscillation frequency *1	f_{OSC3B}	OSC3BFSEL[1:0] = 0x0 (2MHz)	Typ. × 0.95	1.936	Typ. × 1.05	MHz
		OSC3BFSEL[1:0] = 0x1 (1MHz)		1.002		MHz
		OSC3BFSEL[1:0] = 0x2 (500kHz)		0.511		MHz
Dependence of oscillation frequency on temperature *1	Tf_{OSC3B}	OSC3BFSEL[1:0] = 0x0–0x2, Frequency accuracy per ±1°C change in temperature (with reference to 25°C)		0.05	0.07	%/°C

*1 Reference value

27.5 External Clock Input Characteristics



Unless otherwise specified: V_{DD} = 2.0 to 3.6V, V_{SS} = 0V, V_{IH} = 0.8V_{DD}, V_{IL} = 0.2V_{DD}, Ta = -40 to 85°C

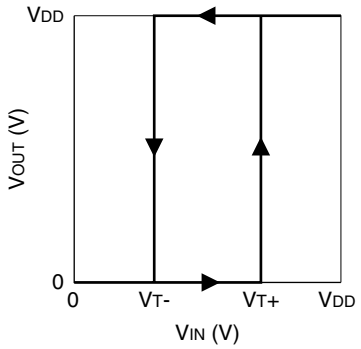
Item	Symbol	Min.	Typ.	Max.	Unit
EXCL*x* input High pulse width	t_{ECH}	60			ns
EXCL*x* input Low pulse width	t_{ECL}	60			ns
UART transfer rate	R_U			230400	bps
UART transfer rate (IrDA mode)	R_{UIrDA}			115200	bps
Input rise time	t_{CR}			80	ns
Input fall time	t_{CF}			80	ns

27.6 Input/Output Pin Characteristics

Unless otherwise specified: V_{DD} = 2.0 to 3.6V, V_{SS} = 0V, Ta = -40 to 85°C

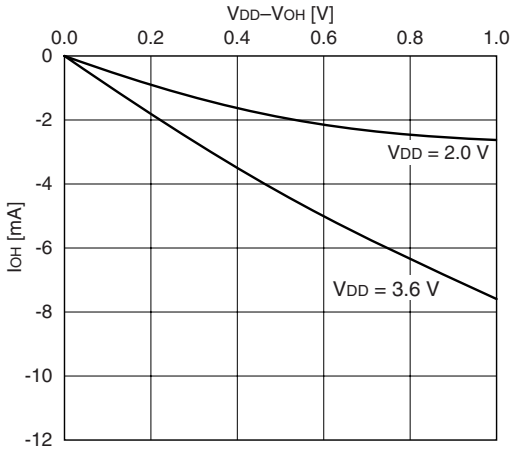
Item	Symbol	Condition	Min.	Typ.	Max.	Unit
High level Schmitt input threshold voltage	V_{T+}	P*xx*, #RESET	0.5V_{DD}		0.9V_{DD}	V
Low level Schmitt input threshold voltage	V_{T-}	P*xx*, #RESET	0.1V_{DD}		0.5V_{DD}	V
Hysteresis voltage	ΔV_T	P*xx*, #RESET	0.1V_{DD}			V
High level output current	I_{OH}	P*xx*, V_{OH} = 0.9V_{DD}			-0.5	mA
Low level output current	I_{OL}	P*xx*, V_{OL} = 0.1V_{DD}	0.5			mA
Leakage current	I_{LEAK}	P*xx*, #RESET	-100		100	nA
Input pull-up resistance	R_{IN}	P*xx*, #RESET	100		500	kΩ
Pin capacitance	C_{IN}	P*xx*, V_{IN} = 0V, f = 1MHz, Ta = 25°C			15	pF
Reset low pulse width	t_{SR}	V_{IH} = 0.8V_{DD}, V_{IL} = 0.2V_{DD}	100			µs
Operating power voltage	V_{SR}		2.0			V
#RESET power-on reset time	t_{PSR}		1.0			ms

Schmitt input threshold voltage



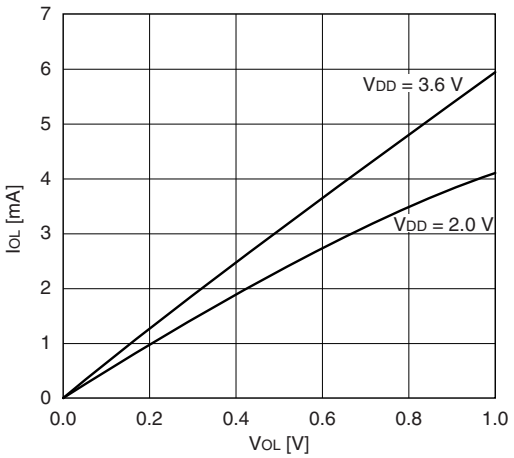
High-level output current characteristic

Ta = 85°C, Max. value

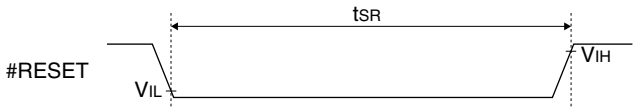


Low-level output current characteristic

Ta = 85°C, Min. value



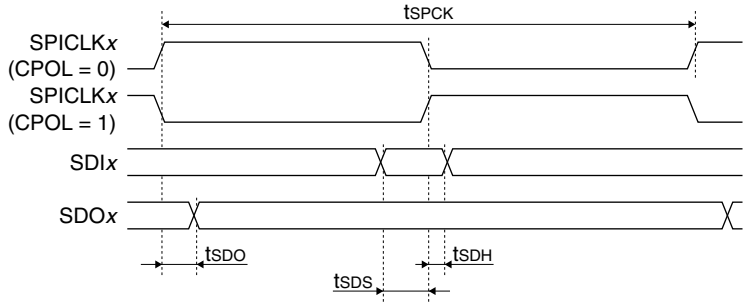
Reset pulse



#RESET power-on reset timing

Note: Be sure to set the #RESET pin to 0.1 VDD or less when performing a power-on reset after the power is turned off.

27.7 SPI Characteristics



Master mode

Unless otherwise specified: VDD = 2.0 to 3.6V, VSS = 0V, Ta = -40 to 85°C

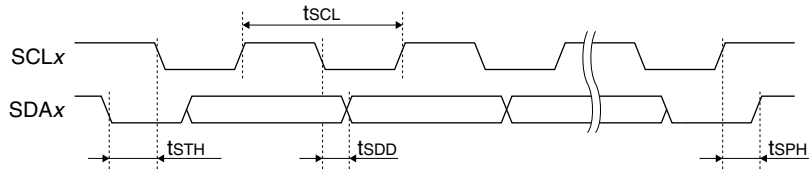
Item	Symbol	Min.	Typ.	Max.	Unit
SPICLK*x* cycle time	tSPCK	500			ns
SDI*x* setup time	tSDS	70			ns
SDI*x* hold time	tSDH	10			ns
SDO*x* output delay time	tSDO			20	ns

Slave mode

Unless otherwise specified: V_{DD} = 2.0 to 3.6V, V_{SS} = 0V, Ta = -40 to 85°C

Item	Symbol	Min.	Typ.	Max.	Unit
SPICLK*x* cycle time	t_{SPCK}	500			ns
SDI*x* setup time	t_{SDS}	10			ns
SDI*x* hold time	t_{SDH}	10			ns
SDO*x* output delay time	t_{SDO}			80	ns

27.8 I²C Characteristics



Unless otherwise specified: V_{DD} = 2.0 to 3.6V, V_{SS} = 0V, Ta = -40 to 85°C

Item	Symbol	Min.	Typ.	Max.	Unit
SCL cycle time	t_{SCL}	2500			ns
Start condition hold time	t_{STH}	1/f_{SYS}			ns
Data output delay time	t_{SDD}	1/f_{SYS}			ns
Stop condition hold time	t_{SPH}	1/f_{SYS}			ns

* f_{SYS}: System operating clock frequency

27.9 EPD Driver Characteristics

The typical values in the following EPD driver characteristics varies depending on the panel load (panel size, drive waveform, number of display pixels and display contents), so evaluate them by connecting to the actually used EPD panel.

V_E regulator output voltage

Unless otherwise specified: V_{DD} = 2.0 to 3.6V, V_{SS} = 0V, Ta = 25°C, C_3–C_{10} = 0.1µF, No panel load

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
V_E regulator output voltage (V_{E1} reference, VESEL = 0)	V_{E1}	VECON[3:0] = 0x0	Typ. - 0.075	1.005	Typ. + 0.075	V
		VECON[3:0] = 0x1		1.055		V
		VECON[3:0] = 0x2		1.106		V
		VECON[3:0] = 0x3		1.156		V
		VECON[3:0] = 0x4		1.206		V
		VECON[3:0] = 0x5		1.256		V
		VECON[3:0] = 0x6		1.307		V
		VECON[3:0] = 0x7		1.357		V
		VECON[3:0] = 0x8		1.407		V
		VECON[3:0] = 0x9		1.457		V
		VECON[3:0] = 0xa		1.508		V
		VECON[3:0] = 0xb		1.558		V
		VECON[3:0] = 0xc		1.608		V
		VECON[3:0] = 0xd		1.658		V
		VECON[3:0] = 0xe		1.709		V
		VECON[3:0] = 0xf		1.759		V

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
V_E regulator output voltage (V_{E2} reference, VESEL = 1)	V_{E2}	VECON[3:0] = 0x0		2.010		V
		VECON[3:0] = 0x1		2.111		V
		VECON[3:0] = 0x2		2.211		V
		VECON[3:0] = 0x3		2.312		V
		VECON[3:0] = 0x4		2.412		V
		VECON[3:0] = 0x5		2.513		V
		VECON[3:0] = 0x6		2.613		V
		VECON[3:0] = 0x7	Typ.	2.714	Typ.	V
		VECON[3:0] = 0x8	- 0.150	2.814	+ 0.150	V
		VECON[3:0] = 0x9		2.915		V
		VECON[3:0] = 0xa		3.015		V
		VECON[3:0] = 0xb		3.116		V
		VECON[3:0] = 0xc		3.216		V
		VECON[3:0] = 0xd		3.317		V
		VECON[3:0] = 0xe		3.417		V
		VECON[3:0] = 0xf		3.518		V

EPD drive voltage

Unless otherwise specified: V_{DD} = 2.0 to 3.6V, V_{SS} = 0V, Ta = 25°C, C_3–C_{10} = 0.1µF, No panel load

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
9V type EPD drive voltage (VHSEL[1:0] = 0x2)	V_{EPD}	VHCON[3:0] = 0x0		8.10		V
		VHCON[3:0] = 0x1		8.28		V
		VHCON[3:0] = 0x2		8.46		V
		VHCON[3:0] = 0x3		8.64		V
		VHCON[3:0] = 0x4		8.82		V
		VHCON[3:0] = 0x5		9.00		V
		VHCON[3:0] = 0x6		9.18		V
		VHCON[3:0] = 0x7	Typ.	9.36	Typ.	V
		VHCON[3:0] = 0x8	- 0.75	9.54	+ 0.75	V
		VHCON[3:0] = 0x9		9.72		V
		VHCON[3:0] = 0xa		9.90		V
		VHCON[3:0] = 0xb		10.08		V
		VHCON[3:0] = 0xc		10.26		V
		VHCON[3:0] = 0xd		10.44		V
		VHCON[3:0] = 0xe		10.62		V
		VHCON[3:0] = 0xf		10.80		V
12V type EPD drive voltage (VHSEL[1:0] = 0x1)	V_{EPD}	VHCON[3:0] = 0x0		10.80		V
		VHCON[3:0] = 0x1		11.04		V
		VHCON[3:0] = 0x2		11.28		V
		VHCON[3:0] = 0x3		11.52		V
		VHCON[3:0] = 0x4		11.76		V
		VHCON[3:0] = 0x5		12.00		V
		VHCON[3:0] = 0x6		12.24		V
		VHCON[3:0] = 0x7	Typ.	12.48	Typ.	V
		VHCON[3:0] = 0x8	- 0.75	12.72	+ 0.75	V
		VHCON[3:0] = 0x9		12.96		V
		VHCON[3:0] = 0xa		13.20		V
		VHCON[3:0] = 0xb		13.44		V
		VHCON[3:0] = 0xc		13.68		V
		VHCON[3:0] = 0xd		13.92		V
		VHCON[3:0] = 0xe		14.16		V
		VHCON[3:0] = 0xf		14.40		V

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
15V type EPD drive voltage (VHSEL[1:0] = 0x0)	V_{EPD}	VHCON[3:0] = 0x0		13.50		V
		VHCON[3:0] = 0x1		13.80		V
		VHCON[3:0] = 0x2		14.10		V
		VHCON[3:0] = 0x3		14.40		V
		VHCON[3:0] = 0x4		14.70		V
		VHCON[3:0] = 0x5		15.00		V
		VHCON[3:0] = 0x6		15.30		V
		VHCON[3:0] = 0x7	Typ.	15.60	Typ.	V
		VHCON[3:0] = 0x8	- 0.75	15.90	+ 0.75	V
		VHCON[3:0] = 0x9		16.20		V
		VHCON[3:0] = 0xa		16.50		V
		VHCON[3:0] = 0xb		16.80		V
		VHCON[3:0] = 0xc		17.10		V
		VHCON[3:0] = 0xd		17.40		V
		VHCON[3:0] = 0xe		17.70		V
		VHCON[3:0] = 0xf		18.00		V

ESEG/ETP/EBP output characteristics

Unless otherwise specified: V_{DD} = 2.0 to 3.6V, V_{SS} = 0V, Ta = -40 to 85°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Segment/top plane/back plane output current	I_{SEGH}	ESEG*xx*, ETP*x*, EBP*x*, V_{SEGH} = V_{EPD}			-10	µA
	I_{SEGL}	ESEG*xx*, ETP*x*, EBP*x*, V_{SEGL} = V_{SS}	10			µA

EPD driver circuit current consumption

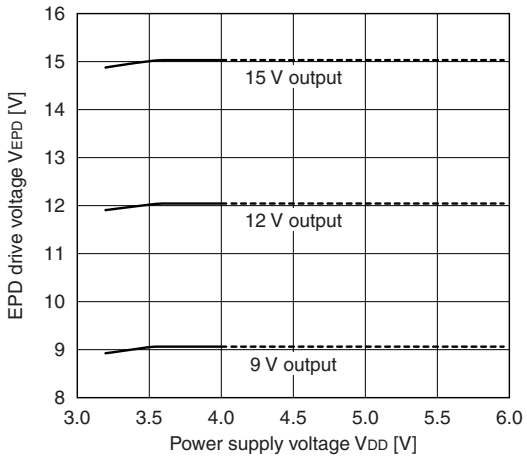
Unless otherwise specified: V_{DD} = 2.0 to 3.6V, V_{SS} = 0V, Ta = 25°C, C_3–C_{10} = 0.1µF, No EPD panel load, PCKEN[1:0] = 0x3 (ON), OSC1A = 32kHz, OSC3B = OFF, OSC3A = OFF, EPDTCLKSRC[1:0] = 0x1 (OSC1), EPDTCLKD[2:0] = 0x0 (1/1), Booster clock = 16kHz, Doubler clock = 32kHz

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
EPD circuit current *1	I_{EPD}	Reference voltage V_{E1} (= 1.759V)		75	100	µA
EPD circuit current in heavy load protection mode *1	I_{EPDH}	Reference voltage V_{E1} (= 1.759V), HVLDVE = 1		85		µA
		Reference voltage V_{E1} (= 1.759V), HVLDVH = 1		120		µA

*1 This value is added to the current consumption during HALT/execution (with or without heavy load protection mode) when the EPD circuit is active. Current consumption increases according to the drive waveforms and panel load.

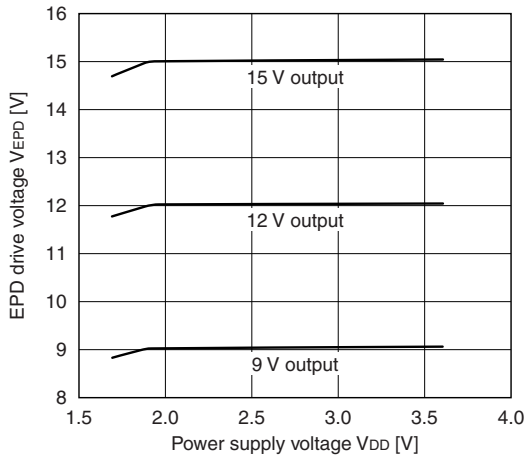
EPD drive voltage-power supply voltage characteristic (V_{E2} reference)

V_{E2} = 3.518V, Ta = 25°C, 20µA load, Typ. value



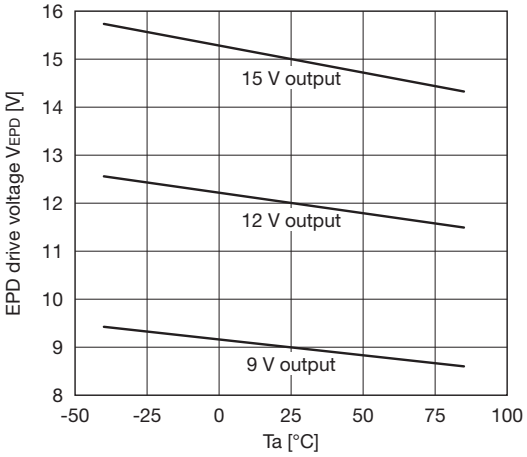
EPD drive voltage-power supply voltage characteristic (V_{E1} reference)

V_{E1} = 1.759V, Ta = 25°C, 10µA load, Typ. value



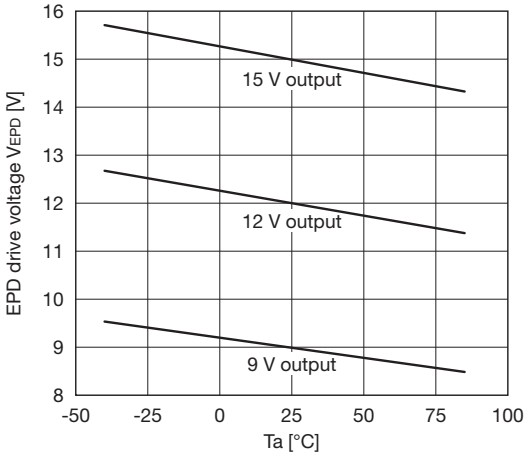
EPD drive voltage-temperature characteristic (V_{E2} reference)

V_{E2} = 3.518V, V_{DD} = 3.6V, 20µA load, Typ. value



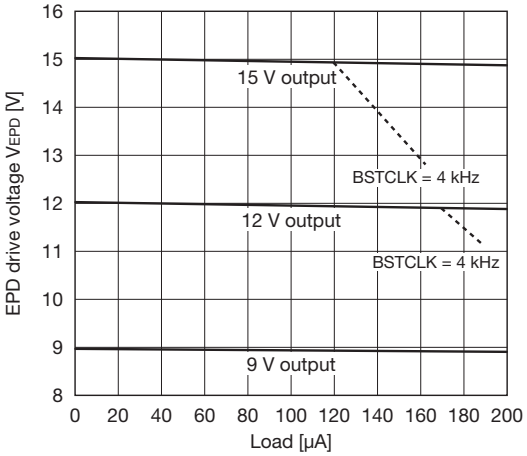
EPD drive voltage-temperature characteristic (V_{E1} reference)

V_{E1} = 1.759V, V_{DD} = 2.0V, 10µA load, Typ. value



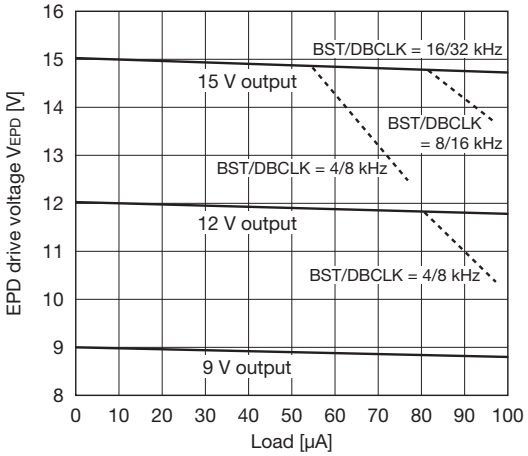
EPD drive voltage-load characteristic (V_{E2} reference)

V_{E2} = 3.518V, Ta = 25°C, V_{DD} = 3.6V, Typ. value
BSTCLK = 4kHz, 8kHz, or 16kHz



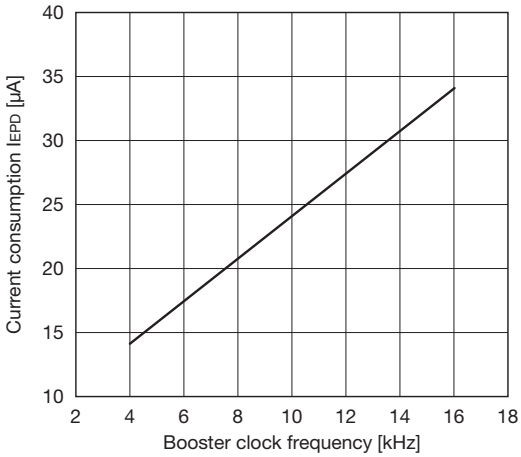
EPD drive voltage-load characteristic (V_{E1} reference)

V_{E1} = 1.759V, Ta = 25°C, V_{DD} = 2.0V, Typ. value
BST/DBCLK = 4/8kHz, 8/16kHz, or 16/32kHz



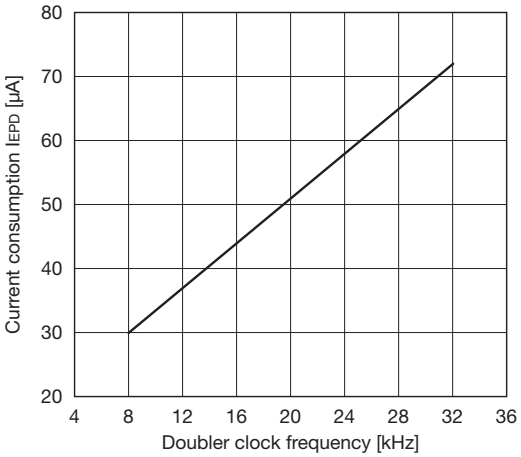
EPD driver circuit current consumption-booster clock frequency dependence (V_{E2} reference)

V_{E2} = 3.518V, Ta = 25°C, V_{DD} = 3.6V, No load, Typ. value



EPD driver circuit current consumption-doubler/booster clock frequency dependence (V_{E1} reference)

V_{E1} = 1.759V, Ta = 25°C, V_{DD} = 2.0V, No load, Typ. value
Booster clock frequency = Doubler clock frequency / 2



27.10 R/F Converter Characteristics

Analog characteristics

Unless otherwise specified: V_{DD} = 2.0 to 3.6V, V_{SS} = 0V, Ta = -40 to 85°C

Item	Symbol	Condition		Min.	Typ.	Max.	Unit
Reference/sensor oscillation frequency *1	f_{RFCLK}			1		4000	kHz
Reference/sensor oscillation frequency IC deviation *2	$\Delta f_{RFCLK}/\Delta IC$	Resistive sensor DC/AC oscillation mode	V_{DD} = 3.6V	-25		25	%
			V_{DD} = 2.0V	-50		50	%
Reference resistor/resistive sensor resistance *3	R_{REF}, R_{SEN}	Resistive sensor DC oscillation mode		10			kΩ
		Resistive sensor AC oscillation mode		10			kΩ
Time base counter clock frequency	f_{TCCLK}					4.2	MHz
RFIN pin high level Schmitt input voltage	V_{T+}			0.5•V_{DD}		0.9•V_{DD}	V
RFIN pin low level Schmitt input voltage	V_{T-}			0.1•V_{DD}		0.5•V_{DD}	V

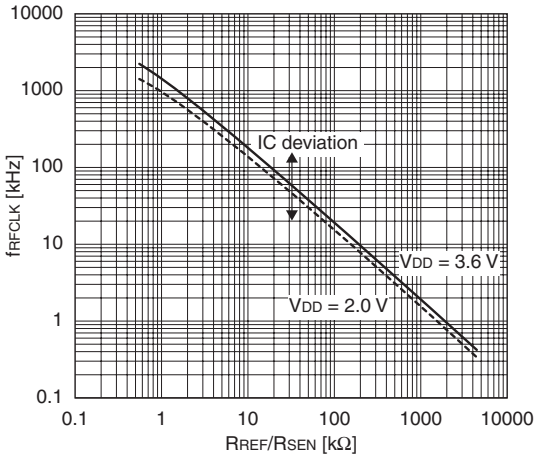
*1 The oscillation frequency/IC deviation characteristic value may increase due to variations in oscillation frequency caused by leakage if the oscillation frequency is 1 kHz or lower.

