EPSON
EXCEED YOUR VISION

CMOS 16-BIT SINGLE CHIP MICROCONTROLLER

S1C17002
Technical Manual

SEIKO EPSON CORPORATION

Rev.1.3

NOTICE

No part of this material may be reproduced or duplicated in any form or by any means without the written permission of Seiko Epson. Seiko Epson reserves the right to make changes to this material without notice. Seiko Epson does not assume any liability of any kind arising out of any inaccuracies contained in this material or due to its application or use in any product or circuit and, further, there is no representation that this material is applicable to products requiring high level reliability, such as medical products. Moreover, no license to any intellectual property rights is granted by implication or otherwise, and there is no representation or warranty that anything made in accordance with this material will be free from any patent or copyright infringement of a third party. This material or portions thereof may contain technology or the subject relating to strategic products under the control of the Foreign Exchange and Foreign Trade Law of Japan and may require an export license from the Ministry of Economy, Trade and Industry or other approval from another government agency.

Configuration of product number

Devices

S1 C 17xxx F 00E1 00

Packing specifications

00 :	Besides tape & reel	
0A :	TCP BL	2 directions
0B :	Tape & reel	BACK
0C :	TCP BR	2 directions
0D :	TCP BT	2 directions
0E :	TCP BD	2 directions
0F :	Tape & reel	FRONT
0G :	TCP BT	4 directions
0H :	TCP BD	4 directions
0J :	TCP SL	2 directions
0K :	TCP SR	2 directions
0L :	Tape & reel	LEFT
0M :	TCP ST	2 directions
0N :	TCP SD	2 directions
0P :	TCP ST	4 directions
0Q :	TCP SD	4 directions
0R :	Tape & reel	RIGHT
99 :	Specs not fixed	

Specification

Package
[D: die form; F: QFP, B: BGA]

Model number

Model name
[C: microcomputer, digital products]

Product classification
[S1: semiconductor]

Development tools

S5U1 C 17000 H2 1 00

Packing specifications
[00: standard packing]

Version
[1: Version 1]

Tool type

Hx : ICE
Dx : Evaluation board
Ex : ROM emulation board
Mx: Emulation memory for external ROM
Tx : A socket for mounting
Cx : Compiler package
Sx : Middleware package
Yx : Writer software

Corresponding model number
[17xxx: for S1C17xxx]

Tool classification
[C: microcomputer use]

Product classification
[S5U1: development tool for semiconductor products]

S1C17002 Technical Manual

– Contents –

I S1C17002 SPECIFICATIONS

II S1C17002 CLOCK SYSTEM

III S1C17002 INTERRUPT/INTERNAL BUS CONTROLLERS

IV S1C17002 TIMER MODULES

V S1C17002 INTERFACE MODULES

VI S1C17002 I/O PORTS

VII S1C17002 ANALOG MODULE

APPENDIX

S1C17002 Technical Manual

I S1C17002 SPECIFICATIONS

I.1 Overview

The S1C17002 is a cost effective, high performance and compact 16-bit RISC application specific controller (ASC). It is suitable for various products that require analog inputs and interfaces for connection, such as healthcare goods, sensor systems, alarms, home electric appliance (rice cookers, microwave ovens and remote controllers).

The S1C17002 consists of a S1C17 16-bit compact RISC CPU Core, a 128K-byte ROM, an 8K-byte RAM, a 10-bit ADC with four analog input channels, a 16-bit multi-function timer, an infrared remote controller, serial interfaces (UART with IrDA 1.0, SPI and I^{2}C), an RTC, 16-bit and 8-bit timers, a watchdog timer, and GPIO ports.

The S1C17002 provides a 16 bits × 16 bits + 32 bits MAC (multiply and accumulate) and 16 bits ÷ 16 bits division functions to implement a DSP function.

The S1C17002 has adopted the EPSON SoC (System on Chip) design technology using 0.18 µm mixed analog low power CMOS process.

Table I.1.1 Product Lineup

	ROM size	RAM size	Package
1	128K bytes	8K bytes	TQFP12-64pin
2			WCSP-48
3			Bare chip

The main functions and features of the S1C17002 are outlined below.

Technology

- 0.18 µm AL-4-layers mixed analog low power CMOS process technology

CPU

- Seiko Epson original 16-bit RISC processor S1C17 Core
- Internal 3-stage pipeline
- Instruction set
  - 16-bit fixed length
  - 111 basic instructions (184 including variations)
  - Compact and fast instruction set optimized for development in C language
- Registers
  - Eight 24-bit general-purpose registers
  - Three special registers (24-bit × 2, 8-bit × 1)
- Memory space
  - Up to 16M bytes accessible (24-bit address)

Internal Memories

- Mask ROM
  - 128K bytes
- RAM
  - 8K bytes

Access Cycles

- Instruction read access cycle
  - Internal RAM   Instruction read: 2 cycles (32-bit read)
  - Internal ROM   Instruction read: 2 cycles (32-bit read) when pre-fetched data is hit
    3 cycles (32-bit read) when pre-fetched data is missed

  * The numbers of cycles listed above are assumed when reading two instructions (16 bits × 2) in sequential access.

- Data read/write access cycle
  - Internal RAM Data write: 1 cycle
    Data read: 2 cycles
  - Internal ROM Data read: 1 cycle (16-bit read) when pre-read data is hit
    2 cycles (16-bit read) when pre-read data is missed
- Branch penalty cycle in one cycle mode for the internal ROM

Current address → branch address	Number of penalty cycles when a 3-cycle branch instruction is executed	Number of penalty cycles when a 4-cycle branch instruction is executed
4-byte boundary → 4-byte boundary	+2 cycles	+1 cycle
4-byte boundary → 2-byte boundary	+3 cycles	+2 cycles
2-byte boundary → 4-byte boundary	+3 cycles	+1 cycle
2-byte boundary → 2-byte boundary	+4 cycles	+2 cycles

- Maximum operating frequency in one cycle mode for the internal ROM: 20 MHz

Operating Clock

- Main clock
  - 20 MHz (max.)
  - On-chip oscillator (crystal or ceramic) or external clock input
- Sub clock
  - 32.768 kHz (typ.) for the RTC, usable as the main clock
  - On-chip oscillator (crystal) or external clock input

Interrupt Controller

- Four non-maskable interrupts
  - Reset (#RESET pin or watchdog timer)
  - Address misaligned
  - Debug
  - NMI (watchdog timer)
- 29 maskable interrupts
  - Port inputs (eight systems)
  - 16-bit multi-function timer (one system)
  - A/D converter (two systems)
  - 16-bit timer of clock generator (one system)
  - 8-bit timers of clock generator (three systems)
  - UART (one system)
  - SPI (one system)
  - I^{2}C master (one system)
  - I^{2}C slave (two systems)
  - RTC (one system)
  - 8-bit programmable timers (four systems)
  - 8-bit OSC1 timers (two systems)
  - Extended SPI (one system)
  - Remote controller (one system)
  - The interrupt level (priority) of each maskable interrupt system is configurable (levels 0 to 7).

Prescaler

- Generates the source clocks for the clock generator.

16-bit Multi-Function Timer

- One channel of 16-bit timer/counter with PWM output function is available.
- Can generate two compare-match interrupts.
- Supports the IGBT output control function using the A/D converter out-of-range signal.

Clock Generator

- One channel of 16-bit timer and three channels of 8-bit timers are available.
- Can be used as the clock source for the UART, SPI, and I^{2}C master.
- Each timer can generate an underflow interrupt.

8-bit Programmable Timers

- Four channels of 8-bit timers (presettable down counter) are available.
- Can be used as an interval timer to trigger the ADC.
- Each timer can generate an underflow interrupt.

8-bit OSC1 Timers

- Two channels of 8-bit timers (presettable down counter) that are driven with the OSC1 clock are available.
- Each timer can generate an underflow interrupt.

Watchdog Timer

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

RTC

- Contains time counters (second, minute, and hour) and calendar counters (day, day of the week, month, and year).
- Periodic interrupts are possible.

UART

- One channel of UART is available.
- Supports IrDA 1.0 interface.
- Two-byte receive data buffer and one-byte transmit buffer are built in to support full-duplex communication.
- Transfer rate: 150 to 460800 bps, character length: seven or eight bits, parity mode: even, odd, or no parity, stop bit: one or two bits
- Parity error, framing error, and overrun error detectable
- Each channel can generate receive buffer full, transmit buffer empty, and receive error interrupts.

SPI

- Supports both master and slave modes.
- One-byte receive data buffer and one-byte transmit buffer are built in.
- Data length: eight bits fixed (MSB first)
- Data transfer timing (clock phase and polarity variations) is selectable from among 4 types.
- Can generate receive buffer full and transmit buffer empty interrupts.

Extended SPI

- Supports both master and slave modes.
- One-byte receive data buffer and one-byte transmit buffer are built in.
- Data length: eight bits fixed (MSB first)
- Data transfer timing (clock phase and polarity variations) is selectable from among 4 types.
- Can generate receive buffer full and transmit buffer empty interrupts.
- Exclusive clock source is available.

I^{2}C Master

- Data format: 8 bits (MSB first)
- Addressing mode: 7-bit addressing (10-bit addressing is not supported.)
- Incorporates a noise rejector (can be enabled by a register).
- Can generate receive buffer full and transmit buffer empty interrupts.

I²C Slave

- Data format: 8 bits (MSB first)
- Addressing mode: 7-bit addressing (10-bit addressing is not supported.)
- Supports a clock stretch function
- Incorporates a noise rejector (can be enabled by a register).
- Can generate receive, transmit, and bus status interrupts.

Infrared Remote Controller

- Outputs a modulated carrier signal and inputs remote control pulses.
- Embedded carrier signal generator and data length counter.
- Can generates REMC interrupts.

General-Purpose I/O Ports

- Maximum 30 I/O ports and four input ports are available.
- Can generate input interrupts from the six ports selected with software.

* The GPIO ports are shared with other peripheral function pins (UART, PWM etc.). Therefore, the number of GPIO ports depends on the peripheral functions used.

A/D Converter

- 10-bit A/D converter with up to four analog input ports
- Can generates an end of conversion interrupt and an out of range interrupt.
- Outputs an out of range signal to the IGBT circuit in the 16-bit multi-function timer module.

Operating Voltage

- HVDD (for I/O): 1.65 to 3.60 V
- LVDD (for Core): 1.65 to 1.95 V
- AVDD (for ADC): 2.70 to 3.60 V (1.65 to 3.60 V*)

* The AVDD voltage range can be changed to 1.65 to 3.60 V only when the ADC is not used and the P0*x* pins are used as digital signal input pins, not analog input pins. However, the high and low level input voltages of the digital signals must be AVDD and GND, respectively.

Operating Temperatures

- -40 to 85°C

Current Consumption

• During SLEEP:	1.8 µA (typ.)	32 kHz/1.8 V, RTC = On
• During HALT:	1.3 mA (typ.)	20 MHz/1.8 V
• During execution:	3.8 mA (typ.)	20 MHz/1.8 V

* By controlling the clocks through the Clock-Gear (CMU), power consumption can be reduced.

Shipping Form

- TQFP12-64pin (7 mm × 7 mm × 1.2 mm, 0.4 mm pin pitch)
- WCSP-48 (3.124 mm × 3.124 mm × 0.78 mm, 0.4 mm ball pitch)
- Bare chip (3.124 mm × 3.124 mm × 0.40 mm)

I.2 Block Diagram

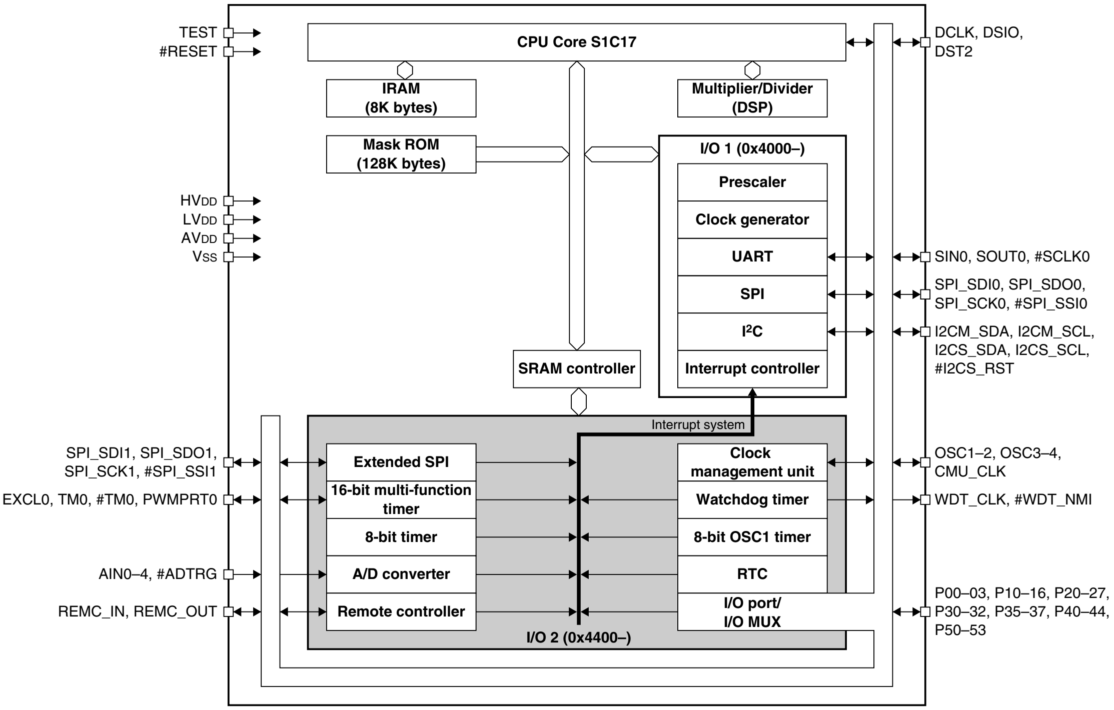


Figure I.2.1 S1C17002 Block Diagram

THIS PAGE IS BLANK.

I.3 Pin Description

I.3.1 Pin Arrangement

The S1C17002 comes in a TQFP12-64pin or a WCSP-48 package.

TQFP12-64pin package

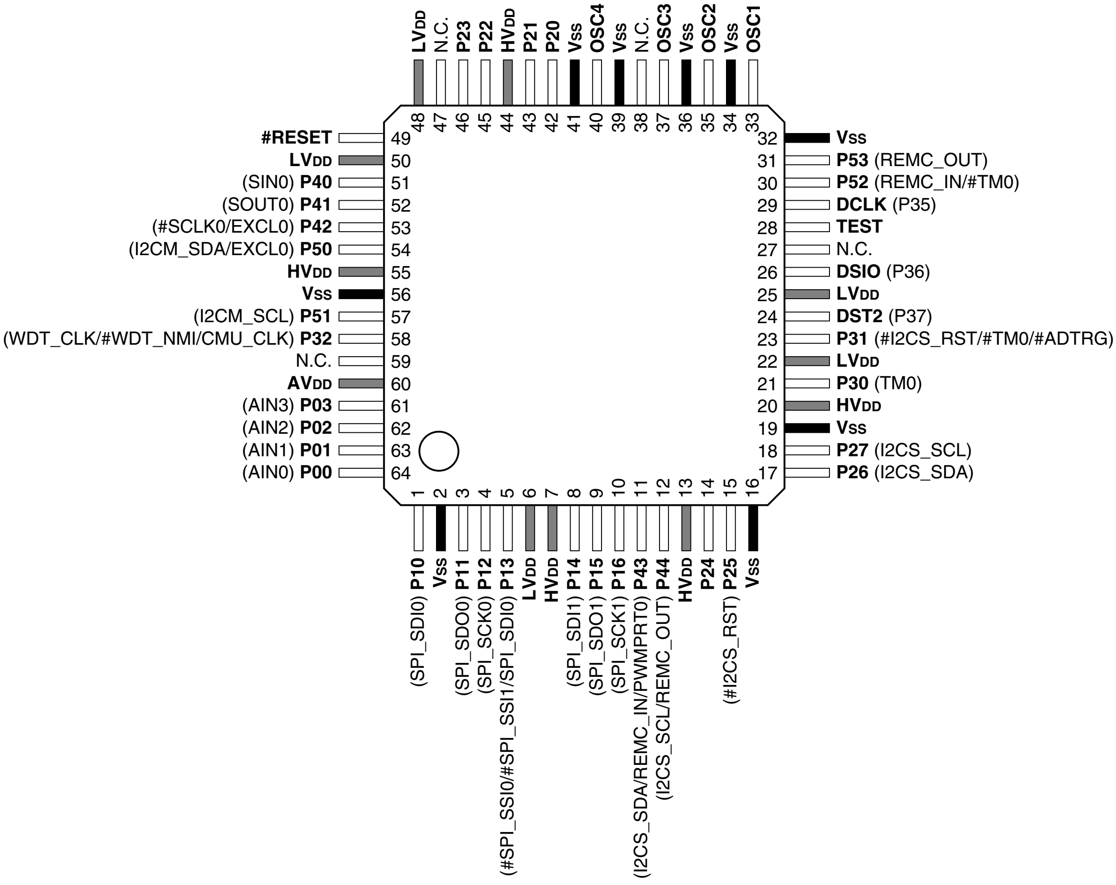
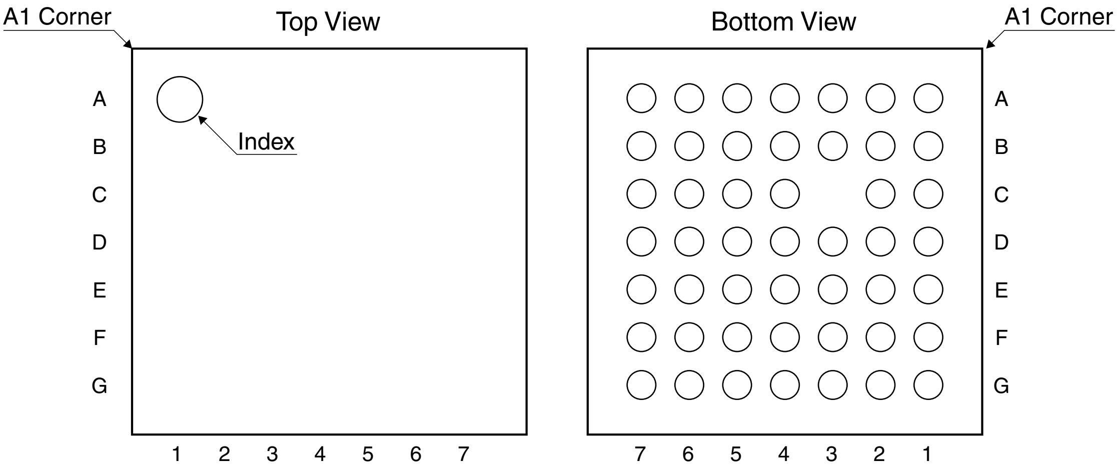


Figure I.3.1.1 Pin Arrangement (TQFP12-64pin)

WCSP-48 package



Top View

	1	2	3	4	5	6	7
A	**P10** SPI_SDI0	**P11** SPI_SDO0	**P13** #SPI_SSI0 #SPI_SSI1 SPI_SDI0	**P14** SPI_SDI1	**P43** I2CS_SDA REMC_IN PWMPRT0	**P24**	**P26** I2CS_SDA
B	**P01** AIN1	**P00** AIN0	**P12** SPI_SCK0	**P15** SPI_SDO1	**P44** I2CS_SCL REMC_OUT	**P25** #I2CS_RST	**P27** I2CS_SCL
C	**P03** AIN3	**P02** AIN2	☒	**HVDD**	**P16** SPI_SCK1	**VSS**	**P30** TM0
D	**P51** I2CM_SCL	**P32** WDT_CLK #WDT_NMI CMU_CLK	**AVDD**	**VSS**	**LVDD**	**P31** #I2CS_RST #TM0 #ADTRG	**DST2** P37
E	**P50** I2CM_SDA EXCL0	**P42** #SCLK0 EXCL0	**P41** SOUT0	**HVDD**	**DCLK** P35	**TEST**	**DSIO** P36
F	**P40** SIN0	**LVDD**	**P22**	**P20**	**VSS**	**P53** REMC_OUT	**P52** REMC_IN #TM0
G	**#RESET**	**P23**	**P21**	**OSC4**	**OSC3**	**OSC2**	**OSC1**

Figure I.3.1.2 Pin Arrangement (WCSP-48)

I.3.2 Pin Functions

Tables I.3.2.1 to I.3.2.4 list the function of each pin on the S1C17002.

Table I.3.2.1 Power Supply Pin List

Pin name	Pin No.		I/O	Type	PU/PD	Description
	TQFP	WCSP				
HV$_{DD}$	7, 13, 20, 44, 55	C4, E4	–	3.3 V	–	I/O power supply (+) (1.8 V/2.5 V/3.3 V)
LV$_{DD}$	6, 22, 25, 48, 50	D5, F2	–	1.8 V	–	Core power supply (+) (1.8 V)
V$_{SS}$	2, 16, 19, 32, 34, 36, 39, 41, 56	D4, C6, F5	–	GND	–	GND
AV$_{DD}$	60	D3	–	3.3 V	–	Analog power supply (3.0 V/3.3 V)

Table I.3.2.2 Clock Pin List

Pin name	Pin No.		I/O	Type	PU/PD	Description
	TQFP	WCSP				
OSC3	37	G5	I	Analog	–	High speed (OSC3) oscillation input (crystal/ceramic oscillator or external clock input)
OSC4	40	G4	O	Analog	–	High speed (OSC3) oscillation output
OSC1	33	G7	I	Analog	–	RTC (OSC1) oscillation input (crystal oscillator or external clock input)
OSC2	35	G6	O	Analog	–	RTC (OSC1) oscillation output

Table I.3.2.3 Input/Output Port and Peripheral Circuit Pin List

Pin name	Pin No.		I/O	Type	PU/PD	Description
	TQFP	WCSP				
P00 AIN0	64	B2	**I**	LVCMOS	–	P00: Input port (default) AIN0: A/D converter CH.0 input
P01 AIN1	63	B1	**I**	LVCMOS	–	P01: Input port (default) AIN1: A/D converter CH.1 input
P02 AIN2	62	C2	**I**	LVCMOS	–	P02: Input port (default) AIN2: A/D converter CH.2 input
P03 AIN3	61	C1	**I**	LVCMOS	–	P03: Input port (default) AIN3: A/D converter CH.3 input
P10 SPI_SDI0	1	A1	**I**/o	LVCMOS	–	P10: I/O port (default) SPI_SDI0: SPI CH.0 data input
P11 SPI_SDO0	3	A2	**I**/o	LVCMOS	–	P11: I/O port (default) SPI_SDO0: SPI CH.0 data output
P12 SPI_SCK0	4	B3	**I**/o	LVCMOS	–	P12: I/O port (default) SPI_SCK0: SPI CH.0 clock input/output
P13 #SPI_SSI0 #SPI_SSI1 SPI_SDI0	5	A3	**I**/o	LVCMOS	–	P13: I/O port (default) SPI_SSL0: SPI CH.0 slave select signal output SPI_SSL1: SPI CH.1 slave select signal output SPI_SDI0: SPI CH.0 data input
P14 SPI_SDI1	8	A4	**I**/o	LVCMOS	–	P14: I/O port (default) SPI_SDI1: SPI CH.1 data input
P15 SPI_SDO1	9	B4	**I**/o	LVCMOS	–	P15: I/O port (default) SPI_SDO1: SPI CH.1 data output
P16 SPI_SCK1	10	C5	**I**/o	LVCMOS	–	P16: I/O port (default) SPI_SCK1: SPI CH.1 clock input/output
P20	42	F4	**I**/o	LVCMOS	–	P20: I/O port (default)
P21	43	G3	**I**/o	LVCMOS	–	P21: I/O port (default)
P22	45	F3	**I**/o	LVCMOS	–	P22: I/O port (default)
P23	46	G2	**I**/o	LVCMOS	–	P23: I/O port (default)
P24	14	A6	**I**/o	LVCMOS	–	P24: I/O port (default)
P25 #I2CS_RST	15	B6	**I**/o	LVCMOS	–	P25: I/O port (default) #I2CS_RST: I^{2}C slave bus free request input
P26 I2CS_SDA	17	A7	**I**/o	LVCMOS	–	P26: I/O port (default) I2CS_SDA: I^{2}C slave data signal
P27 I2CS_SCL	18	B7	**I**/o	LVCMOS	–	P27: I/O port (default) I2CS_SCL: I^{2}C slave clock input
P30 TM0	21	C7	**I**/o	LVCMOS	–	P30: I/O port (default) TM0: 16-bit multi-function timer output
P31 #I2CS_RST #TM0 #ADTRG	23	D6	**I**/o	LVCMOS	–	P31: I/O port (default) #I2CS_RST: I^{2}C slave bus free request input #TM0: 16-bit multi-function timer inverted output #ADTRG: A/D converter trigger input

Pin name	Pin No.		I/O	Type	PU/PD	Description
	TQFP	WCSP				
P32 WDT_CLK #WDT_NMI CMU_CLK	58	D2	**I**/o	LVCMOS	–	P32: I/O port (default) WDT_CLK: Watchdog timer clock output #WDT_NMI: Watchdog timer NMI signal output CMU_CLK: CMU clock output
P40 SIN0	51	F1	**I**/o	LVCMOS	–	P40: I/O port (default) SIN0: UART with IrDA CH.0 data input
P41 SOUT0	52	E3	**I**/o	LVCMOS	–	P41: I/O port (default) SOUT0: UART with IrDA CH.0 data output
P42 #SCLK0 EXCL0	53	E2	**I**/o	LVCMOS	–	P42: I/O port (default) #SCLK0: UART with IrDA CH.0 clock input/output EXCL0: 16-bit multi-function timer event counter input
P43 I2CS_SDA REMC_IN PWMPRT0	11	A5	**I**/o	LVCMOS	–	P43: I/O port (default) I2CS_SDA: I^{2}C slave data signal REMC_IN: Remote controller receive signal input PWMPRT0: 16-bit multi-function timer port protection signal input
P44 I2CS_SCL REMC_OUT	12	B5	**I**/o	LVCMOS	–	P44: I/O port (default) I2CS_SCL: I^{2}C slave clock input REMC_OUT: Remote controller transmit signal output
P50 I2CM_SDA EXCL0	54	E1	**I**/o	LVCMOS	–	P50: I/O port (default) I2CM_SDA: I^{2}C master data signal EXCL0: 16-bit multi-function timer event counter input
P51 I2CM_SCL	57	D1	**I**/o	LVCMOS	–	P51: I/O port (default) I2CM_SCL: I^{2}C master clock output
P52 REMC_IN #TM0	30	F7	**I**/o	LVCMOS	–	P52: I/O port (default) REMC_IN: Remote controller receive signal input #TM0: 16-bit multi-function timer inverted output
P53 REMC_OUT	31	F6	**I**/o	LVCMOS	–	P53: I/O port (default) REMC_OUT: Remote controller transmit signal output

Table I.3.2.4 Other Pin List

Pin name	Pin No.		I/O	Type	PU/PD	Description
	TQFP	WCSP				
#RESET	49	G1	**I**	LVCMOS	100k PU	Reset input (with noise filter)
DCLK P35	29	E5	i/**O**	LVCMOS	–	DCLK: DCLK (Debug SIO Clock) signal output (default) P35: I/O port
DSIO P36	26	E7	**I**/o	LVCMOS	100k PU	DSIO: DSIO (Debug SIO) pin (with noise filter) (default) P36: I/O port
DST2 P37	24	D7	i/**O**	LVCMOS	–	DST2: DST2 (Debug Status) signal output (default) P37: I/O port
TEST	28	E6	**I**	–	50k PD	Test input. Connect to V_{SS} in user mode.

Notes:
- The # prefixed to pin names indicates that input/output signals of the pin are active low.
- The pin names listed in boldface denote the default pin (signal) name.
- The I/O listed in boldface and uppercase denote the default input/output direction.
- “PU” means “Pull-up” and “PD” means “Pull-down.”

I.3.3 Switching Over the Multiplexed Pin Functions

I.3.3.1 Pin Function Select Bits

Each pin is assigned one to four functions, as listed in Table I.3.3.1.1.

When the chip is powered on or reset, each pin defaults to function 0. If any pin must be used for other than this default function, select the desired function by writing data to the corresponding pin function select bits.

Table I.3.3.1.1 List of Pin Function Select Bits

Pin function 0	Pin function 1	Pin function 2	Pin function 3	Function select bit
OSC3				
OSC4				
OSC1				
OSC2				
TEST				
#RESET				
DCLK	P35			CFP35[1:0] (D[3:2]/0x4427)
DSIO	P36			CFP36[1:0] (D[5:4]/0x4427)
DST2	P37			CFP37[1:0] (D[7:6]/0x4427)
P00	AIN0			CFP00[1:0] (D[1:0]/0x4420)
P01	AIN1			CFP01[1:0] (D[3:2]/0x4420)
P02	AIN2			CFP02[1:0] (D[5:4]/0x4420)
P03	AIN3			CFP03[1:0] (D[7:6]/0x4420)
P10	SPI_SDI0			CFP10[1:0] (D[1:0]/0x4422)
P11	SPI_SDO0			CFP11[1:0] (D[3:2]/0x4422)
P12	SPI_SCK0			CFP12[1:0] (D[5:4]/0x4422)
P13	#SPI_SSI0	#SPI_SSI1	SPI_SDI0	CFP13[1:0] (D[7:6]/0x4422)
P14	SPI_SDI1			CFP14[1:0] (D[1:0]/0x4423)
P15	SPI_SDO1			CFP15[1:0] (D[3:2]/0x4423)
P16	SPI_SCK1			CFP16[1:0] (D[5:4]/0x4423)
P20				
P21				
P22				
P23				
P24				
P25		#I2CS_RST		CFP25[1:0] (D[3:2]/0x4425)
P26		I2CS_SDA		CFP26[1:0] (D[5:4]/0x4425)
P27		I2CS_SCL		CFP27[1:0] (D[7:6]/0x4425)
P30	TM0			CFP30[1:0] (D[1:0]/0x4426)
P31	#I2CS_RST	#TM0	#ADTRG	CFP31[1:0] (D[3:2]/0x4426)
P32	WDT_CLK	#WDT_NMI	CMU_CLK	CFP32[1:0] (D[5:4]/0x4426)
P40	SIN0			CFP40[1:0] (D[1:0]/0x4428)
P41	SOUT0			CFP41[1:0] (D[3:2]/0x4428)
P42	#SCLK0	EXCL0		CFP42[1:0] (D[5:4]/0x4428)
P43	I2CS_SDA	REMC_IN	PWMPRT0	CFP43[1:0] (D[7:6]/0x4428)
P44	I2CS_SCL	REMC_OUT		CFP44[1:0] (D[1:0]/0x4429)
P50	I2CM_SDA	EXCL0		CFP50[1:0] (D[1:0]/0x442a)
P51	I2CM_SCL			CFP51[1:0] (D[3:2]/0x442a)
P52	REMC_IN	#TM0		CFP52[1:0] (D[5:4]/0x442a)
P53	REMC_OUT			CFP53[1:0] (D[7:6]/0x442a)

∗ The set values 0 to 3 of the pin function select bits correspond to functions 0 to 3, respectively.

I.3.3.2 List of Port Function Select Registers

Table I.3.3.2.1 List of Port Function Select Registers

Address	Register name		Function
0x4420	P0_03_CFP	P00–P03 Port Function Select Register	Selects the P00–P03 port functions.
0x4422	P1_03_CFP	P10–P13 Port Function Select Register	Selects the P10–P13 port functions.
0x4423	P1_46_CFP	P14–P16 Port Function Select Register	Selects the P14–P16 port functions.
0x4425	P2_57_CFP	P25–P27 Port Function Select Register	Selects the P25–P27 port functions.
0x4426	P3_02_CFP	P30–P32 Port Function Select Register	Selects the P30–P32 port functions.
0x4427	P3_57_CFP	P35–P37 Port Function Select Register	Selects the P35–P37 port functions.
0x4428	P4_03_CFP	P40–P43 Port Function Select Register	Selects the P40–P43 port functions.
0x4429	P4_4_CFP	P44 Port Function Select Register	Selects the P44 port function.
0x442a	P5_03_CFP	P50–P53 Port Function Select Register	Selects the P50–P53 port functions.

The following describes each port function select register. These are all 8-bit registers.

Note: When setting the registers, be sure to write a 0, and not a 1, for all “reserved bits.”

0x4420: P00–P03 Port Function Select Register (P0_03_CFP)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
P00–P03 Port Function Select Register (P0_03_CFP)	**0x4420** (8 bits)	D7–6	**CFP03[1:0]**	P03 port function select	CFP03[1:0]	Function	0x0	R/W	
					0x3–0x2	reserved			
					0x1	AIN3			
					0x0	P03			
		D5–4	**CFP02[1:0]**	P02 port function select	CFP02[1:0]	Function	0x0	R/W	
					0x3–0x2	reserved			
					0x1	AIN2			
					0x0	P02			
		D3–2	**CFP01[1:0]**	P01 port function select	CFP01[1:0]	Function	0x0	R/W	
					0x3–0x2	reserved			
					0x1	AIN1			
					0x0	P01			
		D1–0	**CFP00[1:0]**	P00 port function select	CFP00[1:0]	Function	0x0	R/W	
					0x3–0x2	reserved			
					0x1	AIN0			
					0x0	P00			

This register selects the functions of the P00 to P03 ports.

D[7:6] CFP03[1:0]: P03 Port Function Select Bits

11 (R/W): Reserved
10 (R/W): Reserved
01 (R/W): AIN3
00 (R/W): P03 input port (default)

D[5:4] CFP02[1:0]: P02 Port Function Select Bits

11 (R/W): Reserved
10 (R/W): Reserved
01 (R/W): AIN2
00 (R/W): P02 input port (default)

D[3:2] CFP01[1:0]: P01 Port Function Select Bits

11 (R/W): Reserved
10 (R/W): Reserved
01 (R/W): AIN1
00 (R/W): P01 input port (default)

D[1:0] CFP00[1:0]: P00 Port Function Select Bits

11 (R/W): Reserved
10 (R/W): Reserved
01 (R/W): AIN0
00 (R/W): P00 input port (default)

0x4422: P10–P13 Port Function Select Register (P1_03_CFP)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
P10–P13 Port Function Select Register (P1_03_CFP)	**0x4422** (8 bits)	D7–6	**CFP13[1:0]**	P13 port function select	CFP13[1:0]	Function	0x0	R/W	
					0x3	SPI_SDI0			
					0x2	#SPI_SSI1			
					0x1	#SPI_SSI0			
					0x0	P13			
		D5–4	**CFP12[1:0]**	P12 port function select	CFP12[1:0]	Function	0x0	R/W	
					0x3–0x2	reserved			
					0x1	SPI_SCK0			
					0x0	P12			
		D3–2	**CFP11[1:0]**	P11 port function select	CFP11[1:0]	Function	0x0	R/W	
					0x3–0x2	reserved			
					0x1	SPI_SDO0			
					0x0	P11			
		D1–0	**CFP10[1:0]**	P10 port function select	CFP10[1:0]	Function	0x0	R/W	
					0x3–0x2	reserved			
					0x1	SPI_SDI0			
					0x0	P10			

This register selects the functions of the P10 to P13 ports.

D[7:6] CFP13[1:0]: P13 Port Function Select Bits

11 (R/W): SPI_SDI0
10 (R/W): #SPI_SSI1
01 (R/W): #SPI_SSI0
00 (R/W): P13 I/O port (default)

D[5:4] CFP12[1:0]: P12 Port Function Select Bits

11 (R/W): Reserved
10 (R/W): Reserved
01 (R/W): SPI_SCK0
00 (R/W): P12 I/O port (default)

D[3:2] CFP11[1:0]: P11 Port Function Select Bits

11 (R/W): Reserved
10 (R/W): Reserved
01 (R/W): SPI_SDO0
00 (R/W): P11 I/O port (default)

D[1:0] CFP10[1:0]: P10 Port Function Select Bits

11 (R/W): Reserved
10 (R/W): Reserved
01 (R/W): SPI_SDI0
00 (R/W): P10 I/O port (default)

0x4423: P14–P16 Port Function Select Register (P1_46_CFP)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
P14–P16 Port Function Select Register (P1_46_CFP)	**0x4423** (8 bits)	D7–6	–	reserved	–		–	–	0 when being read.
		D5–4	**CFP16[1:0]**	P16 port function select	CFP16[1:0]	Function	0x0	R/W	
					0x3–0x2 0x1 0x0	reserved SPI_SCK1 P16			
		D3–2	**CFP15[1:0]**	P15 port function select	CFP15[1:0]	Function	0x0	R/W	
					0x3–0x2 0x1 0x0	reserved SPI_SDO1 P15			
		D1–0	**CFP14[1:0]**	P14 port function select	CFP14[1:0]	Function	0x0	R/W	
					0x3–0x2 0x1 0x0	reserved SPI_SDI1 P14			

This register selects the functions of the P14 to P16 ports.

D[7:6] Reserved

D[5:4] CFP16[1:0]: P16 Port Function Select Bits

11 (R/W): Reserved
10 (R/W): Reserved
01 (R/W): SPI_SCK1
00 (R/W): P16 I/O port (default)

D[3:2] CFP15[1:0]: P15 Port Function Select Bits

11 (R/W): Reserved
10 (R/W): Reserved
01 (R/W): SPI_SDO1
00 (R/W): P15 I/O port (default)

D[1:0] CFP14[1:0]: P14 Port Function Select Bits

11 (R/W): Reserved
10 (R/W): Reserved
01 (R/W): SPI_SDI1
00 (R/W): P14 I/O port (default)

0x4425: P25–P27 Port Function Select Register (P2_57_CFP)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
P25–P27 Port Function Select Register (P2_57_CFP)	**0x4425** (8 bits)	D7–6	**CFP27[1:0]**	P27 port function select	CFP27[1:0]	Function	0x0	R/W	
					0x3 0x2 0x1 0x0	reserved I2CS_SCL reserved P27			
		D5–4	**CFP26[1:0]**	P26 port function select	CFP26[1:0]	Function	0x0	R/W	
					0x3 0x2 0x1 0x0	reserved I2CS_SDA reserved P26			
		D3–2	**CFP25[1:0]**	P25 port function select	CFP25[1:0]	Function	0x0	R/W	
					0x3 0x2 0x1 0x0	reserved #I2CS_RST reserved P25			
		D1–0	–	reserved	–		–	–	0 when being read.

This register selects the functions of the P25 to P27 ports.

D[7:6] CFP27[1:0]: P27 Port Function Select Bits

11 (R/W): Reserved
10 (R/W): I2CS_SCL
01 (R/W): Reserved
00 (R/W): P27 I/O port (default)

D[5:4] CFP26[1:0]: P26 Port Function Select Bits

11 (R/W): Reserved
10 (R/W): I2CS_SDA
01 (R/W): Reserved
00 (R/W): P26 I/O port (default)

D[3:2] CFP25[1:0]: P25 Port Function Select Bits

11 (R/W): Reserved
10 (R/W): #I2CS_RST
01 (R/W): Reserved
00 (R/W): P25 I/O port (default)

D[1:0] Reserved

0x4426: P30–P32 Port Function Select Register (P3_02_CFP)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
P30–P32 Port Function Select Register (P3_02_CFP)	**0x4426** (8 bits)	D7–6	–	reserved	–		–	–	0 when being read.
		D5–4	**CFP32[1:0]**	P32 port function select	CFP32[1:0]	Function	0x0	R/W	
					0x3	CMU_CLK			
					0x2	#WDT_NMI			
					0x1	WDT_CLK			
					0x0	P32			
		D3–2	**CFP31[1:0]**	P31 port function select	CFP31[1:0]	Function	0x0	R/W	
					0x3	#ADTRG			
					0x2	#TM0			
					0x1	#I2CS_RST			
					0x0	P31			
		D1–0	**CFP30[1:0]**	P30 port function select	CFP30[1:0]	Function	0x0	R/W	
					0x3–0x2	reserved			
					0x1	TM0			
					0x0	P30			

This register selects the functions of the P30 to P32 ports.

D[7:6] Reserved

D[5:4] CFP32[1:0]: P32 Port Function Select Bits

11 (R/W): CMU_CLK
10 (R/W): #WDT_NMI
01 (R/W): WDT_CLK
00 (R/W): P32 I/O port (default)

D[3:2] CFP31[1:0]: P31 Port Function Select Bits

11 (R/W): #ADTRG
10 (R/W): #TM0
01 (R/W): #I2CS_RST
00 (R/W): P31 I/O port (default)

D[1:0] CFP30[1:0]: P30 Port Function Select Bits

11 (R/W): Reserved
10 (R/W): Reserved
01 (R/W): TM0
00 (R/W): P30 I/O port (default)

0x4427: P35–P37 Port Function Select Register (P3_57_CFP)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
P35–P37 Port Function Select Register (P3_57_CFP)	**0x4427** (8 bits)	D7–6	**CFP37[1:0]**	P37 port function select	CFP37[1:0]	Function	0x0	R/W	
					0x3–0x2 0x1 0x0	reserved P37 DST2			
		D5–4	**CFP36[1:0]**	P36 port function select	CFP36[1:0]	Function	0x0	R/W	
					0x3–0x2 0x1 0x0	reserved P36 DSIO			
		D3–2	**CFP35[1:0]**	P35 port function select	CFP35[1:0]	Function	0x0	R/W	
					0x3–0x2 0x1 0x0	reserved P35 DCLK			
		D1–0	–	reserved	–		–	–	0 when being read.

This register selects the functions of the P35 to P37 ports.

D[7:6] CFP37[1:0]: P37 Port Function Select Bits

11 (R/W): Reserved
10 (R/W): Reserved
01 (R/W): P37 I/O port
00 (R/W): DST2 (default)

D[5:4] CFP36[1:0]: P36 Port Function Select Bits

11 (R/W): Reserved
10 (R/W): Reserved
01 (R/W): P36 I/O port
00 (R/W): DSIO (default)

D[3:2] CFP35[1:0]: P35 Port Function Select Bits

11 (R/W): Reserved
10 (R/W): Reserved
01 (R/W): P35 I/O port
00 (R/W): DCLK (default)

D[1:0] Reserved

0x4428: P40–P43 Port Function Select Register (P4_03_CFP)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
P40–P43 Port Function Select Register (P4_03_CFP)	**0x4428** (8 bits)	D7–6	**CFP43[1:0]**	P43 port function select	CFP43[1:0]	Function	0x0	R/W	
					0x3	PWMPRT0			
					0x2	REMC_IN			
					0x1	I2CS_SDA			
					0x0	P43			
		D5–4	**CFP42[1:0]**	P42 port function select	CFP42[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	EXCL0			
					0x1	#SCLK0			
					0x0	P42			
		D3–2	**CFP41[1:0]**	P41 port function select	CFP41[1:0]	Function	0x0	R/W	
					0x3–0x2	reserved			
					0x1	SOUT0			
					0x0	P41			
		D1–0	**CFP40[1:0]**	P40 port function select	CFP40[1:0]	Function	0x0	R/W	
					0x3–0x2	reserved			
					0x1	SIN0			
					0x0	P40			

This register selects the functions of the P40 to P43 ports.

D[7:6] CFP43[1:0]: P43 Port Function Select Bits

11 (R/W): PWMPRT0
10 (R/W): REMC_IN
01 (R/W): I2CS_SDA
00 (R/W): P43 I/O port (default)

D[5:4] CFP42[1:0]: P42 Port Function Select Bits

11 (R/W): Reserved
10 (R/W): EXCL0
01 (R/W): #SCLK0
00 (R/W): P42 I/O port (default)

D[3:2] CFP41[1:0]: P41 Port Function Select Bits

11 (R/W): Reserved
10 (R/W): Reserved
01 (R/W): SOUT0
00 (R/W): P41 I/O port (default)

D[1:0] CFP40[1:0]: P40 Port Function Select Bits

11 (R/W): Reserved
10 (R/W): Reserved
01 (R/W): SIN0
00 (R/W): P40 I/O port (default)

0x4429: P44 Port Function Select Register (P4_4_CFP)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
P44 Port Function Select Register (P4_4_CFP)	**0x4429** (8 bits)	D7–2	**–**	reserved	–		–	–	0 when being read.
		D1–0	**CFP44[1:0]**	P44 port function select	CFP44[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	REMC_OUT			
					0x1	I2CS_SCL			
					0x0	P44			

This register selects the function of the P44 port.

D[7:2] Reserved

D[1:0] CFP44[1:0]: P44 Port Function Select Bits

11 (R/W): Reserved
10 (R/W): REMC_OUT
01 (R/W): I2CS_SCL
00 (R/W): P44 I/O port (default)

0x442a: P50–P53 Port Function Select Register (P5_03_CFP)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
P50–P53 Port Function Select Register (P5_03_CFP)	**0x442a** (8 bits)	D7–6	**CFP53[1:0]**	P53 port function select	CFP53[1:0]	Function	0x0	R/W	
					0x3–0x2 0x1 0x0	reserved REMC_OUT P53			
		D5–4	**CFP52[1:0]**	P52 port function select	CFP52[1:0]	Function	0x0	R/W	
					0x3 0x2 0x1 0x0	reserved #TM0 REMC_IN P52			
		D3–2	**CFP51[1:0]**	P51 port function select	CFP51[1:0]	Function	0x0	R/W	
					0x3–0x2 0x1 0x0	reserved I2CM_SCL P51			
		D1–0	**CFP50[1:0]**	P50 port function select	CFP50[1:0]	Function	0x0	R/W	
					0x3 0x2 0x1 0x0	reserved EXCL0 I2CM_SDA P50			

This register selects the functions of the P50 to P53 ports.

D[7:6] CFP53[1:0]: P53 Port Function Select Bits

11 (R/W): Reserved
10 (R/W): Reserved
01 (R/W): REMC_OUT
00 (R/W): P53 I/O port (default)

D[5:4] CFP52[1:0]: P52 Port Function Select Bits

11 (R/W): Reserved
10 (R/W): #TM0
01 (R/W): REMC_IN
00 (R/W): P52 I/O port (default)

D[3:2] CFP51[1:0]: P51 Port Function Select Bits

11 (R/W): Reserved
10 (R/W): Reserved
01 (R/W): I2CM_SCL
00 (R/W): P51 I/O port (default)

D[1:0] CFP50[1:0]: P50 Port Function Select Bits

11 (R/W): Reserved
10 (R/W): EXCL0
01 (R/W): I2CM_SDA
00 (R/W): P50 I/O port (default)

I.3.4 Input/Output Cells and Input/Output Characteristics

Table I.3.4.1 Pin Characteristics

Pin name	Direction	Cell name	Input level	I_{OH}/I_{OL} *2	Pull-up/down
OSC3 *1	I	LLINY	–	–	–
OSC4	O	LLOTY	–	–	–
OSC1 *1	I	LLINY	–	–	–
OSC2	O	LLOTY	–	–	–
TEST	I	LITST1Y	–	–	50k pull-down
#RESET	I	HIBCP1TY	LVCMOS	–	100k pull-up
DCLK (P35)	I/O	HBBC1BTY	LVCMOS	1 mA	–
DSIO (P36)	I/O	HBBC1BP1TY	LVCMOS	1 mA	100k pull-up
DST2 (P37)	I/O	HBBC1BTY	LVCMOS	1 mA	–
P00 (AIN0)	I	HIBASP2TY	LVCMOS	–	–
P01 (AIN1)	I	HIBASP2TY	LVCMOS	–	–
P02 (AIN2)	I	HIBASP2TY	LVCMOS	–	–
P03 (AIN3)	I	HIBASP2TY	LVCMOS	–	–
P10 (SPI_SDI0)	I/O	HBBC1BTY	LVCMOS	1 mA	–
P11 (SPI_SDO0)	I/O	HBBC1BTY	LVCMOS	1 mA	–
P12 (SPI_SCK0)	I/O	HBBC1BTY	LVCMOS	1 mA	–
P13 (#SPI_SSI0/#SPI_SSI1/SPI_SDI0)	I/O	HBBC1BTY	LVCMOS	1 mA	–
P14 (SPI_SDI1)	I/O	HBBC1BTY	LVCMOS	1 mA	–
P15 (SPI_SDO1)	I/O	HBBC1BTY	LVCMOS	1 mA	–
P16 (SPI_SCK1)	I/O	HBBC1BTY	LVCMOS	1 mA	–
P20	I/O	HBBC1BTY	LVCMOS	1 mA	–
P21	I/O	HBBC1BTY	LVCMOS	1 mA	–
P22	I/O	HBBC1BTY	LVCMOS	1 mA	–
P23	I/O	HBBC1BTY	LVCMOS	1 mA	–
P24	I/O	HBBC1BTY	LVCMOS	1 mA	–
P25 (#I2CS_RST)	I/O	HBBC1BTY	LVCMOS	1 mA	–
P26 (I2CS_SDA)	I/O	HBBC1BTY	LVCMOS	1 mA	–
P27 (I2CS_SCL)	I/O	HBBC1BTY	LVCMOS	1 mA	–
P30 (TM0)	I/O	HBBC1BTY	LVCMOS	1 mA	–
P31 (#I2CS_RST/#TM0/#ADTRG)	I/O	HBBC1BTY	LVCMOS	1 mA	–
P32 (WDT_CLK/#WDT_NMI/CMU_CLK)	I/O	HBBC1BTY	LVCMOS	1 mA	–
P40 (SIN0)	I/O	HBBC1BTY	LVCMOS	1 mA	–
P41 (SOUT0)	I/O	HBBC1BTY	LVCMOS	1 mA	–
P42 (#SCLK0/EXCL0)	I/O	HBBC1BTY	LVCMOS	1 mA	–
P43 (I2CS_SDA/REMC_IN/PWMPRT0)	I/O	HBBC1BTY	LVCMOS	1 mA	–
P44 (I2CS_SCL/REMC_OUT)	I/O	HBBC1BTY	LVCMOS	1 mA	–
P50 (I2CM_SDA/EXCL0)	I/O	HBBC1BTY	LVCMOS	1 mA	–
P51 (I2CM_SCL)	I/O	HBBC1BTY	LVCMOS	1 mA	–
P52 (REMC_IN/#TM0)	I/O	HBBC1BTY	LVCMOS	1 mA	–
P53 (REMC_OUT)	I/O	HBBC1BTY	LVCMOS	1 mA	–

*1: Input voltage (max.) = LV_{DD} (1.65 to 1.95 V)
*2: When HV_{DD} = 3.3 V

I.3.5 Package

I.3.5.1 TQFP12-64pin Package

(Unit: mm)

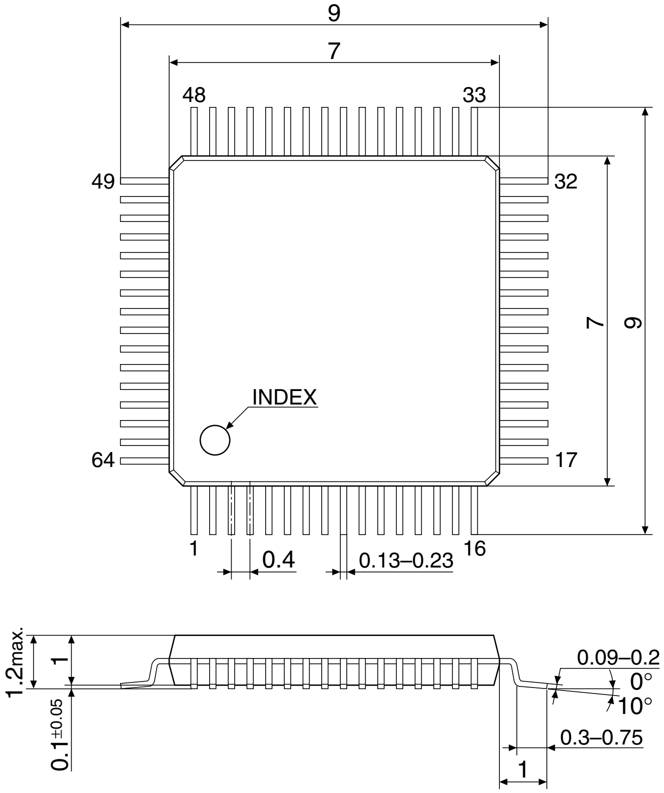
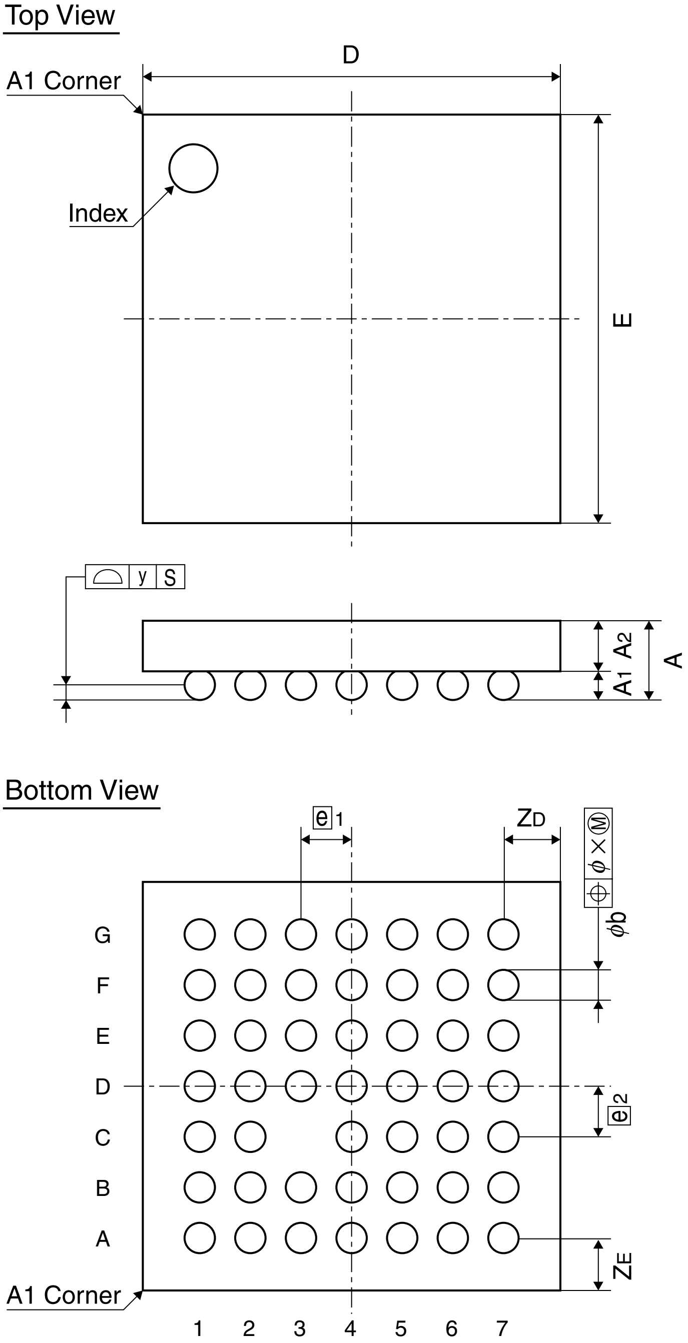


Figure I.3.5.1.1 TQFP12-64pin Package Dimensions

I.3.5.2 WCSP-48 Package



Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	3.024	3.124	3.224
E	3.024	3.124	3.224
A	–	–	0.78
A_1	–	0.23	–
A_2	–	0.49	–
$\boxed{e}_1$	–	0.40	–
$\boxed{e}_2$	–	0.40	–
b	0.23	0.26	0.29
X	–	–	0.08
y	–	–	0.05
Z_D	–	0.362	–
Z_E	–	0.362	–

Figure I.3.5.2.1 WCSP-48 Package Dimensions

I.3.5.3 Thermal Resistance of the Package

The chip temperature of LSI devices tends to increase with the power consumed on the chip. The chip temperature when encapsulated in a package is calculated from its ambient temperature (Ta), the thermal resistance of the package (θ), and power dissipation (P_D).

Chip temperature (Tj) = Ta + ($P_D \times \theta$) [°C]

When used under normal operating conditions, make sure that the chip temperature (Tj) is 125°C or less.

1. When mounted on a board (windless condition)

   Thermal resistance (θj-a) = 33.3°C/W

   This value indicates the thermal resistance of the package when measured under a windless condition, with the sample mounted on a measurement board (size: 114 × 76 × 1.6 mm thick, FR4/4 layered board).

2. When suspended alone (windless condition)

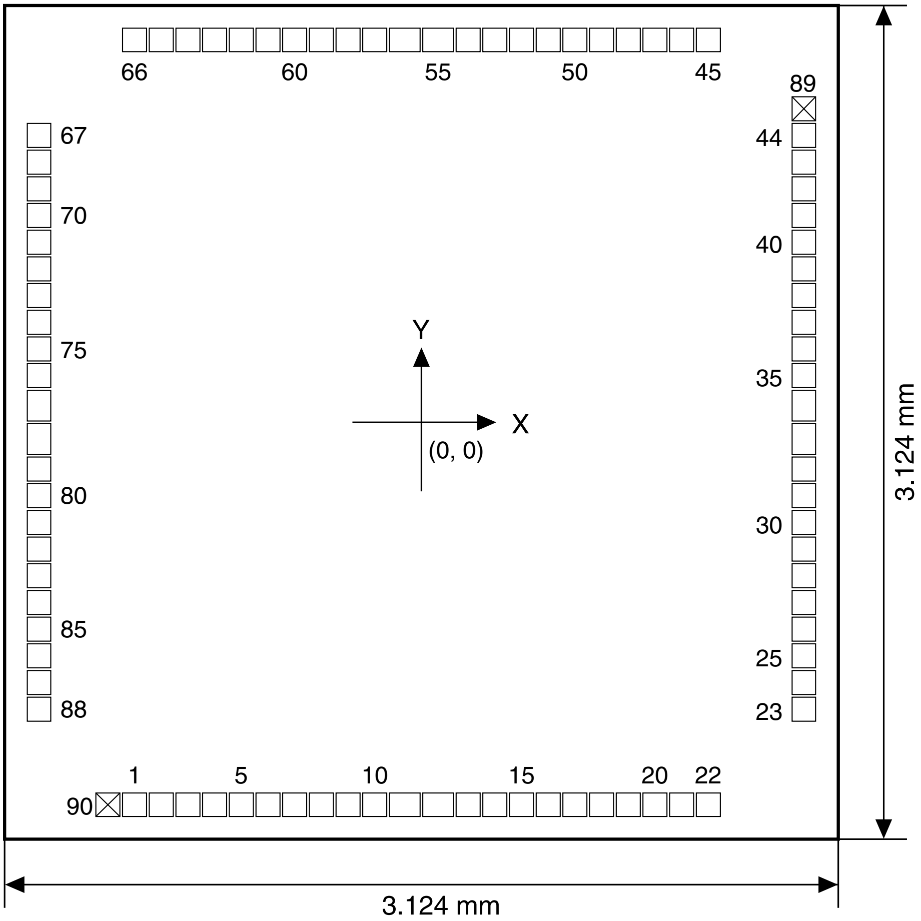
   Thermal resistance = 90–100°C/W

   This value indicates the thermal resistance of the package when measured under a windless condition, with the sample suspended alone.

Note: The thermal resistance of the package varies significantly depending on how it is mounted on the board and whether forcibly air-cooled.

I.3.6 Pad Layout

I.3.6.1 Diagram of Pad Layout



Chip thickness: 400 μm

I.3.6.2 Pad Coordinates

Table I.3.6.2 Pad Coordinates (Unit: μm)

PAD No.	PAD name	PAD coordinates		PAD opening		PAD No.	PAD name	PAD coordinates		PAD opening	
		X	Y	X	Y			X	Y	X	Y
1	P10 (SPI_SDI0)	-1075	-1433	90	88	45	N.C.	1075	1433	90	88
2	VSS	-975	-1433	90	88	46	OSC1	975	1433	90	88
3	VSS	-875	-1433	90	88	47	VSS	875	1433	90	88
4	P11 (SPI_SDO0)	-775	-1433	90	88	48	VSS	775	1433	90	88
5	N.C.	-675	-1433	90	88	49	OSC2	675	1433	90	88
6	P12 (SPI_SCK0)	-575	-1433	90	88	50	N.C.	575	1433	90	88
7	P13 (#SPI_SSI0/#SPI_SSI1/SPI_SDI0)	-475	-1433	90	88	51	VSS	475	1433	90	88
8	LVDD	-375	-1433	90	88	52	OSC3	375	1433	90	88
9	LVDD	-275	-1433	90	88	53	N.C.	275	1433	90	88
10	HVDD	-175	-1433	90	88	54	VSS	175	1433	90	88
11	HVDD	-62.5	-1433	115	88	55	OSC4	62.5	1433	115	88
12	P14 (SPI_SDI1)	62.5	-1433	115	88	56	VSS	-62.5	1433	115	88
13	P15 (SPI_SDO1)	175	-1433	90	88	57	N.C.	-175	1433	90	88
14	P16 (SPI_SCK1)	275	-1433	90	88	58	P20	-275	1433	90	88
15	P43 (I2CS_SDA/REMC_IN/PWMPRT0)	375	-1433	90	88	59	P21	-375	1433	90	88
16	P44 (I2CS_SCL/REMC_OUT)	475	-1433	90	88	60	HVDD	-475	1433	90	88
17	HVDD	575	-1433	90	88	61	P22	-575	1433	90	88
18	P24	675	-1433	90	88	62	P23	-675	1433	90	88
19	P25 (#I2CS_RST)	775	-1433	90	88	63	N.C.	-775	1433	90	88
20	VSS	875	-1433	90	88	64	LVDD	-875	1433	90	88
21	VSS	975	-1433	90	88	65	N.C.	-975	1433	90	88
22	N.C.	1075	-1433	90	88	66	N.C.	-1075	1433	90	88
23	N.C.	1433	-1075	88	90	67	#RESET	-1433	1075	88	90
24	P26 (I2CS_SDA)	1433	-975	88	90	68	N.C.	-1433	975	88	90
25	N.C.	1433	-875	88	90	69	LVDD	-1433	875	88	90
26	P27 (I2CS_SCL)	1433	-775	88	90	70	LVDD	-1433	775	88	90
27	VSS	1433	-675	88	90	71	P40 (SIN0)	-1433	675	88	90
28	VSS	1433	-575	88	90	72	P41 (SOUT0)	-1433	575	88	90
29	HVDD	1433	-475	88	90	73	P42 (#SCLK0/EXCL0)	-1433	475	88	90
30	P30 (TM0)	1433	-375	88	90	74	P50 (I2CM_SDA/EXCL0)	-1433	375	88	90
31	LVDD	1433	-275	88	90	75	HVDD	-1433	275	88	90
32	P31 (#I2CS_RST/#TM0/#ADTRG)	1433	-175	88	90	76	VSS	-1433	175	88	90
33	DST2 (P37)	1433	-62.5	88	115	77	P51 (I2CM_SCL)	-1433	62.5	88	115
34	LVDD	1433	62.5	88	115	78	VSS	-1433	-62.5	88	115
35	LVDD	1433	175	88	90	79	P32 (WDT_CLK/#WDT_NMI/CMU_CLK)	-1433	-175	88	90
36	DSIO (P36)	1433	275	88	90	80	AVDD	-1433	-275	88	90
37	N.C.	1433	375	88	90	81	N.C.	-1433	-375	88	90
38	TEST0	1433	475	88	90	82	AVDD	-1433	-475	88	90
39	DCLK (P35)	1433	575	88	90	83	P03 (AIN3)	-1433	-575	88	90
40	P52 (REMC_IN/#TM0)	1433	675	88	90	84	P02 (AIN2)	-1433	-675	88	90
41	N.C.	1433	775	88	90	85	AVDD	-1433	-775	88	90
42	P53 (REMC_OUT)	1433	875	88	90	86	P01 (AIN1)	-1433	-875	88	90
43	VSS	1433	975	88	90	87	P00 (AIN0)	-1433	-975	88	90
44	VSS	1433	1075	88	90	88	N.C.	-1433	-1075	88	90
–	–	–	–	–	–	89	(Test pad)	1433	1175	88	90
–	–	–	–	–	–	90	(Test pad)	-1433	-1175	88	90

THIS PAGE IS BLANK.