*2 In these characteristics, unevenness between production lots, and variations in board, resistances and capacitances used in the measurement environment are taken into account (variations in temperature are not included).

*3 The CR oscillation can be performed if the resistance or capacitance is out of the range shown in the table (see characteristic curves), note, however, that the oscillation frequency/IC deviation characteristic value may increase due to parasitic elements on the board and those in the IC.

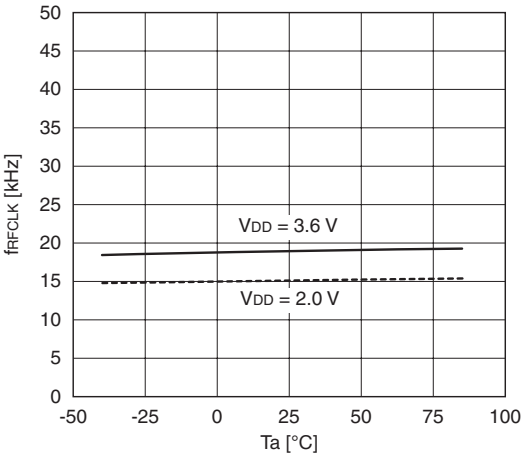
RFC reference/sensor oscillation frequency-resistance characteristic (Resistive sensor DC/AC oscillation mode)

C_{REF} = 1000pF, Ta = 25°C, Typ. value



RFC reference/sensor oscillation frequency-temperature characteristic (Resistive sensor DC/AC oscillation mode)

R_{REF}/R_{SEN} = 100kΩ, C_{REF} = 1000pF, Typ. value



R/F converter current consumption

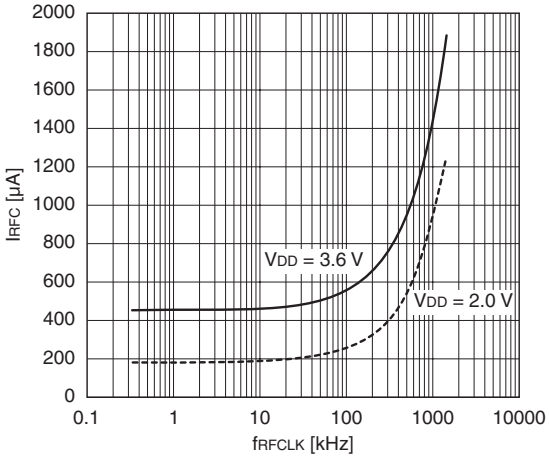
Unless otherwise specified: V_{DD} = 3.6V, V_{SS} = 0V, Ta = 25°C, PCKEN[1:0] = 0x3 (ON), C_{REF} = 1000pF, R_{REF}/R_{SEN} = 100kΩ

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
R/F converter operating current *1	I_{RFC}	Resistive sensor DC/AC oscillation mode		270	330	µA

*1 This value is added to the current consumption during HALT/execution (with or without heavy load protection mode) when the R/F converter is active. Current consumption depends on the V_{DD} voltage, reference capacitance, and reference/sensor oscillation frequency.

RFC reference/sensor oscillation current consumption-frequency characteristic (Resistive sensor DC/AC oscillation mode)

R_{REF}/R_{SEN} = 100kΩ, C_{REF} = 1000pF, Typ. value



27.11 Temperature Detection Circuit Characteristics

Analog characteristics

Unless otherwise specified: V_{DD} = 2.0 to 3.6V, V_{SS} = 0V, Ta = -40 to 85°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Resolution	–			8		bit
Temperature sensor output voltage stabilization time	t_{STAB}				10	ms
Temperature sensor output voltage comparison time	t_{CMP}		150			µs
Temperature detection range	T_{RNG}		-40		85	°C
Temperature detection error	E_{TEM}	T_{RNG} = 0 to 50°C		±2	±5	°C

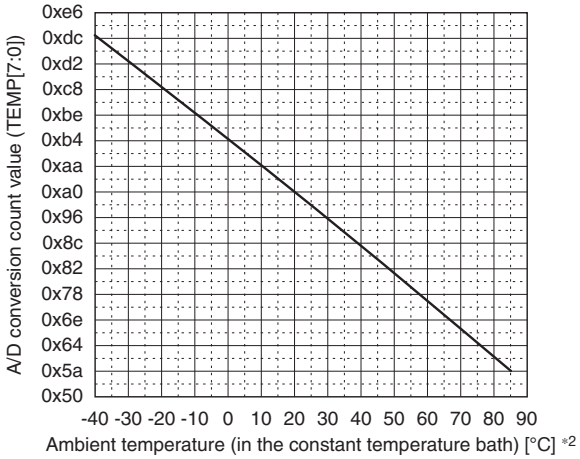
Temperature detection circuit current consumption

Unless otherwise specified: V_{DD} = 2.0 to 3.6V, V_{SS} = 0V, Ta = 25°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Temperature detection circuit operating current *1	I_{TEM}			6	12	µA

*1 This value is added to the current consumption during HALT/execution (with or without heavy load protection mode) when the temperature detection circuit is active.

Temperature detection circuit conversion value -ambient temperature characteristic



*2 This characteristic was measured in 150 µs of the conversion time (comparison time) after canceling SLEEP status by placing the device in the constant temperature bath.
The temperature detection circuit measures temperature inside the device. Depending on the use environment, the difference between the measured temperature and the ambient temperature may be increased. When using the measured temperature as the ambient temperature, prepare an appropriate conversion table according to the use environment.

27.12 SVD Circuit Characteristics

Analog characteristics

Unless otherwise specified: V_{DD} = 2.0 to 3.6V, V_{SS} = 0V, Ta = -40 to 85°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
SVD voltage	V_{SVD}	SVDC[4:0] = 0x0		–		V
		SVDC[4:0] = 0x1		–		V
		SVDC[4:0] = 0x2		–		V
		SVDC[4:0] = 0x3		–		V
		SVDC[4:0] = 0x4		–		V
		SVDC[4:0] = 0x5		–		V
		SVDC[4:0] = 0x6		–		V
		SVDC[4:0] = 0x7		–		V
		SVDC[4:0] = 0x8		–		V
		SVDC[4:0] = 0x9		–		V
		SVDC[4:0] = 0xa		–		V
		SVDC[4:0] = 0xb		–		V
		SVDC[4:0] = 0xc		–		V
		SVDC[4:0] = 0xd		–		V
		SVDC[4:0] = 0xe		2.00		V
		SVDC[4:0] = 0xf	Typ. ×	2.10	Typ. ×	V
		SVDC[4:0] = 0x10	0.96	2.20	1.04	V
		SVDC[4:0] = 0x11		2.30		V
		SVDC[4:0] = 0x12		2.40		V
		SVDC[4:0] = 0x13		2.50		V
		SVDC[4:0] = 0x14		2.60		V
		SVDC[4:0] = 0x15		2.70		V
		SVDC[4:0] = 0x16		2.80		V
		SVDC[4:0] = 0x17		2.90		V
		SVDC[4:0] = 0x18		3.00		V
		SVDC[4:0] = 0x19		3.10		V
		SVDC[4:0] = 0x1a		3.20		V
		SVDC[4:0] = 0x1b		–		V
		SVDC[4:0] = 0x1c		–		V
		SVDC[4:0] = 0x1d		–		V
		SVDC[4:0] = 0x1e		–		V
		SVDC[4:0] = 0x1f		–		V
SVD circuit-enable response time *1	t_{SVDEN}				500	µs
SVD circuit response time *2	t_{SVD}				60	µs

*1 This time is required to obtain stable detection results after SVDEN is altered from 0 to 1.
*2 This time is required to obtain stable detection results after SVDC[4:0] is altered.

SVD circuit current consumption

Unless otherwise specified: V_{DD} = 2.0 to 3.6V, V_{SS} = 0V, Ta = 25°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
SVD circuit current *1	I_{SVD}	V_{DD} = 3.6V, SVDC[4:0] = 0xe (2.0V)		12	17	µA

*1 This value is added to the current consumption during SLEEP/HALT/execution (with or without heavy load protection mode) when the SVD circuit is active.

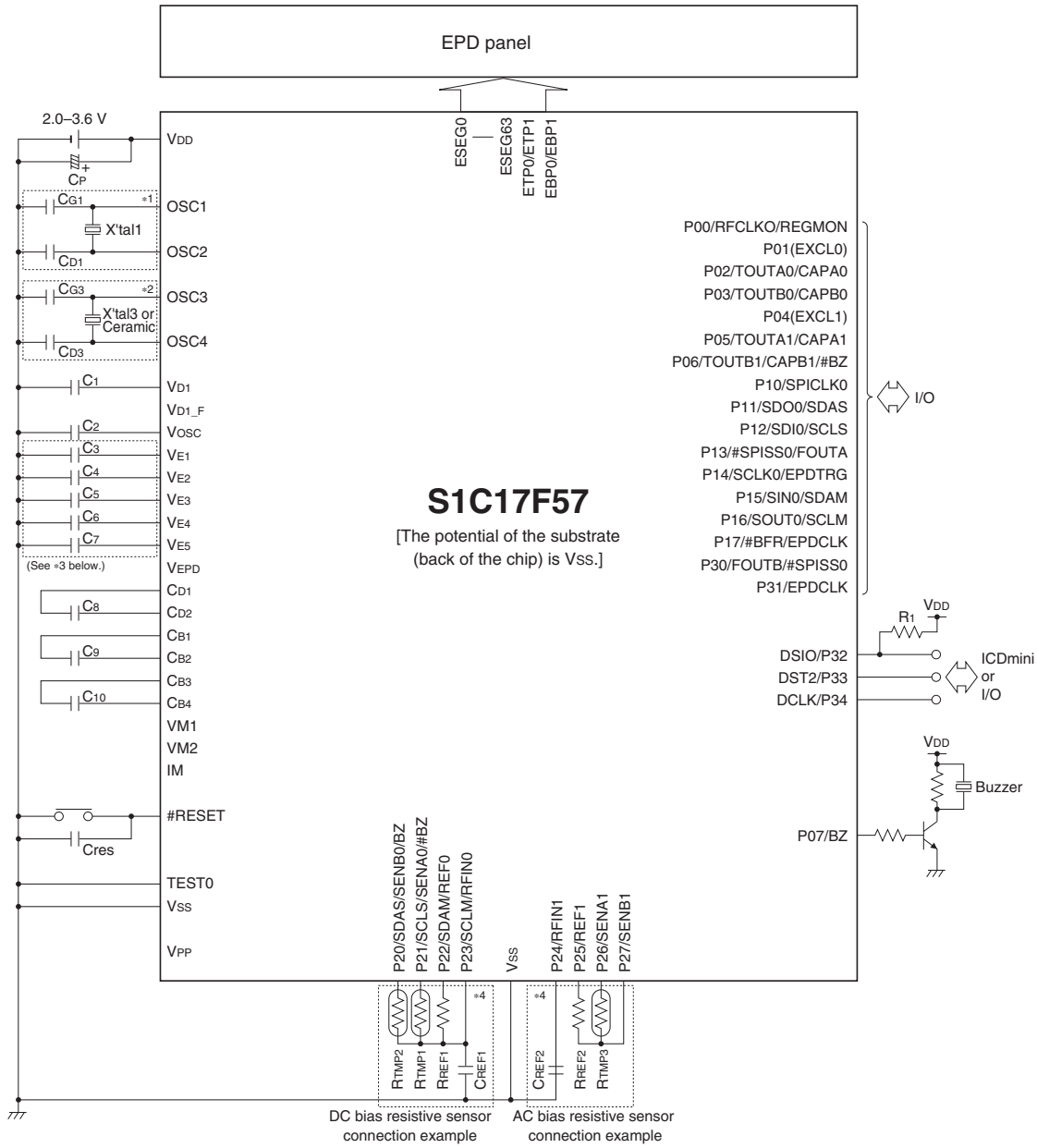
27.13 Flash Memory Characteristics

Unless otherwise specified: V_{DD} = 2.0 to 3.6V, V_{PP} = 7.0V (for programming)/7.5V (for erasing), V_{SS} = 0V *1, Ta = 10 to 40°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Programming count *2	C_{FEP}	Programmed data is guaranteed to be retained for 10 years.	3			times

*1 The potential variation of the V_{SS} voltage should be suppressed to within ±0.3 V on the basis of the ground potential of the MCU mounting board while the Flash is being programmed, as it affects the Flash memory characteristics (programming count).
*2 Assumed that Erasing + Programming as count of 1. The count includes programming in the factory.

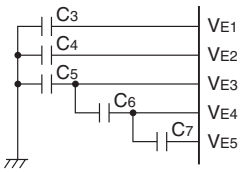
28 Basic External Connection Diagram



*1: This external circuit is required only when the OSC1A oscillator is used.

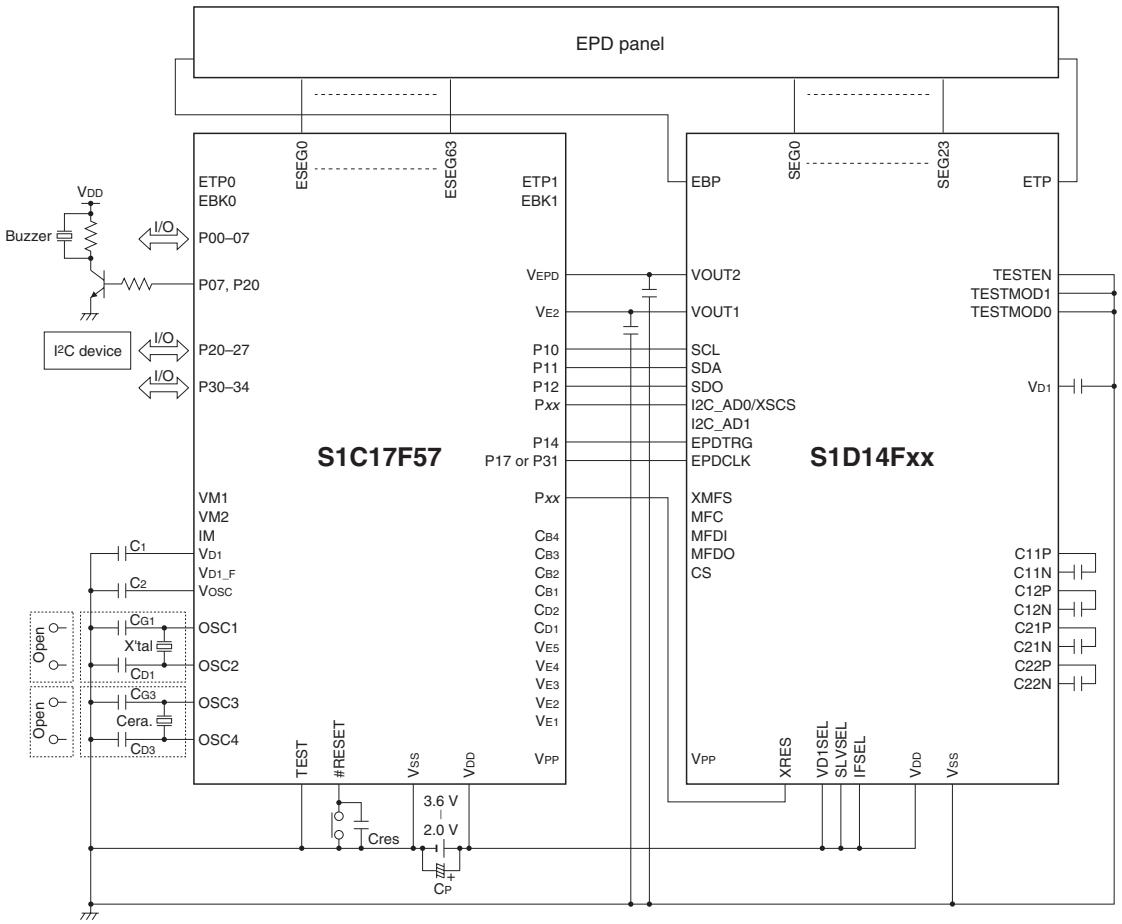
*2: This external circuit is required only when the OSC3A oscillator is used.

*3: The figure below shows a connection example to lower the maximum operating voltage of the external components for the EPD power supply circuit. Note, however, that it decreases the EPD power supply capability. When using this connection method, be sure to evaluate the EPD power supply capability using an actual device.



*4: This external circuit is required only when the R/F converter (RFC) is used.

External EPD driver connection example
(when the EPD power supply in the external EPD driver is used)



Recommended values for external parts

External parts for the theoretical regulation type OSC1A oscillator circuit

Symbol	Resonator	Recommended manufacturer	Frequency [Hz]	Product number	Recommended values		Recommended operating condition
					C_{D1} [pF]	C_{G1} [pF]	Temperature range [°C]
X'tal1	Crystal	Seiko Epson Corporation	32.768k	C-002RX (R_1 = 50 kΩ (Max.), C_L = 7 pF)	3	3	-10 to 60°C
				MC-146 (R_1 = 65 kΩ (Max.), C_L = 7 pF)	3	3	-40 to 85°C

External parts for the OSC3A oscillator circuit

Symbol	Resonator	Recommended manufacturer	Frequency [Hz]	Product number	Recommended values *		Recommended operating condition
					C_{D3} [pF]	C_{G3} [pF]	Temperature range [°C]
X'tal3	Crystal	Seiko Epson Corporation	4M	MA-406	15	15	-20 to 70°C
Ceramic	Ceramic	Murata Manufacturing Co., Ltd.	2M	CSTCC2M00G56-R0 (SMD)	(47)	(47)	-40 to 85°C
			4M	CSTCR4M00G53-R0 (SMD)	(15)	(15)	-40 to 85°C
			4M	CSTLS4M00G53-B0 (lead)	(15)	(15)	-40 to 85°C

* The C_{D3} and C_{G3} values enclosed with ( ) are the built-in capacitances of the resonator.

Other

Symbol	Name	Recommended value	Maximum operating voltage
CP	Capacitor for power supply	3.3 µF	3.6 V
C_{G1}	Gate capacitor	3 pF	3.6 V
C_{D1}	Drain capacitor	3 pF	3.6 V
C_{G3}	Gate capacitor	15 pF	3.6 V
C_{D3}	Drain capacitor	15 pF	3.6 V
Cres	Capacitor for #RESET pin	0.47 µF	3.6 V
C_1	Capacitor between V_{D1} and V_{SS}	0.1 µF	3.6 V
C_2	Capacitor between V_{OSC} and V_{SS}	0.1 µF	3.6 V
C_3	Capacitor between V_{E1} and V_{SS}	0.1 µF	3.6 V
C_4	Capacitor between V_{E2} and V_{SS}	0.1 µF	3.6 V
C_5	Capacitor between V_{E3} and V_{SS}	0.1 µF	7.2 V
C_6	Capacitor between V_{E4} and V_{SS}	0.1 µF	14.4 V (7.2 V*)
C_7	Capacitor between V_{E5} and V_{SS}	0.1 µF	21.6 V (7.2 V*)
C_8	Capacitor between C_{D1} and C_{D2}	0.1 µF	7.2 V
C_9	Capacitor between C_{B1} and C_{B2}	0.1 µF	14.4 V (7.2 V*)
C_{10}	Capacitor between C_{B3} and C_{B4}	0.1 µF	21.6 V (7.2 V*)
R_1	DSIO pull-up resistor	10 kΩ	–

* When the external capacitors for the EPD power supply circuit are connected as shown in the basic external connection diagram (*3)

Notes: • The values in the above table are shown only for reference and not guaranteed.

• Crystal and ceramic resonators are extremely sensitive to influence of external components and printed-circuit boards. Before using a resonator, please contact the manufacturer for further information on conditions of use.

29 Chip

29.1 Pad/Bump Configuration

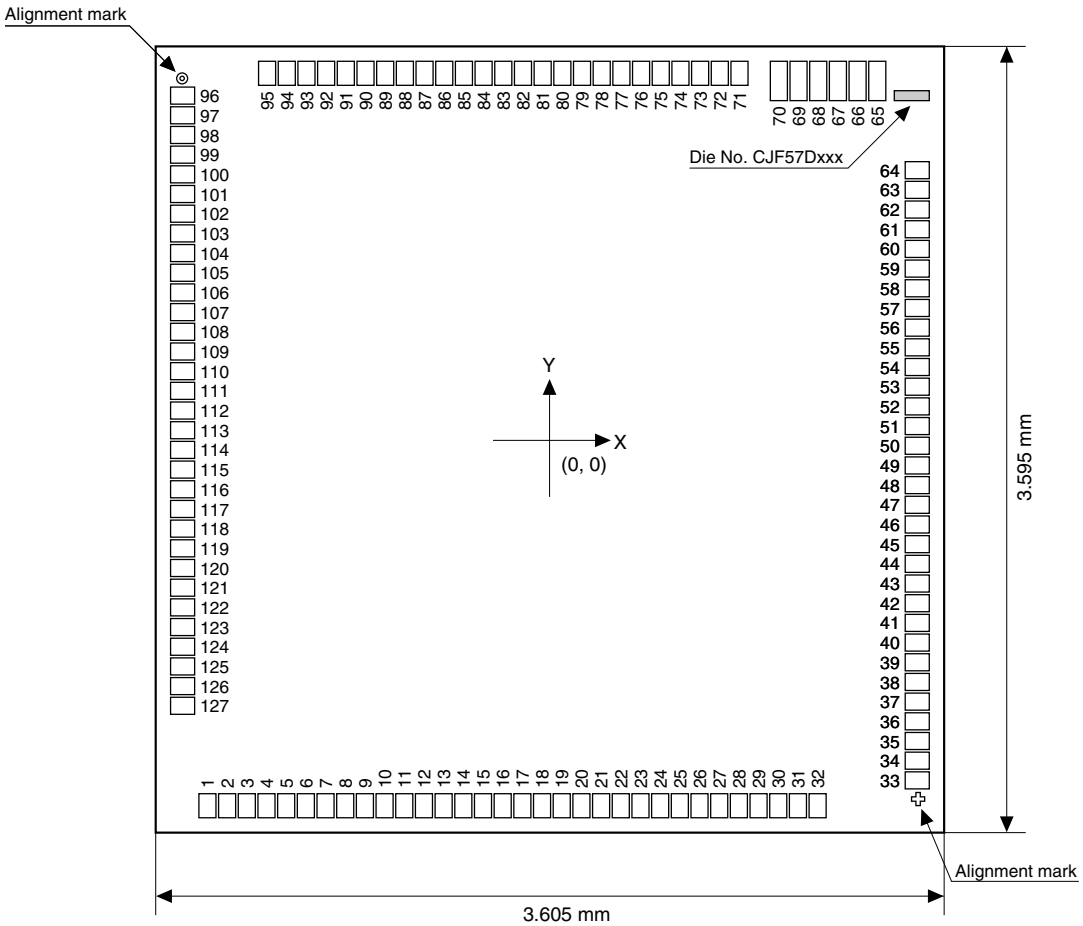


Figure 29.1.1 S1C17F57 Pad/Bump Configuration Diagram

Chip size	X = 3.605 mm, Y = 3.595 mm
Pad opening	No. 1 to 32, 71 to 95: X = 76 µm, Y = 110 µm No. 33 to 64, 96 to 127: X = 110 µm, Y = 76 µm No. 65 to 70: X = 76 µm, Y = 180 µm
Bump size	No. 1 to 32, 71 to 95: X = 70 µm, Y = 104 µm No. 33 to 64, 96 to 127: X = 104 µm, Y = 70 µm No. 65 to 70: X = 70 µm, Y = 174 µm
Chip thickness	200 µm
Alignment mark coordinates	Upper left: X = -1678.1 µm, Y = 1673.1 µm Lower right: X = 1678.1 µm, Y = -1673.1 µm

Alignment marks

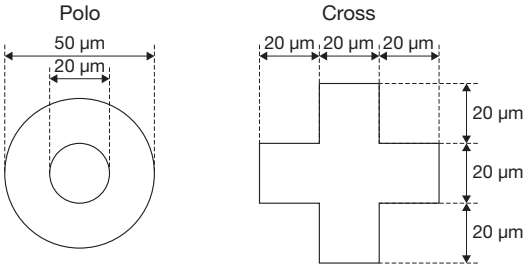


Figure 29.1.2 Alignment Marks

Table 29.1.1 S1C17F57 Pad/Bump Coordinates

No.	Name	X (µm)	Y (µm)
1	**ESEG8**	-1566.0	-1674.5
2	**ESEG9**	-1476.0	-1674.5
3	**ESEG10**	-1386.0	-1674.5
4	**ESEG11**	-1296.0	-1674.5
5	**ESEG12**	-1206.0	-1674.5
6	**ESEG13**	-1116.0	-1674.5
7	**ESEG14**	-1026.0	-1674.5
8	**ESEG15**	-936.0	-1674.5
9	**ESEG16**	-846.0	-1674.5
10	**ESEG17**	-756.0	-1674.5
11	**ESEG18**	-666.0	-1674.5
12	**ESEG19**	-576.0	-1674.5
13	**ESEG20**	-486.0	-1674.5
14	**ESEG21**	-396.0	-1674.5
15	**ESEG22**	-306.0	-1674.5
16	**ESEG23**	-216.0	-1674.5
17	**ESEG24**	-126.0	-1674.5
18	**ESEG25**	-36.0	-1674.5
19	**ESEG26**	54.0	-1674.5
20	**ESEG27**	144.0	-1674.5
21	**ESEG28**	234.0	-1674.5
22	**ESEG29**	324.0	-1674.5
23	**ESEG30**	414.0	-1674.5
24	**ESEG31**	504.0	-1674.5
25	**ESEG32**	594.0	-1674.5
26	**ESEG33**	684.0	-1674.5
27	**ESEG34**	774.0	-1674.5
28	**ESEG35**	864.0	-1674.5
29	**ESEG36**	954.0	-1674.5
30	**ESEG37**	1044.0	-1674.5
31	**ESEG38**	1134.0	-1674.5
32	**ESEG39**	1224.0	-1674.5
33	**ESEG40**	1679.5	-1558.5
34	**ESEG41**	1679.5	-1468.5
35	**ESEG42**	1679.5	-1378.5
36	**ESEG43**	1679.5	-1288.5
37	**ESEG44**	1679.5	-1198.5
38	**ESEG45**	1679.5	-1108.5
39	**ESEG46**	1679.5	-1018.5
40	**ESEG47**	1679.5	-928.5
41	**ESEG48**	1679.5	-838.5
42	**ESEG49**	1679.5	-748.5
43	**ESEG50**	1679.5	-658.5
44	**ESEG51**	1679.5	-568.5
45	**ESEG52**	1679.5	-478.5
46	**ESEG53**	1679.5	-388.5
47	**ESEG54**	1679.5	-298.5
48	**ESEG55**	1679.5	-208.5
49	**ESEG56**	1679.5	-118.5
50	**ESEG57**	1679.5	-28.5
51	**ESEG58**	1679.5	61.5
52	**ESEG59**	1679.5	151.5
53	**ESEG60**	1679.5	241.5
54	**ESEG61**	1679.5	331.5
55	**ESEG62**	1679.5	421.5
56	**ESEG63**	1679.5	511.5
57	**EBP1**	1679.5	601.5
58	**ETP1**	1679.5	691.5
59	**V_{EPD}**	1679.5	781.5
60	**V_{E5}**	1679.5	871.5
61	**V_{E4}**	1679.5	961.5
62	**V_{E3}**	1679.5	1051.5
63	**C_{B4}**	1679.5	1141.5
64	**C_{B3}**	1679.5	1231.5
65	**C_{B2}**	1496.8	1639.5
66	**C_{B1}**	1406.8	1639.5
67	**V_{E2}**	1316.8	1639.5
68	**V_{E1}**	1226.8	1639.5
69	**C_{D2}**	1136.8	1639.5
70	**C_{D1}**	1046.8	1639.5
71	**VM1**	866.8	1674.5
72	**VM2**	776.8	1674.5
73	**IM**	686.8	1674.5
74	**V_{SS}**	596.8	1674.5
75	**OSC1**	506.8	1674.5
76	**OSC2**	416.8	1674.5
77	**V_{OSC}**	326.8	1674.5
78	**OSC3**	236.8	1674.5
79	**OSC4**	146.8	1674.5
80	**V_{DD}**	56.8	1674.5
81	**V_{D1}**	-33.2	1674.5
82	**V_{D1_F}**	-123.2	1674.5
83	**P00**/RFCLKO/REGMON	-213.2	1674.5
84	**P01**/EXCL0	-303.2	1674.5
85	**P02**/TOUTA0/CAPA0	-393.2	1674.5
86	**P03**/TOUTB0/CAPB0	-483.2	1674.5
87	**P04**/EXCL1	-573.2	1674.5
88	**P05**/TOUTA1/CAPA1	-663.2	1674.5
89	**P06**/TOUTB1/CAPB1/#BZ	-753.2	1674.5
90	**P07**/BZ	-843.2	1674.5
91	**P10**/SPICLK0	-933.2	1674.5
92	**P11**/SDO0/SDAS	-1023.2	1674.5
93	**P12**/SDI0/SCLS	-1113.2	1674.5
94	**P13**/#SPISS0/FOUTA	-1203.2	1674.5
95	**P14**/SCLK0/EPDTRG	-1293.2	1674.5
96	**P15**/SIN0/SDAM	-1679.5	1572.5
97	**P16**/SOUT0/SCLM	-1679.5	1482.5
98	**P17**/#BFR/EPDCLK	-1679.5	1392.5
99	**P20**/SDAS/SENB0/BZ	-1679.5	1302.5
100	**P21**/SCLS/SENA0/#BZ	-1679.5	1212.5
101	**P22**/SDAM/REF0	-1679.5	1122.5
102	**P23**/SCLM/RFIN0	-1679.5	1032.5
103	**V_{SS}**	-1679.5	942.5
104	**P24**/RFIN1	-1679.5	852.5
105	**P25**/REF1	-1679.5	762.5
106	**P26**/SENA1	-1679.5	672.5
107	**P27**/SENB1	-1679.5	582.5
108	**P30**/FOUTB/#SPISS0	-1679.5	492.5
109	**P31**/EPDCLK	-1679.5	402.5
110	**DSIO**/P32	-1679.5	312.5
111	**DST2**/P33	-1679.5	222.5
112	**DCLK**/P34	-1679.5	132.5
113	**#RESET**	-1679.5	42.5
114	**TEST0**	-1679.5	-47.5
115	**V_{DD}**	-1679.5	-137.5
116	**V_{PP}**	-1679.5	-227.5
117	**V_{SS}**	-1679.5	-317.5
118	**ETP0**	-1679.5	-407.5
119	**EBP0**	-1679.5	-497.5
120	**ESEG0**	-1679.5	-587.5
121	**ESEG1**	-1679.5	-677.5
122	**ESEG2**	-1679.5	-767.5
123	**ESEG3**	-1679.5	-857.5
124	**ESEG4**	-1679.5	-947.5
125	**ESEG5**	-1679.5	-1037.5
126	**ESEG6**	-1679.5	-1127.5
127	**ESEG7**	-1679.5	-1217.5

29.2 Gold Bump Specifications

Table 29.2.1 Gold Bump Specifications

Characteristic		Specification
Bump shape		Strait bump
Bump height (Distance between Al trace and top of bump)	Central height	17 µm Typ.
	Bump-to-bump variation tolerances in all lots	Central height ± 4 µm
	Bump-to-bump variation tolerances in a chip	R(Max. - Min.) ≤ 3 µm
Bump hardness	All bumps in all lots	30 to 70 HV
Bump strength	All bumps in all lots	0.0067g/µm^2, shearing within a gold bump
Bump surface asperities	Height Max. - Min. in a bump	3.0 µm or less
Bump dimensions	X and Y plane dimension tolerances (at top of bump)	X ± 4 µm, Y ± 4 µm
Clearance between bumps	Minimum value	S = 20 µm

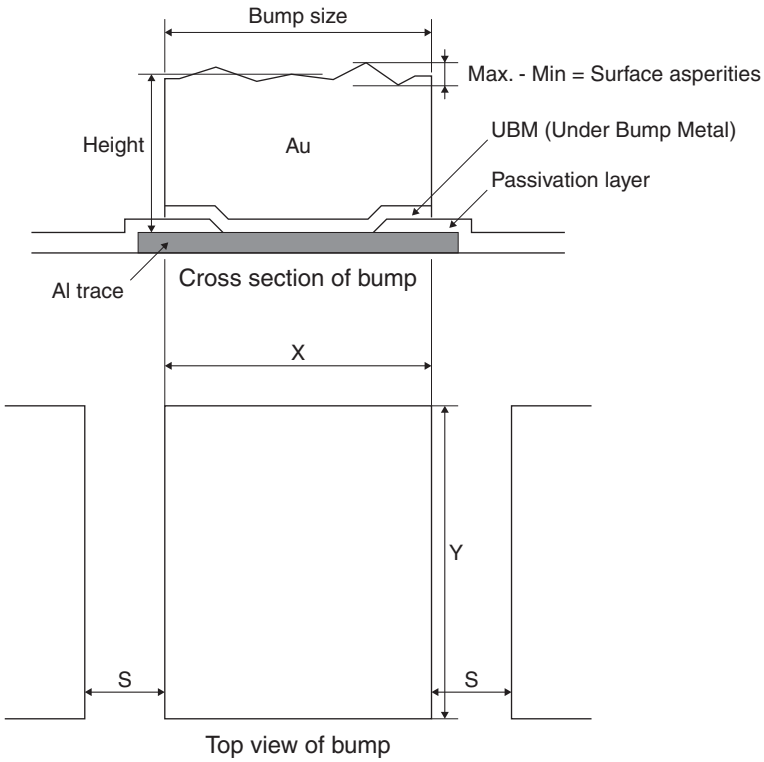


Figure 29.2.1 Gold Bump Specifications

Appendix A List of I/O Registers

Internal peripheral circuit area 1 (0x4000–0x43ff)

Peripheral	Address	Register name		Function
MISC register (8-bit device)	0x4020	MISC_ DMODE1	Debug Mode Control Register 1	Enables peripheral operations in debug mode (PCLK).
UART (with IrDA) Ch.0 (8-bit device)	0x4100	UART_ST0	UART Ch.0 Status Register	Indicates transfer, buffer and error statuses.
	0x4101	UART_TXD0	UART Ch.0 Transmit Data Register	Transmit data
	0x4102	UART_RXD0	UART Ch.0 Receive Data Register	Receive data
	0x4103	UART_MOD0	UART Ch.0 Mode Register	Sets transfer data format.
	0x4104	UART_CTL0	UART Ch.0 Control Register	Controls data transfer.
	0x4105	UART_EXP0	UART Ch.0 Expansion Register	Sets IrDA mode.
	0x4106	UART_BR0	UART Ch.0 Baud Rate Register	Sets baud rate.
	0x4107	UART_FMD0	UART Ch.0 Fine Mode Register	Sets fine mode.
8-bit timer Ch. 0 (16-bit device)	0x4240	T8_CLK0	T8 Ch.0 Count Clock Select Register	Selects a count clock.
	0x4242	T8_TR0	T8 Ch.0 Reload Data Register	Sets reload data.
	0x4244	T8_TC0	T8 Ch.0 Counter Data Register	Counter data
	0x4246	T8_CTL0	T8 Ch.0 Control Register	Sets the timer mode and starts/stops the timer.
	0x4248	T8_INT0	T8 Ch.0 Interrupt Control Register	Controls the interrupt.
8-bit timer Ch. 1 (16-bit device)	0x4260	T8_CLK1	T8 Ch.1 Count Clock Select Register	Selects a count clock.
	0x4262	T8_TR1	T8 Ch.1 Reload Data Register	Sets reload data.
	0x4264	T8_TC1	T8 Ch.1 Counter Data Register	Counter data
	0x4266	T8_CTL1	T8 Ch.1 Control Register	Sets the timer mode and starts/stops the timer.
	0x4268	T8_INT1	T8 Ch.1 Interrupt Control Register	Controls the interrupt.
Interrupt controller (16-bit device)	0x4306	ITC_LV0	Interrupt Level Setup Register 0	Sets the P0 and P2 interrupt levels.
	0x4308	ITC_LV1	Interrupt Level Setup Register 1	Sets the SWT and CT interrupt levels.
	0x430a	ITC_LV2	Interrupt Level Setup Register 2	Sets the RTC interrupt level.
	0x430c	ITC_LV3	Interrupt Level Setup Register 3	Sets the EPD and T16A2 Ch.0 interrupt levels.
	0x4310	ITC_LV5	Interrupt Level Setup Register 5	Sets the T8 Ch.0 and Ch.1 interrupt levels.
	0x4312	ITC_LV6	Interrupt Level Setup Register 6	Sets the UART Ch.0 and I2CS interrupt levels.
	0x4314	ITC_LV7	Interrupt Level Setup Register 7	Sets the SPI Ch.0 and I2CM interrupt levels.
	0x4316	ITC_LV8	Interrupt Level Setup Register 8	Sets the T16A2 Ch.1 interrupt level.
	0x4318	ITC_LV9	Interrupt Level Setup Register 9	Sets the TEM and RFC interrupt levels.
SPI Ch.0 (16-bit device)	0x4320	SPI_ST0	SPI Ch.0 Status Register	Indicates transfer and buffer statuses.
	0x4322	SPI_TXD0	SPI Ch.0 Transmit Data Register	Transmit data
	0x4324	SPI_RXD0	SPI Ch.0 Receive Data Register	Receive data
	0x4326	SPI_CTL0	SPI Ch.0 Control Register	Sets the SPI mode and enables data transfer.
I²C master (16-bit device)	0x4340	I2CM_EN	I²C Master Enable Register	Enables the I²C master module.
	0x4342	I2CM_CTL	I²C Master Control Register	Controls the I²C master operation and indicates transfer status.
	0x4344	I2CM_DAT	I²C Master Data Register	Transmit/receive data
	0x4346	I2CM_ICTL	I²C Master Interrupt Control Register	Controls the I²C master interrupt.
I²C slave (16-bit device)	0x4360	I2CS_TRNS	I²C Slave Transmit Data Register	I²C slave transmit data
	0x4362	I2CS_RECV	I²C Slave Receive Data Register	I²C slave receive data
	0x4364	I2CS_SADRS	I²C Slave Address Setup Register	Sets the I²C slave address.
	0x4366	I2CS_CTL	I²C Slave Control Register	Controls the I²C slave module.
	0x4368	I2CS_STAT	I²C Slave Status Register	Indicates the I²C bus status.
	0x436a	I2CS_ASTAT	I²C Slave Access Status Register	Indicates the I²C slave access status.
	0x436c	I2CS_ICTL	I²C Slave Interrupt Control Register	Controls the I²C slave interrupt.

Internal Peripheral Circuit Area 2 (0x5000–0x5fff)

Peripheral	Address	Register name		Function
Clock timer (8-bit device)	0x5000	CT_CTL	Clock Timer Control Register	Resets and starts/stops the timer.
	0x5001	CT_CNT	Clock Timer Counter Register	Counter data
	0x5002	CT_IMSK	Clock Timer Interrupt Mask Register	Enables/disables interrupt.
	0x5003	CT_IFLG	Clock Timer Interrupt Flag Register	Indicates/resets interrupt occurrence status.
Stopwatch timer (8-bit device)	0x5020	SWT_CTL	Stopwatch Timer Control Register	Resets and starts/stops the timer.
	0x5021	SWT_BCNT	Stopwatch Timer BCD Counter Register	BCD counter data
	0x5022	SWT_IMSK	Stopwatch Timer Interrupt Mask Register	Enables/disables interrupt.
	0x5023	SWT_IFLG	Stopwatch Timer Interrupt Flag Register	Indicates/resets interrupt occurrence status.
Watchdog timer (8-bit device)	0x5040	WDT_CTL	Watchdog Timer Control Register	Resets and starts/stops the timer.
	0x5041	WDT_ST	Watchdog Timer Status Register	Sets the timer mode and indicates NMI status.

Peripheral	Address	Register name		Function
Clock generator /Theoretical regulation (8-bit device) (RFC, T16A2, UART, SND, TEM, EPD, TR)	0x5060	CLG_SRC	Clock Source Select Register	Selects the clock source.
	0x5061	CLG_CTL	Oscillation Control Register	Controls oscillation.
	0x5064	CLG_FOUTA	FOUTA Control Register	Controls FOUTA clock output.
	0x5065	CLG_FOUTB	FOUTB Control Register	Controls FOUTB clock output.
	0x5067	RFC_CLK	RFC Clock Control Register	Controls the RFC clock.
	0x5068	T16A_CLK0	T16A2 Clock Control Register Ch.0	Controls the T16A2 Ch.0 clock.
	0x5069	T16A_CLK1	T16A2 Clock Control Register Ch.1	Controls the T16A2 Ch.1 clock.
	0x506c	UART_CLK0	UART Ch.0 Clock Control Register	Selects the baud rate generator clock.
	0x506e	SND_CLK	SND Clock Control Register	Controls the SND clock.
	0x506f	TEM_CLK	TEM Clock Control Register	Controls the TEM clock.
	0x5070	EPD_TCLK	EPD Timing Clock Control Register	Controls the EPD timing clock.
	0x5071	EPD_DCLK	EPD Doubler Clock Control Register	Controls the EPD doubler clock.
	0x5072	EPD_BCLK	EPD Booster Clock Control Register	Controls the EPD booster clock.
	0x5078	TR_CTL	TR Control Register	Controls theoretical regulation.
	0x5079	TR_VAL	TR Value Register	Sets a regulation value.
	0x507d	CLG_WAIT	Oscillation Stabilization Wait Control Register	Controls oscillation stabilization waiting time.
	0x5080	CLG_PCLK	PCLK Control Register	Controls the PCLK supply.
	0x5081	CLG_CCLK	CCLK Control Register	Configures the CCLK division ratio.
SVD circuit (8-bit device)	0x5100	SVD_EN	SVD Enable Register	Enables/disables the SVD operation.
	0x5101	SVD_CMP	SVD Comparison Voltage Register	Sets the comparison voltage.
	0x5102	SVD_RSLT	SVD Detection Result Register	Voltage detection results
Power generator (8-bit device)	0x5120	VD1_CTL	V_{D1} Control Register	Controls the V_{D1} regulator heavy load protection mode.
Sound generator (8-bit device)	0x5180	SND_CTL	SND Control Register	Controls buzzer outputs.
	0x5181	SND_BZFQ	Buzzer Frequency Control Register	Sets the buzzer frequency.
	0x5182	SND_BZDT	Buzzer Duty Ratio Control Register	Sets the buzzer signal duty ratio.
Temperature detection circuit (8-bit device)	0x51a0	TEM_TIME	TEM Comparison Time Setting Register	Sets the sensor output comparison time.
	0x51a1	TEM_CTL	TEM Control Register	Controls the temperature detection circuit.
	0x51a2	TEM_STAT	TEM Status Register	Indicates the conversion status.
	0x51a3	TEM_RSLT	TEM Conversion Result Register	Temperature conversion results
P port & port MUX (8-bit device)	0x5200	P0_IN	P0 Port Input Data Register	P0 port input data
	0x5201	P0_OUT	P0 Port Output Data Register	P0 port output data
	0x5202	P0_OEN	P0 Port Output Enable Register	Enables P0 port outputs.
	0x5203	P0_PU	P0 Port Pull-up Control Register	Controls the P0 port pull-up resistor.
	0x5205	P0_IMSK	P0 Port Interrupt Mask Register	Enables P0 port interrupts.
	0x5206	P0_EDGE	P0 Port Interrupt Edge Select Register	Selects the signal edge for generating P0 port interrupts.
	0x5207	P0_IFLG	P0 Port Interrupt Flag Register	Indicates/resets the P0 port interrupt occurrence status.
	0x5208	P0_CHAT	P0 Port Chattering Filter Control Register	Controls the P0 port chattering filter.
	0x5209	P0_KRST	P0 Port Key-Entry Reset Configuration Register	Configures the P0 port key-entry reset function.
	0x520a	P0_IEN	P0 Port Input Enable Register	Enables P0 port inputs.
	0x5210	P1_IN	P1 Port Input Data Register	P1 port input data
	0x5211	P1_OUT	P1 Port Output Data Register	P1 port output data
	0x5212	P1_OEN	P1 Port Output Enable Register	Enables P1 port outputs.
	0x5213	P1_PU	P1 Port Pull-up Control Register	Controls the P1 port pull-up resistor.
	0x521a	P1_IEN	P1 Port Input Enable Register	Enables P1 port inputs.
	0x5220	P2_IN	P2 Port Input Data Register	P2 port input data
	0x5221	P2_OUT	P2 Port Output Data Register	P2 port output data
	0x5222	P2_OEN	P2 Output Enable Register	Enables P2 port outputs.
	0x5223	P2_PU	P2 Port Pull-up Control Register	Controls the P2 port pull-up resistor.
	0x5225	P2_IMSK	P2 Port Interrupt Mask Register	Enables P2 port interrupts.
	0x5226	P2_EDGE	P2 Port Interrupt Edge Select Register	Selects the signal edge for generating P2 port interrupts.
	0x5227	P2_IFLG	P2 Port Interrupt Flag Register	Indicates/resets the P2 port interrupt occurrence status.
	0x5228	P2_CHAT	P2 Port Chattering Filter Control Register	Controls the P2 port chattering filter.
	0x522a	P2_IEN	P2 Port Input Enable Register	Enables P2 port inputs.
	0x5230	P3_IN	P3 Port Input Data Register	P3 port input data
	0x5231	P3_OUT	P3 Port Output Data Register	P3 port output data
	0x5232	P3_OEN	P3 Port Output Enable Register	Enables P3 port outputs.
	0x5233	P3_PU	P3 Port Pull-up Control Register	Controls the P3 port pull-up resistor.
	0x523a	P3_IEN	P3 Port Input Enable Register	Enables P3 port inputs.
	0x52a0	P00_03PMUX	P0[3:0] Port Function Select Register	Selects the P0[3:0] port functions.
	0x52a1	P04_07PMUX	P0[7:4] Port Function Select Register	Selects the P0[7:4] port functions.
	0x52a2	P10_13PMUX	P1[3:0] Port Function Select Register	Selects the P1[3:0] port functions.
	0x52a3	P14_17PMUX	P1[7:4] Port Function Select Register	Selects the P1[7:4] port functions.
	0x52a4	P20_23PMUX	P2[3:0] Port Function Select Register	Selects the P2[3:0] port functions.
	0x52a5	P24_27PMUX	P2[7:4] Port Function Select Register	Selects the P2[7:4] port functions.
	0x52a6	P30_33PMUX	P3[3:0] Port Function Select Register	Selects the P3[3:0] port functions.
	0x52a7	P34PMUX	P34 Port Function Select Register	Selects the P34 port functions.