I.4 Power Supply

This section explains the operating voltage of the S1C17002.

I.4.1 Power Supply Pins

The S1C17002 has the power supply pins shown in Table I.4.1.1.

Table I.4.1.1 Power Supply Pins

Pin name	Pin No.		I/O	Type	PU/PD	Description
	TQFP	WCSP				
HV$_{DD}$	7, 13, 20, 44, 55	C4, E4	–	3.3 V	–	I/O power supply (+) (1.8 V/2.5 V/3.3 V)
LV$_{DD}$	6, 22, 25, 48, 50	D5, F2	–	1.8 V	–	Core power supply (+) (1.8 V)
V$_{SS}$	2, 16, 19, 32, 34, 36, 39, 41, 56	D4, C6, F5	–	GND	–	GND
AV$_{DD}$	60	D3	–	3.3 V	–	Analog power supply (3.0 V/3.3 V)

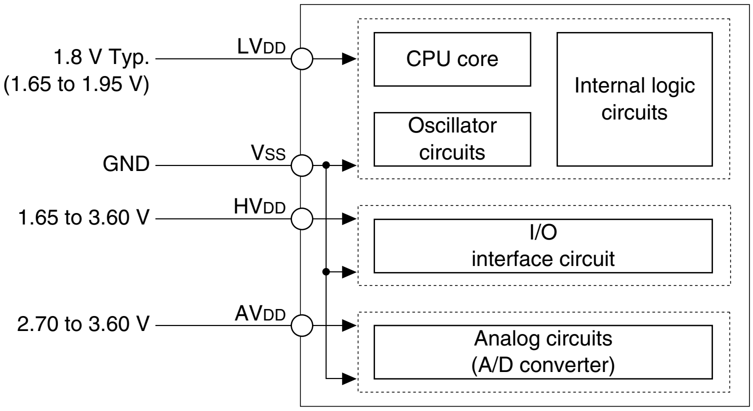


Figure I.4.1.1 Power Supply System

I.4.2 Operating Voltage (LVDD, VSS)

The core CPU and internal logic circuits operate with a voltage supplied between the LVDD and VSS pins.
The following operating voltage can be used:

LVDD = 1.65 V to 1.95 V (1.80 V ± 0.15 V, VSS = GND)

Note: The S1C17002 TQFP package has five LVDD pins and nine VSS pins; the WCSP package has two LVDD pins and three VSS pins. Be sure to supply the operating voltage to all the pins. Do not open any of them.

I.4.3 Power Supply for I/O Interface (HVDD)

The HVDD voltage is used for interfacing with external I/O signals. For the output interface of the S1C17002, the HVDD voltage is used as high level and the VSS voltage as low level. The VSS pin is used for the ground common with LVDD. The following voltage is enabled for HVDD:

HVDD = 1.65 V to 3.60 V (VSS = GND)

Notes:
- The S1C17002 TQFP package has five HVDD pins; the WCSP package has two HVDD pins. Be sure to supply the operating voltage to all the pins. Do not open any of them.
- When an external clock is input to the OSC3 or OSC1 pin, the clock signal level must be LVDD.

I.4.4 Power Supply for Analog Circuits (AVDD)

The analog power supply pin (AVDD) is provided separately from the LVDD and HVDD pins in order that the digital circuits do not affect the analog circuit (A/D converter). The AVDD pin is used to supply an analog power voltage and the VSS pin is used as the analog ground.
The following voltage is enabled for AVDD:

AVDD = 2.70 V to 3.60 V or 1.65 V to 3.60 V (Note) (VSS = GND)

Notes:
- Be sure to supply a voltage within the range from 1.65 to 3.60 V to the AVDD pin even if the analog circuit is not used. It is not necessary to supply a voltage same as the HVDD level.
- The AVDD voltage range can be changed to 1.65 to 3.60 V only when the ADC is not used and the P0*x* pins are used as digital signal input pins, not analog input pins. However, the high and low level input voltages of the digital signals must be AVDD and GND, respectively.

Noise on the analog power lines decrease the A/D converting precision, so use a stabilized power supply and make the board pattern with consideration given to that.

I.4.5 Precautions on Power Supply

Power-on sequence

In order to operate the device normally, supply power in accordance with the following timing.

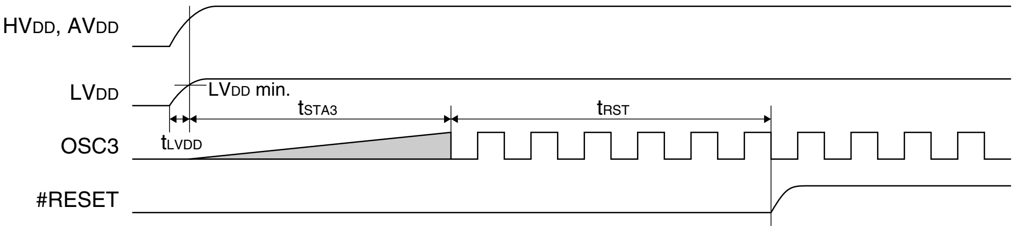


Figure I.4.5.1 Power-On Sequence

(1) t_{LVDD}: Elapsed time until the power supply stabilizes after power-on
Supply power in the following sequence:
Power-on: LV_{DD} → HV_{DD} (I/O), AV_{DD} (A/D) → Apply the input signal
or LV_{DD}, HV_{DD} (I/O), AV_{DD} (A/D) → Apply the input signal
(See Notes in "Power-off sequence" below.)

(2) t_{STA3}: Time at which OSC3 oscillation starts

(3) t_{RST}: Minimum reset pulse width
Time at which the clock supplied to the chip stabilizes plus at least six clocks; Keep the #RESET signal low.

Note: When the HV_{DD} power is turned on from off status, stable internal circuit statuses cannot be guaranteed due to noise in the power line. Therefore, the circuit statuses must be initialized (reset) after the power is turned on.

Power-off sequence

Shut off the power supply in the following sequence:
Power-off: Turn off the input signal → HV_{DD} (I/O), AV_{DD} (A/D) → LV_{DD}
or Turn off the input signal → HV_{DD} (I/O), AV_{DD} (A/D), LV_{DD} (See Notes below.)

Notes:

- Applying only LV_{DD} with other power voltage turned off makes a diode circuit on the path from LV_{DD} to HV_{DD} (AV_{DD}) that results current flowing to the HV_{DD} (AV_{DD}) power supply. In order to avoid this statue, the power supplies should be turned off simultaneously.
- Be sure to avoid applying HV_{DD} or AV_{DD} for a duration of one second or more when the LV_{DD} power is off, as a breakdown may occur in the device or the characteristics may be degraded due to flow-through current of the HV_{DD} or AV_{DD}.

Latch-up

The CMOS device may be in the latch-up condition. This is the phenomenon caused by conduction of the parasitic PNPN junction (thyristor) contained in the CMOS IC, resulting in a large current between HV_{DD} and V_{SS} and leading to breakage.

Latch-up occurs when the voltage applied to the input / output exceeds the rated value and a large current flows into the internal element, or when the voltage at the HV_{DD} pin exceeds the rated value and the internal element is in the breakdown condition. In the latter case, even if the application of a voltage exceeding the rated value is instantaneous, the current remains high between HV_{DD} and V_{SS} once the device is in the latch-up condition. As this may result in heat generation or smoking, the following points must be taken into consideration:

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

THIS PAGE IS BLANK.

I.5 CPU

The S1C17002 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 that do not need a lot of data processing power like the S1C33 Cores the high-end processors, 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."

I.5.1 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 cycles
- 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)

I.5.2 CPU Registers

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

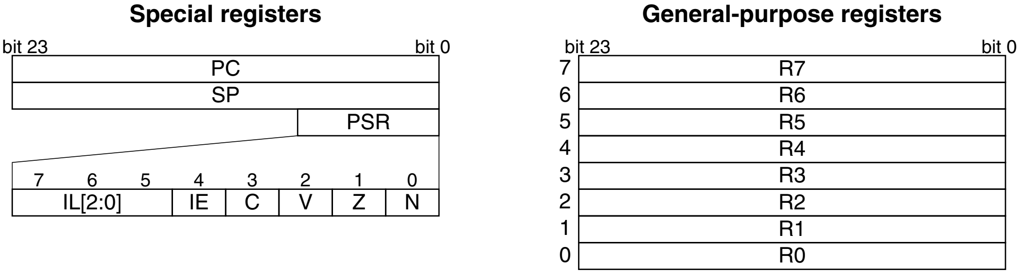


Figure I.5.2.1 Registers

I.5.3 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 I.5.3.1 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) Memory address post-increment, post-decrement, and pre-decrement functions can be used.
		%rd,[%rb]+	
		%rd,[%rb]-	
		%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 Memory address post-increment, post-decrement, and pre-decrement functions can be used.
		[%rb]+,%rs	
		[%rb]-,%rs	
		-[%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) Memory address post-increment, post-decrement, and pre-decrement functions can be used.
		%rd,[%rb]+	
		%rd,[%rb]-	
		%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 Memory address post-increment, post-decrement, and pre-decrement functions can be used.
		%rd,[%rb]+	
		%rd,[%rb]-	
		%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 Memory address post-increment, post-decrement, and pre-decrement functions can be used.
		[%rb]+,%rs	
		[%rb]-,%rs	
		-[%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)
		%rd,[%rb]	Memory (32 bits) → general-purpose register * Memory address post-increment, post-decrement, and pre-decrement functions can be used.
		%rd,[%rb]+	
		%rd,[%rb]-	
		%rd,-[%rb]	
		%rd,[%sp+imm7]	Stack (32 bits) → general-purpose register *
		%rd,[imm7]	Memory (32 bits) → general-purpose register *
		[%rb],%rs	General-purpose register (32 bits, zero-extended) → memory * Memory address post-increment, post-decrement, and pre-decrement functions can be used.
		[%rb]+,%rs	
		[%rb]-,%rs	
		-[%rb],%rs	
		[%sp+imm7],%rs	General-purpose register (32 bits, zero-extended) → stack *
		[imm7],%rs	General-purpose register (32 bits, zero-extended) → memory *
		%rd,%sp	SP → general-purpose register
		%rd,%pc	PC → general-purpose register
		%rd,[%sp]	Stack (32 bits) → general-purpose register * Stack pointer post-increment, post-decrement, and pre-decrement functions can be used.
		%rd,[%sp]+	
		%rd,[%sp]-	
		%rd,-[%sp]	

Classification	Mnemonic		Function
Data transfer	ld.a	[%sp],%rs	General-purpose register (32 bits, zero-extended) → stack * Stack pointer post-increment, post-decrement, and pre-decrement functions can be used.
		[%sp]+,%rs	
		[%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
Logical operation	and	%rd,%rs	Logical AND between general-purpose registers Supports conditional execution (/c: executed if C = 1, /nc: executed if C = 0).
	and/c		
	and/nc		
	and	%rd,sign7	Logical AND of general-purpose register and immediate
	or	%rd,%rs	Logical OR between general-purpose registers Supports conditional execution (/c: executed if C = 1, /nc: executed if C = 0).
	or/c		
	or/nc		
	or	%rd,sign7	Logical OR of general-purpose register and immediate
	xor	%rd,%rs	Exclusive OR between general-purpose registers Supports conditional execution (/c: executed if C = 1, /nc: executed if C = 0).
	xor/c		
	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) Supports conditional execution (/c: executed if C = 1, /nc: executed if C = 0).
	not/c		
	not/nc		
	not	%rd,sign7	Logical inversion of general-purpose register and immediate (1's complement)

Classification	Mnemonic		Function
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*	Convert signed 8-bit data into 24 bits
	cv.as	*%rd,%rs*	Convert signed 16-bit data into 24 bits
	cv.al	*%rd,%rs*	Convert 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
	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	

* The `ld.a` instruction accesses memories in 32-bit length. During data transfer from a register to a memory other than the IRAM area, the 32-bit data in which the eight high-order bits are set to 0 is written to the memory. During data transfer from a register to the IRAM area, the eight high-order bits are not 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 I.5.3.2 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)

I.5.4 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. The boot address from which the program starts running after a reset must be written to the top of the vector table.

Table I.5.4.1 shows the vector table of the S1C17002.

Table I.5.4.1 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	C compiler (reserved)	Used in emulation library for C compiler	5
4 (0x04)	TTBR + 0x10	Port input interrupt 0	P*x*0 input (rising/falling edge or high/low level)	High *1
5 (0x05)	TTBR + 0x14	Port input interrupt 1	P*x*1 input (rising/falling edge or high/low level)	↑
6 (0x06)	TTBR + 0x18	Port input interrupt 2	P*x*2 input (rising/falling edge or high/low level)	
7 (0x07)	TTBR + 0x1c	Port input interrupt 3	P*x*3 input (rising/falling edge or high/low level)	
8 (0x08)	TTBR + 0x20	MFT interrupt	• Compare-match • Period-match • ADC protection input • Port protection input	
9 (0x09)	TTBR + 0x24	reserved	–	
10 (0x0a)	TTBR + 0x28	A/D converter	Out of range results (upper- and lower-limit)	
11 (0x0b)	TTBR + 0x2c		End of conversion	
12 (0x0c)	TTBR + 0x30	CLG_T16U0 timer interrupt	Timer underflow	
		Port input interrupt 4	P*x*4 input (rising/falling edge or high/low level)	
13 (0x0d)	TTBR + 0x34	CLG_T8FU0 timer interrupt	Timer underflow	
		Port input interrupt 5	P*x*5 input (rising/falling edge or high/low level)	
14 (0x0e)	TTBR + 0x38	CLG_T8S timer interrupt	Timer underflow	
		Port input interrupt 6	P*x*6 input (rising/falling edge or high/low level)	
15 (0x0f)	TTBR + 0x3c	CLG_T8I timer interrupt	Timer underflow	
		Port input interrupt 7	P*x*7 input (rising/falling edge or high/low level)	
16 (0x10)	TTBR + 0x40	UART with IrDA CH.0 interrupt	• Transmit buffer empty • Receive buffer full • Receive error	
		Port input interrupt 4	P*x*4 input (rising/falling edge or high/low level)	
17 (0x11)	TTBR + 0x44	Port input interrupt 5	P*x*5 input (rising/falling edge or high/low level)	
18 (0x12)	TTBR + 0x48	SPI CH.0 interrupt	• Transmit buffer empty • Receive buffer full	
		Port input interrupt 6	P*x*6 input (rising/falling edge or high/low level)	
19 (0x13)	TTBR + 0x4c	I^{2}C master interrupt	• Transmit buffer empty • Receive buffer full	
		Port input interrupt 7	P*x*7 input (rising/falling edge or high/low level)	
20 (0x14)	TTBR + 0x50	RTC interrupt	1/64 second, 1 second, 1 minute, or 1 hour count up	
21 (0x15)	TTBR + 0x54	8-bit timer CH.0 interrupt	Timer 0 underflow	
		8-bit OSC1 timer CH.0 interrupt	Compare match	
22 (0x16)	TTBR + 0x58	8-bit timer CH.1 interrupt	Timer 1 underflow	
		8-bit OSC1 timer CH.1 interrupt	Compare match	
23 (0x17)	TTBR + 0x5c	8-bit timer CH.2 interrupt	Timer 2 underflow	
24 (0x18)	TTBR + 0x60	8-bit timer CH.3 interrupt	Timer 3 underflow	
25 (0x19)	TTBR + 0x64	reserved	–	
26 (0x1a)	TTBR + 0x68	SPI CH.1 interrupt	• Transmit buffer empty • Receive buffer full	
27 (0x1b)	TTBR + 0x6c	reserved	–	
28 (0x1c)	TTBR + 0x70	I^{2}C slave interrupt	• Transmit buffer empty • Receive buffer full	
29 (0x1d)	TTBR + 0x74		• Bus status	
30 (0x1e)	TTBR + 0x78	REMC interrupt	• Envelope counter underflow • REMC_IN rising edge detection • REMC_IN falling edge detection	↓
31 (0x1f)	TTBR + 0x7c	reserved	–	Low *1

*1 When the same interrupt level is set

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

The S1C17002 allows the base (starting) address of the vector table to be set using the TTBR_LOW and TTBR_HIGH registers (0x5814, 0x5816). "TTBR" described in Table I.5.4.1 means the value set to these registers. After an initial reset, the TTBR_LOW/HIGH registers are set to 0x20000. Therefore, even when the vector table position is changed, it is necessary that at least the reset vector be written to the above address. Bits 7 to 0 in the TTBR_LOW register are fixed at 0, so the vector table starting address always begins with a 256-byte boundary address.

0x5814–0x5816: Vector Table Base Registers (TTBR_LOW, TTBR_HIGH)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Vector Table Base Register 0 (TTBR_LOW)	**0x5814** (16 bits)	D15–8	**TTBR[15:8]**	Vector table base address A[15:8]	0x0–0xff	0x0	R/W	
		D7–0	**TTBR[7:0]**	Vector table base address A[7:0] (fixed at 0)	0x0	0x0	R	
Vector Table Base Register 1 (TTBR_HIGH)	**0x5816** (16 bits)	D15–8	**–**	reserved	–	–	–	0 when being read.
		D7–0	**TTBR[23:16]**	Vector table base address A[23:16]	0x0–0xff	0x2	R/W	

Note: The Vector Table Base Registers are write-protected. Before these registers can be rewritten, write protection must be removed by writing data 0x96 to the ROMC Protect Register (0x5810). Note that since unnecessary rewrites to the Vector Table Base Registers could lead to erratic system operation, the ROMC Protect Register (0x5810) should be set to other than 0x96 unless the Vector Table Base Registers must be rewritten.

I.5.5 On-chip Debugger

I.5.5.1 Debug Functions

The S1C17 Core has an embedded debug unit to assist in software development by the user.
The debug unit provides the following functions that are used with debugging tools:

- Instruction break
  A debug interrupt is generated before the set instruction address is executed. An instruction break can be set at one address location.
- Single step
  A debug interrupt is generated every instruction executed.
- Forcible break
  A debug interrupt is generated by an external input signal (DSIO = 0).
- Software break
  A debug interrupt is generated when the `brk` instruction is executed.

When a debug interrupt occurs, the processor performs the following processing:

(1) Suspends the instructions currently being executed.

(2) Saves the contents of the PC and PSR, and R0, in that order, to the addresses specified below.
PC/PSR → DBRAM + 0x0
R0 → DBRAM + 0x4 (DBRAM: Start address of the work area for debugging in the user RAM)

(3) Loads address 0xfffc00 to PC and branches to the debug interrupt handler routine.

In the interrupt handler routine, the `retd` instruction should be executed at the end of processing to return to the suspended instructions. When returning from the interrupt by the `retd` instruction, the processor restores the saved data in order of the R0 and the PC and PSR.
Neither hardware interrupts nor NMI interrupts are accepted during a debug interrupt.

I.5.5.2 Work Area for Debugging

A 64-byte work area is required for debugging. In the S1C17002, the address range from 0x0 to 0x3f in the internal RAM is reserved as the work area for debugging. When using the debug functions, do not access this area from the application program.
The debug RAM start address can be read out from the DBRAM register (0xffff90).

0xffff90: 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	0x0	0x0	R	

D[31:24] Unused (fixed at 0)

D[23:0] DBRAM[23:0]: Debug RAM Base Address Bits
This is a read-only register that contains the start address of a work area (64 bytes) for debugging.

I.5.5.3 Debugging Tools

Debugging is performed by connecting the ICD (In-Circuit Debugger) such as S5U1C17001H (ICD Mini) to the debug pins of the S1C17002 and entering debug commands from the debugger being run on a personal computer.
The tools listed below are required for debugging.

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

I.5.5.4 Debug Pins

The ICD (e.g. S5U1C17001H) is connected to the debug pins listed below.

Table I.5.5.4.1 List of Debug Pins

Pin name	I/O	Size	Function
DCLK (P35)	O	1	On-chip debugger clock output pin This pin outputs a clock to the ICD.
DSIO (P36)	I/O	1	On-chip debugger data input/output pin This pin inputs/outputs data for debugging and inputs a break signal.
DST2 (P37)	O	1	On-chip debugger status signal output pin This pin outputs the processor status during debugging (goes low in normal mode or goes high in debug mode).

The on-chip debugger input/output pins (DCLK, DSIO, DST2) are shared with the I/O ports (P35, P36, P37) and they are initialized as debug pins by default. When the debug function is not used, these pins can be configured for general-purpose I/O ports using the P3_57_CFP register (0x4427). For details on switching pin function, see Section I.3.3, “Switching Over the Multiplexed Pin Functions.”

I.5.5.5 Clock for Debugging

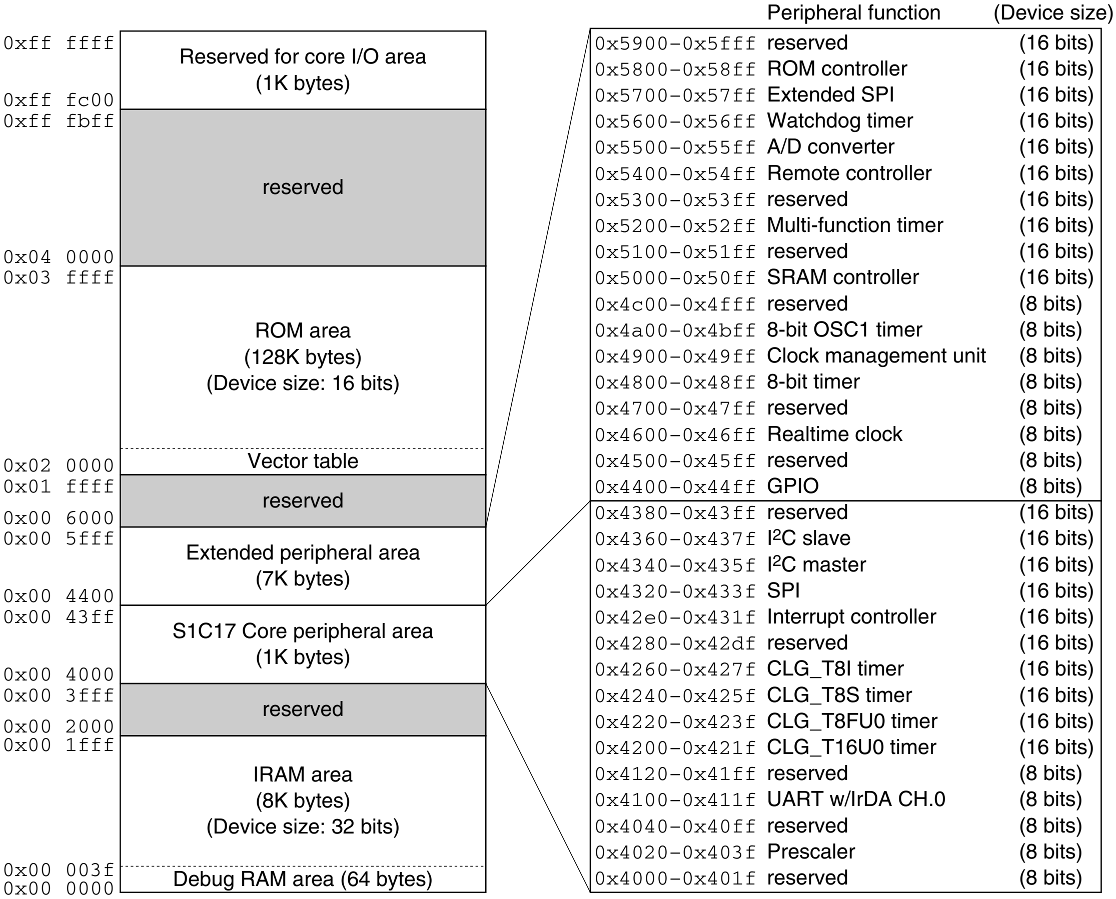
The embedded debug unit communicates with the ICD in serial data transfer. DCLK is the sync clock for transfer and its frequency is always half of the CCLK frequency.

I.5.5.6 Debugger Status Signal (DST2)

The DST2 signal is set to low during normal operation and it goes high when the S1C17 Core enters debug mode by a debug interrupt.

I.6 Memory Map

Figure I.6.1 shows the S1C17002 memory map.


0xff ffff
Reserved for core I/O area (1K bytes)
0xff fc00
0xff fbff
reserved
0x04 0000
0x03 ffff
ROM area (128K bytes) (Device size: 16 bits)
0x02 0000
Vector table
0x01 ffff
reserved
0x00 6000
0x00 5fff
Extended peripheral area (7K bytes)
0x00 4400
0x00 43ff
S1C17 Core peripheral area (1K bytes)
0x00 4000
0x00 3fff
reserved
0x00 2000
0x00 1fff
IRAM area (8K bytes) (Device size: 32 bits)
0x00 003f
Debug RAM area (64 bytes)
0x00 0000

	Peripheral function	(Device size)
0x5900–0x5fff	reserved	(16 bits)
0x5800–0x58ff	ROM controller	(16 bits)
0x5700–0x57ff	Extended SPI	(16 bits)
0x5600–0x56ff	Watchdog timer	(16 bits)
0x5500–0x55ff	A/D converter	(16 bits)
0x5400–0x54ff	Remote controller	(16 bits)
0x5300–0x53ff	reserved	(16 bits)
0x5200–0x52ff	Multi-function timer	(16 bits)
0x5100–0x51ff	reserved	(16 bits)
0x5000–0x50ff	SRAM controller	(16 bits)
0x4c00–0x4fff	reserved	(8 bits)
0x4a00–0x4bff	8-bit OSC1 timer	(8 bits)
0x4900–0x49ff	Clock management unit	(8 bits)
0x4800–0x48ff	8-bit timer	(8 bits)
0x4700–0x47ff	reserved	(8 bits)
0x4600–0x46ff	Realtime clock	(8 bits)
0x4500–0x45ff	reserved	(8 bits)
0x4400–0x44ff	GPIO	(8 bits)
0x4380–0x43ff	reserved	(16 bits)
0x4360–0x437f	I^{2}C slave	(16 bits)
0x4340–0x435f	I^{2}C master	(16 bits)
0x4320–0x433f	SPI	(16 bits)
0x42e0–0x431f	Interrupt controller	(16 bits)
0x4280–0x42df	reserved	(16 bits)
0x4260–0x427f	CLG_T8I timer	(16 bits)
0x4240–0x425f	CLG_T8S timer	(16 bits)
0x4220–0x423f	CLG_T8FU0 timer	(16 bits)
0x4200–0x421f	CLG_T16U0 timer	(16 bits)
0x4120–0x41ff	reserved	(8 bits)
0x4100–0x411f	UART w/IrDA CH.0	(8 bits)
0x4040–0x40ff	reserved	(8 bits)
0x4020–0x403f	Prescaler	(8 bits)
0x4000–0x401f	reserved	(8 bits)

Figure I.6.1 S1C17002 Memory Map

I.6.1 Access Cycle

As shown in Table I.6.1.1, the number of cycles required for one bus access depends on the peripheral or memory module. Furthermore, the number of bus accesses depends on the CPU instruction (access size) and device size.

Table I.6.1.1 Number of Access Cycles for Data Read/Write

Module	Access condition		Write	Read
IRAM	–	8-bit access	1	2
		16-bit access	1	2
		24/32-bit access	1	2
ROM	–	8-bit access	–	2 + w
		16-bit access	–	2 + w
		24/32-bit access	–	1 + (1 + w) × 2
Peripheral control registers	8-bit device	8-bit access	4 + w	4 + w
		16-bit access	7 + w × 2	7 + w × 2
		24-bit access	13 + w × 4	13 + w × 4
	16-bit device	8-bit access	4 + w	4 + w
		16-bit access	4 + w	4 + w
		24-bit access	7 + w × 2	7 + w × 2
MAC operation	–	–	1	–
DIV operation	–	–	17 to 20	–

Note: "w" means the number of wait cycles.

Table I.6.1.2 Number of Access Cycles for Instruction Read

Module	Access condition		Instruction Read
IRAM	CPU read	32-bit read	2 (Note)
ROM		32-bit read	1 + (1 + w) × 2

Notes:
- "w" means the number of wait cycles.
- The CPU can read a 16-bit instruction from the IRAM in 1 clock cycle.

Handling the eight high-order bits during 32-bit accesses

During writing, the eight high-order bits of 32-bit data are written as 0. However, the eight high-order bits are not written when data is written to IRAM using the "`ld.a`" instruction.

During reading from a memory, the eight high-order bits are ignored. However, the eight high-order bits are effective as the PSR value only in the stack operation when an interrupt occurs.

I.6.1.1 Restrictions on Access Size

The modules shown below have a restriction on the access size. Appropriate instructions should be used in programming.

I²C master, WDT

The I²C master and watchdog timer registers allow only 16-bit read/write instructions for accessing.

Other peripheral modules can be accessed with an 8-bit, or 16-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.

I.6.1.2 Simultaneous Access to Instruction and Data by Harvard Architecture

The S1C17 Core has adopted Harvard Architecture. An instruction fetch and a data access are performed simultaneously under one of the conditions listed below, this makes it possible to improve the execution speed.

- When the S1C17002 accesses data in the IRAM area and executes the instruction stored in the ROM area
- When the S1C17002 executes the instruction stored in the IRAM area and accesses data in the ROM area, S1C17 Core peripheral area (0x4000–), or extended peripheral area (0x4400–)
- When the S1C17002 accesses data in the S1C17 Core peripheral area (0x4000–) and executes the instruction stored in the IRAM or ROM area

I.6.2 IRAM Area

The S1C17002 contains a RAM in the 8K-byte area from address 0x0 to address 0x17ff. The RAM is accessed in one cycle for data writing or two cycles for data reading regardless of the access size. An instruction can be read in one cycle from the IRAM.

Notes:
- The 64-byte area at the beginning of the RAM (0x0–0x3f) 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.
- When data is written to IRAM using the "`ld.a`" instruction, the S1C17 Core does not write anything to the eight high-order bits (D[31:24]) of the 32-bit space.
  Example: `ld.a [%rb],%rs`

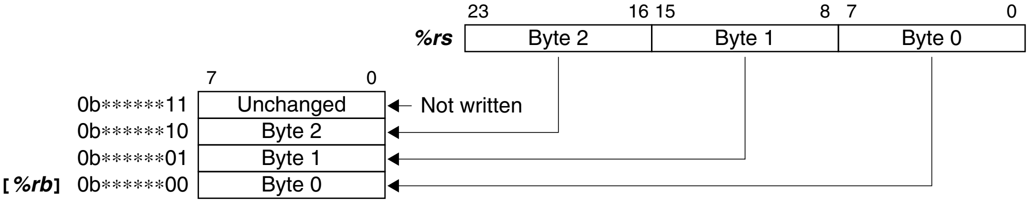


Figure I.6.2.1 24-bit Write to IRAM

I.6.3 ROM Area

The S1C17002 contains a mask ROM in the 128K-byte area from address 0x20000 to address 0x3ffff for storing application programs and data. Address 0x20000 is defined as the vector table base address by default, therefore the reset vector must be placed on this address. The vector table base address can be changed using the TTBR_LOW/HIGH registers (0x5814, 0x5816).