Peripheral	Address	Register name		Function
MISC registers (16-bit device)	0x5322	MISC_ DMODE2	Debug Mode Control Register 2	Enables peripheral operations in debug mode (except PCLK).
	0x5324	MISC_PROT	MISC Protect Register	Enables writing to the MISC registers.
	0x5326	MISC_IRAMSZ	IRAM Size Select Register	Selects the IRAM size.
	0x5328	MISC_TTBRL	Vector Table Address Low Register	Sets vector table address.
	0x532a	MISC_TTBRH	Vector Table Address High Register	
	0x532c	MISC_PSR	PSR Register	Indicates the S1C17 Core PSR values.
R/F converter (16-bit device)	0x53a0	RFC_CTL	RFC Control Register	Controls R/F converter.
	0x53a2	RFC_TRG	RFC Oscillation Trigger Register	Controls oscillations.
	0x53a4	RFC_MCL	RFC Measurement Counter Low Register	Measurement counter data
	0x53a6	RFC_MCH	RFC Measurement Counter High Register	
	0x53a8	RFC_TCL	RFC Time Base Counter Low Register	Time base counter data
	0x53aa	RFC_TCH	RFC Time Base Counter High Register	
	0x53ac	RFC_IMSK	RFC Interrupt Mask Register	Enables/disables interrupts.
	0x53ae	RFC_IFLG	RFC Interrupt Flag Register	Indicates/resets interrupt occurrence status.
16-bit PWM timer Ch.0 (16-bit device)	0x5400	T16A_CTL0	T16A Counter Ch.0 Control Register	Controls the counter.
	0x5402	T16A_TC0	T16A Counter Ch.0 Data Register	Counter data
	0x5404	T16A_CCCTL0	T16A Comparator/Capture Ch.0 Control Register	Controls the comparator/capture block and TOUT.
	0x5406	T16A_CCA0	T16A Compare/Capture Ch.0 A Data Register	Compare A/capture A data
	0x5408	T16A_CCB0	T16A Compare/Capture Ch.0 B Data Register	Compare B/capture B data
	0x540a	T16A_IEN0	T16A Compare/Capture Ch.0 Interrupt Enable Register	Enables/disables interrupts.
	0x540c	T16A_IFLG0	T16A Compare/Capture Ch.0 Interrupt Flag Register	Displays/sets interrupt occurrence status.
16-bit PWM timer Ch.1 (16-bit device)	0x5420	T16A_CTL1	T16A Counter Ch.1 Control Register	Controls the counter.
	0x5422	T16A_TC1	T16A Counter Ch.1 Data Register	Counter data
	0x5424	T16A_CCCTL1	T16A Comparator/Capture Ch.1 Control Register	Controls the comparator/capture block and TOUT.
	0x5426	T16A_CCA1	T16A Compare/Capture Ch.1 A Data Register	Compare A/capture A data
	0x5428	T16A_CCB1	T16A Compare/Capture Ch.1 B Data Register	Compare B/capture B data
	0x542a	T16A_IEN1	T16A Compare/Capture Ch.1 Interrupt Enable Register	Enables/disables interrupts.
	0x542c	T16A_IFLG1	T16A Compare/Capture Ch.1 Interrupt Flag Register	Displays/sets interrupt occurrence status.
Flash controller (16-bit device)	0x54b0	FLASHC_ WAIT	FLASHC Read Wait Control Register	Sets Flash read wait cycle.
EPD controller/ driver (16-bit device)	0x5600	EPD_PWR0	EPD Power Control Register 0	Controls the V_E regulator and doubler.
	0x5602	EPD_PWR1	EPD Power Control Register 1	Controls the V_H regulator and booster.
	0x5604	EPD_CTL	EPD Display Control Register	Controls display on EPD.
	0x5606	EPD_INT	EPD Interrupt Control Register	Controls interrupts.
	0x5620	EPD_PLNDAT	EPD Top/Back Plane Data Register	Top plane/back plane output data
	0x5622	EPD_SEGDAT0	EPD Segment Data Register 0	Segment output data (ESEG0–ESEG15)
	0x5624	EPD_SEGDAT1	EPD Segment Data Register 1	Segment output data (ESEG16–ESEG31)
	0x5626	EPD_SEGDAT2	EPD Segment Data Register 2	Segment output data (ESEG32–ESEG47)
	0x5628	EPD_SEGDAT3	EPD Segment Data Register 3	Segment output data (ESEG48–ESEG63)
	0x5640	EPD_WAVE0	EPD Wave Timing Set 0 Register	Display waveform data (Timing set 0)
	0x5642	EPD_WAVE1	EPD Wave Timing Set 1 Register	Display waveform data (Timing set 1)
	:	:	:	:
	0x567e	EPD_WAVE31	EPD Wave Timing Set 31 Register	Display waveform data (Timing set 31)
Real-time clock (16-bit device)	0x56c0	RTC_CTL	RTC Control Register	Controls the RTC.
	0x56c2	RTC_IEN	RTC Interrupt Enable Register	Enables/disables interrupts.
	0x56c4	RTC_IFLG	RTC Interrupt Flag Register	Displays/sets interrupt occurrence status.
	0x56c6	RTC_MS	RTC Minute/Second Counter Register	Minute/second counter data
	0x56c8	RTC_H	RTC Hour Counter Register	Hour counter data

Core I/O Reserved Area (0xffff84–0xffffd0)

Peripheral	Address	Register name		Function
S1C17 Core I/O	0xffff84	IDIR	Processor ID Register	Indicates the processor ID.
	0xffff90	DBRAM	Debug RAM Base Register	Indicates the debug RAM base address.
	0xffffa0	DCR	Debug Control Register	Controls debugging.
	0xffffb4	IBAR1	Instruction Break Address Register 1	Sets Instruction break address #1.
	0xffffb8	IBAR2	Instruction Break Address Register 2	Sets Instruction break address #2.
	0xffffbc	IBAR3	Instruction Break Address Register 3	Sets Instruction break address #3.
	0xffffd0	IBAR4	Instruction Break Address Register 4	Sets Instruction break address #4.

Note: Addresses marked as “Reserved” or unused peripheral circuit areas not marked in the table must not be accessed by application programs.

0x4100–0x4107, 0x506c UART (with IrDA) Ch.0

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
UART Ch.0 Status Register (UART_ST0)	**0x4100** (8 bits)	D7	**TRED**	End of transmission flag	1 Completed 0 Not completed	0	R/W	Reset by writing 1.
		D6	**FER**	Framing error flag	1 Error 0 Normal	0	R/W	
		D5	**PER**	Parity error flag	1 Error 0 Normal	0	R/W	
		D4	**OER**	Overrun error flag	1 Error 0 Normal	0	R/W	
		D3	**RD2B**	Second byte receive flag	1 Ready 0 Empty	0	R	
		D2	**TRBS**	Transmit busy flag	1 Busy 0 Idle	0	R	Shift register status
		D1	**RDRY**	Receive data ready flag	1 Ready 0 Empty	0	R	
		D0	**TDBE**	Transmit data buffer empty flag	1 Empty 0 Not empty	1	R	
UART Ch.0 Transmit Data Register (UART_TXD0)	**0x4101** (8 bits)	D7–0	**TXD[7:0]**	Transmit data TXD7(6) = MSB TXD0 = LSB	0x0 to 0xff (0x7f)	0x0	R/W	
UART Ch.0 Receive Data Register (UART_RXD0)	**0x4102** (8 bits)	D7–0	**RXD[7:0]**	Receive data in the receive data buffer RXD7(6) = MSB RXD0 = LSB	0x0 to 0xff (0x7f)	0x0	R	Older data in the buffer is read out first.
UART Ch.0 Mode Register (UART_MOD0)	**0x4103** (8 bits)	D7–5	**–**	reserved	–	–	–	0 when being read.
		D4	**CHLN**	Character length select	1 8 bits 0 7 bits	0	R/W	
		D3	**PREN**	Parity enable	1 With parity 0 No parity	0	R/W	
		D2	**PMD**	Parity mode select	1 Odd 0 Even	0	R/W	
		D1	**STPB**	Stop bit select	1 2 bits 0 1 bit	0	R/W	
		D0	**–**	reserved	–	–	–	0 when being read.
UART Ch.0 Control Register (UART_CTL0)	**0x4104** (8 bits)	D7	**TEIEN**	End of transmission int. enable	1 Enable 0 Disable	0	R/W	
		D6	**REIEN**	Receive error int. enable	1 Enable 0 Disable	0	R/W	
		D5	**RIEN**	Receive buffer full int. enable	1 Enable 0 Disable	0	R/W	
		D4	**TIEN**	Transmit buffer empty int. enable	1 Enable 0 Disable	0	R/W	
		D3–2	**–**	reserved	–	–	–	0 when being read.
		D1	**RBFI**	Receive buffer full int. condition setup	1 2 bytes 0 1 byte	0	R/W	
		D0	**RXEN**	UART enable	1 Enable 0 Disable	0	R/W	
UART Ch.0 Expansion Register (UART_EXP0)	**0x4105** (8 bits)	D7–1	**–**	reserved	–	–	–	0 when being read.
		D0	**IRMD**	IrDA mode select	1 On 0 Off	0	R/W	
UART Ch.0 Baud Rate Register (UART_BR0)	**0x4106** (8 bits)	D7–0	**BR[7:0]**	Baud rate setting	0x0 to 0xff	0x0	R/W	
UART Ch.0 Fine Mode Register (UART_FMD0)	**0x4107** (8 bits)	D7–4	**–**	reserved	–	–	–	0 when being read.
		D3–0	**FMD[3:0]**	Fine mode setup	0x0 to 0xf	0x0	R/W	Set a number of times to insert delay into a 16-underflow period.
UART Ch.0 Clock Control Register (UART_CLK0)	**0x506c** (8 bits)	D7–6	**–**	reserved	–	–	–	0 when being read.
		D5–4	**UTCLKD [1:0]**	Clock division ratio select	UTCLKD[1:0] / Division ratio 0x3 / 1/8 0x2 / 1/4 0x1 / 1/2 0x0 / 1/1	0x0	R/W	When the clock source is OSC3B or OSC3A
		D3–2	**UTCLKSRC [1:0]**	Clock source select	UTCLKSRC [1:0] / Clock source 0x3 / External clock 0x2 / OSC3A 0x1 / OSC1 0x0 / OSC3B	0x0	R/W	
		D1	**–**	reserved	–	–	–	0 when being read.
		D0	**UTCLKE**	UART clock enable	1 Enable 0 Disable	0	R/W	

0x4240–0x4248 8-bit Timer Ch.0

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T8 Ch.0 Count Clock Select Register (T8_CLK0)	0x4240 (16 bits)	D15–4	–	reserved	–	–	–	0 when being read.
		D3–0	DF[3:0]	Count clock division ratio select	DF[3:0] / Division ratio: 0xf reserved 0xe 1/16384 0xd 1/8192 0xc 1/4096 0xb 1/2048 0xa 1/1024 0x9 1/512 0x8 1/256 0x7 1/128 0x6 1/64 0x5 1/32 0x4 1/16 0x3 1/8 0x2 1/4 0x1 1/2 0x0 1/1	0x0	R/W	Source clock = PCLK
T8 Ch.0 Reload Data Register (T8_TR0)	0x4242 (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	TR[7:0]	Reload data TR7 = MSB TR0 = LSB	0x0 to 0xff	0x0	R/W	
T8 Ch.0 Counter Data Register (T8_TC0)	0x4244 (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	TC[7:0]	Counter data TC7 = MSB TC0 = LSB	0x0 to 0xff	0xff	R	
T8 Ch.0 Control Register (T8_CTL0)	0x4246 (16 bits)	D15–5	–	reserved	–	–	–	Do not write 1.
		D4	TRMD	Count mode select	1 One shot / 0 Repeat	0	R/W	
		D3–2	–	reserved	–	–	–	0 when being read.
		D1	PRESER	Timer reset	1 Reset / 0 Ignored	0	W	
		D0	PRUN	Timer run/stop control	1 Run / 0 Stop	0	R/W	
T8 Ch.0 Interrupt Control Register (T8_INT0)	0x4248 (16 bits)	D15–9	–	reserved	–	–	–	0 when being read.
		D8	T8IE	T8 interrupt enable	1 Enable / 0 Disable	0	R/W	
		D7–1	–	reserved	–	–	–	0 when being read.
		D0	T8IF	T8 interrupt flag	1 Cause of interrupt occurred / 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.

0x4260–0x4268 8-bit Timer Ch.1

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T8 Ch.1 Count Clock Select Register (T8_CLK1)	0x4260 (16 bits)	D15–4	–	reserved	–	–	–	0 when being read.
		D3–0	DF[3:0]	Count clock division ratio select	DF[3:0] / Division ratio: 0xf reserved 0xe 1/16384 0xd 1/8192 0xc 1/4096 0xb 1/2048 0xa 1/1024 0x9 1/512 0x8 1/256 0x7 1/128 0x6 1/64 0x5 1/32 0x4 1/16 0x3 1/8 0x2 1/4 0x1 1/2 0x0 1/1	0x0	R/W	Source clock = PCLK
T8 Ch.1 Reload Data Register (T8_TR1)	0x4262 (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	TR[7:0]	Reload data TR7 = MSB TR0 = LSB	0x0 to 0xff	0x0	R/W	
T8 Ch.1 Counter Data Register (T8_TC1)	0x4264 (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	TC[7:0]	Counter data TC7 = MSB TC0 = LSB	0x0 to 0xff	0xff	R	
T8 Ch.1 Control Register (T8_CTL1)	0x4266 (16 bits)	D15–5	–	reserved	–	–	–	Do not write 1.
		D4	TRMD	Count mode select	1 One shot / 0 Repeat	0	R/W	
		D3–2	–	reserved	–	–	–	0 when being read.
		D1	PRESER	Timer reset	1 Reset / 0 Ignored	0	W	
		D0	PRUN	Timer run/stop control	1 Run / 0 Stop	0	R/W	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T8 Ch.1 Interrupt Control Register (T8_INT1)	0x4268 (16 bits)	D15–9	–	reserved	–	–	–	0 when being read.
		D8	T8IE	T8 interrupt enable	1 Enable 0 Disable	0	R/W	
		D7–1	–	reserved	–	–	–	0 when being read.
		D0	T8IF	T8 interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.

0x4306–0x4318 Interrupt Controller

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Interrupt Level Setup Register 0 (ITC_LV0)	0x4306 (16 bits)	D15–11	–	reserved	–	–	–	0 when being read.
		D10–8	ILV1[2:0]	P2 interrupt level	0 to 7	0x0	R/W	
		D7–3	–	reserved	–	–	–	0 when being read.
		D2–0	ILV0[2:0]	P0 interrupt level	0 to 7	0x0	R/W	
Interrupt Level Setup Register 1 (ITC_LV1)	0x4308 (16 bits)	D15–11	–	reserved	–	–	–	0 when being read.
		D10–8	ILV3[2:0]	CT interrupt level	0 to 7	0x0	R/W	
		D7–3	–	reserved	–	–	–	0 when being read.
		D2–0	ILV2[2:0]	SWT interrupt level	0 to 7	0x0	R/W	
Interrupt Level Setup Register 2 (ITC_LV2)	0x430a (16 bits)	D15–3	–	reserved	–	–	–	0 when being read.
		D2–0	ILV4[2:0]	RTC interrupt level	0 to 7	0x0	R/W	
Interrupt Level Setup Register 3 (ITC_LV3)	0x430c (16 bits)	D15–11	–	reserved	–	–	–	0 when being read.
		D10–8	ILV7[2:0]	T16A2 Ch.0 interrupt level	0 to 7	0x0	R/W	
		D7–3	–	reserved	–	–	–	0 when being read.
		D2–0	ILV6[2:0]	EPD interrupt level	0 to 7	0x0	R/W	
Interrupt Level Setup Register 5 (ITC_LV5)	0x4310 (16 bits)	D15–11	–	reserved	–	–	–	0 when being read.
		D10–8	ILV11[2:0]	T8 Ch.1 interrupt level	0 to 7	0x0	R/W	
		D7–3	–	reserved	–	–	–	0 when being read.
		D2–0	ILV10[2:0]	T8 Ch.0 interrupt level	0 to 7	0x0	R/W	
Interrupt Level Setup Register 6 (ITC_LV6)	0x4312 (16 bits)	D15–11	–	reserved	–	–	–	0 when being read.
		D10–8	ILV13[2:0]	I2CS interrupt level	0 to 7	0x0	R/W	
		D7–3	–	reserved	–	–	–	0 when being read.
		D2–0	ILV12[2:0]	UART Ch.0 interrupt level	0 to 7	0x0	R/W	
Interrupt Level Setup Register 7 (ITC_LV7)	0x4314 (16 bits)	D15–11	–	reserved	–	–	–	0 when being read.
		D10–8	ILV15[2:0]	I2CM interrupt level	0 to 7	0x0	R/W	
		D7–3	–	reserved	–	–	–	0 when being read.
		D2–0	ILV14[2:0]	SPI Ch.0 interrupt level	0 to 7	0x0	R/W	
Interrupt Level Setup Register 8 (ITC_LV8)	0x4316 (16 bits)	D15–11	–	reserved	–	–	–	0 when being read.
		D10–8	ILV17[2:0]	T16A2 Ch.1 interrupt level	0 to 7	0x0	R/W	
		D7–0	–	reserved	–	–	–	0 when being read.
Interrupt Level Setup Register 9 (ITC_LV9)	0x4318 (16 bits)	D15–11	–	reserved	–	–	–	0 when being read.
		D10–8	ILV19[2:0]	RFC interrupt level	0 to 7	0x0	R/W	
		D7–3	–	reserved	–	–	–	0 when being read.
		D2–0	ILV18[2:0]	TEM interrupt level	0 to 7	0x0	R/W	

0x4320–0x4326 SPI Ch.0

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SPI Ch.0 Status Register (SPI_ST0)	0x4320 (16 bits)	D15–3	–	reserved	–	–	–	0 when being read.
		D2	SPBSY	Transfer busy flag (master)	1 Busy 0 Idle	0	R	
				ss signal low flag (slave)	1 ss = L 0 ss = H			
		D1	SPRBF	Receive data buffer full flag	1 Full 0 Not full	0	R	
		D0	SPTBE	Transmit data buffer empty flag	1 Empty 0 Not empty	1	R	
SPI Ch.0 Transmit Data Register (SPI_TXD0)	0x4322 (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	SPTDB[7:0]	SPI transmit data buffer SPTDB7 = MSB SPTDB0 = LSB	0x0 to 0xff	0x0	R/W	
SPI Ch.0 Receive Data Register (SPI_RXD0)	0x4324 (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	SPRDB[7:0]	SPI receive data buffer SPRDB7 = MSB SPRDB0 = LSB	0x0 to 0xff	0x0	R	
SPI Ch.0 Control Register (SPI_CTL0)	0x4326 (16 bits)	D15–10	–	reserved	–	–	–	0 when being read.
		D9	MCLK	SPI clock source select	1 T8 Ch.0 0 PCLK/4	0	R/W	
		D8	MLSB	LSB/MSB first mode select	1 LSB 0 MSB	0	R/W	
		D7–6	–	reserved	–	–	–	0 when being read.
		D5	SPRIE	Receive data buffer full int. enable	1 Enable 0 Disable	0	R/W	
		D4	SPTIE	Transmit data buffer empty int. enable	1 Enable 0 Disable	0	R/W	
		D3	CPHA	Clock phase select	1 Data out 0 Data in	0	R/W	These bits must be set before setting SPEN to 1.
		D2	CPOL	Clock polarity select	1 Active L 0 Active H	0	R/W	
		D1	MSSL	Master/slave mode select	1 Master 0 Slave	0	R/W	
		D0	SPEN	SPI enable	1 Enable 0 Disable	0	R/W	

0x4340–0x4346 I²C Master

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
I²C Master Enable Register (I2CM_EN)	**0x4340** (16 bits)	D15–1	**–**	reserved	–	–	–	0 when being read.
		D0	**I2CMEN**	I²C master enable	1 Enable 0 Disable	0	R/W	
I²C Master Control Register (I2CM_CTL)	**0x4342** (16 bits)	D15–10	**–**	reserved	–	–	–	0 when being read.
		D9	**RBUSY**	Receive busy flag	1 Busy 0 Idle	0	R	
		D8	**TBUSY**	Transmit busy flag	1 Busy 0 Idle	0	R	
		D7–5	**–**	reserved	–	–	–	0 when being read.
		D4	**NSERM**	Noise remove on/off	1 On 0 Off	0	R/W	
		D3–2	**–**	reserved	–	–	–	0 when being read.
		D1	**STP**	Stop control	1 Stop 0 Ignored	0	R/W	
		D0	**STRT**	Start control	1 Start 0 Ignored	0	R/W	
I²C Master Data Register (I2CM_DAT)	**0x4344** (16 bits)	D15–12	**–**	reserved	–	–	–	0 when being read.
		D11	**RBRDY**	Receive buffer ready flag	1 Ready 0 Empty	0	R	
		D10	**RXE**	Receive execution	1 Receive 0 Ignored	0	R/W	
		D9	**TXE**	Transmit execution	1 Transmit 0 Ignored	0	R/W	
		D8	**RTACK**	Receive/transmit ACK	1 Error 0 ACK	0	R/W	
		D7–0	**RTDT[7:0]**	Receive/transmit data RTDT7 = MSB RTDT0 = LSB	0x0 to 0xff	0x0	R/W	
I²C Master Interrupt Control Register (I2CM_ICTL)	**0x4346** (16 bits)	D15–2	**–**	reserved	–	–	–	0 when being read.
		D1	**RINTE**	Receive interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	**TINTE**	Transmit interrupt enable	1 Enable 0 Disable	0	R/W	

0x4360–0x436c I²C Slave

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
I²C Slave Transmit Data Register (I2CS_TRNS)	**0x4360** (16 bits)	D15–8	**–**	reserved	–	–	–	0 when being read.
		D7–0	**SDATA[7:0]**	I²C slave transmit data	0–0xff	0x0	R/W	
I²C Slave Receive Data Register (I2CS_RECV)	**0x4362** (16 bits)	D15–8	**–**	reserved	–	–	–	0 when being read.
		D7–0	**RDATA[7:0]**	I²C slave receive data	0–0xff	0x0	R	
I²C Slave Address Setup Register (I2CS_SADRS)	**0x4364** (16 bits)	D15–7	**–**	reserved	–	–	–	0 when being read.
		D6–0	**SADRS[6:0]**	I²C slave address	0–0x7f	0x0	R/W	
I²C Slave Control Register (I2CS_CTL)	**0x4366** (16 bits)	D15–9	**–**	reserved	–	–	–	0 when being read.
		D8	**TBUF_CLR**	I2CS_TRNS register clear	1 Clear state 0 Normal	0	R/W	
		D7	**I2CSEN**	I²C slave enable	1 Enable 0 Disable	0	R/W	
		D6	**SOFTRESET**	Software reset	1 Reset 0 Cancel	0	R/W	
		D5	**NAK_ANS**	NAK answer	1 NAK 0 ACK	0	R/W	
		D4	**BFREQ_EN**	Bus free request enable	1 Enable 0 Disable	0	R/W	
		D3	**CLKSTR_EN**	Clock stretch On/Off	1 On 0 Off	0	R/W	
		D2	**NF_EN**	Noise filter On/Off	1 On 0 Off	0	R/W	
		D1	**ASDET_EN**	Async.address detection On/Off	1 On 0 Off	0	R/W	
		D0	**COM_MODE**	I²C slave communication mode	1 Active 0 Standby	0	R/W	
I²C Slave Status Register (I2CS_STAT)	**0x4368** (16 bits)	D15–8	**–**	reserved	–	–	–	0 when being read.
		D7	**BSTAT**	Bus status transition	1 Changed 0 Unchanged	0	R	
		D6	**–**	reserved	–	–	–	0 when being read.
		D5	**TXUDF** **RXOVF**	Transmit data underflow Receive data overflow	1 Occurred 0 Not occurred	0	R/W	Reset by writing 1.
		D4	**BFREQ**	Bus free request	1 Occurred 0 Not occurred	0	R/W	
		D3	**DMS**	Output data mismatch	1 Error 0 Normal	0	R/W	
		D2	**ASDET**	Async. address detection status	1 Detected 0 Not detected	0	R/W	
		D1	**DA_NAK**	NAK receive status	1 NAK 0 ACK	0	R/W	
		D0	**DA_STOP**	STOP condition detect	1 Detected 0 Not detected	0	R/W	
I²C Slave Access Status Register (I2CS_ASTAT)	**0x436a** (16 bits)	D15–5	**–**	reserved	–	–	–	0 when being read.
		D4	**RXRDY**	Receive data ready	1 Ready 0 Not ready	0	R	
		D3	**TXEMP**	Transmit data empty	1 Empty 0 Not empty	0	R	
		D2	**BUSY**	I²C bus status	1 Busy 0 Free	0	R	
		D1	**SELECTED**	I²C slave select status	1 Selected 0 Not selected	0	R	
		D0	**R/W**	Read/write direction	1 Output 0 Input	0	R	
I²C Slave Interrupt Control Register (I2CS_ICTL)	**0x436c** (16 bits)	D15–3	**–**	reserved	–	–	–	0 when being read.
		D2	**BSTAT_IEN**	Bus status interrupt enable	1 Enable 0 Disable	0	R/W	
		D1	**RXRDY_IEN**	Receive interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	**TXEMP_IEN**	Transmit interrupt enable	1 Enable 0 Disable	0	R/W	

0x5000–0x5003 Clock Timer

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Clock Timer Control Register (CT_CTL)	0x5000 (8 bits)	D7–5	–	reserved	–	–	–	0 when being read.
		D4	CTRST	Clock timer reset	1 Reset 0 Ignored	0	W	
		D3–1	–	reserved	–	–	–	
		D0	CTRUN	Clock timer run/stop control	1 Run 0 Stop	0	R/W	
Clock Timer Counter Register (CT_CNT)	0x5001 (8 bits)	D7–0	CTCNT[7:0]	Clock timer counter value	0x0 to 0xff	0	R	
Clock Timer Interrupt Mask Register (CT_IMSK)	0x5002 (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.
		D3	CTIE32	32 Hz interrupt enable	1 Enable 0 Disable	0	R/W	
		D2	CTIE8	8 Hz interrupt enable	1 Enable 0 Disable	0	R/W	
		D1	CTIE2	2 Hz interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	CTIE1	1 Hz interrupt enable	1 Enable 0 Disable	0	R/W	
Clock Timer Interrupt Flag Register (CT_IFLG)	0x5003 (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.
		D3	CTIF32	32 Hz interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D2	CTIF8	8 Hz interrupt flag		0	R/W	
		D1	CTIF2	2 Hz interrupt flag		0	R/W	
		D0	CTIF1	1 Hz interrupt flag		0	R/W	

0x5020–0x5023 Stopwatch Timer

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Stopwatch Timer Control Register (SWT_CTL)	0x5020 (8 bits)	D7–5	–	reserved	–	–	–	0 when being read.
		D4	SWTRST	Stopwatch timer reset	1 Reset 0 Ignored	0	W	
		D3–1	–	reserved	–	–	–	
		D0	SWTRUN	Stopwatch timer run/stop control	1 Run 0 Stop	0	R/W	
Stopwatch Timer BCD Counter Register (SWT_BCNT)	0x5021 (8 bits)	D7–4	BCD10[3:0]	1/10 sec. BCD counter value	0 to 9	0	R	
		D3–0	BCD100[3:0]	1/100 sec. BCD counter value	0 to 9	0	R	
Stopwatch Timer Interrupt Mask Register (SWT_IMSK)	0x5022 (8 bits)	D7–3	–	reserved	–	–	–	0 when being read.
		D2	SIE1	1 Hz interrupt enable	1 Enable 0 Disable	0	R/W	
		D1	SIE10	10 Hz interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	SIE100	100 Hz interrupt enable	1 Enable 0 Disable	0	R/W	
Stopwatch Timer Interrupt Flag Register (SWT_IFLG)	0x5023 (8 bits)	D7–3	–	reserved	–	–	–	0 when being read.
		D2	SIF1	1 Hz interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D1	SIF10	10 Hz interrupt flag		0	R/W	
		D0	SIF100	100 Hz interrupt flag		0	R/W	

0x5040–0x5041 Watchdog Timer

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Watchdog Timer Control Register (WDT_CTL)	0x5040 (8 bits)	D7–5	–	reserved	–	–	–	0 when being read.
		D4	WDTRST	Watchdog timer reset	1 Reset 0 Ignored	0	W	
		D3–0	WDTRUN[3:0]	Watchdog timer run/stop control	Other than 1010 Run / 1010 Stop	1010	R/W	
Watchdog Timer Status Register (WDT_ST)	0x5041 (8 bits)	D7–2	–	reserved	–	–	–	0 when being read.
		D1	WDTMD	NMI/Reset mode select	1 Reset 0 NMI	0	R/W	
		D0	WDTST	NMI status	1 NMI occurred 0 Not occurred	0	R	

0x5060–0x5081 Clock Generator

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Clock Source Select Register (CLG_SRC)	0x5060 (8 bits)	D7–6	OSC3B FSEL[1:0]	OSC3B frequency select	OSC3BFSEL[1:0] / Frequency: 0x3 reserved; 0x2 500 kHz; 0x1 1 MHz; 0x0 2 MHz	0x0	R/W	
		D5	–	reserved	–	–	–	0 when being read.
		D4	OSC1SEL	OSC1 source select	1 OSC1B 0 OSC1A	1	R/W	
		D3–2	–	reserved	–	–	–	0 when being read.
		D1–0	CLKSRC[1:0]	System clock source select	CLKSRC[1:0] / Clock source: 0x3 reserved; 0x2 OSC3A; 0x1 OSC1; 0x0 OSC3B	0x0	R/W	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Oscillation Control Register (CLG_CTL)	**0x5061** (8 bits)	D7–3	**–**	reserved	–	–	–	0 when being read.
		D2	**OSC3BEN**	OSC3B enable	1 Enable 0 Disable	1	R/W	
		D1	**OSC1EN**	OSC1 enable	1 Enable 0 Disable	0	R/W	
		D0	**OSC3AEN**	OSC3A enable	1 Enable 0 Disable	0	R/W	
FOUTA Control Register (CLG_FOUTA)	**0x5064** (8 bits)	D7	**–**	reserved	–	–	–	0 when being read.
		D6–4	**FOUTAD [2:0]**	FOUTA clock division ratio select	FOUTAD[2:0] / Division ratio 0x7 1/128 0x6 1/64 0x5 1/32 0x4 1/16 0x3 1/8 0x2 1/4 0x1 1/2 0x0 1/1	0x0	R/W	
		D3–2	**FOUTASRC [1:0]**	FOUTA clock source select	FOUTASRC[1:0] / Clock source 0x3 reserved 0x2 OSC3A 0x1 OSC1 0x0 OSC3B	0x0	R/W	
		D1	**–**	reserved	–	–	–	0 when being read.
		D0	**FOUTAE**	FOUTA output enable	1 Enable 0 Disable	0	R/W	
FOUTB Control Register (CLG_FOUTB)	**0x5065** (8 bits)	D7	**–**	reserved	–	–	–	0 when being read.
		D6–4	**FOUTBD [2:0]**	FOUTB clock division ratio select	FOUTBD[2:0] / Division ratio 0x7 1/128 0x6 1/64 0x5 1/32 0x4 1/16 0x3 1/8 0x2 1/4 0x1 1/2 0x0 1/1	0x0	R/W	
		D3–2	**FOUTBSRC [1:0]**	FOUTB clock source select	FOUTBSRC[1:0] / Clock source 0x3 reserved 0x2 OSC3A 0x1 OSC1 0x0 OSC3B	0x0	R/W	
		D1	**–**	reserved	–	–	–	0 when being read.
		D0	**FOUTBE**	FOUTB output enable	1 Enable 0 Disable	0	R/W	
Oscillation Stabilization Wait Control Register (CLG_WAIT)	**0x507d** (8 bits)	D7–6	**OSC3BWT [1:0]**	OSC3B stabilization wait cycle select	OSC3BWT[1:0] / Wait cycle 0x3 8 cycles 0x2 16 cycles 0x1 32 cycles 0x0 64 cycles	0x0	R/W	
		D5–4	**OSC3AWT [1:0]**	OSC3A stabilization wait cycle select	OSC3AWT[1:0] / Wait cycle 0x3 128 cycles 0x2 256 cycles 0x1 512 cycles 0x0 1024 cycles	0x0	R/W	
		D3–2	**OSC1BWT [1:0]**	OSC1B stabilization wait cycle select	OSC1BWT[1:0] / Wait cycle 0x3 8 cycles 0x2 16 cycles 0x1 32 cycles 0x0 64 cycles	0x0	R/W	
		D1–0	**OSC1AWT [1:0]**	OSC1A stabilization wait cycle select	OSC1AWT[1:0] / Wait cycle 0x3 2048 cycles 0x2 4096 cycles 0x1 8192 cycles 0x0 16384 cycles	0x0	R/W	
PCLK Control Register (CLG_PCLK)	**0x5080** (8 bits)	D7–2	**–**	reserved	–	–	–	0 when being read.
		D1–0	**PCKEN[1:0]**	PCLK enable	PCKEN[1:0] / PCLK supply 0x3 Enable 0x2 Not allowed 0x1 Not allowed 0x0 Disable	0x3	R/W	
CCLK Control Register (CLG_CCLK)	**0x5081** (8 bits)	D7–2	**–**	reserved	–	–	–	0 when being read.
		D1–0	**CCLKGR[1:0]**	CCLK clock gear ratio select	CCLKGR[1:0] / Gear ratio 0x3 1/8 0x2 1/4 0x1 1/2 0x0 1/1	0x0	R/W	

0x5078–0x5079 Theoretical Regulation Circuit

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
TR Control Register (TR_CTL)	0x5078 (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.
		D3	RCLKFSEL	Monitor clock frequency select	1 1 Hz 0 256 Hz	0	R/W	
		D2	RCLKMON	Regulated clock monitor enable	1 Enable 0 Disable	0	R/W	
		D1	–	reserved	–	–	–	0 when being read.
		D0	REGTRIG	Regulation trigger	1 Trigger 0 Ignored	0	W	
TR Value Register (TR_VAL)	0x5079 (8 bits)	D7–5	–	reserved	–	–	–	0 when being read.
		D4–0	TRIM[4:0]	Regulation value	TRIM[4:0] / Regulation value: 0xf +16 0xe +15 : : 0x1 +2 0x0 +1 0x1f 0 0x1e -1 : : 0x11 -14 0x10 -15	0x0	R/W	

0x5100–0x5102 SVD Circuit

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SVD Enable Register (SVD_EN)	0x5100 (8 bits)	D7–1	–	reserved	–	–	–	0 when being read.
		D0	SVDEN	SVD enable	1 Enable 0 Disable	0	R/W	
SVD Comparison Voltage Register (SVD_CMP)	0x5101 (8 bits)	D7–5	–	reserved	–	–	–	0 when being read.
		D4–0	SVDC[4:0]	SVD comparison voltage select	SVDC[4:0] / Voltage: 0x1f–0x1b reserved 0x1a 3.20 V 0x19 3.10 V 0x18 3.00 V 0x17 2.90 V 0x16 2.80 V 0x15 2.70 V 0x14 2.60 V 0x13 2.50 V 0x12 2.40 V 0x11 2.30 V 0x10 2.20 V 0xf 2.10 V 0xe 2.00 V 0xd–0x0 reserved	0x0	R/W	
SVD Detection Result Register (SVD_RSLT)	0x5102 (8 bits)	D7–1	–	reserved	–	–	–	0 when being read.
		D0	SVDDT	SVD detection result	1 Low 0 Normal	×	R	

0x5120 Power Generator

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
V_{D1} Control Register (VD1_CTL)	0x5120 (8 bits)	D7–6	–	reserved	–	–	–	0 when being read.
		D5	HVLD	V_{D1} heavy load protection mode	1 On 0 Off	0	R/W	
		D4–0	–	reserved	–	–	–	0 when being read.

0x506e, 0x5180–0x5182 Sound Generator

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SND Clock Control Register (SND_CLK)	0x506e (8 bits)	D7–1	–	reserved	–	–	–	0 when being read.
		D0	SNDCLKE	SND clock enable	1 Enable 0 Disable	0	R/W	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SND Control Register (SND_CTL)	**0x5180** (8 bits)	D7–6	**–**	reserved	–	–	–	0 when being read.
		D5–4	**BZTM[1:0]**	Buzzer envelope time/one-shot output time select	BZTM[1:0] Time 0x3 125 ms 0x2 62.5 ms 0x1 31.25 ms 0x0 15.63 ms	0x0	R/W	
		D3–2	**BZMD[1:0]**	Buzzer mode select	BZMD[1:0] Mode 0x3 reserved 0x2 Envelope 0x1 One-shot 0x0 Normal	0x0	R/W	
		D1	**–**	reserved	–	–	–	0 when being read.
		D0	**BZEN**	Buzzer output control	1 On/Trigger 0 Off	0	R/W	
Buzzer Frequency Control Register (SND_BZFQ)	**0x5181** (8 bits)	D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**BZFQ[2:0]**	Buzzer frequency select	BZFQ[2:0] Frequency 0x7 1170.3 Hz 0x6 1365.3 Hz 0x5 1638.4 Hz 0x4 2048.0 Hz 0x3 2340.6 Hz 0x2 2730.7 Hz 0x1 3276.8 Hz 0x0 4096.0 Hz	0x0	R/W	
Buzzer Duty Ratio Control Register (SND_BZDT)	**0x5182** (8 bits)	D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**BZDT[2:0]**	Buzzer duty ratio select	BZDT[2:0] Duty (volume) 0x7 Level 8 (Min.) : : 0x0 Level 1 (Max.)	0x0	R/W	

0x506f, 0x51a0–0x51a3 Temperature Detection Circuit

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
TEM Clock Control Register (TEM_CLK)	**0x506f** (8 bits)	D7–6	**–**	reserved	–	–	–	0 when being read.
		D5–4	**TEMCLKD [1:0]**	TEM clock division ratio select	TEMCLKD[1:0] Division ratio 0x3 1/8 0x2 1/4 0x1 1/2 0x0 1/1	0x0	R/W	When the clock source is OSC3B or OSC3A
		D3–2	**TEMCLK SRC[1:0]**	TEM clock source select	TEMCLK SRC[1:0] Clock source 0x3 reserved 0x2 OSC3A 0x1 OSC1 0x0 OSC3B	0x0	R/W	
		D1	**–**	reserved	–	–	–	0 when being read.
		D0	**TEMCLKE**	TEM clock enable	1 Enable 0 Disable	0	R/W	
TEM Comparison Time Setting Register (TEM_TIME)	**0x51a0** (8 bits)	D7–0	**CVTM[7:0]**	Comparison time select $\frac{CVTM[7:0]+1}{f_{TEMCLK}} \geq 150\ \mu s$	0 to 255 clocks	0	R/W	
TEM Control Register (TEM_CTL)	**0x51a1** (8 bits)	D7–5	**–**	reserved	–	–	–	0 when being read.
		D4	**TEMIE**	Conversion completion int. enable	1 Enable 0 Disable	0	R/W	
		D3–2	**–**	reserved	–	–	–	0 when being read.
		D1	**TEMTRG**	Conversion trigger	1 Start 0 Stop	0	W	
		D0	**TEMEN**	TEM enable	1 Enable 0 Disable	0	R/W	
TEM Status Register (TEM_STAT)	**0x51a2** (8 bits)	D7–5	**–**	reserved	–	–	–	0 when being read.
		D4	**TEMIF**	Conversion completion interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D3–1	**–**	reserved	–	–	–	0 when being read.
		D0	**TEMST**	Conversion status	1 Busy 0 Idle	0	R	
TEM Conversion Result Register (TEM_RSLT)	**0x51a3** (8 bits)	D7–0	**TEMP[7:0]**	Conversion result	0 to 255	0	R	