The ROM can be read in a minimum of one cycle.
The ROM controller can insert a wait cycle in the ROM read cycle. The number of system clock cycles to be inserted as a wait cycle can be specified using ROM_WAIT[2:0] (D[2:0]/ROMC_WAIT register).

0x5804: ROMC Wait Register (ROMC_WAIT)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
ROMC Wait Register (ROMC_WAIT)	**0x5804** (16 bits)	D15–3	–	reserved	–		–	–	0 when being read.
		D2–0	**ROM_WAIT [2:0]**	ROM read access wait cycle setup	ROM_WAIT[2:0]	Wait cycle	0x7	R/W	
					0x7 : 0x0	7 cycles : 0 cycles			

D[15:3] Reserved

D[2:0] ROM_WAIT[2:0]: ROM Read Access Wait Cycle Setup Bits

These bits set the number of wait cycles to be inserted when the ROM is read.

Table I.1.6.3.1 Setting Read Access Wait Cycle

ROM_WAIT[2:0]	Number of wait cycles	Number of read access cycles	System clock frequency
0x7	7 cycles	8 cycles	Less than 20 MHz
0x6	6 cycles	7 cycles	
0x5	5 cycles	6 cycles	
0x4	4 cycles	5 cycles	
0x3	3 cycles	4 cycles	
0x2	2 cycles	3 cycles	
0x1	1 cycle	2 cycles	
0x0	0 cycles	1 cycle	

(Default: 0x7)

The S1C17002 can operate with 0 wait cycles by setting ROM_WAIT[2:0] to 0x0.

I.6.4 Internal Peripheral Area

The I/O and control registers for the internal peripheral modules are located in two areas beginning with addresses 0x4000 and 0x4400.

I.6.4.1 S1C17 Core Peripheral Area (0x4000–)

The S1C17 Core peripheral area beginning with address 0x4000 contains the I/O memory for the peripheral functions included in the Core module listed below and this area can be accessed in a minimum of four cycles.

- Prescaler (PSC, 8-bit device)
- UART (UART, 8-bit device)
- Clock generator (CLG, 16-bit device)
- Interrupt controller (ITC, 16-bit device)
- SPI (SPI, 16-bit device)
- I^{2}C master (I2CM, 16-bit device)
- I^{2}C slave (I2CS, 16-bit device)

I.6.4.2 Extended Peripheral Area (0x4400–)

The extended peripheral area beginning with address 0x4400 contains the I/O memory for the peripheral functions listed below and this area can be accessed in a minimum of four cycles.

- GPIO (GPIO, 8-bit device)
- Realtime clock (RTC, 8-bit device)
- 8-bit timer (PT8, 8-bit device)
- Clock management unit (CMU, 8-bit device)
- 8-bit OSC1 timer (T8OSC1, 8-bit device)
- SRAM controller (SRAMC, 16-bit device)
- Multi-function timer (MFT, 16-bit device)
- Remote controller (REMC, 16-bit device)
- A/D converter (ADC, 16-bit device)
- Watchdog timer (WDT, 16-bit device)
- Extended SPI (ESPI, 16-bit device)
- ROM controller (ROMC, 16-bit device)

I.6.4.3 I/O Map

This section shows the I/O map table for the internal peripheral area. For details of each control register, see the I/O register list in Appendix or description for each peripheral module.

Table I.6.4.3.1 I/O Map (S1C17 Core Peripheral Area)

Peripheral	Address	Register name		Function
Prescaler (8-bit device)	0x4020	PSC_CTL	Prescaler Control Register	Starts/stops the prescaler.
	0x4021–0x403f	–	–	Reserved
UART (with IrDA) (8-bit device)	0x4100	UART_ST	UART Status Register	Indicates transfer, buffer and error statuses.
	0x4101	UART_TXD	UART Transmit Data Register	Transmit data
	0x4102	UART_RXD	UART Receive Data Register	Receive data
	0x4103	UART_MOD	UART Mode Register	Sets transfer data format.
	0x4104	UART_CTL	UART Control Register	Controls data transfer.
	0x4105	UART_EXP	UART Expansion Register	Sets IrDA mode.
	0x4106–0x411f	–	–	Reserved
CLG_T16U0 timer (16-bit device)	0x4200	CLG_T16U0_CLK	CLG_T16U0 Input Clock Select Register	Selects a prescaler output clock.
	0x4202	CLG_T16U0_TR	CLG_T16U0 Reload Data Register	Sets reload data.
	0x4204	CLG_T16U0_TC	CLG_T16U0 Counter Data Register	Counter data
	0x4206	CLG_T16U0_CTL	CLG_T16U0 Control Register	Sets the timer mode and starts/stops the timer.
	0x4208–0x421f	–	–	Reserved
CLG_T8FU0 timer (16-bit device)	0x4220	CLG_T8FU0_CLK	CLG_T8S Input Clock Select Register	Selects a prescaler output clock.
	0x4222	CLG_T8FU0_TR	CLG_T8S Reload Data Register	Sets reload data.
	0x4224	CLG_T8FU0_TC	CLG_T8S Counter Data Register	Counter data
	0x4226	CLG_T8FU0_CTL	CLG_T8S Control Register	Sets the timer mode and starts/stops the timer.
	0x4228–0x423f	–	–	Reserved

Peripheral	Address	Register name		Function
CLG_T8S timer (16-bit device)	0x4240	CLG_T8S_CLK	CLG_T8S Input Clock Select Register	Selects a prescaler output clock.
	0x4242	CLG_T8S_TR	CLG_T8S Reload Data Register	Sets reload data.
	0x4244	CLG_T8S_TC	CLG_T8S Counter Data Register	Counter data
	0x4246	CLG_T8S_CTL	CLG_T8S Control Register	Sets the timer mode and starts/stops the timer.
	0x4248–0x425f	–	–	Reserved
CLG_T8I timer (16-bit device)	0x4260	CLG_T8I_CLK	CLG_T8I Input Clock Select Register	Selects a prescaler output clock.
	0x4262	CLG_T8I_TR	CLG_T8I Reload Data Register	Sets reload data.
	0x4264	CLG_T8I_TC	CLG_T8I Counter Data Register	Counter data
	0x4266	CLG_T8I_CTL	CLG_T8I Control Register	Sets the timer mode and starts/stops the timer.
	0x4268–0x427f	–	–	Reserved
Interrupt controller (16-bit device)	0x42e0	ITC_AIFLG	Additional Interrupt Flag Register	Indicates/resets interrupt occurrence status.
	0x42e2	ITC_AEN	Additional Interrupt Enable Register	Enables/disables each maskable interrupt.
	0x42e4	–	–	Reserved
	0x42e6	ITC_AILV0	Additional Interrupt Level Setup Register 0	Sets the MFT interrupt level.
	0x42e8	ITC_AILV1	Additional Interrupt Level Setup Register 1	Sets the ADC interrupt level.
	0x42ea	ITC_AILV2	Additional Interrupt Level Setup Register 2	Sets the RTC and PT8 CH.0/T8OSC1 CH.0 interrupt levels.
	0x42ec	ITC_AILV3	Additional Interrupt Level Setup Register 3	Sets the PT8 CH.1/T8OSC1 CH.1 and PT8 CH.2 interrupt levels.
	0x42ee	ITC_AILV4	Additional Interrupt Level Setup Register 4	Sets the PT8 CH.3 interrupt level.
	0x42f0	ITC_AILV5	Additional Interrupt Level Setup Register 5	Sets the SPI CH.1 interrupt level.
	0x42f2	ITC_AILV6	Additional Interrupt Level Setup Register 6	Sets the I^{2}C slave interrupt levels.
	0x42f4	ITC_AILV7	Additional Interrupt Level Setup Register 7	Sets the REMC interrupt level.
	0x42f6–0x42ff	–	–	Reserved
	0x4300	ITC_IFLG	Interrupt Flag Register	Indicates/resets interrupt occurrence status.
	0x4302	ITC_EN	Interrupt Enable Register	Enables/disables each maskable interrupt.
	0x4304	ITC_CTL	ITC Control Register	Enables/disables the ITC.
	0x4306	ITC_ELV0	External Interrupt Level Setup Register 0	Sets the port 0 and port 1 interrupt levels and trigger modes.
	0x4308	ITC_ELV1	External Interrupt Level Setup Register 1	Sets the port 2 and port 3 interrupt levels and trigger modes.
	0x430a	ITC_ELV2	External Interrupt Level Setup Register 2	Sets the port 4 and port 5 interrupt levels and trigger modes.
	0x430c	ITC_ELV3	External Interrupt Level Setup Register 3	Sets the port 6 and port 7 interrupt levels and trigger modes.
	0x430e	ITC_ILV0	Internal Interrupt Level Setup Register 0	Sets the CLG_T16U0 and CLG_T8FU0 timer interrupt levels.
	0x4310	ITC_ILV1	Internal Interrupt Level Setup Register 1	Sets the CLG_T8S and CLG_T8I timer interrupt levels.
	0x4312	ITC_ILV2	Internal Interrupt Level Setup Register 2	Sets the UART interrupt level.
	0x4314	ITC_ILV3	Internal Interrupt Level Setup Register 3	Sets the SPI CH.0 and I^{2}C master interrupt levels.
	0x4316–0x431f	–	–	Reserved
SPI (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 CH.0 mode and enables data transfer.
	0x4328–0x433f	–	–	Reserved
I^{2}C master (16-bit device)	0x4340	I2CM_EN	I^{2}C Master Enable Register	Enables the I^{2}C master module.
	0x4342	I2CM_CTL	I^{2}C Master Control Register	Controls the I^{2}C master operation and indicates transfer status.
	0x4344	I2CM_DAT	I^{2}C Master Data Register	I^{2}C master transmit/receive data
	0x4346	I2CM_ICTL	I^{2}C Master Interrupt Control Register	Controls the I^{2}C master interrupt.
	0x4348–0x435f	–	–	Reserved
I^{2}C slave (16-bit device)	0x4360	I2CS_TRNS	I^{2}C Slave Transmit Data Write Register	I^{2}C slave transmit data
	0x4362	I2CS_RECV	I^{2}C Slave Receive Data Read Register	I^{2}C slave receive data
	0x4364	I2CS_SADRS	I^{2}C Slave Address Setup Register	Sets the I^{2}C slave address.
	0x4366	I2CS_CTL	I^{2}C Slave Control Register	Controls the I^{2}C slave module.
	0x4368	I2CS_STAT	I^{2}C Slave Status Register	Indicates the I^{2}C slave bus status.
	0x436a	I2CS_ASTAT	I^{2}C Slave Access Status Register	Indicates the I^{2}C slave access status.
	0x436c	I2CS_ICTL	I^{2}C Slave Interrupt Control Register	Controls the I^{2}C slave interrupt.
	0x436e–0x437f	–	–	Reserved

Table I.6.4.3.2 I/O Map (Extended Peripheral Area)

Peripheral	Address	Register name		Function
GPIO (8-bit device)	0x4400	P0_DAT	P0 Port Input Data Register	P0 port input data
	0x4401	–	–	Reserved
	0x4402	P1_DAT	P1 Port Input/Output Data Register	P1 port input/output data
	0x4403	P1_IOC	P1 Port I/O Control Register	Selects the P1 port I/O direction.
	0x4404	P2_DAT	P2 Port Input/Output Data Register	P2 port input/output data
	0x4405	P2_IOC	P2 Port I/O Control Register	Selects the P2 port I/O direction.
	0x4406	P3_DAT	P3 Port Input/Output Data Register	P3 port input/output data
	0x4407	P3_IOC	P3 Port I/O Control Register	Selects the P3 port I/O direction.
	0x4408	P4_DAT	P4 Port Input/Output Data Register	P4 port input/output data
	0x4409	P4_IOC	P4 Port I/O Control Register	Selects the P4 port I/O direction.
	0x440a	P5_DAT	P5 Port Input/Output Data Register	P5 port input/output data
	0x440b	P5_IOC	P5 Port I/O Control Register	Selects the P5 port I/O direction.
	0x440c–0x441f	–	–	Reserved
	0x4420	P0_03_CFP	P00–P03 Port Function Select Register	Selects the P00–P03 port functions.
	0x4421	–	–	Reserved
	0x4422	P1_03_CFP	P10–P13 Port Function Select Register	Selects the P10–P13 port functions.
	0x4423	P1_46_CFP	P14–P16 Port Function Select Register	Selects the P14–P16 port functions.
	0x4424	–	–	Reserved
	0x4425	P2_57_CFP	P25–P27 Port Function Select Register	Selects the P25–P27 port functions.
	0x4426	P3_02_CFP	P30–P32 Port Function Select Register	Selects the P30–P32 port functions.
	0x4427	P3_57_CFP	P35–P37 Port Function Select Register	Selects the P35–P37 port functions.
	0x4428	P4_03_CFP	P40–P43 Port Function Select Register	Selects the P40–P43 port functions.
	0x4429	P4_4_CFP	P44 Port Function Select Register	Selects the P44 port functions.
	0x442a	P5_03_CFP	P50–P53 Port Function Select Register	Selects the P50–P53 port functions.
	0x442b–0x443f	–	–	Reserved
	0x4440	PINTSEL0	Port Input Interrupt 0 Select Register	Selects a P*x*0 port for input interrupt.
	0x4441	PINTSEL1	Port Input Interrupt 1 Select Register	Selects a P*x*1 port for input interrupt.
	0x4442	PINTSEL2	Port Input Interrupt 2 Select Register	Selects a P*x*2 port for input interrupt.
	0x4443	PINTSEL3	Port Input Interrupt 3 Select Register	Selects a P*x*3 port for input interrupt.
	0x4444	PINTSEL4	Port Input Interrupt 4 Select Register	Selects a P*x*4 port for input interrupt.
	0x4445	PINTSEL5	Port Input Interrupt 5 Select Register	Selects a P*x*5 port for input interrupt.
	0x4446	PINTSEL6	Port Input Interrupt 6 Select Register	Selects a P*x*6 port for input interrupt.
	0x4447	PINTSEL7	Port Input Interrupt 7 Select Register	Selects a P*x*7 port for input interrupt.
	0x4448–0x44ff	–	–	Reserved
Real-time clock (8-bit device)	0x4600	RTC_INTSTAT	RTC Interrupt Status Register	Indicates RTC interrupt status.
	0x4601	RTC_INTMODE	RTC Interrupt Mode Register	Sets up RTC interrupt modes.
	0x4602	RTC_CNTL0	RTC Control 0 Register	Controls the RTC.
	0x4603	RTC_CNTL1	RTC Control 1 Register	
	0x4604–0x4613	–	–	Reserved
	0x4614	RTC_SEC	RTC Second Register	Second counter data
	0x4615	RTC_MIN	RTC Minute Register	Minute counter data
	0x4616	RTC_HOUR	RTC Hour Register	Hour counter data
	0x4617	RTC_DAY	RTC Day Register	Day counter data
	0x4618–0x4627	–	–	Reserved
	0x4628	RTC_MONTH	RTC Month Register	Month counter data
	0x4629	RTC_YEAR	RTC Year Register	Year counter data
	0x462a	RTC_WEEK	RTC Days of Week Register	Days of week counter data
	0x462b–0x46ff	–	–	Reserved
8-bit programmable timer CH.0 (8-bit device)	0x4800	PT8_CLK0	PT8 CH.0 Input Clock Select Register	Selects the count clock.
	0x4801	PT8_RLD0	PT8 CH.0 Reload Data Register	Sets reload data.
	0x4802	PT8_PTD0	PT8 CH.0 Counter Data Register	Counter data
	0x4803	PT8_CTL0	PT8 CH.0 Control Register	Sets the timer mode and starts/stops the timer.
8-bit programmable timer CH.1 (8-bit device)	0x4804	PT8_CLK1	PT8 CH.1 Input Clock Select Register	Selects the count clock.
	0x4805	PT8_RLD1	PT8 CH.1 Reload Data Register	Sets reload data.
	0x4806	PT8_PTD1	PT8 CH.1 Counter Data Register	Counter data
	0x4807	PT8_CTL1	PT8 CH.1 Control Register	Sets the timer mode and starts/stops the timer.
8-bit programmable timer CH.2 (8-bit device)	0x4808	PT8_CLK2	PT8 CH.2 Input Clock Select Register	Selects the count clock.
	0x4809	PT8_RLD2	PT8 CH.2 Reload Data Register	Sets reload data.
	0x480a	PT8_PTD2	PT8 CH.2 Counter Data Register	Counter data
	0x480b	PT8_CTL2	PT8 CH.2 Control Register	Sets the timer mode and starts/stops the timer.
8-bit programmable timer CH.3 (8-bit device)	0x480c	PT8_CLK3	PT8 CH.3 Input Clock Select Register	Selects the count clock.
	0x480d	PT8_RLD3	PT8 CH.3 Reload Data Register	Sets reload data.
	0x480e	PT8_PTD3	PT8 CH.3 Counter Data Register	Counter data
	0x480f	PT8_CTL3	PT8 CH.3 Control Register	Sets the timer mode and starts/stops the timer.
	0x4810–0x48ff	–	–	Reserved

Peripheral	Address	Register name		Function
Clock management unit (8-bit device)	0x4900	CMU_SYSCLKCTL	System Clock Control Register	Controls the system clock.
	0x4901	CMU_OSC3_WCNT	OSC3 Wait Timer Register	Sets the OSC3 wait timer for system wake-up.
	0x4902	CMU_NF	Noise Filter Control Register	Enables noise filters.
	0x4903	CMU_OSC3DIV	OSC3 Clock Divider Register	Selects an OSC3 system clock frequency.
	0x4904	–	–	Reserved
	0x4905	CMU_CMUCLK	CMU_CLK Select Register	Selects the output CMU_CLK frequency.
	0x4906	CMU_GATEDCLK0	Gated Clock Control 0 Register	Controls clock supply to peripheral modules.
	0x4907	CMU_GATEDCLK1	Gated Clock Control 1 Register	
	0x4908	CMU_GATEDCLK2	Gated Clock Control 2 Register	
	0x4909–0x491f	–	–	Reserved
	0x4920	CMU_PROTECT	CMU Write Protect Register	Enables writing to the CMU registers (0x4900–0x4908).
	0x4921–0x49ff	–	–	Reserved
8-bit OSC1 timer CH.0 (8-bit device)	0x4a00	T8OSC1_CTL0	T8OSC1 CH.0 Control Register	Sets the timer mode and starts/stops the timer.
	0x4a01	T8OSC1_CNT0	T8OSC1 CH.0 Timer Counter Data Register	Counter data
	0x4a02	T8OSC1_CMP0	T8OSC1 CH.0 Timer Compare Data Register	Sets compare data.
	0x4a03	T8OSC1_IMSK0	T8OSC1 CH.0 Timer Interrupt Mask Register	Enables/disables interrupt.
	0x4a04	T8OSC1_IFLG0	T8OSC1 CH.0 Timer Interrupt Flag Register	Indicates/resets interrupt occurrence status.
	0x4a05–0x4aff	–	–	Reserved
8-bit OSC1 timer CH.1 (8-bit device)	0x4b00	T8OSC1_CTL1	T8OSC1 CH.1 Timer Control Register	Sets the timer mode and starts/stops the timer.
	0x4b01	T8OSC1_CNT1	T8OSC1 CH.1 Timer Counter Data Register	Counter data
	0x4b02	T8OSC1_CMP1	T8OSC1 CH.1 Timer Compare Data Register	Sets compare data.
	0x4b03	T8OSC1_IMSK1	T8OSC1 CH.1 Timer Interrupt Mask Register	Enables/disables interrupt.
	0x4b04	T8OSC1_IFLG1	T8OSC1 CH.1 Timer Interrupt Flag Register	Indicates/resets interrupt occurrence status.
	0x4b05–0x4bff	–	–	Reserved
SRAM controller (16-bit device)	0x5000–0x5017	–	–	Reserved
	0x5018	RTC_WAIT	RTC Wait Control Register	Sets up RTC access cycle.
	0x501a–0x50ff	–	–	Reserved
Multi-function timer (16-bit device)	0x5200	MFT_TC	MFT Counter Data Register	Counter data
	0x5202	MFT_PRD	MFT Period Data Register	Sets period data.
	0x5204	MFT_CMP	MFT Compare Data Register	Sets compare data.
	0x5206	MFT_CTL	MFT Control Register	Sets the timer mode and starts/stops the timer.
	0x5208–0x521d	–	–	Reserved
	0x521e	MFT_IOCTL	MFT Input/Output Control Register	Controls the clock input/output.
	0x5230	MFT_IE	MFT Interrupt Enable Register	Enables the MFT interrupt.
	0x5238	MFT_IF	MFT Interrupt Flag Register	Indicates the MFT interrupt status.
	0x523a–0x527d	–	–	Reserved
	0x527e	MFT_TST	MFT Test Register	Controls the MFT test.
	0x5280–0x52ff	–	–	Reserved
Remote controller (16-bit device)	0x5400	REMC_PSC	REMC Prescaler Control Register	Sets up the REMC prescaler.
	0x5404	REMC_CFG	REMC Configuration Register	Sets the REMC modes and controls the REMC interrupt.
	0x5406	–	–	Reserved
	0x5408	REMC_CTL	REMC Control Register	Starts/stops transmission.
	0x540a	–	–	Reserved
	0x540c	REMC_CARL	REMC Carrier Load Register	Configures the carrier signal.
	0x540e	REMC_ENVL	REMC Envelope Load Register	Configures the envelope pulse width.
	0x5410	REMC_ENVC	REMC Envelope Capture Register	Input envelope pulse width
	0x5412–0x54ff	–	–	Reserved
A/D converter (16-bit device)	0x5500–0x551f	–	–	Reserved
	0x5520	AD_CLKCTL	A/D Clock Control Register	Controls A/D converter clock.
	0x5522–0x553f	–	–	Reserved
	0x5540	AD_DAT	A/D Conversion Result Register	A/D converted data
	0x5542	AD_TRIG_CH	A/D Trigger/Channel Select Register	Sets start/end channels and conversion mode.
	0x5544	AD_CTL	A/D Control/Status Register	Controls A/D converter and indicates conversion status.
	0x5546	AD_CH_STAT	A/D Channel Status Flag Register	Indicates overwrite error and conversion complete status.
	0x5548	AD_CH0_BUF	A/D CH.0 Conversion Result Buffer Register	A/D CH.0 converted data
	0x554a	AD_CH1_BUF	A/D CH.1 Conversion Result Buffer Register	A/D CH.1 converted data
	0x554c	AD_CH2_BUF	A/D CH.2 Conversion Result Buffer Register	A/D CH.2 converted data
	0x554e	AD_CH3_BUF	A/D CH.3 Conversion Result Buffer Register	A/D CH.3 converted data
	0x5550–0x5557	–	–	Reserved
	0x5558	AD_UPPER	A/D Upper Limit Value Register	Specifies A/D conversion upper limit value.
	0x555a	AD_LOWER	A/D Lower Limit Value Register	Specifies A/D conversion lower limit value.
	0x555c	AD_INTMASK	A/D Conversion Complete Interrupt Mask Register	Masks A/D conversion complete interrupt.
	0x555e	AD_ADVMODE	A/D Converter Mode Select/Internal Status Register	Selects A/D operating mode and indicates internal status and internal counter value.

Peripheral	Address	Register name		Function
Watchdog timer (16-bit device)	0x5600–0x565f	–	–	Reserved
	0x5660	WD_WP	WDT Write Protect Register	Enables WDT control registers for writing.
	0x5662	WD_EN	WDT Enable and Setup Register	Configures and starts watchdog timer.
	0x5664	WD_CMP_L	WDT Comparison Data L Register	Comparison data
	0x5666	WD_CMP_H	WDT Comparison Data H Register	
	0x5668	WD_CNT_L	WDT Count Data L Register	Watchdog timer counter data
	0x566a	WD_CNT_H	WDT Count Data H Register	
	0x566c	WD_CTL	WDT Control Register	Resets watchdog timer.
	0x566e–0x56ff	–	–	Reserved
Extended SPI (16-bit device)	0x5700	SPI_ST1	SPI CH.1 Status Register	Indicates transfer and buffer statuses.
	0x5702	SPI_TXD1	SPI CH.1 Transmit Data Register	Transmit data
	0x5704	SPI_RXD1	SPI CH.1 Receive Data Register	Receive data
	0x5706	SPI_CTL1	SPI CH.1 Control Register	Sets the SPI CH.1 mode and enables data transfer.
	0x5708	SPI_CLK1	SPI CH.1 Clock Control Register	Sets up the SPI clock.
	0x570a–0x57ff	–	–	Reserved
ROM controller (16-bit device)	0x5800–0x5803	–	–	Reserved
	0x5804	ROMC_WAIT	ROMC Wait Register	Sets the wait cycle for ROM read.
	0x5806–0x580f	–	–	Reserved
	0x5810	ROMC_PRT	ROMC Protect Register	Enables ROMC registers for writing.
	0x5812–0x5813	–	–	Reserved
	0x5814	TTBR_LOW	Trap Table Base Register 0	Sets the vector table address.
	0x5816	TTBR_HIGH	Trap Table Base Register 1	
	0x5818–0x58ff	–	–	Reserved

Note: Do not access the “Reserved” address in the table above and unused areas in the peripheral area that are not described in the table from the application program.

I.6.5 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 I.6.5.1 I/O Map (S1C17 Core I/O Area)

Peripheral	Address	Register name		Function
S1C17 Core I/O	0xffff90	DBRAM	Debug RAM Base Register	Indicates the debug RAM base address.

See Section I.5.5.2, “Work Area for Debugging,” for DBRAM.

I.7 Electrical Characteristics

I.7.1 Absolute Maximum Rating

(V_{SS} = 0V)

Item	Symbol	Condition	Rated value	Unit
Core power source voltage	LV_{DD}		-0.3 to 2.5	V
I/O power source voltage	HV_{DD}		-0.3 to 4.0	V
Analog power supply voltage	AV_{DD}		-0.3 to 4.0	V
Input voltage	V_I		-0.3 to HV_{DD} + 0.5	V
Analog input voltage	AV_{IN}		-0.3 to AV_{DD} + 0.3	V
High level output current	I_{OH}	1 pin	-10	mA
		Total of all pins	-40	mA
Low level output current	I_{OL}	1 pin	10	mA
		Total of all pins	40	mA
Storage temperature	Tstg		-65 to 150	°C

I.7.2 Recommended Operating Conditions

(Ta = -40 to 85°C)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Core power source voltage	LV_{DD}		1.65	1.80	1.95	V
I/O power source voltage	HV_{DD}		1.65	–	3.60	V
Analog power supply voltage *1	AV_{DD}	P0*x* = analog inputs	2.70	–	3.60	V
		P0*x* = digital inputs	1.65	–	3.60	V
Input voltage	HV_I		V_{SS}	–	HV_{DD}	V
	LV_I		V_{SS}	–	LV_{DD}	V
Analog input voltage	AV_{IN}		V_{SS}	–	AV_{DD}	V
CPU operating frequency	f_{CPU}		–	–	20	MHz
Internal bus operating frequency	f_{BUS}		–	–	20	MHz
OSC3 oscillation frequency	f_{OSC3}		1	–	20	MHz
OSC3 external input clock frequency	f_{ECLK3}		–	–	20	MHz
OSC1 oscillation frequency	f_{OSC1}		–	32.768	–	kHz
OSC1 external input clock frequency	f_{ECLK1}		–	32.768	–	kHz
Operating temperature	Ta		-40	25	85	°C
Input rise time (normal input)	t_{ri}		–	–	50	ns
Input fall time (normal input)	t_{fi}		–	–	50	ns
Input rise time (Schmitt input)	t_{ri}		–	–	5	ms
Input fall time (Schmitt input)	t_{fi}		–	–	5	ms

*1) The AV_{DD} voltage range can be extended into 1.65 to 3.60 V only when the ADC is not used and the P0*x* pins are used as digital signal input pins, not analog input pins. However, the high and low level input voltages of the digital signals must be AV_{DD} and GND, respectively.

I.7.3 DC Characteristics

Unless otherwise specified: LV_{DD} = HV_{DD} = 1.8V±0.15V, V_{SS} = 0V, Ta = -40 to 85°C

Item	Symbol	Condition		Min.	Typ.	Max.	Unit
Input leakage current	I_{LI}			-5	–	5	μA
Off-state leakage current	I_{OZ}			-5	–	5	μA
High level output current	I_{OH1}	Type 1	V_{OH} = HV_{DD}-0.4V HV_{DD} = Min.	-1	–	–	mA
	I_{OH2}	Type 2		-2	–	–	mA
	I_{OH3}	Type 3		-4	–	–	mA
	I_{OH4}	Type 4		-6	–	–	mA
Low level output current	I_{OL1}	Type 1	V_{OL} = 0.4V HV_{DD} = Min.	1	–	–	mA
	I_{OL2}	Type 2		2	–	–	mA
	I_{OL3}	Type 3		4	–	–	mA
	I_{OL4}	Type 4		6	–	–	mA
High level input voltage	V_{IH}	LVCMOS level, HV_{DD} = Max.		1.27	–	–	V
Low level input voltage	V_{IL}	LVCMOS level, HV_{DD} = Min.		–	–	0.57	V
Positive trigger input voltage	V_{T1+}	LVCMOS Schmitt		0.6	–	1.4	V
Negative trigger input voltage	V_{T1-}	LVCMOS Schmitt		0.3	–	1.1	V
Hysteresis voltage	ΔV	LVCMOS Schmitt		0.02	–	–	V
Pull-up resistor	R_{PU}	Type 1	Vin = 0V	48	120	300	kΩ
		Type 2		96	240	600	kΩ
Pull-down resistor	R_{PD}	Type 1	Vin = HV_{DD}	48	120	300	kΩ
		Type 2		96	240	600	kΩ
High level bus hold current	I_{BHH}	HV_{DD} = Min., Vin = 1.27V		–	–	-2	μA
Low level bus hold current	I_{BHL}	HV_{DD} = Min., Vin = 0.57V		–	–	2	μA
High level bus drive current	I_{BHHO}	HV_{DD} = Max., VI = 0.57V		-100	–	–	μA
Low level bus drive current	I_{BHLO}	HV_{DD} = Max., VI = 1.27V		100	–	–	μA
Input pin capacitance	C_I	f = 1MHz, HV_{DD} = 0V		–	–	8	pF
Output pin capacitance	C_O	f = 1MHz, HV_{DD} = 0V		–	–	8	pF
I/O pin capacitance	C_{IO}	f = 1MHz, HV_{DD} = 0V		–	–	8	pF

Unless otherwise specified: LV_{DD} = 1.8V±0.15V, HV_{DD} = 2.7 to 3.6V, V_{SS} = 0V, Ta = -40 to 85°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Input leakage current	I_{LI}		-5	–	5	μA
Off-state leakage current	I_{OZ}		-5	–	5	μA
High level output voltage	V_{OH}	I_{OH} = -1.7mA (2mA type), I_{OH} = -3.5mA (4mA type), V_{DD} = Min.	V_{DD} - 0.4	–	–	V
Low level output voltage	V_{OL}	I_{OL} = 1.7mA (2mA type), I_{OL} = 3.5mA (4mA type), V_{DD} = Min.	–	–	0.4	V
High level input voltage	V_{IH}	LVTTL level, HV_{DD} = Max.	2.0	–	HV_{DD}	V
Low level input voltage	V_{IL}	LVTTL level, HV_{DD} = Min.	V_{SS}	–	0.8	V
Positive trigger input voltage	V_{T+}	LVCMOS Schmitt	1.2	–	2.7	V
Negative trigger input voltage	V_{T-}	LVCMOS Schmitt	0.5	–	1.8	V
Hysteresis voltage	V_H	LVCMOS Schmitt	0.2	–	–	V
Pull-up resistor	R_{PU}	100kΩ type, V_I = 0V	50	100	288	kΩ
		50kΩ type, V_I = 0V	25	50	144	kΩ
Pull-down resistor	R_{PD}	120kΩ type, V_I = 0V	60	120	346	kΩ
		50kΩ type, V_I = 0V	25	50	144	kΩ
High level bus hold current	I_{BHH}	HV_{DD} = Min., V_I = 1.9V	–	–	-20	μA
Low level bus hold current	I_{BHL}	HV_{DD} = Min., V_I = 0.8V	–	–	17	μA
High level bus drive current	I_{BHHO}	HV_{DD} = Max., V_I = 0.8V	-350	–	–	μA
Low level bus drive current	I_{BHLO}	HV_{DD} = Max., V_I = 1.9V	300	–	–	μA
Input pin capacitance	C_I	f = 1MHz, HV_{DD} = 0V	–	–	8	pF
Output pin capacitance	C_O	f = 1MHz, HV_{DD} = 0V	–	–	8	pF
I/O pin capacitance	C_{IO}	f = 1MHz, HV_{DD} = 0V	–	–	8	pF

Note: See Section I.3.4, “Input/Output Cells and Input/Output Characteristics,” for pin characteristics.