0x5200–0x52a7 P Port & Port MUX

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
P0 Port Input Data Register (P0_IN)	**0x5200** (8 bits)	D7–0	**P0IN[7:0]**	P0[7:0] port input data	1 1 (H) 0 0 (L)	×	R	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
P0 Port Output Data Register (P0_OUT)	**0x5201** (8 bits)	D7–0	**P0OUT[7:0]**	P0[7:0] port output data	1 1 (H) 0 0 (L)	0	R/W	
P0 Port Output Enable Register (P0_OEN)	**0x5202** (8 bits)	D7–0	**P0OEN[7:0]**	P0[7:0] port output enable	1 Enable 0 Disable	0	R/W	
P0 Port Pull-up Control Register (P0_PU)	**0x5203** (8 bits)	D7–0	**P0PU[7:0]**	P0[7:0] port pull-up enable	1 Enable 0 Disable	1 (0xff)	R/W	
P0 Port Interrupt Mask Register (P0_IMSK)	**0x5205** (8 bits)	D7–0	**P0IE[7:0]**	P0[7:0] port interrupt enable	1 Enable 0 Disable	0	R/W	
P0 Port Interrupt Edge Select Register (P0_EDGE)	**0x5206** (8 bits)	D7–0	**P0EDGE[7:0]**	P0[7:0] port interrupt edge select	1 Falling edge 0 Rising edge	0	R/W	
P0 Port Interrupt Flag Register (P0_IFLG)	**0x5207** (8 bits)	D7–0	**P0IF[7:0]**	P0[7:0] port interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
P0 Port Chattering Filter Control Register (P0_CHAT)	**0x5208** (8 bits)	D7	**–**	reserved	–	–	–	0 when being read.
		D6–4	**P0CF2[2:0]**	P0[7:4] chattering filter time	P0CF2[2:0] Filter time 0x7 16384/fPCLK 0x6 8192/fPCLK 0x5 4096/fPCLK 0x4 2048/fPCLK 0x3 1024/fPCLK 0x2 512/fPCLK 0x1 256/fPCLK 0x0 None	0x0	R/W	
		D3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**P0CF1[2:0]**	P0[3:0] chattering filter time	P0CF1[2:0] Filter time 0x7 16384/fPCLK 0x6 8192/fPCLK 0x5 4096/fPCLK 0x4 2048/fPCLK 0x3 1024/fPCLK 0x2 512/fPCLK 0x1 256/fPCLK 0x0 None	0x0	R/W	
P0 Port Key-Entry Reset Configuration Register (P0_KRST)	**0x5209** (8 bits)	D7–2	**–**	reserved	–	–	–	0 when being read.
		D1–0	**P0KRST[1:0]**	P0 port key-entry reset configuration	P0KRST[1:0] Configuration 0x3 P0[3:0] = 0 0x2 P0[2:0] = 0 0x1 P0[1:0] = 0 0x0 Disable	0x0	R/W	
P0 Port Input Enable Register (P0_IEN)	**0x520a** (8 bits)	D7–0	**P0IEN[7:0]**	P0[7:0] port input enable	1 Enable 0 Disable	1 (0xff)	R/W	
P1 Port Input Data Register (P1_IN)	**0x5210** (8 bits)	D7–0	**P1IN[7:0]**	P1[7:0] port input data	1 1 (H) 0 0 (L)	×	R	
P1 Port Output Data Register (P1_OUT)	**0x5211** (8 bits)	D7–0	**P1OUT[7:0]**	P1[7:0] port output data	1 1 (H) 0 0 (L)	0	R/W	
P1 Port Output Enable Register (P1_OEN)	**0x5212** (8 bits)	D7–0	**P1OEN[7:0]**	P1[7:0] port output enable	1 Enable 0 Disable	0	R/W	
P1 Port Pull-up Control Register (P1_PU)	**0x5213** (8 bits)	D7–0	**P1PU[7:0]**	P1[7:0] port pull-up enable	1 Enable 0 Disable	1 (0xff)	R/W	
P1 Port Input Enable Register (P1_IEN)	**0x521a** (8 bits)	D7–0	**P1IEN[7:0]**	P1[7:0] port input enable	1 Enable 0 Disable	1 (0xff)	R/W	
P2 Port Input Data Register (P2_IN)	**0x5220** (8 bits)	D7–0	**P2IN[7:0]**	P2[7:0] port input data	1 1 (H) 0 0 (L)	×	R	
P2 Port Output Data Register (P2_OUT)	**0x5221** (8 bits)	D7–0	**P2OUT[7:0]**	P2[7:0] port output data	1 1 (H) 0 0 (L)	0	R/W	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
P2 Port Output Enable Register (P2_OEN)	**0x5222** (8 bits)	D7–0	**P2OEN[7:0]**	P2[7:0] port output enable	1 Enable 0 Disable	0	R/W	
P2 Port Pull-up Control Register (P2_PU)	**0x5223** (8 bits)	D7–0	**P2PU[7:0]**	P2[7:0] port pull-up enable	1 Enable 0 Disable	1 (0xff)	R/W	
P2 Port Interrupt Mask Register (P2_IMSK)	**0x5225** (8 bits)	D7–0	**P2IE[7:0]**	P2[7:0] port interrupt enable	1 Enable 0 Disable	0	R/W	
P2 Port Interrupt Edge Select Register (P2_EDGE)	**0x5226** (8 bits)	D7–0	**P2EDGE[7:0]**	P2[7:0] port interrupt edge select	1 Falling edge 0 Rising edge	0	R/W	
P2 Port Interrupt Flag Register (P2_IFLG)	**0x5227** (8 bits)	D7–0	**P2IF[7:0]**	P2[7:0] port interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
P2 Port Chattering Filter Control Register (P2_CHAT)	**0x5228** (8 bits)	D7	**–**	reserved	–	–	–	0 when being read.
		D6–4	**P2CF2[2:0]**	P2[7:4] chattering filter time	P2CF2[2:0] Filter time 0x7 16384/fPCLK 0x6 8192/fPCLK 0x5 4096/fPCLK 0x4 2048/fPCLK 0x3 1024/fPCLK 0x2 512/fPCLK 0x1 256/fPCLK 0x0 None	0 0x0	R/W R/W	
		D3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**P2CF1[2:0]**	P2[3:0] chattering filter time	P2CF1[2:0] Filter time 0x7 16384/fPCLK 0x6 8192/fPCLK 0x5 4096/fPCLK 0x4 2048/fPCLK 0x3 1024/fPCLK 0x2 512/fPCLK 0x1 256/fPCLK 0x0 None	0x0	R/W	
P2 Port Input Enable Register (P2_IEN)	**0x522a** (8 bits)	D7–0	**P2IEN[7:0]**	P2[7:0] port input enable	1 Enable 0 Disable	1 (0xff)	R/W	
P3 Port Input Data Register (P3_IN)	**0x5230** (8 bits)	D7–5	**–**	reserved	–	–	–	0 when being read.
		D4–0	**P3IN[4:0]**	P3[4:0] port input data	1 1 (H) 0 0 (L)	×	R	
P3 Port Output Data Register (P3_OUT)	**0x5231** (8 bits)	D7–5	**–**	reserved	–	–	–	0 when being read.
		D4–0	**P3OUT[4:0]**	P3[4:0] port output data	1 1 (H) 0 0 (L)	0	R/W	
P3 Port Output Enable Register (P3_OEN)	**0x5232** (8 bits)	D7–6	**–**	reserved	–	–	–	0 when being read.
		D5	**–**	reserved	–	–	–	1 when being read.
		D4–0	**P3OEN[4:0]**	P3[4:0] port output enable	1 Enable 0 Disable	0	R/W	
P3 Port Pull-up Control Register (P3_PU)	**0x5233** (8 bits)	D7–5	**–**	reserved	–	–	–	0 when being read.
		D4–0	**P3PU[4:0]**	P3[4:0] port pull-up enable	1 Enable 0 Disable	1 (0x1f)	R/W	
P3 Port Input Enable Register (P3_IEN)	**0x523a** (8 bits)	D7–5	**–**	reserved	–	–	–	0 when being read.
		D4–0	**P3IEN[4:0]**	P3[4:0] port input enable	1 Enable 0 Disable	1 (0x1f)	R/W	
P0[3:0] Port Function Select Register (P00_03PMUX)	**0x52a0** (8 bits)	D7–6	**P03MUX[1:0]**	P03 port function select	P03MUX[1:0] Function 0x3 reserved 0x2 reserved 0x1 TOUTB0/CAPB0 0x0 P03	0x0	R/W	
		D5–4	**P02MUX[1:0]**	P02 port function select	P02MUX[1:0] Function 0x3 reserved 0x2 reserved 0x1 TOUTA0/CAPA0 0x0 P02	0x0	R/W	
		D3–2	**P01MUX[1:0]**	P01 port function select	P01MUX[1:0] Function 0x3 reserved 0x2 reserved 0x1 reserved 0x0 P01/EXCL0	0x0	R/W	
		D1–0	**P00MUX[1:0]**	P00 port function select	P00MUX[1:0] Function 0x3 reserved 0x2 REGMON 0x1 RFCLKO 0x0 P00	0x0	R/W	

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
P0[7:4] Port Function Select Register (P04_07PMUX)	**0x52a1** (8 bits)	D7–6	**P07MUX[1:0]**	P07 port function select	P07MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	BZ			
					0x0	P07			
		D5–4	**P06MUX[1:0]**	P06 port function select	P06MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	#BZ			
					0x1	TOUTB1/CAPB1			
					0x0	P06			
		D3–2	**P05MUX[1:0]**	P05 port function select	P05MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	TOUTA1/CAPA1			
					0x0	P05			
		D1–0	**P04MUX[1:0]**	P04 port function select	P04MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	reserved			
					0x0	P04/EXCL1			
P1[3:0] Port Function Select Register (P10_13PMUX)	**0x52a2** (8 bits)	D7–6	**P13MUX[1:0]**	P13 port function select	P13MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	FOUTA			
					0x1	#SPISS0			
					0x0	P13			
		D5–4	**P12MUX[1:0]**	P12 port function select	P12MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	SCLS			
					0x1	SDI0			
					0x0	P12			
		D3–2	**P11MUX[1:0]**	P11 port function select	P11MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	SDAS			
					0x1	SDO0			
					0x0	P11			
		D1–0	**P10MUX[1:0]**	P10 port function select	P10MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	SPICLK0			
					0x0	P10			
P1[7:4] Port Function Select Register (P14_17PMUX)	**0x52a3** (8 bits)	D7–6	**P17MUX[1:0]**	P17 port function select	P17MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	EPDCLK			
					0x1	#BFR			
					0x0	P17			
		D5–4	**P16MUX[1:0]**	P16 port function select	P16MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	SCLM			
					0x1	SOUT0			
					0x0	P16			
		D3–2	**P15MUX[1:0]**	P15 port function select	P15MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	SDAM			
					0x1	SIN0			
					0x0	P15			
		D1–0	**P14MUX[1:0]**	P14 port function select	P14MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	EPDTRG			
					0x1	SCLK0			
					0x0	P14			

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
P2[3:0] Port Function Select Register (P20_23PMUX)	**0x52a4** (8 bits)	D7–6	**P23MUX[1:0]**	P23 port function select	P23MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	RFIN0			
					0x1	SCLM			
					0x0	P23			
		D5–4	**P22MUX[1:0]**	P22 port function select	P22MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	REF0			
					0x1	SDAM			
					0x0	P22			
		D3–2	**P21MUX[1:0]**	P21 port function select	P21MUX[1:0]	Function	0x0	R/W	
					0x3	#BZ			
					0x2	SENA0			
					0x1	SCLS			
					0x0	P21			
		D1–0	**P20MUX[1:0]**	P20 port function select	P20MUX[1:0]	Function	0x0	R/W	
					0x3	BZ			
					0x2	SENB0			
					0x1	SDAS			
					0x0	P20			
P2[7:4] Port Function Select Register (P24_27PMUX)	**0x52a5** (8 bits)	D7–6	**P27MUX[1:0]**	P27 port function select	P27MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	SENB1			
					0x0	P27			
		D5–4	**P26MUX[1:0]**	P26 port function select	P26MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	SENA1			
					0x0	P26			
		D3–2	**P25MUX[1:0]**	P25 port function select	P25MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	REF1			
					0x0	P25			
		D1–0	**P24MUX[1:0]**	P24 port function select	P24MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	RFIN1			
					0x0	P24			
P3[3:0] Port Function Select Register (P30_33PMUX)	**0x52a6** (8 bits)	D7–6	**P33MUX[1:0]**	P33 port function select	P33MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	P33			
					0x0	DST2			
		D5–4	**P32MUX[1:0]**	P32 port function select	P32MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	P32			
					0x0	DSIO			
		D3–2	**P31MUX[1:0]**	P31 port function select	P31MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	EPDCLK			
					0x0	P31			
		D1–0	**P30MUX[1:0]**	P30 port function select	P30MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	#SPISS0			
					0x1	FOUTB			
					0x0	P30			
P34 Port Function Select Register (P34PMUX)	**0x52a7** (8 bits)	D7–2	**–**	reserved	–		–	–	0 when being read.
		D1–0	**P34MUX[1:0]**	P34 port function select	P34MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	P34			
					0x0	DCLK			

0x4020, 0x5322–0x532c MISC Registers

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Debug Mode Control Register 1 (MISC_DMODE1)	**0x4020** (8 bits)	D7–2	**–**	reserved	–				–	–	0 when being read.
		D1	**DBRUN1**	Run/stop select in debug mode	1	Run	0	Stop	0	R/W	
		D0	**–**	reserved	–				–	–	0 when being read.

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Debug Mode Control Register 2 (MISC_DMODE2)	**0x5322** (16 bits)	D15–1	**–**	reserved	–	–	–	0 when being read.
		D0	**DBRUN2**	Run/stop select in debug mode (except PCLK peripheral circuits)	1 Run 0 Stop	0	R/W	
MISC Protect Register (MISC_PROT)	**0x5324** (16 bits)	D15–0	**PROT[15:0]**	MISC register write protect	Writing 0x96 removes the write protection of the MISC registers (0x5326–0x532a). Writing another value set the write protection.	0x0	R/W	
IRAM Size Select Register (MISC_ IRAMSZ)	**0x5326** (16 bits)	D15–9	**–**	reserved	–	–	–	0 when being read.
		D8	**DBADR**	Debug base address select	1 0x0 0 0xfffc00	0	R/W	
		D7	**–**	reserved	–	–	–	0 when being read.
		D6–4	**IRAMACTSZ [2:0]**	IRAM actual size	0x3 (= 2KB)	0x3	R	
		D3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**IRAMSZ[2:0]**	IRAM size select	IRAMSZ[2:0] / Size 0x5 / 512B 0x4 / 1KB 0x3 / 2KB Other / reserved	0x3	R/W	
Vector Table Address Low Register (MISC_TTBRL)	**0x5328** (16 bits)	D15–8	**TTBR[15:8]**	Vector table base address A[15:8]	0x0–0xff	0x80	R/W	
		D7–0	**TTBR[7:0]**	Vector table base address A[7:0] (fixed at 0)	0x0	0x0	R	
Vector Table Address High Register (MISC_TTBRH)	**0x532a** (16 bits)	D15–8	**–**	reserved	–	–	–	0 when being read.
		D7–0	**TTBR[23:16]**	Vector table base address A[23:16]	0x0–0xff	0x0	R/W	
PSR Register (MISC_PSR)	**0x532c** (16 bits)	D15–8	**–**	reserved	–	–	–	0 when being read.
		D7–5	**PSRIL[2:0]**	PSR interrupt level (IL) bits	0x0 to 0x7	0x0	R	
		D4	**PSRIE**	PSR interrupt enable (IE) bit	1 1 (enable) 0 0 (disable)	0	R	
		D3	**PSRC**	PSR carry (C) flag	1 1 (set) 0 0 (cleared)	0	R	
		D2	**PSRV**	PSR overflow (V) flag	1 1 (set) 0 0 (cleared)	0	R	
		D1	**PSRZ**	PSR zero (Z) flag	1 1 (set) 0 0 (cleared)	0	R	
		D0	**PSRN**	PSR negative (N) flag	1 1 (set) 0 0 (cleared)	0	R	

0x5067, 0x53a0–0x53ae R/F Converter

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RFC Clock Control Register (RFC_CLK)	**0x5067** (8 bits)	D7–6	**–**	reserved	–	–	–	0 when being read.
		D5–4	**RFCCLKD [1:0]**	RFC clock division ratio select	RFCCLKD[1:0] / Division ratio 0x3 / 1/8 0x2 / 1/4 0x1 / 1/2 0x0 / 1/1	0x0	R/W	When the clock source is OSC3B or OSC3A
		D3–2	**RFCCLK SRC[1:0]**	RFC clock source select	RFCCLK SRC[1:0] / Clock source 0x3 / reserved 0x2 / OSC3A 0x1 / OSC1 0x0 / OSC3B	0x0	R/W	
		D1	**–**	reserved	–	–	–	0 when being read.
		D0	**RFCCLKE**	RFC clock enable	1 Enable 0 Disable	0	R/W	
RFC Control Register (RFC_CTL)	**0x53a0** (16 bits)	D15–8	**–**	reserved	–	–	–	0 when being read.
		D7	**CONEN**	Continuous oscillation enable	1 Enable 0 Disable	0	R/W	
		D6	**EVTEN**	Event counter mode enable	1 Enable 0 Disable	0	R/W	
		D5–4	**SMODE[1:0]**	Sensor oscillation mode select	SMODE[1:0] / Sensor 0x3–0x2 / reserved 0x1 / AC resistive 0x0 / DC resistive	0x0	R/W	
		D3–2	**–**	reserved	–	–	–	0 when being read.
		D1	**CHSEL**	Conversion channel select	1 Ch.1 0 Ch.0	0	R/W	
		D0	**RFCEN**	RFC enable	1 Enable 0 Disable	0	R/W	
RFC Oscillation Trigger Register (RFC_TRG)	**0x53a2** (16 bits)	D15–3	**–**	reserved	–	–	–	0 when being read.
		D2	**SSENB**	Sensor B oscillation control/status	1 Start/Run 0 Stop	0	R/W	
		D1	**SSENA**	Sensor A oscillation control/status	1 Start/Run 0 Stop	0	R/W	
		D0	**SREF**	Reference oscillation control/status	1 Start/Run 0 Stop	0	R/W	
RFC Measurement Counter Low Register (RFC_MCL)	**0x53a4** (16 bits)	D15–0	**MC[15:0]**	Measurement counter low-order 16-bit data	0x0–0xffff	0x0	R/W	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RFC Measurement Counter High Register (RFC_MCH)	**0x53a6** (16 bits)	D15–8	**–**	reserved	–	–	–	0 when being read.
		D7–0	**MC[23:16]**	Measurement counter high-order 8-bit data	0x0–0xff	0x0	R/W	
RFC Time Base Counter Low Register (RFC_TCL)	**0x53a8** (16 bits)	D15–0	**TC[15:0]**	Time base counter low-order 16-bit data	0x0–0xffff	0x0	R/W	
RFC Time Base Counter High Register (RFC_TCH)	**0x53aa** (16 bits)	D15–8	**–**	reserved	–	–	–	0 when being read.
		D7–0	**TC[23:16]**	Time base counter high-order 8-bit data	0x0–0xff	0x0	R/W	
RFC Interrupt Mask Register (RFC_IMSK)	**0x53ac** (16 bits)	D15–5	**–**	reserved	–	–	–	0 when being read.
		D4	**OVTCIE**	TC overflow error interrupt enable	1 Enable 0 Disable	0	R/W	
		D3	**OVMCIE**	MC overflow error interrupt enable	1 Enable 0 Disable	0	R/W	
		D2	**ESENBIE**	Sensor B oscillation completion interrupt enable	1 Enable 0 Disable	0	R/W	
		D1	**ESENAIE**	Sensor A oscillation completion interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	**EREFIE**	Reference oscillation completion interrupt enable	1 Enable 0 Disable	0	R/W	
RFC Interrupt Flag Register (RFC_IFLG)	**0x53ae** (16 bits)	D15–5	**–**	reserved	–	–	–	0 when being read.
		D4	**OVTCIF**	TC overflow error interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D3	**OVMCIF**	MC overflow error interrupt flag		0	R/W	
		D2	**ESENBIF**	Sensor B oscillation completion interrupt flag		0	R/W	
		D1	**ESENAIF**	Sensor A oscillation completion interrupt flag		0	R/W	
		D0	**EREFIF**	Reference oscillation completion interrupt flag		0	R/W	

0x5068, 0x5400–0x540c 16-bit PWM Timer Ch.0

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16A Clock Control Register Ch.0 (T16A_CLK0)	**0x5068** (8 bits)	D7–4	**T16ACLKD [3:0]**	Clock division ratio select	T16ACLKD[3:0]: Division ratio — OSC3A or OSC3B / OSC1 0xf: reserved / reserved 0xe: 1/16384 / reserved 0xd: 1/8192 / reserved 0xc: 1/4096 / reserved 0xb: 1/2048 / reserved 0xa: 1/1024 / reserved 0x9: 1/512 / F256 0x8: 1/256 / 1/256 0x7: 1/128 / 1/128 0x6: 1/64 / 1/64 0x5: 1/32 / 1/32 0x4: 1/16 / 1/16 0x3: 1/8 / 1/8 0x2: 1/4 / 1/4 0x1: 1/2 / 1/2 0x0: 1/1 / 1/1	0x0	R/W	F256: Regulated 256 Hz clock
		D3–2	**T16ACLK SRC[1:0]**	Clock source select	T16ACLKSRC[1:0]: Clock source 0x3: External clock 0x2: OSC3A 0x1: OSC1 0x0: OSC3B	0x0	R/W	
		D1	**MULTIMD**	Multi-comparator/capture mode select	1 Multi 0 Normal	0	R/W	
		D0	**T16ACLKE**	Count clock enable	1 Enable 0 Disable	0	R/W	
T16A Counter Ch.0 Control Register (T16A_CTL0)	**0x5400** (16 bits)	D15–7	**–**	reserved	–	–	–	0 when being read.
		D6	**HCM**	Half clock mode enable	1 Enable 0 Disable	0	R/W	
		D5–4	**CCABCNT [1:0]**	Counter select	CCABCNT[1:0]: Counter Ch. 0x3, 0x2: reserved 0x1: Ch.1 0x0: Ch.0	0x0	R/W	
		D3	**CBUFEN**	Compare buffer enable	1 Enable 0 Disable	0	R/W	
		D2	**TRMD**	Count mode select	1 One-shot 0 Repeat	0	R/W	
		D1	**PRESET**	Counter reset	1 Reset 0 Ignored	0	W	0 when being read.
		D0	**PRUN**	Counter run/stop control	1 Run 0 Stop	0	R/W	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16A Counter Ch.0 Data Register (T16A_TC0)	**0x5402** (16 bits)	D15–0	**T16ATC [15:0]**	Counter data T16ATC15 = MSB T16ATC0 = LSB	0x0 to 0xffff	0x0	R	
T16A Comparator/ Capture Ch.0 Control Register (T16A_CCCTL0)	**0x5404** (16 bits)	D15–14	**CAPBTRG [1:0]**	Capture B trigger select	CAPBTRG[1:0] / Trigger edge 0x3 / ↑ and ↓ 0x2 / ↓ 0x1 / ↑ 0x0 / None	0x0	R/W	
		D13–12	**TOUTBMD [1:0]**	TOUT B mode select	TOUTBMD[1:0] / Mode 0x3 / cmp B: ↑ or ↓ 0x2 / cmp A: ↑ or ↓ 0x1 / cmp A: ↑, B: ↓ 0x0 / Off	0x0	R/W	
		D11–10	**–**	reserved	–	–	–	0 when being read.
		D9	**TOUTBINV**	TOUT B invert	1 Invert, 0 Normal	0	R/W	
		D8	**CCBMD**	T16A_CCB register mode select	1 Capture, 0 Comparator	0	R/W	
		D7–6	**CAPATRG [1:0]**	Capture A trigger select	CAPATRG[1:0] / Trigger edge 0x3 / ↑ and ↓ 0x2 / ↓ 0x1 / ↑ 0x0 / None	0x0	R/W	
		D5–4	**TOUTAMD [1:0]**	TOUT A mode select	TOUTAMD[1:0] / Mode 0x3 / cmp B: ↑ or ↓ 0x2 / cmp A: ↑ or ↓ 0x1 / cmp A: ↑, B: ↓ 0x0 / Off	0x0	R/W	
		D3–2	**–**	reserved	–	–	–	0 when being read.
		D1	**TOUTAINV**	TOUT A invert	1 Invert, 0 Normal	0	R/W	
		D0	**CCAMD**	T16A_CCA register mode select	1 Capture, 0 Comparator	0	R/W	
T16A Comparator/ Capture Ch.0 A Data Register (T16A_CCA0)	**0x5406** (16 bits)	D15–0	**CCA[15:0]**	Compare/capture A data CCA15 = MSB CCA0 = LSB	0x0 to 0xffff	0x0	R/W	
T16A Comparator/ Capture Ch.0 B Data Register (T16A_CCB0)	**0x5408** (16 bits)	D15–0	**CCB[15:0]**	Compare/capture B data CCB15 = MSB CCB0 = LSB	0x0 to 0xffff	0x0	R/W	
T16A Comparator/ Capture Ch.0 Interrupt Enable Register (T16A_IEN0)	**0x540a** (16 bits)	D15–6	**–**	reserved	–	–	–	0 when being read.
		D5	**CAPBOWIE**	Capture B overwrite interrupt enable	1 Enable, 0 Disable	0	R/W	
		D4	**CAPAOWIE**	Capture A overwrite interrupt enable	1 Enable, 0 Disable	0	R/W	
		D3	**CAPBIE**	Capture B interrupt enable	1 Enable, 0 Disable	0	R/W	
		D2	**CAPAIE**	Capture A interrupt enable	1 Enable, 0 Disable	0	R/W	
		D1	**CBIE**	Compare B interrupt enable	1 Enable, 0 Disable	0	R/W	
		D0	**CAIE**	Compare A interrupt enable	1 Enable, 0 Disable	0	R/W	
T16A Comparator/ Capture Ch.0 Interrupt Flag Register (T16A_IFLG0)	**0x540c** (16 bits)	D15–6	**–**	reserved	–	–	–	0 when being read.
		D5	**CAPBOWIF**	Capture B overwrite interrupt flag	1 Cause of interrupt occurred, 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D4	**CAPAOWIF**	Capture A overwrite interrupt flag		0	R/W	
		D3	**CAPBIF**	Capture B interrupt flag		0	R/W	
		D2	**CAPAIF**	Capture A interrupt flag		0	R/W	
		D1	**CBIF**	Compare B interrupt flag		0	R/W	
		D0	**CAIF**	Compare A interrupt flag		0	R/W	

0x5069, 0x5420–0x542c 16-bit PWM Timer Ch.1

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16A Clock Control Register Ch.1 (T16A_CLK1)	**0x5069** (8 bits)	D7–4	**T16ACLKD [3:0]**	Clock division ratio select	T16ACLKD[3:0] / Division ratio: OSC3A or OSC3B / OSC1 0xf / reserved / reserved 0xe / 1/16384 / reserved 0xd / 1/8192 / reserved 0xc / 1/4096 / reserved 0xb / 1/2048 / reserved 0xa / 1/1024 / reserved 0x9 / 1/512 / F256 0x8 / 1/256 / 1/256 0x7 / 1/128 / 1/128 0x6 / 1/64 / 1/64 0x5 / 1/32 / 1/32 0x4 / 1/16 / 1/16 0x3 / 1/8 / 1/8 0x2 / 1/4 / 1/4 0x1 / 1/2 / 1/2 0x0 / 1/1 / 1/1	0x0	R/W	F256: Regulated 256 Hz clock
		D3–2	**T16ACLK SRC[1:0]**	Clock source select	T16ACLKSRC [1:0] / Clock source 0x3 / External clock 0x2 / OSC3A 0x1 / OSC1 0x0 / OSC3B	0x0	R/W	
		D1	**–**	reserved	–	–	–	0 when being read.
		D0	**T16ACLKE**	Count clock enable	1 Enable 0 Disable	0	R/W	
T16A Counter Ch.1 Control Register (T16A_CTL1)	**0x5420** (16 bits)	D15–7	**–**	reserved	–	–	–	0 when being read.
		D6	**HCM**	Half clock mode enable	1 Enable 0 Disable	0	R/W	
		D5–4	**CCABCNT [1:0]**	Counter select	CCABCNT[1:0] / Counter Ch. 0x3, 0x2 / reserved 0x1 / Ch.1 0x0 / Ch.0	0x0	R/W	
		D3	**CBUFEN**	Compare buffer enable	1 Enable 0 Disable	0	R/W	
		D2	**TRMD**	Count mode select	1 One-shot 0 Repeat	0	R/W	
		D1	**PRESET**	Counter reset	1 Reset 0 Ignored	0	W	0 when being read.
		D0	**PRUN**	Counter run/stop control	1 Run 0 Stop	0	R/W	
T16A Counter Ch.1 Data Register (T16A_TC1)	**0x5422** (16 bits)	D15–0	**T16ATC [15:0]**	Counter data T16ATC15 = MSB T16ATC0 = LSB	0x0 to 0xffff	0x0	R	
T16A Comparator/ Capture Ch.1 Control Register (T16A_CCCTL1)	**0x5424** (16 bits)	D15–14	**CAPBTRG [1:0]**	Capture B trigger select	CAPBTRG[1:0] / Trigger edge 0x3 / ↑ and ↓ 0x2 / ↓ 0x1 / ↑ 0x0 / None	0x0	R/W	
		D13–12	**TOUTBMD [1:0]**	TOUT B mode select	TOUTBMD[1:0] / Mode 0x3 / cmp B: ↑ or ↓ 0x2 / cmp A: ↑ or ↓ 0x1 / cmp A: ↑, B: ↓ 0x0 / Off	0x0	R/W	
		D11–10	**–**	reserved	–	–	–	0 when being read.
		D9	**TOUTBINV**	TOUT B invert	1 Invert 0 Normal	0	R/W	
		D8	**CCBMD**	T16A_CCB register mode select	1 Capture 0 Comparator	0	R/W	
		D7–6	**CAPATRG [1:0]**	Capture A trigger select	CAPATRG[1:0] / Trigger edge 0x3 / ↑ and ↓ 0x2 / ↓ 0x1 / ↑ 0x0 / None	0x0	R/W	
		D5–4	**TOUTAMD [1:0]**	TOUT A mode select	TOUTAMD[1:0] / Mode 0x3 / cmp B: ↑ or ↓ 0x2 / cmp A: ↑ or ↓ 0x1 / cmp A: ↑, B: ↓ 0x0 / Off	0x0	R/W	
		D3–2	**–**	reserved	–	–	–	0 when being read.
		D1	**TOUTAINV**	TOUT A invert	1 Invert 0 Normal	0	R/W	
		D0	**CCAMD**	T16A_CCA register mode select	1 Capture 0 Comparator	0	R/W	
T16A Comparator/ Capture Ch.1 A Data Register (T16A_CCA1)	**0x5426** (16 bits)	D15–0	**CCA[15:0]**	Compare/capture A data CCA15 = MSB CCA0 = LSB	0x0 to 0xffff	0x0	R/W	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16A Comparator/ Capture Ch.1 B Data Register (T16A_CCB1)	**0x5428** (16 bits)	D15–0	**CCB[15:0]**	Compare/capture B data CCB15 = MSB CCB0 = LSB	0x0 to 0xffff	0x0	R/W	
T16A Comparator/ Capture Ch.1 Interrupt Enable Register (T16A_IEN1)	**0x542a** (16 bits)	D15–6	**–**	reserved	–	–	–	0 when being read.
		D5	**CAPBOWIE**	Capture B overwrite interrupt enable	1 Enable 0 Disable	0	R/W	
		D4	**CAPAOWIE**	Capture A overwrite interrupt enable	1 Enable 0 Disable	0	R/W	
		D3	**CAPBIE**	Capture B interrupt enable	1 Enable 0 Disable	0	R/W	
		D2	**CAPAIE**	Capture A interrupt enable	1 Enable 0 Disable	0	R/W	
		D1	**CBIE**	Compare B interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	**CAIE**	Compare A interrupt enable	1 Enable 0 Disable	0	R/W	
T16A Comparator/ Capture Ch.1 Interrupt Flag Register (T16A_IFLG1)	**0x542c** (16 bits)	D15–6	**–**	reserved	–	–	–	0 when being read.
		D5	**CAPBOWIF**	Capture B overwrite interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D4	**CAPAOWIF**	Capture A overwrite interrupt flag		0	R/W	
		D3	**CAPBIF**	Capture B interrupt flag		0	R/W	
		D2	**CAPAIF**	Capture A interrupt flag		0	R/W	
		D1	**CBIF**	Compare B interrupt flag		0	R/W	
		D0	**CAIF**	Compare A interrupt flag		0	R/W	

0x54b0 Flash Controller

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
FLASHC Read Wait Control Register (FLASHC_ WAIT)	**0x54b0** (16 bits)	D15–8	**–**	reserved	–	–	–	0 when being read.
		D7	**–**	reserved	–	X	–	X when being read.
		D6-2	**–**	reserved	–	–	–	0 when being read.
		D1–0	**RDWAIT [1:0]**	Flash read wait cycle	RDWAIT[1:0] Wait 0x3 3 wait 0x2 2 wait 0x1 1 wait 0x0 No wait	0x3	R/W	

0x5070–0x5072, 0x5600–0x567e EPD Controller/Driver

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
EPD Timing Clock Control Register (EPD_TCLK)	**0x5070** (8 bits)	D7	**–**	reserved	–	–	–	0 when being read.
		D6–4	**EPDTCLKD [2:0]**	EPD timing clock division ratio select	EPDTCLKD [2:0] / Division ratio: OSC3B/OSC3A, OSC1 0x7 1/16384 1/128 0x6 1/8192 1/64 0x5 1/4096 1/32 0x4 1/2048 1/16 0x3 1/1024 1/8 0x2 1/512 1/4 0x1 1/256 1/2 0x0 1/128 1/1	0x0	R/W	
		D3–2	**EPDTCLK SRC[1:0]**	EPD timing clock source select	EPDTCLK SRC[1:0] / Clock source 0x3 reserved 0x2 OSC3A 0x1 OSC1 0x0 OSC3B	0x0	R/W	
		D1	**–**	reserved	–	–	–	0 when being read.
		D0	**EPDTCLKE**	EPD timing clock enable	1 Enable 0 Disable	0	R/W	
EPD Doubler Clock Control Register (EPD_DCLK)	**0x5071** (8 bits)	D7	**–**	reserved	–	–	–	0 when being read.
		D6–4	**EPD-DCLKD [2:0]**	EPD doubler clock division ratio select	EPDD CLKD [2:0] / Division ratio: OSC3B, OSC3A, OSC1 0x5 1/128 1/256 reserved 0x4 1/64 1/128 reserved 0x3 1/32 1/64 1/8 0x2 1/16 1/32 1/4 0x1 reserved 1/16 1/2 0x0 reserved reserved 1/1 Other reserved	0x0	R/W	
		D3–2	**EPDDCLK SRC[1:0]**	EPD doubler clock source select	EPDDCLK SRC[1:0] / Clock source 0x3 reserved 0x2 OSC3A 0x1 OSC1 0x0 OSC3B	0x0	R/W	
		D1	**–**	reserved	–	–	–	0 when being read.
		D0	**EPDDCLKE**	EPD doubler clock enable	1 Enable 0 Disable	0	R/W	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
EPD Booster Clock Control Register (EPD_BCLK)	**0x5072** (8 bits)	D7	**–**	reserved	–	–	–	0 when being read.
		D6–4	**EPDBCLKD [2:0]**	EPD booster clock division ratio select	EPDB CLKD [2:0] / Division ratio: OSC3B / OSC3A / OSC1 0x5: 1/128 / 1/256 / reserved 0x4: 1/64 / 1/128 / reserved 0x3: 1/32 / 1/64 / 1/8 0x2: 1/16 / 1/32 / 1/4 0x1: reserved / 1/16 / 1/2 0x0: reserved / reserved / 1/1 Other: reserved	0x0	R/W	
		D3–2	**EPDBCLK SRC[1:0]**	EPD booster clock source select	EPDBCLK SRC[1:0] / Clock source 0x3: reserved 0x2: OSC3A 0x1: OSC1 0x0: OSC3B	0x0	R/W	
		D1	**–**	reserved	–	–	–	0 when being read.
		D0	**EPDBCLKE**	EPD booster clock enable	1 Enable / 0 Disable	0	R/W	
EPD Power Control Register 0 (EPD_PWR0)	**0x5600** (16 bits)	D15–10	**–**	reserved	–	–	–	0 when being read.
		D9	**DBSRT**	V_{E1}–V_{E2} doubler short	1 Short / 0 Open	0	R/W	
		D8	**DBON**	V_{E1} doubler on/off	1 On / 0 Off	0	R/W	
		D7–4	**VECON[3:0]**	V_E regulator contrast setting	VECON[3:0] / Contrast 0xf : High 0x0 : Low	0x7	R/W	
		D3	**–**	reserved	–	–	–	0 when being read.
		D2	**HVLDVE**	V_E heavy load protection mode	1 On / 0 Off	0	R/W	
		D1	**VESEL**	Reference voltage select	1 V_{E2} / 0 V_{E1}	0	R/W	
		D0	**VEON**	V_E regulator on/off	1 On / 0 Off	0	R/W	
EPD Power Control Register 1 (EPD_PWR1)	**0x5602** (16 bits)	D15–12	**VHCON [3:0]**	V_H regulator contrast setting	VHCON[3:0] / Contrast 0xf : High 0x0 : Low	0x5	R/W	
		D11–10	**VHSEL[1:0]**	V_H regulator range select	VHSEL[1:0] / Voltage 0x3: reserved 0x2: 9 V 0x1: 12 V 0x0: 15 V	0x0	R/W	
		D9	**HVLDVH**	V_H heavy load protection mode	1 On / 0 Off	0	R/W	
		D8	**VHON**	V_H regulator on/off	1 On / 0 Off	0	R/W	
		D7–2	**–**	reserved	–	–	–	0 when being read.
		D1	**BSTPLD**	Booster pull-down on/off	1 On / 0 Off	0	R/W	
		D0	**BSTON**	Booster on/off	1 On / 0 Off	0	R/W	
EPD Display Control Register (EPD_CTL)	**0x5604** (16 bits)	D15–5	**–**	reserved	–	–	–	0 when being read.
		D4	**DIRCTL**	Wave/direct mode select	1 Direct / 0 Wave	0	R/W	
		D3–2	**DSPMD [1:0]**	Display mode select	DSPMD[1:0] / Display mode 0x3: All black 0x2: All white 0x1: Reverse 0x0: Normal	0x0	R/W	Effective only in wave mode
		D1	**UPDST**	Display update status	1 Busy / 0 Idle	0	R	Always set to 0 in direct control mode.
		D0	**UPDTRG**	Display update trigger	1 Trigger / 0 Ignored	0	W	0 when being read.
EPD Interrupt Control Register (EPD_INT)	**0x5606** (16 bits)	D15–9	**–**	reserved	–	–	–	0 when being read.
		D8	**DUPDIF**	Display update interrupt flag	1 Occurred / 0 Not occurred	0	R/W	Reset by writing 1.
		D7–1	**–**	reserved	–	–	–	0 when being read.
		D0	**DUPDIE**	Display update interrupt enable	1 Enable / 0 Disable	0	R/W	
EPD Top/Back Plane Data Register (EPD_PLNDAT)	**0x5620** (16 bits)	D15–10	**–**	reserved	–	–	–	0 when being read.
		D9	**SEGHZ**	Segment/back plane Hi-Z control	1 Hi-Z / 0 Normal	0	R/W	Effective only in direct mode
		D8	**TPHZ**	Top plane Hi-Z control	1 Hi-Z / 0 Normal	0	R/W	
		D7–5	**–**	reserved	–	–	–	0 when being read.
		D4	**TP**	Top plane control data	1 High / 0 Low	0	R/W	Effective only in direct mode
		D3–1	**–**	reserved	–	–	–	0 when being read.
		D0	**BP**	Back plane display data	1 Black / 0 White	0	R/W	Wave mode
					1 High / 0 Low			Direct mode
EPD Segment Data Register 0 (EPD_SEGDAT0)	**0x5622** (16 bits)	D15–0	**SEG[15:0]**	ESEG[15:0] display data	1 Black / 0 White	0	R/W	Wave mode
					1 High / 0 Low			Direct mode
EPD Segment Data Register 1 (EPD_SEGDAT1)	**0x5624** (16 bits)	D15–0	**SEG[31:16]**	ESEG[31:16] display data	1 Black / 0 White	0	R/W	Wave mode
					1 High / 0 Low			Direct mode
EPD Segment Data Register 2 (EPD_SEGDAT2)	**0x5626** (16 bits)	D15–0	**SEG[47:32]**	ESEG[47:32] display data	1 Black / 0 White	0	R/W	Wave mode
					1 High / 0 Low			Direct mode

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
EPD Segment Data Register 3 (EPD_SEGDAT3)	0x5628 (16 bits)	D15–0	SEG[63:48]	ESEG[63:48] display data	1 Black 0 White	0	R/W	Wave mode
					1 High 0 Low			Direct mode
EPD Waveform Timing Set 0 Register (EPD_WAVE0) | EPD Waveform Timing Set 31 Register (EPD_WAVE31)	0x5640 | 0x567e (16 bits)	D15	EOW	End of wave	1 End 0 Continue	0	R/W	
		D14	–	reserved	–	–	–	0 when being read.
		D13	HIZ	High impedance	1 Hi-Z 0 Output	0	R/W	
		D12	TP	Top plane	1 High 0 Low	0	R/W	
		D11	BB	Black to black	1 High 0 Low	0	R/W	
		D10	BW	Black to white	1 High 0 Low	0	R/W	
		D9	WB	White to black	1 High 0 Low	0	R/W	
		D8	WW	White to white	1 High 0 Low	0	R/W	
		D7–0	INTV[7:0]	Interval	0 to 255	0x0	R/W	

0x56c0–0x56c8 Real-time Clock

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RTC Control Register (RTC_CTL)	0x56c0 (16 bits)	D15–9	–	reserved	–	–	–	0 when being read.
		D8	RTCST	RTC run/stop status	1 Running 0 Stop	0	R	
		D7–6	–	reserved	–	–	–	0 when being read.
		D5	BCDMD	BCD mode select	1 BCD mode 0 Binary mode	0	R/W	
		D4	RTC24H	24H/12H mode select	1 12H 0 24H	0	R/W	
		D3–1	–	reserved	–	–	–	0 when being read.
		D0	RTCRUN	RTC run/stop control	1 Run 0 Stop	0	R/W	
RTC Interrupt Enable Register (RTC_IEN)	0x56c2 (16 bits)	D15–10	–	reserved	–	–	–	0 when being read.
		D9	INT1DEN	1-day interrupt enable	1 Enable 0 Disable	0	R/W	
		D8	INTHDEN	Half-day interrupt enable	1 Enable 0 Disable	0	R/W	
		D7	INT1HEN	1-hour interrupt enable	1 Enable 0 Disable	0	R/W	
		D6	INT10MEN	10-minute interrupt enable	1 Enable 0 Disable	0	R/W	
		D5	INT1MEN	1-minute interrupt enable	1 Enable 0 Disable	0	R/W	
		D4	INT10SEN	10-second interrupt enable	1 Enable 0 Disable	0	R/W	
		D3	INT1HZEN	1 Hz interrupt enable	1 Enable 0 Disable	0	R/W	
		D2	INT4HZEN	4 Hz interrupt enable	1 Enable 0 Disable	0	R/W	
		D1	INT8HZEN	8 Hz interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	INT32HZEN	32 Hz interrupt enable	1 Enable 0 Disable	0	R/W	
RTC Interrupt Flag Register (RTC_IFLG)	0x56c4 (16 bits)	D15–10	–	reserved	–	–	–	0 when being read.
		D9	INT1D	1-day interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D8	INTHD	Half-day interrupt flag		0	R/W	
		D7	INT1H	1-hour interrupt flag		0	R/W	
		D6	INT10M	10-minute interrupt flag		0	R/W	
		D5	INT1M	1-minute interrupt flag		0	R/W	
		D4	INT10S	10-second interrupt flag		0	R/W	
		D3	INT1HZ	1 Hz interrupt flag		0	R/W	
		D2	INT4HZ	4 Hz interrupt flag		0	R/W	
		D1	INT8HZ	8 Hz interrupt flag		0	R/W	
		D0	INT32HZ	32 Hz interrupt flag		0	R/W	
RTC Minute/Second Counter Register (RTC_MS)	0x56c6 (16 bits)	D15	–	reserved	–	–	–	0 when being read.
		D14–8	RTCMIN [6:0]	Minute counter	0x0 to 0x3b (binary mode) 0x00 to 0x59 (BCD mode)	X	R/W	
		D7	–	reserved	–	–	–	0 when being read.
		D6–0	RTCSEC [6:0]	Second counter	0x0 to 0x3b (binary mode) 0x00 to 0x59 (BCD mode)	X	R/W	
RTC Hour Counter Register (RTC_H)	0x56c8 (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7	AMPM	AM/PM	1 PM 0 AM	X	R/W	
		D6	–	reserved	–	–	–	0 when being read.
		D5–0	RTCHOUR [5:0]	Hour counter	0x0 to 0x17 (binary mode) 0x00 to 0x23 (BCD mode)	X	R/W	

0xffff84–0xffffd0 S1C17 Core I/O

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Processor ID Register (IDIR)	0xffff84 (8 bits)	D7–0	IDIR[7:0]	Processor ID 0x10: S1C17 Core	0x10	0x10	R	
Debug RAM Base Register (DBRAM)	0xffff90 (32 bits)	D31–24	–	Unused (fixed at 0)	0x0	0x0	R	
		D23–0	DBRAM[23:0]	Debug RAM base address	0x7c0	0x7c0	R	

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Debug Control Register (DCR)	**0xffffa0** (8 bits)	D7	**IBE4**	Instruction break #4 enable	1	Enable	0	Disable	0	R/W	
		D6	**IBE3**	Instruction break #3 enable	1	Enable	0	Disable	0	R/W	
		D5	**IBE2**	Instruction break #2 enable	1	Enable	0	Disable	0	R/W	
		D4	**DR**	Debug request flag	1	Occurred	0	Not occurred	0	R/W	Reset by writing 1.
		D3	**IBE1**	Instruction break #1 enable	1	Enable	0	Disable	0	R/W	
		D2	**IBE0**	Instruction break #0 enable	1	Enable	0	Disable	0	R/W	
		D1	**SE**	Single step enable	1	Enable	0	Disable	0	R/W	
		D0	**DM**	Debug mode	1	Debug mode	0	User mode	0	R	
Instruction Break Address Register 1 (IBAR1)	**0xffffb4**	D31–24	**–**	reserved	–				–	–	0 when being read.
	(32 bits)	D23–0	**IBAR1[23:0]**	Instruction break address #1 IBAR123 = MSB IBAR10 = LSB	0x0 to 0xffffff				0x0	R/W	
Instruction Break Address Register 2 (IBAR2)	**0xffffb8**	D31–24	**–**	reserved	–				–	–	0 when being read.
	(32 bits)	D23–0	**IBAR2[23:0]**	Instruction break address #2 IBAR223 = MSB IBAR20 = LSB	0x0 to 0xffffff				0x0	R/W	
Instruction Break Address Register 3 (IBAR3)	**0xffffbc**	D31–24	**–**	reserved	–				–	–	0 when being read.
	(32 bits)	D23–0	**IBAR3[23:0]**	Instruction break address #3 IBAR323 = MSB IBAR30 = LSB	0x0 to 0xffffff				0x0	R/W	
Instruction Break Address Register 4 (IBAR4)	**0xffffd0**	D31–24	**–**	reserved	–				–	–	0 when being read.
	(32 bits)	D23–0	**IBAR4[23:0]**	Instruction break address #4 IBAR423 = MSB IBAR40 = LSB	0x0 to 0xffffff				0x0	R/W	

Appendix B Power Saving

Current consumption will vary dramatically, depending on CPU operating mode, operation clock frequency, and the peripheral circuits being operated. Listed below are the control methods for saving power.