I.7.4 Current Consumption

Unless otherwise specified: LV_{DD} = HV_{DD} = 1.8V, AV_{DD} = 3.3V, V_{SS} = 0V, Ta = 25°C
Peripheral modules: stopped

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Current consumption in SLEEP mode	I_{SLP1}	OSC1: Off*3, OSC3: Off, RTC: Stop	–	0.5	–	µA
	I_{SLP2}	OSC1: 32kHz, OSC3: Off, RTC: Run	–	1.8	–	µA
Current consumption in HALT mode	I_{HALT1}	OSC1: 32kHz, OSC3: Off, RTC: Run	–	3.3	–	µA
	I_{HALT2}	OSC1: 32kHz, OSC3: 1MHz, RTC: Run	–	170	–	µA
	I_{HALT3}	OSC1: 32kHz, OSC3: 4MHz, RTC: Run	–	290	–	µA
	I_{HALT4}	OSC1: 32kHz, OSC3: 8MHz, RTC: Run	–	530	–	µA
	I_{HALT5}	OSC1: 32kHz, OSC3: 20MHz, RTC: Run	–	1.3	–	mA
Current consumption during execution*1	I_{EXE1}	OSC1: 32kHz, OSC3: Off, RTC: Run	–	8.0	–	µA
	I_{EXE2}	OSC1: 32kHz, OSC3: 1MHz, RTC: Run	–	310	–	µA
	I_{EXE3}	OSC1: 32kHz, OSC3: 4MHz, RTC: Run	–	840	–	µA
	I_{EXE4}	OSC1: 32kHz, OSC3: 8MHz, RTC: Run	–	1.7	–	mA
	I_{EXE5}	OSC1: 32kHz, OSC3: 20MHz, RTC: Run	–	3.8	–	mA
ADC operating current*2	I_{ADC1}	When the ADC is enabled	–	260	–	µA

*1) The current consumption during execution in the above table indicates the value when a test program that consists of 60.5% ALU instructions, 17% branch instructions, 12% memory read instructions and 10.5% memory write instructions is being executed in the ROM.

*2) AV_{DD} power current consumption

*3) When no resonator is connected (The OSC1 oscillator circuit cannot be turned off.)

I.7.5 A/D Converter Characteristics

Unless otherwise specified: $HV_{DD} = AV_{DD}$ = 2.7 to 3.6V, LV_{DD} = 1.65 to 1.95V, V_{SS} = 0V, Ta = -40 to 85°C, ST[1:0] = 11

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Resolution	–		–	10	–	bit
Conversion time *1	–		10	–	1250	µs
Zero scale error	E_{ZS}		-2	–	2	LSB
Full scale error	E_{FS}		-2	–	2	LSB
Integral linearity error	E_L		-3	–	3	LSB
Differential linearity error	E_D		-3	–	3	LSB
Permissible signal source impedance	–		–	–	5	kΩ
Analog input capacitance	–		–	–	45	pF

*1) Indicates the minimum value when A/D clock = 2MHz. Indicates the maximum value when A/D clock = 16kHz.

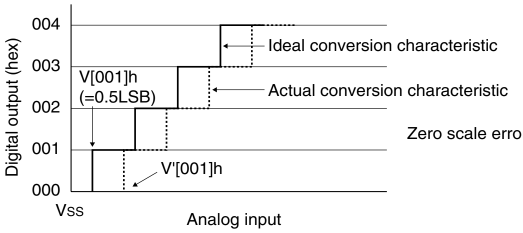
A/D conversion error

V[001]h = Ideal voltage at zero-scale point (=0.5LSB)
V'[001]h = Actual voltage at zero-scale point
V[3FF]h = Ideal voltage at full-scale point (=1022.5LSB)
V'[3FF]h = Actual voltage at full-scale point

$$1LSB = \frac{AV_{DD} - V_{SS}}{2^{10} - 1}$$

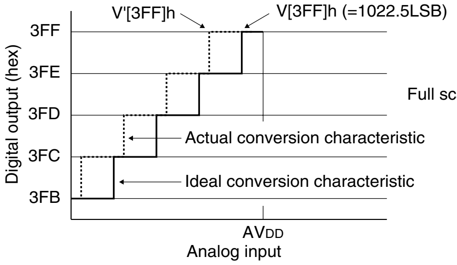
$$1LSB' = \frac{V'[3FF]h - V'[001]h}{2^{10} - 2}$$

■ Zero scale error



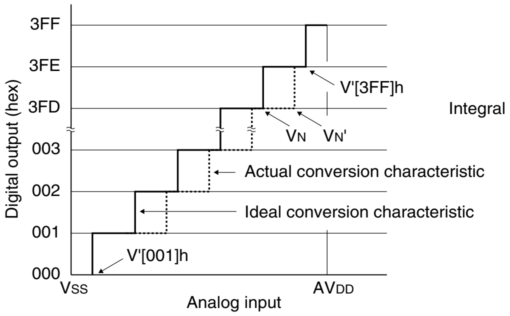
$$\text{Zero scale error } E_{ZS} = \frac{(V'[001]h - 0.5LSB') - (V[001]h - 0.5LSB)}{1LSB} \text{ [LSB]}$$

■ Full scale error



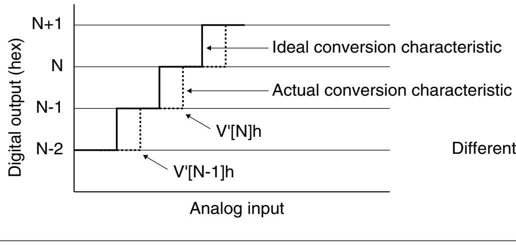
$$\text{Full scale error } E_{FS} = \frac{(V'[3FF]h + 0.5LSB') - (V[3FF]h + 0.5LSB)}{1LSB} \text{ [LSB]}$$

■ Integral linearity error



$$\text{Integral linearity error } E_L = \frac{V_N' - V_N}{1LSB'} \text{ [LSB]}$$

■ Differential linearity error


N+1
N
N-1
N-2
Digital output (hex)
Ideal conversion characteristic
Actual conversion characteristic
V'[N]h
V'[N-1]h
Analog input

$$\text{Differential linearity error } E_D = \frac{V'[N]h - V'[N-1]h}{1LSB'} - 1 \text{ [LSB]}$$

I.7.6 Oscillation Characteristics

Oscillation characteristics change depending on conditions such as components used (resonator, R_f, R_d, C_G, C_D) and board pattern. Use the following characteristics as reference values. In particular, when a ceramic or crystal resonator is used, evaluate the components adequately under real operating conditions by mounting them on the board before the external register (R_f, R_d) and capacitor (C_G, C_D) values are finally decided.

OSC1 crystal oscillation

Unless otherwise specified: LV_{DD} = 1.8V, V_{SS} = 0V, Ta = 25°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Oscillation start time	t_{STA1}	∗1			1	s

∗1) When the recommended parts shown in Section I.8, "Basic External Wiring Diagram," are used

OSC3 crystal oscillation

Note: A "crystal resonator that uses a fundamental" should be used for the OSC3 crystal oscillation circuit.

Unless otherwise specified: LV_{DD} = 1.8V, V_{SS} = 0V, Ta = 25°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Oscillation start time	t_{STA3}	∗1			10	ms

∗1) When the recommended parts shown in Section I.8, "Basic External Wiring Diagram," are used

OSC3 ceramic oscillation

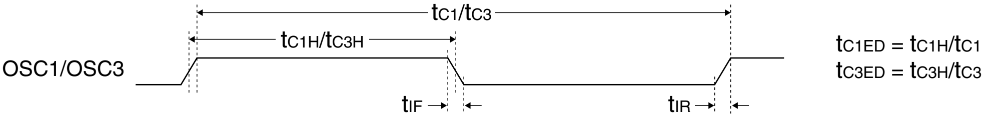
Unless otherwise specified: LV_{DD} = 1.8V, V_{SS} = 0V, Ta = 25°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Oscillation start time	t_{STA3}	∗1			1	ms

∗1) When the recommended parts shown in Section I.8, "Basic External Wiring Diagram," are used

I.7.7 AC Characteristics

I.7.7.1 External Clock Input Characteristics



OSC1 external clock

Unless otherwise specified: $HV_{DD} = AV_{DD}$ = 2.7 to 3.6V, LV_{DD} = 1.65 to 1.95V, V_{SS} = 0V, Ta = -40 to 85°C

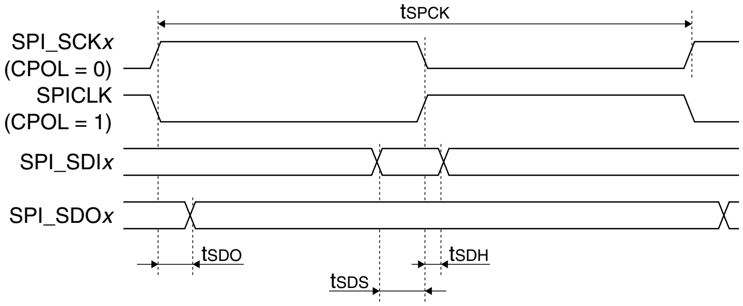
Item	Symbol	Min.	Typ.	Max.	Unit
OSC1 external clock cycle time	t_{C1}		30.51		μs
OSC1 external clock input duty	t_{C1ED}	45		55	%
OSC1 external clock input rise time	t_{IF}			200	ns
OSC1 external clock input fall time	t_{IR}			200	ns

OSC3 external clock

Unless otherwise specified: $HV_{DD} = AV_{DD}$ = 2.7 to 3.6V, LV_{DD} = 1.65 to 1.95V, V_{SS} = 0V, Ta = -40 to 85°C

Item	Symbol	Min.	Typ.	Max.	Unit
OSC3 external clock cycle time	t_{C3}	50		1000	ns
OSC3 external clock input duty	t_{C3ED}	45		55	%
OSC3 external clock input rise time	t_{IF}			10	ns
OSC3 external clock input fall time	t_{IR}			10	ns

I.7.7.2 SPI AC Characteristics



Master mode

Unless otherwise specified: $HV_{DD} = AV_{DD}$ = 2.7 to 3.6V, LV_{DD} = 1.65 to 1.95V, V_{SS} = 0V, Ta = -40 to 85°C

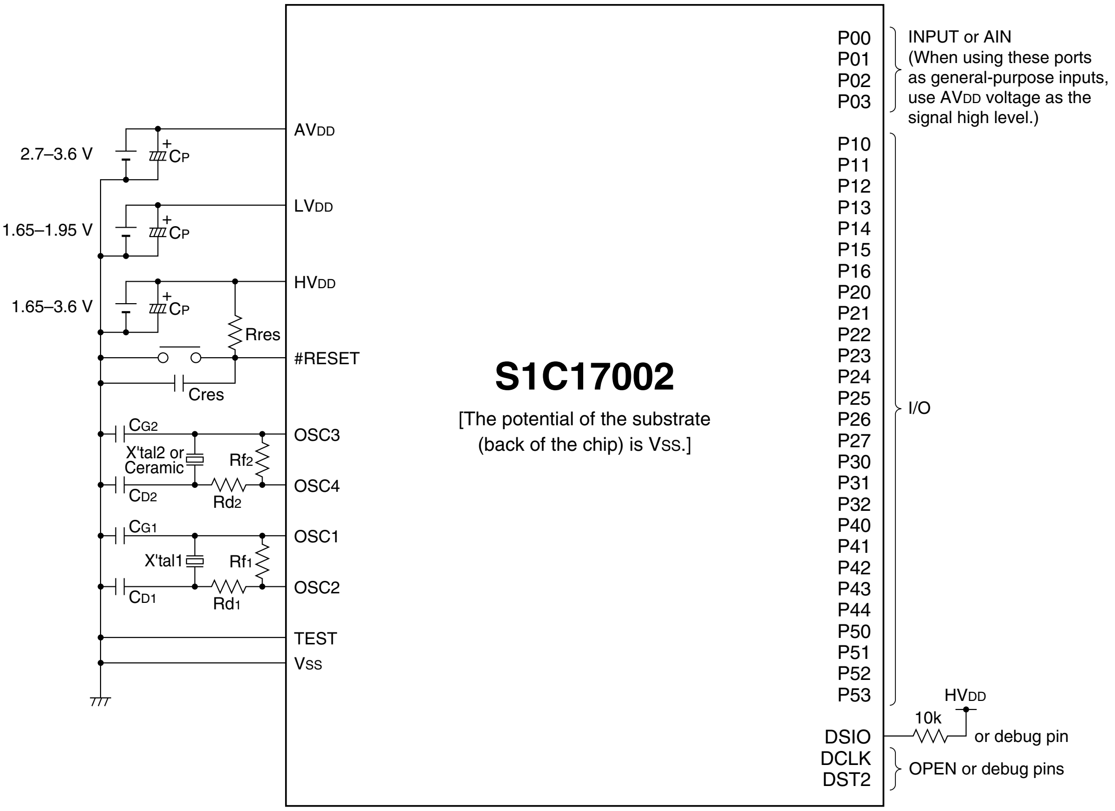
Item	Symbol	Min.	Typ.	Max.	Unit
SPI_SCK*x* cycle time	t_{SPCK}	500			ns
SPI_SDI*x* setup time	t_{SDS}	70			ns
SPI_SDI*x* hold time	t_{SDH}	10			ns
SPI_SDO*x* output delay time	t_{SDO}			20	ns

Slave mode

Unless otherwise specified: $HV_{DD} = AV_{DD}$ = 2.7 to 3.6V, LV_{DD} = 1.65 to 1.95V, V_{SS} = 0V, Ta = -40 to 85°C

Item	Symbol	Min.	Typ.	Max.	Unit
SPI_SCK*x* cycle time	t_{SPCK}	500			ns
SPI_SDI*x* setup time	t_{SDS}	10			ns
SPI_SDI*x* hold time	t_{SDH}	10			ns
SPI_SDO*x* output delay time	t_{SDO}			80	ns

I.8 Basic External Wiring Diagram



Recommended values for external parts

External parts for the OSC1 oscillator circuit

Symbol	Resonator	Recommended manufacturer	Frequency [Hz]	Product number	Recommended values				Recommended operating condition
					C_{D1} [pF]	C_{G1} [pF]	R_{f1} [Ω]	R_{d1} [Ω]	Temperature range [°C]
X'tal1	Crystal	Epson Toyocom Corporation	32.768k	MC-146 (C_L = 7.0 pF)	7	7	1M	0	-40 to 85

External parts for the OSC3 oscillator circuit

Symbol	Resonator	Recommended manufacturer	Frequency [Hz]	Product number	Recommended values				Recommended operating condition
					C_{D2} [pF]	C_{G2} [pF]	R_{f2} [Ω]	R_{d2} [Ω]	Temperature range [°C]
X'tal2	Crystal	Epson Toyocom Corporation	4M	MA-406 (C_L = 16 pF)	27	27	1M	0	-40 to 85
			8M	MA-406 (C_L = 7.0 pF)	18	18	1M	0	-40 to 85
			16M	FA-238 (C_L = 7.0 pF)	4	4	1M	0	-40 to 85
			20M	FA-238 (C_L = 7.0 pF)	4	4	1M	0	-40 to 85
Ceramic	Ceramic	Murata Manufacturing Co., Ltd.	4M	CSTCR4M00G55	(39)	(39)	1M	100	-20 to 80
			8M	CSTCE8M00G55	(33)	(33)	1M	0	-20 to 80
			12M	CSTCE12M0G55	(33)	(33)	1M	0	-20 to 80
			16M	CSTCE16M0V53	(15)	(15)	1M	0	-20 to 80
			20M	CSTCE20M0V53	(15)	(15)	1M	0	-20 to 80
			4M	CSTCR4M00G55Z	(39)	(39)	1M	100	-40 to 85
			8M	CSTCE8M00G55Z	(33)	(33)	1M	0	-40 to 85
			12M	CSTCE12M0G55Z	(33)	(33)	1M	0	-40 to 85
			16M	CSTCE16M0V53Z	(15)	(15)	1M	0	-40 to 85
			20M	CSTCE20M0V53Z	(15)	(15)	1M	0	-40 to 85

The C_{D2} and C_{G2} values enclosed with ( ) are the built-in capacitances of the resonator.

Other

Symbol	Name	Recommended value
CP	Capacitor for power supply	3.3 µF
Cres	Capacitor for #RESET pin	0.47 µF
Rres	Resistor for #RESET pin	10 kΩ

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.

I.9 Precautions on Mounting

The following shows the precautions when designing the board and mounting the IC.

Oscillation Circuit

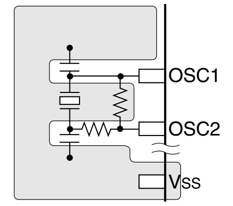
- Oscillation characteristics change depending on conditions such as components used (resonator, Rf, CG, CD) and board pattern. In particular, when a ceramic or crystal resonator is used, evaluate the components adequately under real operating conditions by mounting them on the board before the external register (Rf) and capacitor (CG, CD) values are finally decided.
- Disturbances of the oscillation clock due to noise may cause a malfunction. To prevent this, the following points should be taken into consideration. In particular, the latest devices are more sensitive to noise, as they are more finely processed.
  The measures against noise for the OSC2 pin, and the components and lines connected to this pin is most essential, and similar measures must also be taken for the OSC1 pin. The measures for the OSC1 and OSC2 pins are described below.
  We recommend taking measures similar to those for the high-speed oscillation system, including the OSC3 and OSC4 pins and the components and lines connected to these pins.

  (1) Components that are connected to the OSC1 (OSC3) and OSC2 (OSC4) pins, such as resonators, resistors, and capacitors, should be connected in the shortest line.

  (2) Whenever possible, configure digital signal lines with at least three millimeters clearance from the OSC1 (OSC3) and OSC2 (OSC4) pins and the components and lines connected to these pins. In particular, signals that are switched frequently must not be placed near these pins, components, and lines. The same applies to all layers on the multi-layered board as the distance between the layers is around 0.1 to 0.2 mm.
  Furthermore, do not configure digital signal lines in parallel with these components and lines when arranging them on the same or another layer of the board. Such an arrangement is strictly prohibited, even with clearance of three millimeters or more. Also, avoid arranging digital signal lines across these components and signal lines.

  (3) Shield the OSC1 (OSC3) and OSC2 (OSC4) pins and lines connected to those pins as well as the adjacent layers of the board using Vss.
  As shown in the figure on the right, shield the wired layers as much as possible.
  Whenever possible, make the whole adjacent layers the ground layers, or ensure there is adequate shielding to a radius of five millimeters around the above pins and lines. As described in (2), do not configure digital signal lines in parallel with components and lines even if such precautionary measures are taken, and avoid configuring signal lines that are switched frequently across components and lines on other layers.

Sample Vss pattern
OSC1 and OSC2



  (4) When an external clock is supplied to the OSC1 or OSC3 pin, the clock source should be connected to the OSC1 or OSC3 pin in the shortest line. Furthermore, do not connect anything else to the OSC2 or OSC4 pin.

  (5) After taking the above precautions, check the output clock waveform while operating the actual application program in the actual device.
  To do this, measure the output of the CMU_CLK pins with an oscilloscope.
  Check the waveform quality at the OSC3 output clock by measuring the CMU_CLK output. Ensure that the frequencies are as designed and that there is no noise or jitters.
  Check the waveform quality at the OSC1 clock by measuring the CMU_CLK output (after switching the system clock source to OSC1). Scale up the ranges around the rising and falling edges of the clock pulse to ensure that there is no noise, such as clock and spike, in the 100 ns ranges.

If conditions (1) to (3) are not satisfied, the OSC3 output may be jittery and the OSC1 output may be noisy. When the OSC3 output is jittery, the operating frequency will be lowered. When the OSC1 output is noisy, operation of the RTC using the OSC1 clock and the CPU core after the system clock is switched to OSC1 will be unstable.

Reset Circuit

- The power-on reset signal which is input to the #RESET pin changes depending on conditions (power rise time, components used, board pattern, etc.). Decide the time constant of the capacitor and resistor after enough tests have been completed with the application product.
- In order to prevent any occurrences of unnecessary resetting caused by noise during operating, components such as capacitors and resistors should be connected to the #RESET pin in the shortest line.

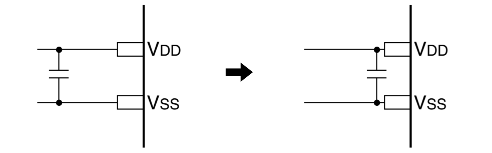
Power Supply Circuit

- Sudden power supply variation due to noise may cause malfunction. Consider the following points to prevent this:

(1) The power supply should be connected to the LVDD, HVDD, VSS, and AVDD pins with patterns as short and large as possible. In particular, the power supply for AVDD affects A/D conversion precision.

(2) When connecting between the VDD and VSS pins with a bypass capacitor, the pins should be connected as short as possible.

Bypass capacitor connection example



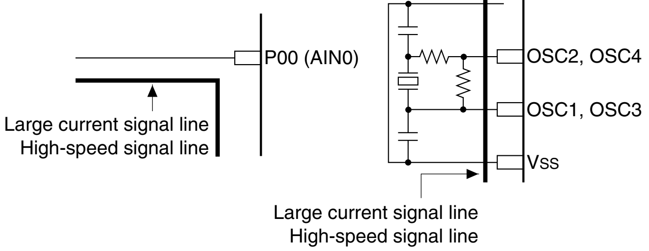
A/D Converter

- When the A/D converter is not used, the power supply pin AVDD for the analog system should be connected to VDD.

Arrangement of Signal Lines

- In order to prevent generation of electromagnetic induction noise caused by mutual inductance, do not arrange a large current signal line near the circuits that are sensitive to noise such as the oscillation unit and analog input unit.
- When a signal line is parallel with a high-speed line in long distance or intersects a high-speed line, noise may generated by mutual interference between the signals and it may cause a malfunction. Do not arrange a high-speed signal line especially near circuits that are sensitive to noise such as the oscillation unit and analog input unit.

Prohibited pattern



Noise-Induced Erratic Operations

If erratic IC operations appear to be attributable to noise, consider the following four points.

(1) TEST pin
If this pin is exposed to high-level noise, the entire IC enters test mode or a high-impedance state and becomes inoperable. In such cases, the IC will not be restored, even when the pin is returned to a low level. Therefore, always make sure the TEST pin is connected to GND on the circuit board. Although the IC contains internal pull-down resistors, it is susceptible to noise because these resistors are high impedance (approximately 50 to 100 kΩ).

(2) DSIO pin
Exposure of this pin to low-level noise causes the IC to enter debug mode. In debug mode, the clock is output from the DCLK pin and the DST2 pin is high, indicating that the IC is in debug mode.
In product versions, it is recommended that the DSIO pin be pulled high by connecting it directly to HV_{DD} or through a resistor of 10 kΩ or less.
Although the IC contains internal pull-up resistors, it is susceptible to noise because these resistors are high impedance (approximately 50 to 100 kΩ).

(3) #RESET pin
Low-level noise on this pin resets the IC. However, the IC may not always be reset normally, depending on the input waveform.
Due to circuit design, this situation tends to occur when the reset input is in the high state, with high impedance.

(4) LV_{DD}, V_{SS}, and HV_{DD} power supplies
If noise lower than the rated voltage enters one of these power-supply lines, the IC may operate erratically.
Take corrective measures in board design; for example, by using solid patterns for power supply lines, adding decoupling capacitors to eliminate noise, or incorporating surge/noise counteracting devices into the power supply lines.

To confirm the above, use an oscilloscope capable of observing higher-frequency waveforms of 200 MHz. The generation of fast noise may not be observed with a low-frequency oscilloscope.

If potential noise-induced erratic operations are detected through waveform observations using an oscilloscope, connect the suspected pin to the GND or power supply with low impedance (1 kΩ or less) and check once again. If erratic operations are no longer detected or occur at reduced frequency, or if different symptoms of erratic operations are observed, said pin may with reasonably certainty be considered to be the source of the erratic operations.

The TEST, DSIO, and #RESET input circuits described above are designed to detect the edges of the input signal, so that even spike noise may result in erratic operations. Among the digital signal circuits, these pins are most susceptible to noise.
In the design of the circuit board, take the following two points into consideration to protect the signal from noise.

(A) The most important measure is to lower the signal-driving impedance, as described in each item above. Connect pins to the power supply or GND, with impedance of 1 kΩ or less, preferably 0 Ω. In addition, limit the length of the connected signal lines to approximately 5 cm.

(B) Parallel routing of said signal lines with other digital lines on the board is undesirable, since the noise generated when the signal changes from high to low or vice versa may adversely affect signals. The signal may be subject to the most noise when signal lines are laid between multiple signal lines whose states change simultaneously. Take corrective measures by shortening the parallel distance (to several cm) or separating signal lines (2 mm or more).

Other

The 0.18 µm fine-pattern process is employed to manufacture this series of products.

Although the product is designed to meet EIAJ and MIL standards regarding basic IC reliability, please pay careful attention to the following points when actually mounting the chip on a board.

Since all the oscillator input/output pins are constructed to use the internal 0.18 µm transistors directly, the pins are susceptible to mechanical damage during the board-mounting process. Moreover, the pins may also be susceptible to electrical damage caused by such disturbances (listed below) whose electrical strength, varying gradually with time, could exceed the absolute maximum rated voltage (2.5 V) of the IC:

(1) Electromagnetic induction noise from the utility power supply in the reflow process during board-mounting, rework process after board-mounting, or individual characteristic evaluation (experimental confirmation), and

(2) Electromagnetic induction noise from the tip of a soldering iron

Especially when using a soldering iron, make sure that the IC GND and soldering iron GND are at the same potential before soldering.

S1C17002 Technical Manual

II S1C17002 CLOCK SYSTEM

II.1 Clock System Diagram

Figure II.1.1 shows the clock system in the S1C17002.

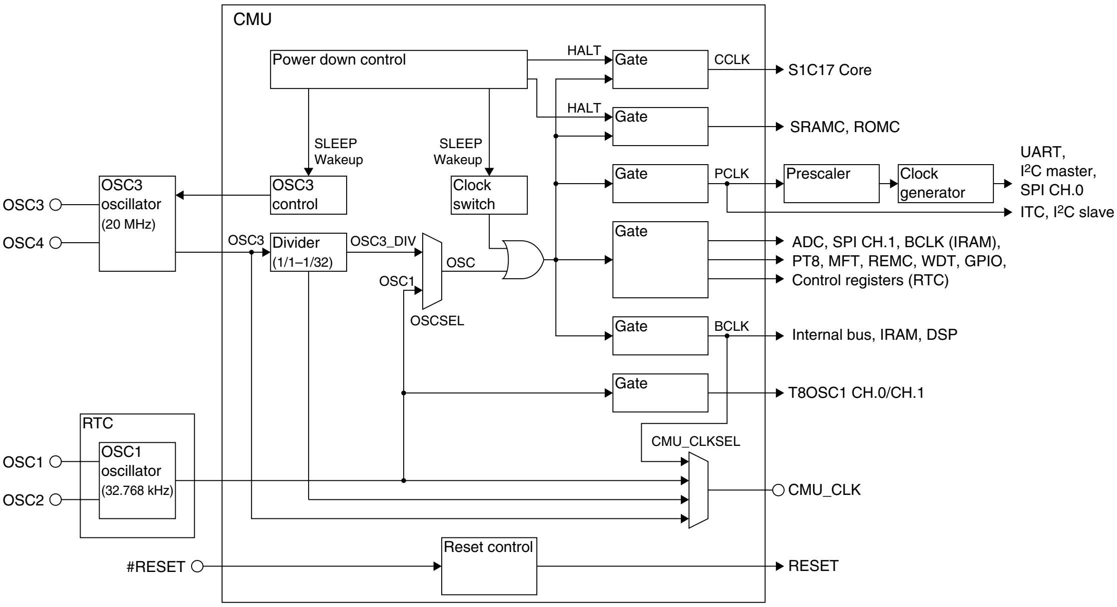


Figure II.1.1 Clock System Diagram

The S1C17002 controls the core and bus operating clocks using the clock management unit (CMU). The peripheral clocks are generated by the prescaler and clock generator.

Current consumption can be reduced by controlling the clocks according to the processing requirements as well as by using the standby mode. For methods to reduce current consumption, see Appendix C, “Power Saving.”

THIS PAGE IS BLANK.

II.2 Clock Management Unit (CMU)

II.2.1 Overview of the CMU

The Clock Management Unit (CMU) controls the operating clock supplied to each functional module. The main functions of the CMU are outlined below.

- Controls the reset input
- Selects the system clock source (OSC3 or OSC1)
- Controls the OSC3 oscillator circuit
- Clock control corresponding to standby modes (SLEEP and HALT)
- Selects divide ratio of the main system clock
- Selects an external bus clock
- Controls on/off of clock supply for each functional module

Through system clock selection, oscillator circuit, main system clock divide ratio selection and clock on/off control for each functional module, the CMU enables the most suitable operating clock frequency to be selected for the processing involved, as well as to turn off unnecessary clock supply, which combined with standby mode, helps to significantly reduce power consumption on the chip.