B.1 Clock Control Power Saving

This section describes clock systems that can be controlled via software and power-saving control details. For more information on control registers and control methods, refer to the respective module sections.

System SLEEP

- Execute the `slp` instruction (when RTC is stopped)
  When the entire system can be stopped, stop the RTC and execute the `slp` instruction. The CPU enters SLEEP mode and the OSC1/OSC3A/OSC3B clocks stop. This also stops all peripheral circuits using the OSC1/OSC3A/OSC3B clocks. Starting up the CPU from SLEEP mode is therefore limited to startup using a port (described later).
- Execute the `slp` instruction (when RTC is running)
  When the system except the RTC for time keeping can be stopped, maintain the RTC in running state and execute the `slp` instruction. The CPU enters SLEEP mode and the OSC3A/OSC3B clocks stop. This also stops all peripheral circuits using the OSC3A/OSC3B clocks. Starting up the CPU from SLEEP mode is therefore limited to startup using a port or RTC (described later).

* The RAM retains data even in SLEEP mode.

System clocks

- Select a low-speed clock source (CLG module)
  Select a low-speed oscillator for the system clock source. You can reduce current consumption by selecting the OSC1 clock when low-speed processing is possible.
- Disable unnecessary oscillator circuits (CLG module)
  Operate the oscillator comprising the system clock source. Where possible, stop the other oscillators. You can reduce current consumption by using OSC1 as the system clock and disable the OSC3B and OSC3A oscillators.

CPU clock (CCLK)

- Execute the `halt` instruction
  Execute the `halt` instruction when program execution by the CPU is not required, for example, when the system is waiting for an interrupt. The CPU enters HALT mode and suspends operations, but the peripheral circuits maintain the status in place at the time of the `halt` instruction, enabling use of peripheral circuits for generating interrupts and the EPD controller/driver. You can reduce power consumption even further by suspending unnecessary oscillator and peripheral circuits before executing the `halt` instruction. The CPU is started from HALT mode by an interrupt from a port or the peripheral circuit operating in HALT mode.
- Select a low-speed clock gear (CLG module)
  The CLG module can reduce CPU clock speeds to between 1/1 and 1/8 of the system clock via the clock gear settings. You can reduce current consumption by operating the CPU at the minimum speed required for the application.

Regulated clock (F256)

- Use an interrupt from a peripheral timer module that runs with the regulated clock (F256) to execute theoretical regulation. An interrupt from the timer that runs all the time should be used to reduce current consumption.

Peripheral clock (PCLK)

- Stop PCLK (CLG module)
  Stop the PCLK clock supplied from the CLG to peripheral circuits if none of the following peripheral circuits is required.

Peripheral circuits that use PCLK
- Interrupt controller
- 8-bit timer Ch.0 and Ch.1
- SPI Ch.0
- I^2C master
- I^2C slave
- Power generator
- P ports and port MUX (control registers, chattering filters)
- MISC registers

PCLK is not required for the peripheral modules/functions shown below.

Peripheral circuits/functions that do not use PCLK
- Real-time clock
- Clock timer
- Stopwatch timer
- Watchdog timer
- EPD controller/driver
- Sound generator
- SVD circuit
- Temperature detection circuit
- R/F converter
- 16-bit PWM timer Ch.0 and Ch.1
- UART Ch.0
- FOUTA/FOUTB outputs

Table B.1.1 shows a list of methods for clock control and starting/stopping the CPU.

Table B.1.1 Clock Control List

Current consumption	OSC1	OSC3B/ OSC3A	CPU (CCLK)	PCLK peripheral	RTC	OSC1 peripheral	CPU stop method	CPU startup method
↑ Low	Stop	Stop	Stop	Stop	Stop	Stop	Execute `slp` instruction	1
	Oscillation (for RTC)	Stop	Stop	Stop	Run	Stop	Execute `slp` instruction	1, 2
	Oscillation (for RTC)	Stop	Stop	Stop	Run	Stop	Execute `halt` instruction	1, 2
	Oscillation (system CLK)	Stop	Stop	Stop	Run	Run	Execute `halt` instruction	1, 2, 3
	Oscillation (system CLK)	Stop	Stop	Run	Run	Run	Execute `halt` instruction	1, 2, 3, 4
	Oscillation (system CLK)	Stop	Run (1/1)	Run	Run	Run		
	Oscillation	Oscillation (system CLK)	Stop	Run	Run	Run	Execute `halt` instruction	1, 2, 3, 4
	Oscillation	Oscillation (system CLK)	Run (low gear)	Run	Run	Run		
High ↓	Oscillation	Oscillation (system CLK)	Run (1/1)	Run	Run	Run		

HALT and SLEEP mode cancelation methods (CPU startup method)

1. Startup by port
   Started up by an I/O port interrupt or a debug interrupt (ICD forced break).
2. Startup by RTC
   Started up by an RTC interrupt.
3. Startup by OSC1 peripheral circuit
   Started up by a clock timer, stopwatch timer, or watchdog timer interrupt.
4. Startup by PCLK peripheral circuit
   Started up by a PCLK peripheral circuit interrupt.

B.2 Reducing Power Consumption via Power Supply Control

The available power supply controls are listed below.

V_{D1} regulator

- Note that turning on internal voltage regulator heavy load protection will increase current consumption. Turn off heavy load protection for normal operations. Turn on only if operations are unstable.

EPD power supply circuit

- Setting VESEL to 0 (V_{E1} reference voltage) will increase current consumption.
  Set VESEL to 1 (V_{E2} reference voltage) if the power supply voltage V_{DD} is higher than V_{E2} (setting) + 0.2 V.

Table B.2.1 Example of V_E Regulator Settings

EPD power supply circuit output voltage	(A) When V_{DD} = 3 V and reference voltage = V_{E1}	(B) When V_{DD} = 3 V and reference voltage = V_{E2}
V_{E1}	1.35 V	–
V_{E2}	V_{E1} × 2 = 2.7 V	2.7 V
V_{E5}	V_{E2} × 6 = 16.2 V	V_{E2} × 6 = 16.2 V
V_{EPD}	15 V is output using V_{E5} = 16.2 V as the power source.	15 V is output using V_{E5} = 16.2 V as the power source.

The V_{EPD} voltage is 15 V in both Examples (A) and (B) above. Note, however, boosting voltage in Example (A) consumes current about twice as large as Example (B). Therefore, if V_{DD} is within the range that allows selection of V_{E2} reference voltage, the EPD power supply circuit should be operated with the V_{E2} reference voltage to reduce current consumption.

- Turning on the EPD power supply heavy load protection will increase current consumption.
  Turn off heavy load protection for normal operations. Turn on only if the display is unstable.
- If no EPD driving is required, turn off the EPD power supply.

Power supply voltage detection (SVD) circuit

- Operating the SVD circuit will increase current consumption.
  Turn off power supply voltage detection unless it is required.

Temperature detection circuit

- Activating the temperature detection circuit will increase current consumption.
  Disable temperature detection unless it is required.

R/F converter

- Activating the R/F converter will increase current consumption.
  Disable R/F conversion unless it is required.

B.3 Other Power Saving Methods

Theoretical regulation

- When data input from the I/O port is used to set the theoretical regulation value register (TR_VAL), place the port into output mode and set the read data as the output data after data is read. This reduces the pull-up resistor current that constantly flows.

Appendix C Mounting Precautions

This section describes various precautions for circuit board design and IC mounting.

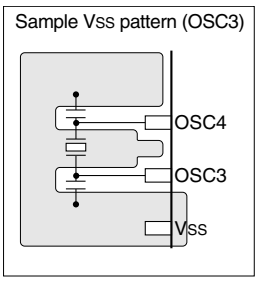
Oscillator circuit

- Oscillation characteristics depend on factors such as components used (resonator, CG, CD) and circuit board patterns. In particular, with ceramic or crystal resonators, select the appropriate capacitors (CG, CD) only after fully evaluating components actually mounted on the circuit board.
- Oscillator clock disturbances caused by noise may cause malfunctions. To prevent such disturbances, consider the following points. The latest devices, in particular, are manufactured by microscopic processes, making them especially susceptible to noise.
  Areas in which noise countermeasures are especially important include the OSC2 pin and related circuit components and wiring. OSC1 pin handling is equally important. The noise precautions required for the OSC1 and OSC2 pins are described below. We also recommend applying similar noise countermeasures to the high-speed oscillator circuit, such as the OSC3 and OSC4 pins and wiring.

(1) Components such as a resonator, resistors, and capacitors connected to the OSC1 (OSC3) and OSC2 (OSC4) pins should have the shortest connections possible.

(2) Wherever possible, avoid locating digital signal lines within 3 mm of the OSC1 (OSC3) and OSC2 (OSC4) pins or related circuit components and wiring. Rapidly-switching signals, in particular, should be kept at a distance from these components. Since the spacing between layers of multi-layer printed circuit boards is a mere 0.1 mm to 0.2 mm, the above precautions also apply when positioning digital signal lines on other layers.
  Never place digital signal lines alongside such components or wiring, even if more than 3 mm distance or located on other layers. Avoid crossing wires.

(3) Use VSS to shield 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 output clock waveform by running the actual application program within the product. Use an oscilloscope to check the FOUTA or FOUTB pin output.
  You can check the quality of the OSC3 output waveform via the FOUTA/B output. Confirm that the frequency is as designed, is free of noise, and has minimal jitter.
  You can check the quality of the OSC1 waveform via the FOUTA/B output. In particular, enlarge the areas before and after the clock rising and falling edges and take special care to confirm that the regions approximately 100 ns to either side are free of clock or spiking noise.

  Failure to observe precautions (1) to (3) adequately may lead to jitter in the OSC3 output and noise in the OSC1 output. Jitter in the OSC3 output will reduce operating frequencies, while noise in the OSC1 output will destabilize timers operated by the OSC1 clock as well as CPU Core operations when the system clock switches to OSC1.

Reset circuit

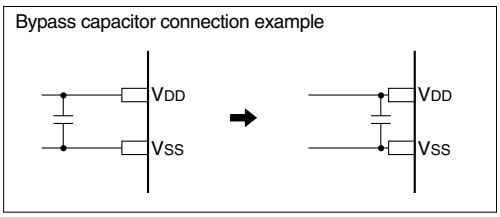
- The reset signal input to the #RESET pin when power is turned on will vary, depending on various factors, such as power supply start-up time, components used, and circuit board patterns. Constants such as capacitance and resistance should be determined through testing with real-world products.
- Components such as capacitors and resistors connected to the #RESET pin should have the shortest connections possible to prevent noise-induced resets.

Power supply circuit

Sudden power supply fluctuations due to noise will cause malfunctions. Consider the following issues.

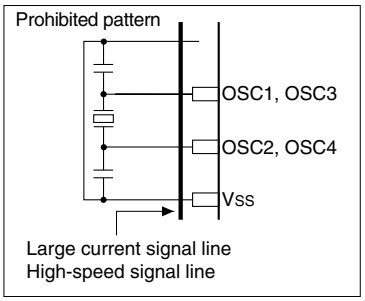
(1) Connections from the power supply to the VDD and VSS pins should be implemented via the shortest, thickest patterns possible.

(2) If a bypass capacitor is connected between VDD and VSS, connections between the VDD and VSS 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 circuits susceptible to noise, such as oscillators.
- Locating signal lines in parallel over significant distances or crossing signal lines operating at high speed will cause malfunctions due to noise generated by mutual interference. Specifically, avoid positioning crossing signal lines operating at high speed close to circuits susceptible to noise, such as oscillators.



Handling of light (for bare chip mounting)

The characteristics of semiconductor components can vary when exposed to light. ICs may malfunction or nonvolatile memory data may be corrupted if ICs are exposed to light.
Consider the following precautions for circuit boards and products in which this IC is mounted to prevent IC malfunctions attributable to light exposure.

(1) Design and mount the product so that the IC is shielded from light during use.

(2) Shield the IC from light during inspection processes.

(3) Shield the IC on the upper, underside, and side faces of the IC chip.

(4) Mount the IC chip within one week of opening the package. If the IC chip must be stored before mounting, take measures to ensure light shielding.

(5) Adequate evaluations are required to assess nonvolatile memory data retention characteristics before product delivery if the product is subjected to heat stress exceeding regular reflow conditions during mounting processes.

Unused pins

(1) I/O port (P) pins
Unused pins should be left open. The control registers should be fixed at the initial status (input with pull-up enabled).

(2) OSC1, OSC2, OSC3, and OSC4 pins
If the OSC1A or OSC3A oscillator circuit is not used, the OSC1 and OSC2 pins or the OSC3 and OSC4 pins should be left open. The control registers should be fixed at the initial status (oscillation disabled).

(3) V_{E1-5}, V_{EPD}, C_{D1}, C_{D2}, C_{B1}–C_{B4}, ESEG*x*, ETP*x*, and EBP*x* pins
If the EPD controller/driver is not used, these pins should be left open. The control registers should be fixed at the initial status (display off). The unused ESEG*x* pins that are not required to connect should be left open even if the EPD controller/driver is used.

Handling gold bump chip products (subjecting to high temperature stress)

If an IC is subjected to high temperature stress such as when a gold bump chip is mounted on COF, the internal Flash memory characteristics may vary. Confirm the heat conditions (temperature and time) for mounting using the table below. If any of the "Data reprogramming required" conditions apply, be sure to reprogram the Flash memory using the corresponding fls program or the standalone Flash programmer. For details of the fls program or the standalone Flash programmer, refer to the respective manual.

Time / Temperature	Not affected to Flash memory	Affected to Flash memory	
		Data reprogramming required	Allowable time
Lower than 250°C	≤ 5 hours	5 hours to 450 hours	Max. 450 hours
250°C to 300°C	≤ 400 seconds	400 seconds to 10 hours	Max. 10 hours
300°C to 350°C	≤ 20 seconds	20 seconds to 0.5 hour	Max. 0.5 hour
350°C to 400°C	≤ 1 second	1 second to 100 seconds	Max. 100 seconds
400°C to 450°C	≤ 0.1 second	0.1 second to 10 seconds	Max. 10 seconds
Higher than 450°C			Max. 0 seconds

Miscellaneous

This product series is manufactured using microscopic processes.
Although it is designed to ensure basic IC reliability meeting EIAJ and MIL standards, minor variations over time may result in electrical damage arising from disturbances in the form of voltages exceeding the absolute maximum rating when mounting the product in addition to physical damage. The following factors can give rise to these variations:

(1) Electromagnetically-induced noise from industrial power supplies used in mounting reflow, reworking after mounting, and individual characteristic evaluation (testing) processes
(2) Electromagnetically-induced noise from a solder iron when soldering

In particular, during soldering, take care to ensure that the soldering iron GND (tip potential) has the same potential as the IC GND.

Appendix D Measures Against Noise

To improve noise immunity, take measures against noise as follows:

Noise Measures for VDD and VSS Power Supply Pins

The IC will malfunction at the instant when noise falling below the rated voltage is input. Take measures on the circuit board, including close patterns for circuit board power supply circuits, noise-filtering decoupling capacitors, and surge/noise prevention components on the power supply line.
For the recommended patterns on the circuit board, see "Mounting Precautions" in Appendix.

Noise Measures for #RESET Pin

The pull-up resistor for the #RESET pin included in this product has a relatively high impedance of 100 kΩ to 500 kΩ and is not noise-resistant. Extraneous noise may pull the #RESET pin down to a low level and this will cause the IC to reset. Therefore, the circuit board must be designed properly taking noise measures into consideration.
For the recommended patterns on the circuit board, see "Mounting Precautions" in Appendix.

Noise Measures for Oscillator Pins

The oscillator input pins must pass a signal of small amplitude, so they are hypersensitive to noise. Therefore, the circuit board must be designed properly taking noise measures into consideration.
For the recommended patterns on the circuit board, see "Mounting Precautions" in Appendix.

Noise Measures for Debug Pins

This product provides the input/output pins (DCLK, DST2, and DSIO) to connect ICDmini (S5U1C17001H) for debugging. If noise is input to these pins, the S1C17 Core may enter debug mode. To prevent unexpected transitions to debug mode caused by extraneous noise, switch the DCLK, DST2, and DSIO pins to general-purpose I/O port pins within the initialization routine when the debug functions are not used.
For details of the pin functions and the function switch control, see the "I/O Ports (P)" chapter.

Note: Do not perform the function switching shown above when the application is under development, as the debug functions must be used. The debugging cannot be performed after the pin function is switched. The above processing must be added after the application development has completed and debugging is no longer necessary.

The DSIO pin should be pulled up with a 10 kΩ resistor when using the debug pin functions. The pull-up resistor for the DSIO pin included in this product has a relatively high impedance of 100 kΩ to 500 kΩ and is not noise-resistant.

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 (P)" chapter.

Noise Measures for UART Pins

This product includes a UART module for asynchronous communications. The UART starts receive operation when it detects a low level input from the SIN*x* pin. Therefore, a receive operation may be started if the SIN*x* 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 UART from malfunction caused by extraneous noise, take the following measures:

- Stop the UART operations (RXEN/UART_CTL*x* register = 0) 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 (P)" chapter. For the UART control and details of receive errors, see the "UART" chapter.

Appendix E Initialization Routine

The following lists typical vector tables and initialization routines:

boot.s

```
.org    0x8000
.section .rodata                                                          ...(1)

; ======================================================================
;       Vector table
; ======================================================================
                                ; interrupt  vector  interrupt
                                ; number     offset  source

.long BOOT                      ; 0x00       0x00    reset                ...(2)
.long unalign_handler           ; 0x01       0x04    unalign
.long nmi_handler               ; 0x02       0x08    NMI
.long int03_handler             ; 0x03       0x0c    -
.long p0_handler                ; 0x04       0x10    P0 port
.long p2_handler                ; 0x05       0x14    P2 port
.long swt_handler               ; 0x06       0x18    SWT
.long ct_handler                ; 0x07       0x1c    CT
.long rtc_handler               ; 0x08       0x20    RTC
.long int09_handler             ; 0x09       0x24    -
.long epd_handler               ; 0x0a       0x28    EPD
.long t16a2_0_handler           ; 0x0b       0x2c    T16A2 ch0
.long int0c_handler             ; 0x0c       0x30    -
.long int0d_handler             ; 0x0d       0x34    -
.long t8_0_handler              ; 0x0e       0x38    T8 ch0
.long t8_1_handler              ; 0x0f       0x3c    T8 ch1
.long uart_0_handler            ; 0x10       0x40    UART ch0
.long i2cs_handler              ; 0x11       0x44    I2C slave
.long spi_0_handler             ; 0x12       0x48    SPI ch0
.long i2cm_handler              ; 0x13       0x4c    I2C master
.long int14_handler             ; 0x14       0x50    -
.long t16a2_1_handler           ; 0x15       0x54    T16A2 ch1
.long tem_handler               ; 0x16       0x58    TEM
.long rfc_handler               ; 0x17       0x5c    RFC
.long int18_handler             ; 0x18       0x60    -
.long int19_handler             ; 0x19       0x64    -
.long int1a_handler             ; 0x1a       0x68    -
.long int1b_handler             ; 0x1b       0x6c    -
.long int1c_handler             ; 0x1c       0x70    -
.long int1d_handler             ; 0x1d       0x74    -
.long int1e_handler             ; 0x1e       0x78    -
.long int1f_handler             ; 0x1f       0x7c    -

; ======================================================================
;       Program code
; ======================================================================

.text                                                                     ...(3)
.align 1

BOOT:
        ; ===== Initialize ============================================

        ; ----- Stack pointer --------------------
        Xld.a   %sp, 0x07c0                                               ...(4)

        ; ----- Memory controller ----------------
        Xld.a   %r1, 0x54b0     ; FLASHC register address

        ; Flash read wait cycle
        Xld.a   %r0, 0x00       ; No wait
        ld.b    [%r1], %r0      ; [0x54b0] <= 0x00                        ...(5)

        ; ===== Main routine ==========================================
         ...
```

```
; ========================================================================
;       Interrupt handler
; ========================================================================
; ----- Address unalign --------------------------
unalign_handler:
        ...

; ----- NMI ---------------------------------------
nmi_handler:
        ...
```

(1) A "`.rodata`" section is declared to locate the vector table in the "`.vector`" section.

(2) Interrupt handler routine addresses are defined as vectors.
"`int`*`XX`*`_handler`" can be used for software interrupts.

(3) The program code is written in the "`.text`" section.

(4) Sets the stack pointer.

(5) Sets the number of Flash memory wait cycles.
Can be set to no wait in the S1C17F57.
(See the "Memory Map, Bus Control" chapter.)

Revision History

Code No.	Page	Contents
412445700	All	New establishment
412445701	1-1 to 2	1.1 Features Added the following annotations to Table 1.1.1. I²C master (I2CM) *1 I²C slave (I2CS) *1 *1 The input filters in I2CM/I2CS (SDA and SCL inputs) do not comply with the standard for removing noise spikes less than 50 ns. SLEEP state *2 *2 The RAM retains data even in SLEEP mode. Modified Table 1.1.1. Shipping form: The QFP package was deleted.
	1-4	1.3.1 Pin Configuration Diagram The QFP package was deleted.
	1-5 to 6	1.3.2 Pin Descriptions Modified Table 1.3.2.1. The QFP pin numbers were deleted.
	18-1	18.1 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.
	19-1	19.1 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.
	27-1	27.2 Recommended Operating Conditions Added 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). *3 The component values should be determined after evaluating operations using an actual mounting board.
	27-14	22.6 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).
	29-1 to 3	29 Package/Chip The chapter title was changed to “Chip.” 29.1 QFP Package → Deleted the section. 29.2.1 Pad/Bump Configuration → Changed to Section 29.1. 29.2.2 Gold Bump Specifications → Changed to Section 29.2.
	29-1	29.1 Pad/Bump Configuration Added Alignment mark coordinates.
	AP-B-1	B.1 Clock Control Power Saving System SLEEP Added the following description: * The RAM retains data even in SLEEP mode.

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

29th Floor, JR Shinjuku Miraina Tower, 4-1-6 Shinjuku,
Shinjuku-ku, Tokyo 160-8801, Japan

Document Code: 412445701
First Issue November 2012
Revised December 2021 in JAPAN Ⓛ