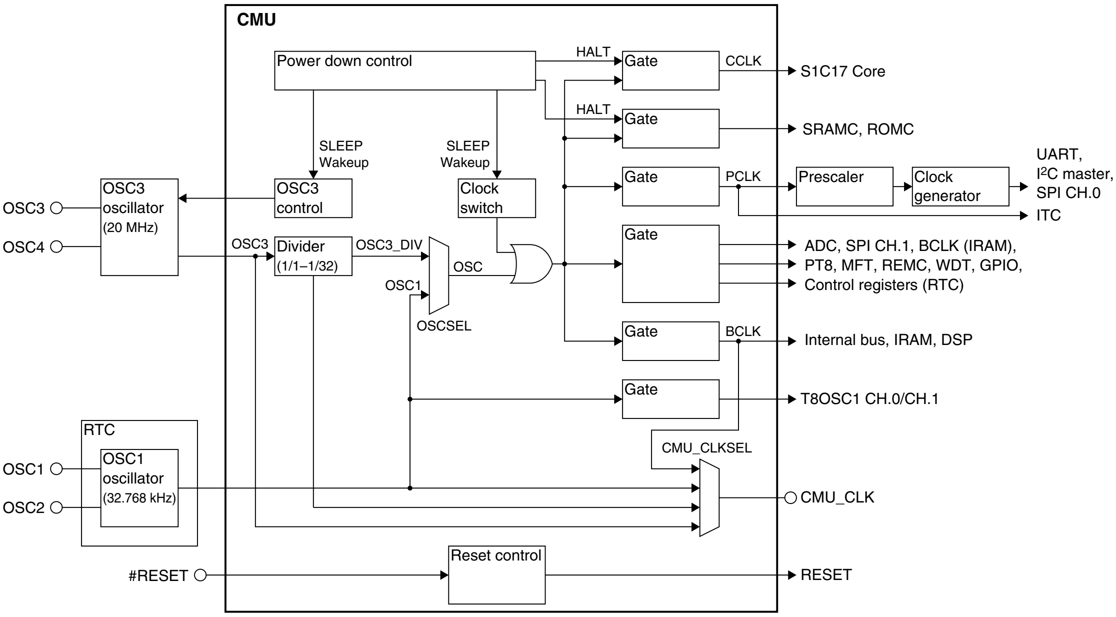


Figure II.2.1.1 CMU Block Diagram

Note: The CMU Control Registers at addresses 0x4900–0x4908 are write-protected. Before the CMU control registers can be rewritten, write protection of these registers must be removed by writing data 0x96 to the CMU Write Protect Register (0x4920). Note that since unnecessary rewrites to addresses 0x4900–0x4908 could lead to erratic system operation, the CMU Write Protect Register (0x4920) should be set to other than 0x96 unless said CMU control registers must be rewritten.

II.2.2 Reset Input and Initial Reset

The CMU also has a function to generate an internal reset signal from external reset input (#RESET).

II.2.2.1 Initial Reset Pin

The #RESET pin is used for initial reset input from outside the IC. Set the #RESET pin to 0 (low) to reset the IC.
The #RESET input signal is sampled with the OSC3 clock. Therefore, the chip cannot be reset when the OSC3 clock is not input or generated. Moreover, to assert the internal reset signal #RESET = 0 must be continuously detected at least three times in this sampling. The #RESET signal should be held low for at least three OSC3 clock cycles to confirm that the chip is reset. Also the internal reset signal is negated when #RESET = 1 (high) is continuously detected three times.
The S1C17002 is reset by the low state (= 0) on the internal reset signal, and starts operating when the reset signal is released back to high (= 1).

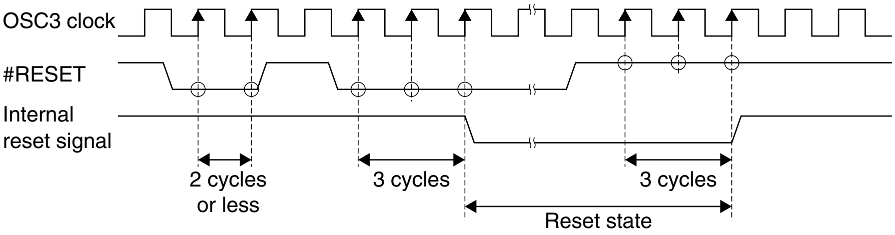


Figure II.2.2.1.1 #RESET Sampling

II.2.2.2 Initial Reset Status

The S1C17 Core and internal peripheral circuits are initialized while the internal reset signal is kept 0. The following shows the internal reset status:

CPU PC:	The reset vector at address 0x20000 is loaded to the PC.
CPU PSR:	All the PSR bits are reset to 0.
Other CPU registers:	All the registers are cleared to 0.
TTBR:	Initialized to 0x20000
CPU operating clock:	The CPU operates with the OSC3 × 1/1 clock.
Oscillator circuit:	The OSC3 oscillator circuit is turned on. The OSC1 oscillator circuit is always on.
Clock supply to peripheral modules:	All clocks are enabled.
I/O pin status:	Initialized (see Section I.3.2, “Pin Functions.”)
Other peripheral modules:	Initialized or undefined (see each I/O map.)

Note: The S1C17002 does not support a hot reset feature that maintains I/O pin status and the TTBR value.

II.2.2.3 Power-on Reset

When turning on the power for the chip, always be sure to reset the chip to ensure that it will start operating normally.

Since the #RESET pin is a gate input, a power-on reset circuit should be configured external to the chip.

Initial reset (#RESET = 0) causes the OSC3 oscillator circuit to start oscillating, and when the reset signal is released back high, the CPU starts operating with the OSC3 clock. The OSC3 oscillator circuit requires a finite time until its oscillation stabilizes after it starts operating. To confirm that the CPU is started, the initial reset can only be deasserted after this oscillation stabilization time elapses.

Note: The oscillation start time of the OSC3 oscillator circuit varies with the device used, board patterns, and operating environment. Therefore, sufficient time should be provided before the reset signal is deasserted.

Power-on sequence

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

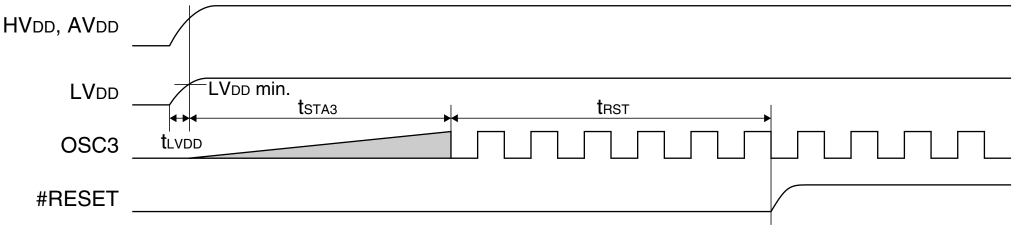


Figure II.2.2.3.1 Power-on Sequence

(1) t_{LVDD}: Elapsed time until the power supply stabilizes after power-on
Supply power in the following sequence.
Power-on: LV_{DD} → HV_{DD} (I/O), AV_{DD} (A/D) → Apply the input signal
or LV_{DD}, HV_{DD} (I/O), AV_{DD} (A/D) → Apply the input signal

(2) t_{STA3}: Time at which OSC3 oscillation starts

(3) t_{RST}: Minimum reset pulse width
Time at which the clock supplied to the chip stabilizes plus at least six clocks; Keep the #RESET signal low.

Make sure #RESET is held low (= 0) for at least 6 clock cycles after the OSC3 clock supplied to the CMU has stabilized.

Note: When the HV_{DD} power is turned on from off status, stable internal circuit statuses cannot be guaranteed due to noise in the power line. Therefore, the circuit statuses must be initialized (reset) after the power is turned on.

II.2.2.4 Precautions to be Taken during Initial Reset

Core CPU

When initially reset, all internal registers of the core CPU are cleared to 0. The Stack Pointer (SP) also becomes 0 when it is initialized upon reset. Note that normal operation of the program cannot be guaranteed if an interrupt occurs before the stack is set up, as the PC or PSR value may be saved to an indeterminate location. To prevent such a problem, set the SP before an interrupt occurs.

Internal RAM

The content of internal RAM becomes undefined when initially reset. Internal RAM must be initialized as required.

OSC3 oscillator circuit

When initially reset, the OSC3 oscillator circuit starts oscillating, and when the reset signal is deasserted, the CPU starts operating with the OSC3 clock. To prevent erratic operation due to an instable clock when the chip is reset at power-on or while the OSC3 oscillator circuit is idle, the reset signal should not be deasserted until after oscillation stabilizes.

OSC1 oscillator circuit

When the chip is reset at power-on, the OSC1 oscillator circuit also starts oscillating. The OSC1 oscillator circuit requires a longer time for oscillation to stabilize than the OSC3 oscillator circuit. (See the electrical characteristics table.) To prevent erratic operation due to an instable clock, the OSC1 clock should not be used until after this stabilization time elapses.

Input/output ports and input/output pins

Initial reset initializes the control and data registers of the input/output ports, therefore, be set up back again in a program.

Other internal peripheral circuits

The control and data registers of other peripheral circuits are initialized or made unstable by initial reset. Therefore, these registers should be set up as required in a program.

For details on how peripheral circuits are initialized by initial reset, see each I/O map or circuit description.

II.2.3 NMI Input

The NMI signal generated by the watchdog timer is input to the CMU, then forwarded to the CPU. For details about NMI exception handling by the CPU, refer to the S1C17 Family S1C17 Core Manual.

Note: NMI cannot be nested. The CPU keeps NMI input masked out until the reti instruction is executed after an NMI exception occurred.

II.2.4 Selecting the System Clock Source

The CMU has the following two clock inputs, one of which can be selected as the source clock (OSC) for the system.

1. OSC3 clock

This clock is generated by the OSC3 oscillator circuit or supplied from an external source through the OSC3 pin. For details about the OSC3 oscillator circuit, see Section II.2.5.1, “OSC3 Oscillator Circuit.”

2. OSC1 clock

This is the source clock (32.768 kHz, typ.) for the Real Time Clock (RTC). When high-speed operation is unnecessary, this low-speed clock may be used to operate the system, thus helping to reduce power consumption on the chip. For details about the OSC1 oscillator circuit, see Section II.2.5.3, “OSC1 Oscillator Circuit.”

The clock source can be selected using OSCSEL (D2/CMU_SYSCLKCTL register).

∗ **OSCSEL**: OSC Clock Selection Bit in the System Clock Control (CMU_SYSCLKCTL) Register (D2/0x4900)

When OSCSEL is set to 0 (default), OSC3 is selected as the system clock source; when OSCSEL is set to 1, OSC1 is selected.

The clock source changed here is not reflected until after the CPU returns from SLEEP mode. Therefore, the slp instruction must be executed once after setting OSCSEL. Although the CPU returns from SLEEP mode to normal operation by an external interrupt from a port, for example, several functions are provided for use in clock source changes, thus automatically returning the CPU from SLEEP mode a certain time after slp instruction execution or leaving the OSC3 oscillator circuit turned on during SLEEP mode. Section II.2.8, “Standby Modes,” describes these methods of control in detail.

Note: When clock sources are changed, the CMU control registers must be set so that the CMU is supplied with a clock from the selected clock source upon returning from SLEEP mode immediately after the change. Otherwise, the chip does not restart after the return from SLEEP mode.

II.2.5 Controlling the Oscillator Circuit

II.2.5.1 OSC3 Oscillator Circuit

The OSC3 oscillator circuit generates the main clock with which to operate the S1C17 Core and internal peripheral circuits.

Input/output pins of the OSC3 oscillator circuit

Table II.2.5.1.1 lists the input/output pins of the OSC3 oscillator circuit.

Table II.2.5.1.1 Input/Output Pins of the OSC3 Oscillator Circuit

Pin name	I/O	Function
OSC3	I	OSC3 oscillator input pin: Crystal/ceramic oscillator or external clock input
OSC4	O	OSC3 oscillator output pin: Crystal/ceramic oscillator (left open when using external clock input)

Structure of the oscillator circuit

The OSC3 oscillator circuit accommodates a crystal/ceramic oscillator and external clock input.
Figure II.2.5.1.1 shows the structure of the OSC3 oscillator circuit.

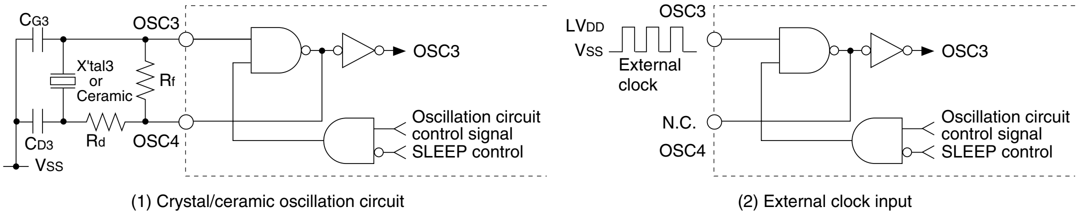


Figure II.2.5.1.1 OSC3 Oscillator Circuit

For use as a crystal or ceramic oscillator circuit, connect a crystal (X'tal3) or ceramic resonator and a feedback resistor (R_f), two capacitors (C_{G3}, C_{D3}) and, if necessary, a drain resistor (R_d) to the OSC3 and OSC4 pins and V_{SS}.

To use an external clock, leave the OSC4 pin open and input a LV_{DD}-level clock (with a 50% duty cycle) to the OSC3 pin.

The range of oscillation frequencies is as follows:

- Crystal/ceramic oscillator: 1 MHz (min.) to 20 MHz (max.)
- External clock input: 20 MHz (max.)

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

Oscillation control

CMU register control bit SOSC3 (D1/CMU_SYSCLKCTL register) is used to control OSC3 oscillation.

∗ **SOSC3**: OSC3 Oscillator On/Off Bit in the System Clock Control (CMU_SYSCLKCTL) Register (D1/0x4900)

Setting this control bit to 0 causes the OSC3 oscillator circuit to stop; setting it to 1 causes the OSC3 oscillator circuit to start oscillating, thereby outputting a clock signal waveform. When initially reset, this bit is set to 1 for enabling OSC3 oscillation.

Note: When the oscillator is made to start oscillating by setting SOSC3 from 0 to 1, a finite time is required until oscillation stabilizes (see “Electrical Characteristics”). To prevent system malfunction, do not use the oscillator-derived clock until this oscillation stabilization time elapses.

II.2.5.2 Setting the OSC3 Divider

An OSC3 divided clock can be used as the system clock when OSC3 is selected as the system clock source. Setting the system clock to the lowest frequency possible according to the processing can reduce current consumption.
The OSC3 divider generates six kinds of clocks from OSC3•1/1 to OSC3•1/32. Select a divided clock from those six clocks using OSC3DIV[2:0] (D[2:0]/CMU_OSC3DIV register).

* **OSC3DIV[2:0]**: OSC3 Clock Divider Selection Bits in the OSC3 Clock Divider (CMU_OSC3DIV) Register (D[2:0]/0x4903)

Table II.2.5.2.1 Selecting an OSC3 Divided Clock

OSC3DIV[2:0]	OSC3_DIV clock
0x7	OSC3•1/1
0x6	OSC3•1/1
0x5	OSC3•1/32
0x4	OSC3•1/16
0x3	OSC3•1/8
0x2	OSC3•1/4
0x1	OSC3•1/2
0x0	OSC3•1/1

(Default: 0x0 = OSC3•1/1)

A divided clock can be selected at any time. However, up to 32 OSC3 clock cycles are required before the clocks are actually changed after altering the register values.

II.2.5.3 OSC1 Oscillator Circuit

The S1C17002 contains an oscillator circuit (OSC1) used to generate a 32.768 kHz (typ.) clock as the clock source for timekeeping operation of the RTC and counting operation of the T8OSC1 CH.0/CH.1.
The OSC1 clock can also be used as a power-saving operating clock for the core system or peripheral circuits.

Input/output pins of the OSC1 oscillator circuit

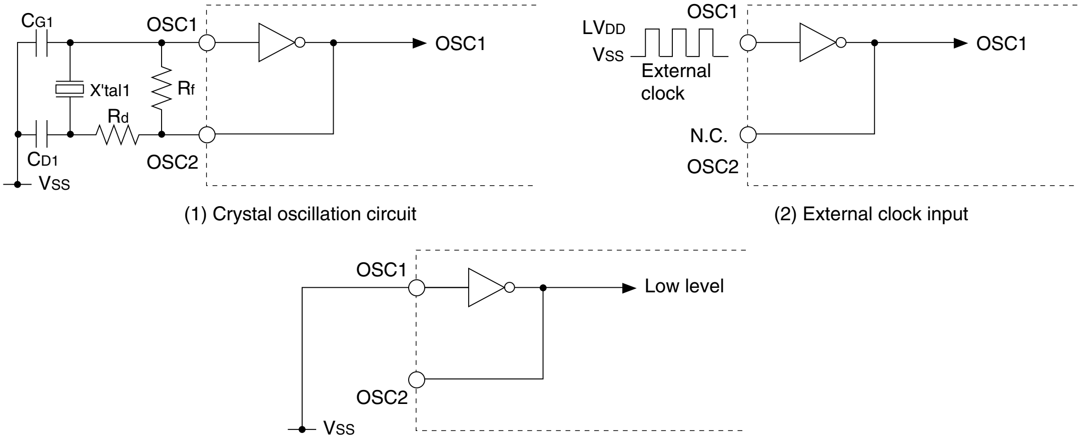
Table II.2.5.3.1 lists the input/output pins of the OSC1 oscillator circuit.

Table II.2.5.3.1 Input/Output Pins of the Low-speed (OSC1) Oscillator Circuit

Pin name	I/O	Function
OSC1	I	OSC1 oscillator input pin: Crystal oscillator or external clock input
OSC2	O	OSC1 oscillator output pin: Crystal oscillator (left open when using external clock input)

Structure of the OSC1 oscillator circuit

The OSC1 oscillator circuit accommodates a crystal oscillator and external clock input. As for the RTC, LV$_{DD}$ is used to supply power to this circuit. Figure II.2.5.3.1 shows the structure of the OSC1 oscillator circuit.



(1) Crystal oscillation circuit

(2) External clock input

(3) When not used

Figure II.2.5.3.1 OSC1 Oscillator Circuit

For use as a crystal oscillator circuit, connect a crystal resonator X'tal1 (32.768 kHz, typ.), feedback resistor (R_f), two capacitors (C_{G1}, C_{D1}), and, if necessary, a drain resistor (R_d) to the OSC1 and OSC2 pins and V_{SS}, as shown in Figure II.2.5.3.1 (1).

To use an external clock, leave the OSC2 pin open and input an LV_{DD} level clock (whose duty cycle is 50%) to the OSC1 pin.

The oscillator frequency/input clock frequency is 32.768 kHz (typ.). Make sure the crystal resonator or external clock used in the RTC has this clock frequency. With any other clock frequencies, the RTC cannot be used for timekeeping purposes.

For details of oscillation characteristics and the input characteristics of external clock, see "Electrical Characteristics."

When not using the OSC1 oscillator circuit, connect the OSC1 pin to V_{SS} and leave the OSC2 pin open.

Oscillation control

The OSC1 oscillator always operates without controlling using a register.

Note: When the oscillator is made to start oscillating at power-on, a finite time (of up to 3 seconds) is required until oscillation stabilizes. To prevent system malfunction, do not use the oscillator-derived clock until this oscillation stabilization time elapses.

II.2.6 Controlling Clock Supply

To reduce power consumption on the chip, a function is provided to turn off clock supply independently for each functional module.

II.2.6.1 Clock Supply to the S1C17 Core

In normal mode, the CMU always supplies the operating clock (CCLK) to the S1C17 Core.
When the S1C17 Core executes the halt or slp instruction, the CMU stops supplying the clock to the S1C17 Core and the S1C17 Core enters a standby (HALT or SLEEP) mode. The CMU resumes the clock supply to the S1C17 Core when the standby mode is cancelled by occurrence of an interrupt. For details of the standby mode, see Section II.2.8, "Standby Modes."

II.2.6.2 Clock Supply to Core Peripheral Modules

The core peripheral modules shown below use the core peripheral clock (PCLK).

- Prescaler
- Clock generator (16-bit/8-bit timers)
- UART
- SPI CH.0
- I^{2}C master
- I^{2}C slave
- Interrupt controller

The PCLK supply can be controlled using PCLK_EN (D0/CMU_GATEDCLK0 register)

* **PCLK_EN**: Core Peripheral Clock Control Bit in the Gated Clock Control 0 (CMU_GATEDCLK0) Register (D0/0x4906)

When initially reset, PCLK_EN is set to 1 (on), with the clock supplied to the core peripheral modules. If all the core peripheral modules are not used, set PCLK_EN to 0 to reduce current consumption.

Note: In HALT mode, PCLK does not stop if PCLK_EN is set to 1 (on). To stop supplying the clock in HALT mode, PCLK_EN should be set to 0 before executing the halt instruction. PCLK will stop in SLEEP mode regardless of how the CMU registers are set (see Section II.2.8.2 for more information).

II.2.6.3 Clock Supply to Extended Peripheral Modules

Table II.2.6.3.1 lists the control bits used for controlling the operating clock supply to the extended peripheral modules. The modules listed here have one controllable clock path, so they can be turned on/off using the corresponding control bit only. See Sections II.2.6.4 to II.2.6.7 for controlling the SRAMC, ROMC, RTC, and T8OSC1 operating clocks.

Table II.2.6.3.1 Extended Peripheral Clock Supply Control Bits

Module	Clock	Control bit	Register
8-bit programmable timers	PT8_CLK	PT8_CLK_EN (D1)	Gated Clock Control 1 (CMU_GATEDCLK1) Register (0x4907)
Multi-function timer	MFT_CLK	MFT_CLK_EN (D0)	
SPI CH.1 *	SPI_CLK	SPI_CLK_EN (D5)	Gated Clock Control 2 (CMU_GATEDCLK2) Register (0x4908)
Remote controller	REMC_CLK	REMC_CLK_EN (D4)	
A/D converter	ADC_CLK	ADC_CLK_EN (D3)	
Watchdog timer	WDT_CLK	WDT_CLK_EN (D2)	
I/O ports	PORT_CLK	PORT_CLK_EN (D1)	

* The prescaler for SPI CH.1 is included in the 8-bit programmable timer module. Therefore, the PT8_CLK supply must be enabled in addition to the SPI_CLK supply when SPI CH.1 is used.

When initially reset, these control bits are set to 1 (on), with clocks supplied to each module. If any module is unused, set the corresponding control bit to 0, thus turning the clock for that module off.

Note: These clocks do not stop in HALT mode if the corresponding control bits are set to 1 (on). To stop supplying the clock in HALT mode, the control bit should be set to 0 before executing the halt instruction. All these clocks will stop in SLEEP mode regardless of how the CMU registers are set (see Section II.2.8.2 for more information).

II.2.6.4 Clock Supply to the SRAMC

The SRAMC provides SRAMC_CLK_EN (D7/CMU_GATEDCLK1 register) for controlling the SRAMC clock (SRAMC_CLK). However, the SRAMC controls the internal peripheral bus (SAPB), so SRAMC_CLK cannot be stopped while the IC is running. In other words, SRAMC_CLK does not stop in normal operation mode (except when the halt or slp instruction is executed) even if SRAMC_CLK_EN is set to 0. However, SRAMC_CLK can be automatically turned off in HALT mode (after the halt instruction is executed) by setting SRAMC_CLK_EN to 0 (default: on).

∗ **SRAMC_CLK_EN**: SRAMC Clock Control (in HALT mode) Bit in the Gated Clock Control 1 (CMU_GATEDCLK1) Register (D7/0x4907)

When initially reset, SRAMC_CLK_EN is set to 1 (on) to enable the SRAMC_CLK supply. If SRAMC_CLK is unused in HALT mode, set SRAMC_CLK_EN to 0 (off). The SRAMC_CLK supply will be stopped after the CPU executes the halt instruction. SRAMC_CLK will stop in SLEEP mode regardless of how the CMU registers are set (see Section II.2.8.2 for more information).

II.2.6.5 Clock Supply to the ROMC

The ROMC is required for executing the program, so the ROMC clock (ROMC_CLK) cannot be stopped while the IC is running. However, the ROMC clock can be automatically turned off in HALT mode by setting ROMC_CLK_EN (D7/CMU_GATEDCLK0 register) to 0 (default: on).

The ROMC provides ROMC_CLK_EN (D7/CMU_GATEDCLK0 register) for controlling the ROMC clock (ROMC_CLK). However, The ROMC is required for executing the program in the ROM, so the ROMC_CLK cannot be stopped while the IC is running. In other words, the ROMC_CLK does not stop in normal operation mode (except when the halt or slp instruction is executed) even if ROMC_CLK_EN is set to 0. However, ROMC_CLK can be automatically turned off in HALT mode (after the halt instruction is executed) by setting ROMC_CLK_EN to 0 (default: on).

∗ **ROMC_CLK_EN**: ROMC Clock Control (in HALT mode) Bit in the Gated Clock Control 0 (CMU_GATEDCLK0) Register (D7/0x4906)

When initially reset, ROMC_CLK_EN is set to 1 (on) to enable the ROMC_CLK supply. If ROMC_CLK is unused in HALT mode, set ROMC_CLK_EN to 0 (off). The ROMC_CLK supply will stop after the CPU executes the halt instruction. ROMC_CLK will stop in SLEEP mode regardless of how the CMU registers are set (see Section II.2.8.2 for more information).

II.2.6.6 Clock Supply to the RTC

The RTC use two clocks for its operation.

(1) 32.768 kHz clock (OSC1)

This clock (OSC1 = 32.768 kHz) is used for timekeeping operations of the RTC. This clock is always supplied to the RTC (even in the standby mode).

(2) Control register clock (RTC_SAPB_CLK)

This clock (MCLK) is used to control the RTC registers. This clock is required for accessing the RTC registers and it can be stopped when not in use. RTC_SAPB_CLK_EN (D0/CMU_GATEDCLK2 register) is used for clock supply control (default: on).

∗ **RTC_SAPB_CLK_EN**: RTC SAPB I/F Clock Control Bit in the Gated Clock Control 2 (CMU_GATEDCLK2) Register (D0/0x4908)

The clock supply turns on when RTC_SAPB_CLK_EN is set to 1 and it turns off when it is set to 0. In HALT mode, RTC_SAPB_CLK does not stop if RTC_SAPB_CLK_EN is set to 1 (on). To stop supplying the clock in HALT mode, RTC_SAPB_CLK_EN should be set to 0 (off) before executing the halt instruction. RTC_SAPB_CLK will stop in SLEEP mode regardless of how the CMU registers are set (see Section II.2.8.2 for more information).

II.2.6.7 Clock Supply to the 8-bit OSC1 Timer

To operate the 8-bit OSC1 timer, two clock supplies must be controlled. The clock supply turns on when the control bit is set to 1 and it turns off when the control bit is set to 0.

(1) 8-bit OSC1 timer clock (T8OSC1_OSC1_CLK)

This clock (OSC1 = 32.768 kHz) is used for counting operation of the 8-bit OSC1 timer. T8OSC1_CLK_EN (D3/CMU_GATEDCLK1 register) is used for clock supply control (default: on). In HALT mode, T8OSC1_OSC1_CLK does not stop if T8OSC1_CLK_EN is set to 1 (on). To stop supplying the clock in HALT mode, T8OSC1_CLK_EN should be set to 0 (off) before executing the halt instruction. T8OSC1_OSC1_CLK will stop in SLEEP mode regardless of how the CMU registers are set (see Section II.2.8.2 for more information).

∗ **T8OSC1_CLK_EN**: 8-bit OSC1 Timer Clock Control Bit in the Gated Clock Control 1 (CMU_GATEDCLK1) Register (D3/0x4907)

(2) Control register clock (T8OSC1_SAPB_CLK)

This clock (MCLK) is used to control the 8-bit OSC1 timer registers. This clock is required for accessing the 8-bit OSC1 timer registers and it can be stopped when not in use. RTC_SAPB_CLK_EN (D0/CMU_GATEDCLK2 register), which is the clock control bit for the RTC, is used for clock supply control (default: on). In HALT mode, T8OSC1_SAPB_CLK does not stop if RTC_SAPB_CLK_EN is set to 1 (on). To stop supplying the clock in HALT mode, RTC_SAPB_CLK_EN should be set to 0 (off) before executing the halt instruction. Note, however, that the RTC_SAPB_CLK supply to the RTC is also disabled when RTC_SAPB_CLK_EN is set to 0 (off). T8OSC1_SAPB_CLK will stop in SLEEP mode regardless of how the CMU registers are set (see Section II.2.8.2 for more information).

∗ **RTC_SAPB_CLK_EN**: RTC SAPB I/F Clock Control Bit in the Gated Clock Control 2 (CMU_GATEDCLK2) Register (D0/0x4908)

II.2.7 Setting the External Clock Output (CMU_CLK)

CMU_CLK is an external output clock for the external devices.
CMU_CLK can be selected from 9 clocks using CMU_CLKSEL[3:0] (D[3:0]/CMU_CMUCLK register).

∗ **CMU_CLKSEL[3:0]**: CMU_CLK Selection Bits in the CMU_CLK Select (CMU_CMUCLK) Register (D[3:0]/0x4905)

Table II.2.7.1 Selecting CMU_CLK

CMU_CLKSEL[3:0]	CMU_CLK
0xf–0xa	Reserved
0x9	OSC3•1/32
0x8	OSC3•1/16
0x7	OSC3•1/8
0x6	OSC3•1/4
0x5	OSC3•1/2
0x4	OSC3•1/1
0x3	Reserved
0x2	BCLK
0x1	OSC1
0x0	OSC3

(Default: 0x0 = OSC3)

CMU_CLK can be selected at any time. However, switching over the clocks creates hazards.
When CMU_CLK must be output to external devices, it is also necessary to select a port function. For details on how to control clock output and the port to be used, see Section I.3.3, “Switching Over the Multiplexed Pin Functions.”

Note: Settings other than those listed in Table II.2.7.1 are reserved for testing. Do not set undescribed values to CMU_CLKSEL[3:0] as undesired clocks may output.

II.2.8 Standby Modes

The S1C17002 supports two standby modes: HALT and SLEEP. Power consumption on the chip can be greatly reduced by placing the CPU in one of these standby modes. Moreover, the CPU must be placed in SLEEP mode before clock sources for the system (OSC3 or OSC1) are switched over (see Sections II.2.9.1 and II.2.9.2 for more information).

II.2.8.1 HALT Mode

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

In HALT mode, the CPU stops operating. Furthermore, the SRAMC clock (SRAMC_CLK) and ROMC clock (ROMC_CLK) can be stopped in HALT mode (after the halt instruction is executed) by setting SRAMC_CLK_EN (D7/CMU_GATEDCLK1 register) and ROMC_CLK_EN (D7/CMU_GATEDCLK0 register) to 0, respectively (see Sections II.2.6.4 and II.2.6.5 for the SRAMC and ROMC clocks). The other internal peripheral circuits remain in the state (idle or operating) held when the halt instruction was executed.

The CPU is released from HALT mode by initial reset, an NMI or other interrupt, or a forcible break from the debugger.

HALT mode is effective in reducing power consumption on the chip when running the CPU is unnecessary, such as when waiting for external input or responses from peripheral circuits. When the CPU is released from HALT mode by an interrupt, it enters a program executable state by interrupt processing and executes an interrupt handling routine for the interrupt generated. In interrupt processing of the CPU, the address for the instruction next to halt is saved to the stack as a return address from the interrupt handling routine, so that the reti instruction in the interrupt handling routine branches to the instruction next to halt. The CPU is released from HALT mode when the interrupt controller (ITC) asserts the interrupt signal to be sent to the CPU. In other words, when an interrupt flag of the interrupts that have been enabled by the interrupt enable bits in the ITC is set to 1, the CPU can be released from HALT mode even if the PSR is set to disable interrupts. However, in this case the CPU does not execute the interrupt handling routine.

II.2.8.2 SLEEP Mode

The CPU suspends program execution upon executing the slp instruction and enters SLEEP mode. In SLEEP mode, the CPU stops operating and the CMU can stop supplying a clock to each functional module (see Section II.2.6 for more information). Therefore, all peripheral circuits (except the OSC1 oscillator circuit and RTC) stop operating. Note that before the CMU actually stops clock output after initiating processing to enter SLEEP mode, up to 8 clock cycles of the source clock (OSC) then selected are required.

The CPU is reawaken from SLEEP mode (when WAKEUPWT = 1) by initial reset, RTC interrupt (level triggered), or other interrupt from an external device (port input interrupt with level triggered). See Sections II.2.9.3 and II.2.9.4 for more information on the CMU register settings.

When the CPU is reawaken from SLEEP mode by an interrupt, it enters a program executable state by interrupt processing and executes an interrupt handling routine for the interrupt generated. In interrupt processing of the CPU, the address for the instruction next to slp is saved to the stack as a return address from the interrupt handling routine, so that the reti instruction in the interrupt handling routine branches to the instruction next to slp.

Cause-of-interrupt flags in the interrupt controller (ITC) cannot be set in SLEEP mode as the clock is not supplied to the ITC in SLEEP mode. Therefore, when the clock is not supplied to the ITC, the level triggered interrupt signals from the interrupt sources that have been enabled to generate an interrupt are input to the CMU through the ITC and used to wake up the CPU from SLEEP mode. In this case, the interrupt flag is set after the clock has started supplying to the ITC. The CPU can wake up from SLEEP mode by a cause of interrupt as described above even if the PSR is set to disable interrupts, note however, that the CPU does not execute the interrupt handling routine.

Note: In SLEEP mode, there is a time lag between input of an interrupt signal for wakeup and the start of the clock supply to the ITC, so a delay will occur until the interrupt controller (ITC) sets the interrupt flag. Therefore, no interrupt will occur if the interrupt signal is deasserted before the clock is supplied to the ITC, as the interrupt flag in the ITC is not set.

Furthermore, additional time is needed for the CPU to accept the interrupt request from the ITC, the CPU may execute a few instructions that follow the slp instruction before it starts the interrupt processing.

When using a port input interrupt is used to wake up the CPU from SLEEP mode, set the interrupt trigger mode to level trigger (see Section III.1.3.5) and assert the input signal until the clock supply has started.

Stopping OSC3 oscillation and waiting for oscillation stabilization at wakeup

By default, the OSC3 oscillator circuit does not stop operating in SLEEP mode. OSC3 oscillation can be made to stop during SLEEP mode by setting OSC3OFF (D7/CMU_SYSCLKCTL register).

∗ **OSC3OFF**: OSC3 Disable During SLEEP Bit in the System Clock Control (CMU_SYSCLKCTL) Register (D7/0x4900)

Setting OSC3OFF to 1 causes OSC3 oscillation to stop during SLEEP mode. In this case, the OSC3 oscillator circuit starts oscillating when the CPU is reawaken from SLEEP mode. However, since the CPU may operate erratically if it starts operating with the OSC3 clock before the oscillation stabilizes, an OSC oscillation start wait timer is provided to keep the CPU waiting a while before it starts operating. The wait time can be set by using OSCTM[7:0] (D[7:0]/CMU_OSC3_WCNT register) and TMHSP (D6/CMU_SYSCLKCTL register).

∗ **OSCTM[7:0]**: OSC3 Oscillation Stabilization-Wait Timer Bit in the OSC3 Wait Timer (CMU_OSC3_WCNT) Register (D[7:0]/0x4901)

∗ **TMHSP**: Wait-Timer High-Speed Mode Bit in the System Clock Control (CMU_SYSCLKCTL) Register (D6/0x4900)

Table II.2.8.2.1 Oscillation Stabilization Wait Time at Wakeup

TMHSP	OSCTM[7:0]	Number of clocks	Time
1	0x0	0	0
	0x1	16	800 ns
	0x2	32	1.6 µs
	:	:	:
	0xff	4080	0.204 ms
0	0x0	0	0
	0x1	8192	0.409 ms
	0x2	16384	0.819 ms
	:	:	:
	0xff	2M	104.5 ms

(The time shown here is an example when operating with a 20 MHz OSC3.)

SLEEP control when clock sources are switched over

When the CPU reawakes from SLEEP mode, the clock sources (OSC3 or OSC1) also are switched over depending on how OSCSEL (D2/CMU_SYSCLKCTL register) is set. Before the clock sources can be switched over, the CPU must be placed once in SLEEP mode, then released. Therefore, a function is provided that automatically reawakes the CPU from SLEEP mode without using an interrupt, etc. To use this function, set WAKEUPWT (D4/CMU_SYSCLKCTL register) to 0. (By default, it is set to 0.)

∗ **OSCSEL**: OSC Clock Selection Bit in the System Clock Control (CMU_SYSCLKCTL) Register (D2/0x4900)

∗ **WAKEUPWT**: Wakeup-Wait Function Enable Bit in the System Clock Control (CMU_SYSCLKCTL) Register (D4/0x4900)

When the slp instruction is executed with WAKEUPWT set to 0, the CPU automatically reawakes from SLEEP mode several 10 clock cycles after that time, then restarts with the source clock selected by OSCSEL after the oscillation stabilization time described above has elapsed.

The OSC oscillation start wait timer configured using OSCTM[7:0] and TMHSP is effective even if WAKEUPWT is 0. To restart the CPU in the shortest time possible, set OSCTM[7:0] to 0x0 and TMHSP to 1.

When WAKEUPWT is set to 1, the CPU is reawaken from SLEEP mode by initial reset, RTC interrupt (level triggered), or other interrupt from an external device (port input interrupt with level triggered).

For details about clock switchover and SLEEP control procedures, see Section II.2.9, “Clock Setup Procedure.”

II.2.8.3 Precautions

Interrupt

The standby mode is released by an interrupt from the ITC, NMI, or reset. Note that the ITC must be configured so that the interrupt to be used for releasing the standby mode can be generated to the CPU. When the clock has not been supplied to the ITC, the interrupt signal from the interrupt source that has been enabled to interrupt is passed through the ITC and is input to the CMU. This signal is used to release the standby mode and to start supplying clocks. The ITC can operate with the supplied clock in HALT mode, so the cause-of-interrupt flag is set immediately after the interrupt source asserts the interrupt signal and the ITC requests an interrupt to the CPU without a delay. In SLEEP mode, the ITC will be able to set the cause-of-interrupt flag and to request an interrupt to the CPU after the CMU starts supplying the clock to the ITC. Therefore, the delay in the interrupt request to the CPU after waking up from SLEEP mode may cause the CPU to execute a few instructions that follows the slp instruction before the CPU executes the interrupt processing. Moreover, if the interrupt source deasserts the interrupt signal before the CMU starts supplying the clock to the ITC, an interrupt does not occur since the cause-of-interrupt flag is not set. The IE and IL[2:0] bits in the CPU's PSR register do not affect the releasing of standby mode by an interrupt. For example, by setting the ITC to enable the interrupt used for releasing and setting the IE bit to disable interrupts, the CPU can wake up from SLEEP mode without an interrupt processing.

To ensure that the interrupt handler routine will be executed when a port input interrupt (level interrupt) is used to cancel standby mode, the port input signal must be asserted longer than the time shown below.

(1) When the clock is stopped during SLEEP mode
OSC3 oscillation start time + OSC3 oscillation stabilization wait time (set by the user) + 10 system clock cycles

(2) When the clock is not stopped during SLEEP mode, or in HALT mode
10 system clock cycles

Oscillator circuits

When OSC3 oscillation is set to stop during SLEEP mode, the OSC3 oscillator circuit starts oscillating upon exiting SLEEP mode. This is because the OSC3 oscillator circuit requires a finite time before its oscillation stabilizes after starting operation. To restart the CPU using the OSC3 as the source clock, OSCTM[7:0] (D[7:0]/CMU_OSC3_WCNT register) and TMHSP (D6/CMU_SYSCLKCTL register) must be properly set so that the CPU starts operating after this oscillation stabilization time elapses. The oscillation start time of the OSC3 oscillator circuit varies with the device used, board patterns, and operating environment. Therefore, the set time above should have a sufficient allowance.

Switching over the clock sources

Use the automatic SLEEP cancellation function when executing the slp instruction for switching over the clock sources. When the SLEEP mode is cancelled, the OSC oscillation start wait timer that has been configured using OSCTM[7:0] and TMHSP starts operating with the clock source after switch over. Use the switched clock frequency for calculating the oscillation wait time.

Other

The core CPU register contents are retained even during standby mode. In SLEEP mode, the input/output pins keep the status at the time the S1C17002 enters SLEEP mode. Also in HALT mode, the input/output pins keep the status at the time the S1C17002 enters HALT mode. However, some input/output pin statuses may be changed according to the module operation if the CMU has been configured to supply the module operating clock in HALT mode.

II.2.9 Clock Setup Procedure

This section describes the procedure for setting up clocks or altering clock settings.

When initially reset, the clocks are set to the following states:

OSC3 oscillator circuit:	On
OSC1 oscillator circuit:	On
System clock source:	OSC3
System clock:	OSC3•1/1
CMU_CLK:	OSC3•1/1

II.2.9.1 Changing the Clock Source from OSC3 to OSC1, then Turning Off OSC3

1. CMU Write Protect Register (0x4920) = 0x96
   Disable write protection of the CMU registers.

2. OSCSEL (D2/0x4900) = 1
   Select OSC1 for the clock source.

3. Setting the OSC3 Wait Timer Register (0x4901) and System Clock Control Register (0x4900)
   - OSCTM[7:0] (D[7:0]/0x4901) = *
   - OSC3OFF (D7/0x4900) = 0
   - TMHSP (D6/0x4900) = *
   - WAKEUPWT (D4/0x4900) = 0

   * Set appropriate values so that the wait timer exceeds the stabilization time of OSC1 oscillation (e.g., 3 seconds in the S1C17002). Be aware that the wait timer operates with the OSC1 clock. For details about the OSC1 oscillation start time, see “Electrical Characteristics.”

   This setting causes the CPU to automatically exit SLEEP mode and restart after the set time has passed without waiting for an interrupt.

4. Stop any peripheral circuits that are operating.

5. Execute the slp instruction.
   The chip enters SLEEP mode and the CMU temporarily stops clock output. The CPU automatically reawakens from SLEEP mode after the set time has passed from execution of the slp instruction, and restarts using OSC1 as the clock source.

6. SOSC3 (D1/0x4900) = 0
   Turn off the OSC3 oscillator circuit.

7. Newly setting the CMU registers again
   Newly alter the CMU_CLK settings, and set other CMU registers again, as required.

8. CMU Write Protect Register (0x4920) = other than 0x96
   Reenable write protection of the CMU registers.

II.2.9.2 Changing the Clock Source from OSC1 to OSC3

1. CMU Write Protect Register (0x4920) = 0x96
   Disable write protection of the CMU registers.

2. SOSC3 (D1/0x4900) = 1
   Turn on the OSC3 oscillator circuit if turned off.

3. OSCSEL (D2/0x4900) = 0
   Select OSC3 for the clock source.

4. Setting the OSC3 Wait Timer Register (0x4901) and System Clock Control Register (0x4900)
   • OSCTM[7:0] (D[7:0]/0x4901) = *
   • OSC3OFF (D7/0x4900) = 0
   • TMHSP (D6/0x4900) = *
   • WAKEUPWT (D4/0x4900) = 0

   * Set appropriate values so that the wait timer exceeds the stabilization time of OSC3 oscillation (e.g., 25 ms in the S1C17002). Be aware that the wait timer operates with the OSC3 clock. For details about the OSC3 oscillation start time, see "Electrical Characteristics."

   This setting causes the CPU to automatically exit SLEEP mode and restart after the set time has passed without waiting for an interrupt.

5. Stop any peripheral circuits that are operating, except the RTC.

6. Execute the slp instruction.
   The chip enters SLEEP mode and the CMU temporarily stops clock output. The CPU automatically reawakens from SLEEP mode after the set time has passed from execution of the slp instruction, and restarts using OSC3 as the clock source.

7. Newly setting the clock control registers again
   Newly alter the system clock or CMU_CLK settings, and set other CMU registers newly again, as required.

8. CMU Write Protect Register (0x4920) = other than 0x96
   Reenable write protection of the CMU registers.

II.2.9.3 Turning Off OSC3 during SLEEP

To turn off OSC3 oscillation during SLEEP mode when operating with OSC3 as the clock source, follow the control procedure described below.

1. CMU Write Protect Register (0x4920) = 0x96
   Disable write protection of the CMU registers.
2. Setting the OSC3 Wait Timer Register (0x4901) and System Clock Control Register (0x4900)
   - OSCTM[7:0] (D[7:0]/0x4901) and TMHSP (D6/0x4900)
     Set the wait time until the oscillation stabilizes after exiting SLEEP mode.
     Example: TMHSP = 1, OSCTM[7:0] = 0x40 (wait time = about 26 ms when OSC3 = 20 MHz)
   - OSC3OFF (D7/0x4900) = 1
     Turn off OSC3 oscillation when in SLEEP mode.
   - WAKEUPWT (D4/0x4900) = 1
     Set the CPU to awake from SLEEP mode by using an RTC interrupt (level triggered), or other interrupt from an external device (port input interrupt with level triggered).
3. CMU Write Protect Register (0x4920) = other than 0x96
   Reenable write protection of the CMU registers.
4. Stop any peripheral circuits that are operating, except the RTC.
5. Execute the slp instruction.
   The chip enters SLEEP mode and the CMU temporarily stops clock output.

   The CPU is brought out of SLEEP mode by an RTC interrupt (level triggered), forced break from the debugger, or other interrupt from an external device (port input interrupt with level triggered), and it restarts using the clock source (OSC3) selected with OSCSEL (D2/0x4900).

II.2.9.4 SLEEP Keeping Oscillation On (without Clock Change)

To enter SLEEP mode without a clock source change and turning off the oscillation, follow the control procedure described below. This is the control to reduce power consumption as much as possible by stopping the core and peripheral functions, with no restart time penalty.

1. CMU Write Protect Register (0x4920) = 0x96
   Disable write protection of the CMU registers.

2. Setting the OSC3 Wait Timer Register (0x4901) and System Clock Control Register (0x4900)
   • OSCTM[7:0] (D[7:0]/0x4901) = 0x0
   • OSC3OFF (D7/0x4900) = 0
   • TMHSP (D6/0x4900) = 1
   • WAKEUPWT (D4/0x4900) = 1
     This setting causes the CPU to exit SLEEP mode using an RTC interrupt (level triggered), or other interrupt from an external device (port input interrupt with level triggered), and to restart in the shortest time possible (several 10 clock cycles).

3. CMU Write Protect Register (0x4920) = other than 0x96
   Reenable write protection of the CMU registers.

4. Stop any peripheral circuits that are operating, except the RTC.

5. Execute the slp instruction.
   The chip enters SLEEP mode and the CMU temporarily stops clock output.

   The CPU is brought out of SLEEP mode by an RTC interrupt (level triggered), forced break from the debugger, or other interrupt from an external device (port input interrupt with level triggered), and it restarts using the clock source selected with OSCSEL (D2/0x4900).

II.2.10 Noise Filters

II.2.10.1 Noise Filter for DSIN Input

If the DSIO signal becomes active due to noise, the S1C17 Core suspends the program execution and enters debug mode. To avoid this, the S1C17002 incorporates a noise filter that operates with the system clock to remove noise from the signal before it is input to the S1C17 Core.

When using this noise filter, set DSINNF (D4/CMU_NF register) to 1. When DSINNF is set to 0 (default), the DSIO signal bypasses the noise filter.

* **DSINNF**: DSIO Input Noise Filter Enable Bit in the Noise Filter Control (CMU_NF) Register (D4/0x4902)

II.2.10.2 Noise Filters for Input Ports

The CMU module provides the noise filters to remove noise on the signals input from the ports shown below.

SPI:	SPI_SDI0, SPI_SCK0, SPI_SDI1, SPI_SCK1
UART:	#SCLK0
SRAMC:	#WAIT
I^{2}C:	I2CM_SDA, I2CM_SCL, I2CS_SDA, I2CS_SCL, #I2CS_RST
CG_T16U0:	EXCL0

When using these noise filters, set INPORTNF (D1/CMU_NF register) to 0. When INPORTNF is set to 1 (default), the input signals bypass the noise filters.

* **INPORTNF**: Input Port Noise Filter Enable Bit in the Noise Filter Control (CMU_NF) Register (D1/0x4902)

Notes:
- These noise filters cannot be enabled individually.
- The noise filters are not effective if these ports are used as general-purpose input port.

II.2.10.3 Noise Filter for OSC3 Clock Input

To stabilize the operation when an external clock is input to the OSC3 pin, the CMU provides a noise filter to remove noise from the input clock.

When using this noise filter, set OSC3NF (D0/CMU_NF register) to 0. When OSC3NF is set to 1 (default), the input clock bypasses the noise filter.

* **OSC3NF**: OSC3 Input Noise Filter Enable Bit in the Noise Filter Control (CMU_NF) Register (D0/0x4902)

II.2.11 Details of Control Registers

Table II.2.11.1 List of CMU Registers

Address	Register name		Function
0x4900	CMU_SYSCLKCTL	System Clock Control Register	Controls the system clock.
0x4901	CMU_OSC3_WCNT	OSC3 Wait Timer Register	Sets the OSC3 wait timer for system wake-up.
0x4902	CMU_NF	Noise Filter Control Register	Enables noise filters.
0x4903	CMU_OSC3DIV	OSC3 Clock Divider Register	Selects an OSC3 system clock frequency.
0x4905	CMU_CMUCLK	CMU_CLK Select Register	Selects the output CMU_CLK frequency.
0x4906	CMU_GATEDCLK0	Gated Clock Control 0 Register	Controls clock supply to peripheral modules.
0x4907	CMU_GATEDCLK1	Gated Clock Control 1 Register	
0x4908	CMU_GATEDCLK2	Gated Clock Control 2 Register	
0x4920	CMU_PROTECT	CMU Write Protect Register	Enables writing to the CMU registers (0x4900–0x4908).

The following describes each CMU control register.

The CMU control registers are mapped as an 8-bit device at addresses 0x4900 to 0x4920, and can be accessed in units of 16 bits or bytes.

Note: The CMU registers (0x4900–0x4908) are write-protected. Before these register can be rewritten, write protection must be removed by writing data 0x96 to the CMU Write Protect Register (0x4920). Note that since unnecessary rewrites to addresses 0x4900–0x4908 could lead to erratic system operation, the CMU Write Protect Register (0x4920) should be set to other than 0x96 unless said CMU registers must be rewritten.

0x4900: System Clock Control Register (CMU_SYSCLKCTL)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
System Clock Control Register (CMU _SYSCLKCTL)	**0x4900** (8 bits)	D7	**OSC3OFF**	OSC3 disable during SLEEP	1	Stop	0	Run	0	R/W	
		D6	**TMHSP**	Wait-timer high-speed mode	1	High speed	0	Normal	0	R/W	
		D5	–	reserved	–				–	–	0 when being read.
		D4	**WAKEUPWT**	Wakeup-wait function enable	1	Wait interrupt	0	No wait	0	R/W	
		D3	–	reserved	–				–	–	0 when being read.
		D2	**OSCSEL**	OSC clock selection	1	OSC1	0	OSC3	0	R/W	
		D1	**SOSC3**	OSC3 oscillator on/off	1	On	0	Off	1	R/W	
		D0	–	reserved	–				–	–	0 when being read.

D7 OSC3OFF: OSC3 Disable During SLEEP Bit

Selects whether to turn off the OSC3 oscillator circuit during SLEEP mode.

1 (R/W): Stop
0 (R/W): Operating (default)

Continue operating OSC3 when entering SLEEP mode to switch over the clock sources (OSC), or turn it off when entering SLEEP mode for power-down purposes.

D6 TMHSP: Wait-Timer High-Speed Mode Bit

Sets count mode for the oscillation stabilization wait timer (CMU_OSC3_WCNT register).

1 (R/W): High-speed mode
0 (R/W): Normal mode (default)

The oscillation stabilization wait timer counts from 0 to 2M in units of 8,192 system clock cycles during normal mode, or from 0 to 4,080 in units of 16 system clock cycles during high-speed mode. Select either mode in which the OSC3 oscillation start time can be secured with the OSC frequency used.

D5 Reserved

D4 WAKEUPWT: Wakeup-Wait Function Enable Bit

Enables the SLEEP mode wakeup-wait function used for switching over the clocks.

1 (R/W): Wait an interrupt
0 (R/W): No wait (default)

When the slp instruction is executed while WAKEUPWT is set to 0, the CPU automatically reawakes from SLEEP mode several 10 clock cycles after instruction execution, and restarts with the source clock selected by OSCSEL (D2). Since even in this case the oscillation stabilization wait time set by OSCTM[7:0] (D[7:0]/CMU_OSC3_WCNT register) is effective, OSCTM[7:0] should be set to 0x0 when clocks must be switched over in the shortest time possible.

When WAKEUPWT is set to 1, the CPU can only be reawaken from SLEEP mode by initial reset, RTC interrupt (level triggered), forced break from the debugger, and other interrupt from an external source (port input interrupt with level triggered).

D3 Reserved

D2 OSCSEL: OSC Clock Selection Bit

Selects the system clock source (OSC).

1 (R/W): OSC1
0 (R/W): OSC3 (default)

The clock sources changed here are not switched over immediately, but are actually switched over upon returning from SLEEP mode. Therefore, the CPU must be placed in SLEEP mode after setting up OSCSEL.

Note: When the clock source is changed, the clock control registers must be set so that the CMU is supplied with a clock from the selected clock source upon returning from SLEEP mode immediately after the change. Otherwise, the chip does not restart after return from SLEEP mode.

D1 **SOSC3: OSC3 Oscillator On/Off Bit**

Turns the OSC3 oscillator circuit on or off.

1 (R/W): On (default)
0 (R/W): Off

Note: When SOSC3 is set from 0 to 1 for initiating oscillation by the oscillator, a finite time is required until the oscillation stabilizes. To prevent erratic operation, do not use the oscillator-derived clock until the oscillation start time stipulated in the electrical characteristics table elapses.

D0 **Reserved**

0x4901: OSC3 Wait Timer Register (CMU_OSC3_WCNT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
OSC3 Wait Timer Register (CMU_OSC3_ WCNT)	**0x4901** (8 bits)	D7–0	**OSCTM[7:0]**	OSC oscillation stabilization-wait timer	0–255	0x0	R/W	

D[7:0] OSCTM[7:0]: OSC Oscillation Stabilization-Wait Timer Bits

Sets an oscillation stabilization wait time during which the CPU is kept waiting before it starts operating upon returning from SLEEP mode. This wait time can be set in increments of 16 system clock cycles when TMHSP (D6/CMU_SYSCLKCTL register) = 1, or 8,192 clock cycles when TMHSP = 0. (Default: 0x0 = no wait time)

Table II.2.11.2 Oscillation Stabilization Wait Time at Wakeup

TMHSP	OSCTM[7:0]	Number of clocks	Time
1	0x0	0	0
	0x1	16	800 ns
	0x2	32	1.6 µs
	:	:	:
	0xff	4080	0.204 ms
0	0x0	0	0
	0x1	8192	0.409 ms
	0x2	16384	0.819 ms
	:	:	:
	0xff	2M	104.5 ms

(The time shown here is an example when operating with a 20 MHz OSC3.)

When the OSC3 oscillation is to be turned off during SLEEP mode, make sure the wait time set by these bits is equal to or greater than the OSC3 oscillation start time stipulated in the electrical characteristics table.

Note: The OSC oscillation start wait timer operates with the operating clock activated after the SLEEP mode is released. Therefore, use the switched clock frequency for calculating the oscillation wait time to be set to OSCTM[7:0] when executing the slp instruction for switching over the clock sources.

0x4902: Noise Filter Control Register (CMU_NF)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Noise Filter Control Register (CMU_NF)	**0x4902** (8 bits)	D7–5	–	reserved	–				–	–	0 when being read.
		D4	**DSINNF**	DSIO input noise filter enable	1	Enable	0	Disable	0	R/W	
		D3–2	–	reserved	–				–	–	0 when being read.
		D1	**INPORTNF**	Input port noise filter enable	1	Disable	0	Enable	1	R/W	
		D0	**OSC3NF**	OSC3 input noise filter enable	1	Disable	0	Enable	1	R/W	

D[7:5] Reserved

D4 DSINNF: DSIO Input Noise Filter Enable Bit

Enables/disables the noise filter for the DSIO input.

1 (R/W): Enable (reject noise)

0 (R/W): Disable (bypass) (default)

When using this noise filter, set DSINNF to 1. When DSINNF is set to 0 (default), the DSIO signal bypasses the noise filter.

D[3:2] Reserved

D1 INPORTNF: Input Port Noise Filter Enable Bit

Enables/disables the noise filters for the input ports.

1 (R/W): Disable (bypass) (default)

0 (R/W): Enable (reject noise)

The CMU module provides the noise filters to remove noise on the signals input from the ports shown below.

SPI:	SPI_SDI0, SPI_SCK0, SPI_SDI1, SPI_SCK1
UART:	#SCLK0
SRAMC:	#WAIT
I^{2}C:	I2CM_SDA, I2CM_SCL, I2CS_SDA, I2CS_SCL, #I2CS_RST
CG_T16U0:	EXCL0

When using these noise filters, set INPORTNF to 0. When INPORTNF is set to 1 (default), the input signals bypass the noise filters.

D0 OSC3NF: OSC3 Input Noise Filter Enable Bit

Enables/disables the noise filter for the OSC3 external clock input.

1 (R/W): Disable (bypass) (default)

0 (R/W): Enable (reject noise)

When using this noise filter, set OSC3NF to 0. When OSC3NF is set to 1 (default), the input clock bypasses the noise filter.

0x4903: OSC3 Clock Divider Register (CMU_OSC3DIV)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
OSC3 Clock Divider Register (CMU_OSC3DIV)	**0x4903** (8 bits)	D7–3	–	reserved	–		–	–	0 when being read.
		D2–0	**OSC3DIV [2:0]**	OSC3 clock divider selection	OSC3DIV[2:0]	Divider	0x0	R/W	
					0x7	OSC3•1/1			
					0x6	OSC3•1/1			
					0x5	OSC3•1/32			
					0x4	OSC3•1/16			
					0x3	OSC3•1/8			
					0x2	OSC3•1/4			
					0x1	OSC3•1/2			
					0x0	OSC3•1/1			

D[7:3] Reserved

D[2:0] OSC3DIV[2:0]: OSC3 Clock Divider Selection Bits

OSC3DIV[2:0] is used to select the system clock frequency when OSC3 is selected as the system clock source. It is derived from the OSC3 clock by dividing its frequency by a given value. Use OSC3DIV[2:0] to select this clock divide ratio.

Table II.2.11.3 Selecting an OSC3 Divided Clock

OSC3DIV[2:0]	OSC3_DIV clock
0x7	OSC3•1/1
0x6	OSC3•1/1
0x5	OSC3•1/32
0x4	OSC3•1/16
0x3	OSC3•1/8
0x2	OSC3•1/4
0x1	OSC3•1/2
0x0	OSC3•1/1

(Default: 0x0 = OSC3•1/1)

A divided clock can be selected at any time. However, up to 32 OSC3 clock cycles are required before the clocks are actually changed after altering the register values.

0x4905: CMU_CLK Select Register (CMU_CMUCLK)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
CMU_CLK Select Register (CMU_CMUCLK)	**0x4905** (8 bits)	D7–4	–	reserved	–		–	–	0 when being read.
		D3–0	**CMU_ CLKSEL[3:0]**	CMU_CLK selection	CMU_CLKSEL[3:0]	Clock source	0x0	R/W	
					0xf–0xa	reserved			
					0x9	OSC3•1/32			
					0x8	OSC3•1/16			
					0x7	OSC3•1/8			
					0x6	OSC3•1/4			
					0x5	OSC3•1/2			
					0x4	OSC3•1/1			
					0x3	reserved			
					0x2	BCLK			
					0x1	OSC1			
					0x0	OSC3			

D[7:4] Reserved

D[3:0] CMU_CLKSEL[3:0]: CMU_CLK Selection Bits

CMU_CLK is the clock for the external bus. It can be selected from the 9 clocks listed in Table II.2.11.4.

Table II.2.11.4 Selecting CMU_CLK

CMU_CLKSEL[3:0]	CMU_CLK
0xf–0xa	Reserved
0x9	OSC3•1/32
0x8	OSC3•1/16
0x7	OSC3•1/8
0x6	OSC3•1/4
0x5	OSC3•1/2
0x4	OSC3•1/1
0x3	Reserved
0x2	BCLK
0x1	OSC1
0x0	OSC3

(Default: 0x0 = OSC3)

CMU_CLK can be selected at any time. However, switching over the clocks creates hazards. When CMU_CLK must be output to external devices, it is also necessary to select a port function. For details on how to control clock output and about the port to be used, see Section I.3.3, “Switching Over the Multiplexed Pin Functions.”

Note: Other settings than that listed in Table II.2.11.4 are reserved for testing. Do not set undescribed values to CMU_CLKSEL[3:0] as undesired clocks may output.

0x4906: Gated Clock Control 0 Register (CMU_GATEDCLK0)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Gated Clock Control 0 Register (CMU _GATEDCLK0)	**0x4906** (8 bits)	D7	**ROMC _CLK_EN**	ROMC clock control (in HALT mode)	1	On	0	Off	1	R/W	
		D6–1	–	reserved	–				–	–	0 when being read.
		D0	**PCLK_EN**	Core peripheral clock control	1	On	0	Off	1	R/W	

D7 ROMC_CLK_EN: ROMC Clock Control (in HALT mode) Bit

Controls clock (ROMC_CLK) supply to the ROMC in HALT mode.

1 (R/W): On (default)

0 (R/W): Off

D[6:1] Reserved

D0 PCLK_EN: Core Peripheral Clock Control Bit

Controls clock (PCLK) supply to the core peripheral modules.

1 (R/W): On (default)

0 (R/W): Off

Be sure to leave this bit 1 (default) and to avoid setting to this bit to 0.

0x4907: Gated Clock Control 1 Register (CMU_GATEDCLK1)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Gated Clock Control 1 Register (CMU_GATEDCLK1)	**0x4907** (8 bits)	D7	**SRAMC_CLK_EN**	SRAMC clock control (in HALT mode)	1	On	0	Off	1	R/W	
		D6–4	–	reserved	–				–	–	0 when being read.
		D3	**T8OSC1_CLK_EN**	8-bit OSC1 timer clock control	1	On	0	Off	1	R/W	
		D2	–	reserved	–				–	–	0 when being read.
		D1	**PT8_CLK_EN**	8-bit programmable timer clock control	1	On	0	Off	1	R/W	
		D0	**MFT_CLK_EN**	Multi-function timer clock control					1	R/W	

D7 SRAMC_CLK_EN: SRAMC Clock Control (in HALT mode) Bit

Controls clock (SRAMC_CLK) supply to the SRAMC in HALT mode.

1 (R/W): On (default)

0 (R/W): Off

D[6:4] Reserved

D3 T8OSC1_CLK_EN: 8-bit OSC1 Timer Clock Control Bit

Controls clock (T8OSC1_CLK) supply to the 8-bit OSC1 timers.

1 (R/W): On (default)

0 (R/W): Off

D2 Reserved

D1 PT8_CLK_EN: 8-bit Programmable Timer Clock Control Bit

Controls clock (PT8_CLK) supply to the 8-bit programmable timers.

1 (R/W): On (default)

0 (R/W): Off

D0 MFT_CLK_EN: Multi-Function Timer Clock Control Bit

Controls clock (MFT_CLK) supply to the multi-function timer.

1 (R/W): On (default)

0 (R/W): Off

0x4908: Gated Clock Control 2 Register (CMU_GATEDCLK2)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Gated Clock Control 2 Register (CMU_GATEDCLK2)	**0x4908** (8 bits)	D7–6	–	reserved	–				–	–	0 when being read.
		D5	**SPI_CLK_EN**	SPI CH.1 module clock control	1	On	0	Off	1	R/W	
		D4	**REMC_CLK_EN**	REMC module clock control					1	R/W	
		D3	**ADC_CLK_EN**	ADC module clock control					1	R/W	
		D2	**WDT_CLK_EN**	WDT module clock control					1	R/W	
		D1	**PORT_CLK_EN**	I/O port module clock control					1	R/W	
		D0	**RTC_SAPB_CLK_EN**	RTC SAPB I/F clock control					1	R/W	

D[7:6] Reserved

D5 SPI_CLK_EN: SPI CH.1 Module Clock Control Bit

Controls clock (SPI_CLK) supply to the SPI module.

1 (R/W): On (default)

0 (R/W): Off

D4 REMC_CLK_EN: REMC Module Clock Control Bit

Controls clock (REMC_CLK) supply to the remote controller module.

1 (R/W): On (default)

0 (R/W): Off

D3 ADC_CLK_EN: ADC Module Clock Control Bit

Controls clock (ADC_CLK) supply to the A/D converter module.

1 (R/W): On (default)

0 (R/W): Off

D2 WDT_CLK_EN: WDT Module Clock Control Bit

Controls clock (WDT_CLK) supply to the watchdog timer module.

1 (R/W): On (default)

0 (R/W): Off

D1 PORT_CLK_EN: I/O Port Module Clock Control Bit

Controls clock (PORT_CLK) supply to the I/O ports.

1 (R/W): On (default)

0 (R/W): Off

D0 RTC_SAPB_CLK_EN: RTC SAPB I/F Clock Control Bit

Controls clock (RTC_SAPB_CLK, T8OSC1_SAPB_CLK) supply to the SAPB interface of the RTC and 8-bit OSC1 timer.

1 (R/W): On (default)

0 (R/W): Off

0x4920: CMU Write Protect Register (CMU_PROTECT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
CMU Write Protect Register (CMU _PROTECT)	**0x4920** (8 bits)	D7–0	**CLGP[7:0]**	CMU register protect flag	Writing 10010110 (0x96) removes the write protection of the CMU registers (0x4900–0x4908). Writing another value set the write protection.	0x0	R/W	

D[7:0] CLGP[7:0]: CMU Register Protect Flag Bits

Enables/disables write protection of the CMU registers (0x4900–0x4908).

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

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

II.2.12 Precautions

Precautions regarding clock control

- The CMU registers (0x4900–0x4908) are write-protected. Before these registers can be rewritten, write protection must be removed by writing data 0x96 to the CMU Write Protect Register (0x4920). Once write protection is removed, the CMU registers can be written to any number of times until the protect register is reset to other than 0x96. Note that since unnecessary rewriting of the CMU registers could lead to erratic system operation, the CMU Write Protect Register (0x4920) should be set to other than 0x96 unless the CMU registers must be rewritten.
- When the clock source is changed, the CMU registers must be set so that the CMU is supplied with a clock from the selected clock source upon returning from SLEEP mode immediately after the change. Otherwise, the chip may not restart after return from SLEEP mode.
  Furthermore, note that the timer, which generates an oscillation stabilization wait time after the SLEEP mode is released, operates with the clock after switching over. Be sure to use the correct clock frequency for calculating the wait time to be set to OSCTM[7:0] (D[7:0]/CMU_OSC3_WCNT register) and TMHSP (D6/CMU_SYSCLKCTL register).
- When SOSC3 (D1/CMU_SYSCLKCTL register) is set from 0 to 1 for initiating oscillation by the oscillator, a finite time is required until the oscillation stabilizes (e.g., 25 ms in the S1C17002). To prevent erratic operation, do not use the oscillator-derived clock until the oscillation start time stipulated in the electrical characteristics table elapses.

Precautions regarding reset input

- Even if the #RESET pin is pulled low (= 0), the chip may not be reset unless supplied with a clock. To reset the chip for sure, #RESET should be held low for at least 3 OSC3 clock cycles. However, the input/output port pins will be initialized by reset regardless of whether the chip is supplied with a clock.
- The oscillation start time of the OSC3 oscillator circuit varies with the device used, board patterns, and operating environment. Therefore, a sufficient time should be provided before the reset signal is deasserted.

Precautions regarding NMI input

NMI cannot be nested. The CPU keeps NMI input masked out until the reti instruction is executed after an NMI exception occurred.

THIS PAGE IS BLANK.

II.3 Prescaler (PSC)

II.3.1 Configuration of the Prescaler

The S1C17002 incorporates a prescaler for generating the source clock of the clock generator that generates the operating clocks for the UART (CH.0), SPI (CH.0) and I^{2}C master modules. The prescaler divides the PCLK clock, which is supplied from the CMU, by 1 to 16K to generate 15 clocks with different frequencies. A clock select register is provided for each destination peripheral module allowing selection of a prescaler output clock as the count clock.

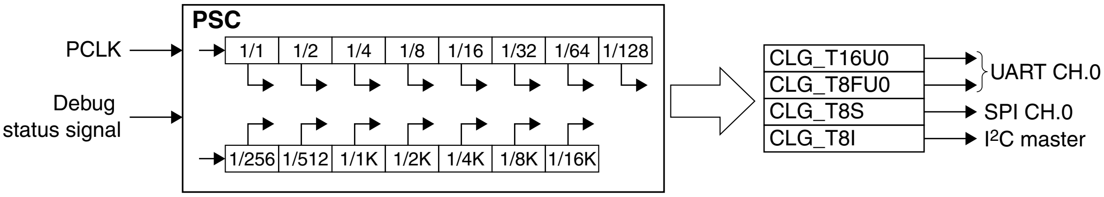


Figure II.3.1.1 Prescaler

The prescaler is controlled by the PRUN bit (D0/PSC_CTL register). Write 1 to PRUN to run the prescaler and write 0 to stop the prescaler. When the clock generator and interface modules are idle, stop the prescaler to reduce current consumption. At initial reset, the prescaler stops operating.

∗ **PRUN**: Prescaler Run/Stop Control Bit in the Prescaler Control (PSC_CTL) Register (D0/0x4020)

Note: Supply PCLK from the CMU before the prescaler can be used.

The prescaler provides one more control bit PRUND (D1/PSC_CTL register). This bit is used to specify the prescaler operation in debug mode. If PRUND is set to 1, the prescaler operates in debug mode. If PRUND is set to 0, the prescaler stops operating when the S1C17 Core enters debug mode. Set PRUND to 1 when using the clock generator and interface modules in debug mode.

∗ **PRUND**: Prescaler Run/Stop in Debug Mode Bit in the Prescaler Control (PSC_CTL) Register (D1/0x4020)

II.3.2 Details of Control Register

Table II.3.2.1 Prescaler Register

Address	Register name		Function
0x4020	PSC_CTL	Prescaler Control Register	Starts/stops the prescaler.

The prescaler register is an 8-bit register.

Note: When setting the register, be sure to write a 0, and not a 1, for all “reserved bits.”

0x4020: Prescaler Control Register (PSC_CTL)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Prescaler Control Register (PSC_CTL)	**0x4020** (8 bits)	D7–2	–	reserved	–				–	–	0 when being read.
		D1	**PRUND**	Prescaler run/stop in debug mode	1	Run	0	Stop	0	R/W	
		D0	**PRUN**	Prescaler run/stop control	1	Run	0	Stop	0	R/W	

D[7:2] Reserved

D1 PRUND: Prescaler Run/Stop in Debug Mode Bit

Selects the prescaler operation in debug mode.

1 (R/W): Run
0 (R/W): Stop (default)

If PRUND is set to 1, the prescaler operates in debug mode. If PRUND is set to 0, the prescaler stops operating when the S1C17 Core enters debug mode. Set PRUND to 1 when using the clock generator and interface modules in debug mode.

D0 PRUN: Prescaler Run/Stop Control Bit

Runs/stops the prescaler.

1 (R/W): Run
0 (R/W): Stop (default)

Write 1 to PRUN to run the prescaler and write 0 to stop the prescaler. When the clock generator and interface modules are idle, stop the prescaler to reduce current consumption.

II.3.3 Precaution

Supply PCLK from the CMU before the prescaler can be used.

THIS PAGE IS BLANK.

II.4 Clock Generator (CLG)

II.4.1 Configuration of the Clock Generator

The S1C17002 is equipped with a clock generator that consists of a 16-bit timer and three 8-bit timers. The 16-bit timer and the 8-bit timer with fine mode generate the clock for UART CH.0 and two other 8-bit timers generate the clock for SPI CH.0 and I^{2}C master. The timers count down from the initial value set in the software using a prescaler output clock as the count clock and output an underflow signal when the counter underflows. The underflow signal is used to generate an interrupt and an internal serial interface clock. This allows the application program to get any desired time intervals and programmable serial transfer rates.
Figure II.4.4.1 shows the configuration of the clock generator.

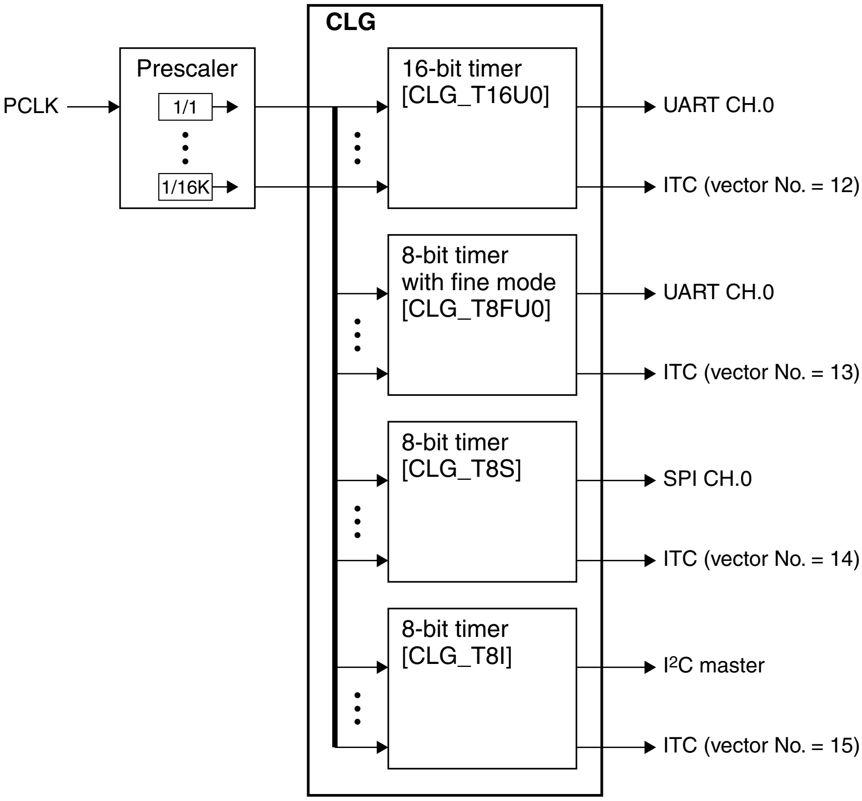


Figure II.4.4.1 Configuration of the Clock Generator

When the serial interface is not used, the timer can be used as a general-purpose programmable timer with an interrupt function.

II.4.2 16-bit Timer (CLG_T16U0)

II.4.2.1 Outline of the 16-bit Timer

The S1C17002 CLG is equipped with a 16-bit timer (CLG_T16U0).
The 16-bit timer includes a 16-bit presettable down counter and a 16-bit reload data register for setting 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 the serial interface clock for UART CH.0. The underflow period can be programmed by selecting a prescaler clock and setting reload data. This allows the application program to get any desired time intervals and programmable serial transfer rates.
Figure II.4.2.1.1 shows the structure of the 16-bit timer.

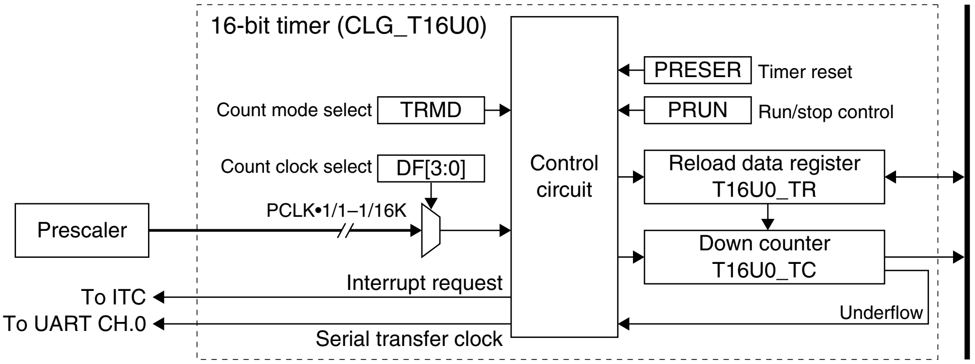


Figure II.4.2.1.1 Structure of 16-bit Timer

Note: Either the CLG_T16U0 clock or the CLG_T8FU0 clock can be selected in the UART module as the serial clock.

II.4.2.2 Count Mode

The 16-bit timer has two count modes: repeat mode and one-shot mode. It can be selected using TRMD (D4/CLG_T16U0_CTL register).

∗ **TRMD**: Count Mode Select Bit in the CLG_T16U0 Control (CLG_T16U0_CTL) Register (D4/0x4226)

Repeat mode (TRMD = 0, default)

The 16-bit timer is set in repeat mode when TRMD is set to 0.
In this mode, the 16-bit timer does not stop after it starts counting until the application program stops the timer. When the counter underflows, the timer presets the reload data register value to the counter and continues counting. The timer outputs the underflow pulses periodically. Set the 16-bit timer in this mode when generating periodical interrupts with a given interval or generating the serial transfer clock.

One-shot mode (TRMD = 1)

The 16-bit timer is set in one-shot mode when TRMD is set to 1.
In this mode, the 16-bit timer automatically stops counting when the counter underflows, so only one interrupt can be generated after starting the timer. When an underflow occurs, the counter is preset with the reload data register value before the timer operation stops. Set the 16-bit timer in this mode when a certain waiting time must be generated.

Note: When setting the count mode, make sure the 16-bit timer counter is stopped.

II.4.2.3 Count Clock

The 16-bit timer uses a prescaler output clock as the count clock. The prescaler divides PCLK by 1 to 16K to generate 15 clocks. Select one of the prescaler output clocks using DF[3:0] (D[3:0]/CLG_T16U0_CLK register).

∗ **DF[3:0]**: Timer Input Clock Select Bits in the CLG_T16U0 Input Clock Select (CLG_T16U0_CLK) Register (D[3:0]/0x4220)

Table II.4.2.3.1 Selecting the Count Clock

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

(Default: 0x0)

Notes:
- Before the 16-bit timer can start counting, the prescaler must be run.
- When setting the count clock, make sure the 16-bit timer counter is stopped.

For controlling the prescaler, refer to Section II.3, "Prescaler (PSC)."

II.4.2.4 16-bit Timer Reload Register and Underflow Period

The Reload Data (CLG_T16U0_TR) Register (0x4222) is used to set the initial value to the down counter.

The counter initial value set in the reload data register is preset to the down counter when the 16-bit timer is reset or when the counter underflows. When starting the 16-bit timer after resetting, the timer counts down from the reload value. So the reload value and the input clock frequency determine the period of time from starting the timer until an underflow occurs (and between underflows). This makes it possible to obtain a desired wait time, a periodical interrupt interval, or programmable transfer clock for the serial interface.

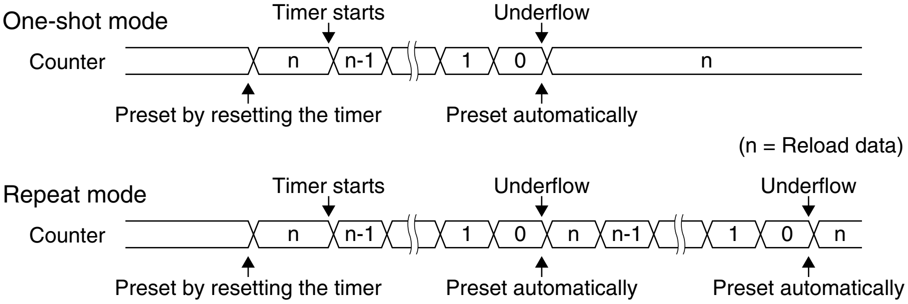


Figure II.4.2.4.1 Preset Timing

The underflow period is calculated by the expression below.

$$\text{Underflow period} = \frac{\text{TR} + 1}{\text{clk\_in}} \text{ [s]} \qquad \text{Underflow cycle} = \frac{\text{clk\_in}}{\text{TR} + 1} \text{ [Hz]}$$

clk_in: Count clock (prescaler output clock) frequency [Hz]
TR: Reload data (0–65535)

Note: The UART divides the 16-bit timer output by 16 to generate the sampling clock. Make sure of the division ratio when setting a transfer rate.

II.4.2.5 Resetting the 16-bit Timer

To reset the 16-bit timer, write 1 to PRESER (D1/CLG_T16U0_CTL register). This initializes the counter by presetting the reload data register value.

∗ **PRESER**: Timer Reset Bit in the CLG_T16U0 Control (CLG_T16U0_CTL) Register (D1/0x4226)

II.4.2.6 16-bit Timer Run/Stop Control

Before starting the 16-bit timer, set up the conditions as shown below.

(1) Select a count mode (one-shot or repeat). See Section II.4.2.2.
(2) Select the count clock (prescaler output clock). See Section II.4.2.3.
(3) Calculate the counter initial value and set it to the reload data register. See Section II.4.2.4.
(4) Reset the timer to preset the initial value to the counter. See Section II.4.2.5.
(5) Set up the interrupt level and enable the interrupt of the timer channel if the timer interrupt is used. See Section II.4.2.8.

To start the 16-bit timer, write 1 to PRUN (D0/CLG_T16U0_CTL register).

∗ **PRUN**: Timer Run/Stop Control Bit in the CLG_T16U0 Control (CLG_T16U0_CTL) Register (D0/0x4226)

The timer starts counting down from the initial value or the current counter value if the initial value has not been preset. When the counter underflows, the timer outputs an underflow pulse and presets the initial value again. At the same time, an interrupt request is sent to the interrupt controller (ITC).
If the timer is set in one-shot mode, the timer stops counting.
If the timer is set in repeat mode, the timer continues counting from the reloaded initial value.

To stop the 16-bit timer from the application program, write 0 to the PRUN bit. The counter stops counting and holds the current counter value until the timer is reset or restarted. To restart counting from the initial value, reset the timer before writing 1 to the PRUN bit.

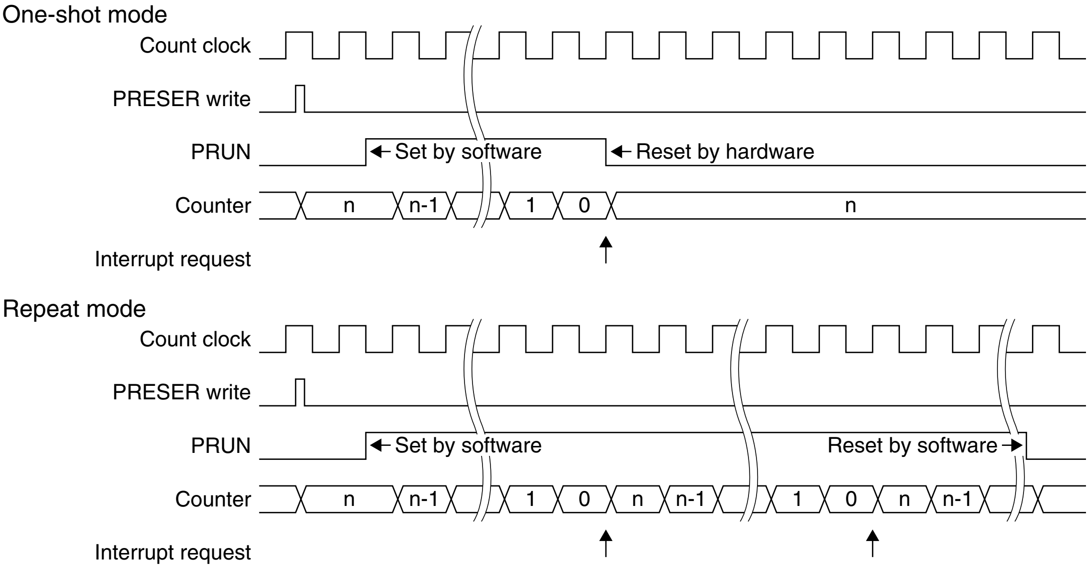


Figure II.4.2.6.1 Count Operation

II.4.2.7 16-bit Timer Output Signal

The 16-bit timer outputs an underflow pulse when the counter underflows.
This pulse is used to request a timer interrupt.
Also this pulse is used to generate the serial transfer clock for UART CH.0.

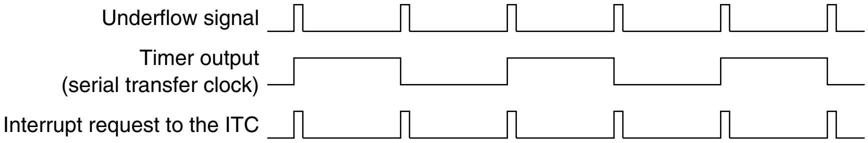


Figure II.4.2.7.1 Timer Output Clock

The reload data register value to obtain a desired transfer rate is calculated by the expression below.

$$\text{bps} = \frac{\text{clk\_in}}{(\text{TR} + 1) \times 16}$$

$$\text{TR} = \left(\frac{\text{clk\_in}}{\text{bps}} - 16\right) \div 16$$

clk_in: Count clock (prescaler output clock) frequency [Hz]
TR: Reload data (0–65535)
bps: Transfer rate (bits/second)

II.4.2.8 16-bit Timer Interrupt

The 16-bit timer outputs an interrupt request signal to the interrupt controller (ITC) when the counter underflows. To generate a timer underflow interrupt, set up the interrupt level and enable the interrupt using the ITC registers.

ITC registers for timer interrupts

The following shows the control bits of the ITC provided for the 16-bit timer:

Interrupt flag IIFT0

* **IIFT0**: CLG_T16U0 Timer Interrupt Flag Bit in the Interrupt Flag (ITC_IFLG) Register (D8/0x4300)

Interrupt enable bit IIEN0

* **IIEN0**: CLG_T16U0 Timer Interrupt Enable Bit in the Interrupt Enable (ITC_EN) Register (D8/0x4302)

Interrupt level setup bits IILV0

* **IILV0[2:0]**: CLG_T16U0 Timer Interrupt Level Bits in the Internal Interrupt Level Setup (ITC_ILV0) Register 0 (D[2:0]/0x430e)

When an underflow occurs in the timer, the corresponding interrupt flag is set to 1.
If the interrupt enable bit corresponding to that interrupt flag has been set to 1, the ITC sends an interrupt request to the S1C17 Core. To disable the timer interrupt, set the interrupt enable bit to 0.
The interrupt flag is always set to 1 by the timer underflow pulse, regardless of how the interrupt enable bit is set (even when set to 0).
The interrupt level setup bits set the interrupt level (0 to 7) of the timer interrupt.
An interrupt request to the S1C17 Core is accepted only when all the conditions described below are met.

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

For details on these interrupt control registers, as well as the device operation when an interrupt has occurred, see Section III.1, “Interrupt Controller (ITC).”

Interrupt vector

The following shows the vector number and vector address for the timer interrupt:

Vector number: 12 (0x0c)
Vector address: TTBR + 0x30

II.4.2.9 Details of Control Registers

Table II.4.2.9.1 List of 16-bit Timer Registers

Address	Register name		Function
0x4220	CLG_T16U0_CLK	CLG_T16U0 Input Clock Select Register	Selects a prescaler output clock.
0x4222	CLG_T16U0_TR	CLG_T16U0 Reload Data Register	Sets reload data.
0x4224	CLG_T16U0_TC	CLG_T16U0 Counter Data Register	Counter data
0x4226	CLG_T16U0_CTL	CLG_T16U0 Control Register	Sets the timer mode and starts/stops the timer.

The following describes each 16-bit timer register. These are all 16-bit registers.

Note: When setting the registers, be sure to write a 0, and not a 1, for all “reserved bits.”

0x4220: CLG_T16U0 Input Clock Select Register (CLG_T16U0_CLK)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
CLG_T16U0 Input Clock Select Register (CLG_T16U0_CLK)	**0x4220** (16 bits)	D15–4	–	reserved	–		–	–	0 when being read.
		D3–0	**DF[3:0]**	Timer input clock select (Prescaler output clock)	DF[3:0]	Clock	0x0	R/W	
					0xf	reserved			
					0xe	PCLK•1/16384			
					0xd	PCLK•1/8192			
					0xc	PCLK•1/4096			
					0xb	PCLK•1/2048			
					0xa	PCLK•1/1024			
					0x9	PCLK•1/512			
					0x8	PCLK•1/256			
					0x7	PCLK•1/128			
					0x6	PCLK•1/64			
					0x5	PCLK•1/32			
					0x4	PCLK•1/16			
					0x3	PCLK•1/8			
					0x2	PCLK•1/4			
					0x1	PCLK•1/2			
					0x0	PCLK•1/1			

D[15:4] Reserved

D[3:0] DF[3:0]: Timer Input Clock Select Bits

These bits select the count clock of the 16-bit timer from 15 prescaler output clocks.

Table II.4.2.9.2 Selecting the Count Clock

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

(Default: 0x0)

Note: When setting the count clock, make sure the 16-bit timer counter is stopped.

0x4222: CLG_T16U0 Reload Data Register (CLG_T16U0_TR)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
CLG_T16U0 Reload Data Register (CLG_T16U0 _TR)	**0x4222** (16 bits)	D15–0	**TR[15:0]**	16-bit timer reload data TR15 = MSB TR0 = LSB	0x0 to 0xffff	0x0	R/W	

D[15:0] TR[15:0]: 16-bit Timer Reload Data Bits

Set the initial value for the counter. (Default: 0x0)

The reload data written in this register is preset to the respective counter when the timer is reset or when the counter underflows.

When starting the 16-bit timer after resetting, the timer counts down from the reload value. So the reload value and the input clock frequency determine the period of time from starting the timer until an underflow occurs (and between underflows). This makes it possible to obtain a desired wait time, a periodical interrupt interval, or programmable transfer clock for the serial interface.

0x4224: CLG_T16U0 Counter Data Register (CLG_T16U0_TC)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
CLG_T16U0 Counter Data Register (CLG_T16U0 _TC)	**0x4224** (16 bits)	D15–0	**TC[15:0]**	16-bit timer counter data TC15 = MSB TC0 = LSB	0x0 to 0xffff	0xffff	R	

D[15:0] TC[15:0]: 16-bit Timer Counter Data Bits

The counter data can be read from this register. (Default: 0xffff)

This is a read-only register, so the writing operation is invalid.

0x4226: CLG_T16U0 Control Register (CLG_T16U0_CTL)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
CLG_T16U0 Control Register (CLG_T16U0 _CTL)	**0x4226** (16 bits)	D15–5	–	reserved	–				–	–	0 when being read.
		D4	**TRMD**	Count mode select	1	One shot	0	Repeat	0	R/W	
		D3–2	–	reserved	–				–	–	0 when being read.
		D1	**PRESER**	Timer reset	1	Reset	0	Ignored	0	W	
		D0	**PRUN**	Timer run/stop control	1	Run	0	Stop	0	R/W	

D[15:5] Reserved

D4 TRMD: Count Mode Select Bit

Selects the count mode of the 16-bit timer.

1 (R/W): One-shot mode

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

The 16-bit timer is set in repeat mode when TRMD is set to 0. In this mode, the 16-bit timer does not stop after it starts counting until the application program stops the timer. When the counter underflows, the timer presets the reload data register value to the counter and continues counting. The timer outputs the underflow pulses periodically. Set the 16-bit timer in this mode when generating periodical interrupts with a given interval or generating the serial transfer clock.

The 16-bit timer is set in one-shot mode when TRMD is set to 1. In this mode, the 16-bit timer automatically stops counting when the counter underflows, so only one interrupt can be generated after starting the timer. When an underflow occurs, the counter is preset with the reload data register value before the timer operation stops. Set the 16-bit timer in this mode when a certain waiting time must be generated.

Note: When setting the count mode, make sure the 16-bit timer counter is stopped.

D[3:2] Reserved

D1 PRESER: Timer Reset Bit

Resets the 16-bit timer.

1 (W): Reset

0 (W): Has no effect

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

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

D0 PRUN: Timer Run/Stop Control Bit

Controls the timer's Run/Stop state.

1 (R/W): Run

0 (R/W): Stop (default)

The timer starts counting by writing 1 to PRUN and stops counting by writing 0.
In the stop state, the counter data is retained until the timer is reset or placed in a run state.

II.4.2.10 Precautions

- Before the 16-bit timer can start counting, the prescaler must be run.
- When setting the count clock or count mode, make sure the 16-bit timer is turned off.

II.4.3 8-bit Timer with Fine Mode (CLG_T8FU0)

II.4.3.1 Outline of the 8-bit Timer with Fine Mode

The S1C17002 incorporates one channel of 8-bit timer with fine mode (CLG_T8FU0).
The timer includes an 8-bit presettable down counter and an 8-bit reload data register for setting 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 the clock for UART CH.0. The underflow period can be programmed by selecting a prescaler clock and setting reload data. This allows the application program to get any desired time intervals and programmable serial transfer rates. The fine mode provides a function to minimize transfer rate error.
Figure II.4.3.1.1 shows the structure of the 8-bit timer.

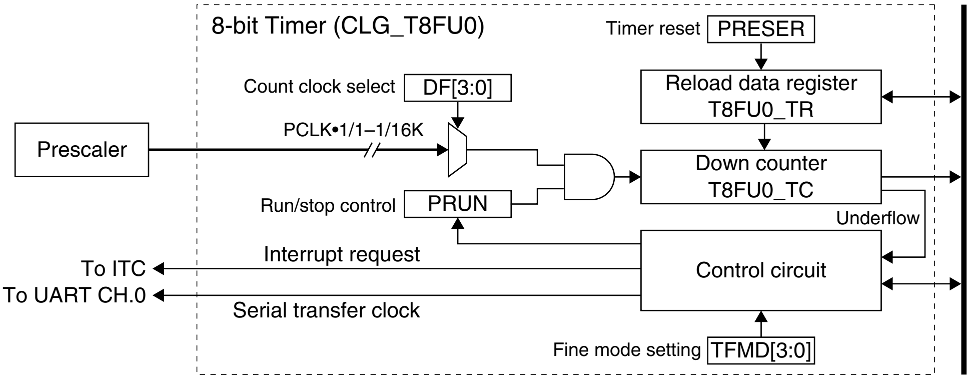


Figure II.4.3.1.1 Structure of 8-bit Timer

Note: Either the CLG_T16U0 clock or the CLG_T8FU0 clock can be selected in the UART module as the serial clock.

II.4.3.2 Count Mode

The 8-bit timer with fine mode has two count modes: repeat mode and one-shot mode. It can be selected using the TRMD bit (D4/CLG_T8FU0_CTL register).

* **TRMD**: Count Mode Select Bit in the CLG_T8FU0 Control (CLG_T8FU0_CTL) Register (D4/0x4226)

Repeat mode (TRMD = 0, default)

The timer is set in repeat mode when TRMD is set to 0.
In this mode, the timer does not stop after it starts counting until the application program stops the timer. When the counter underflows, the timer presets the reload data register value to the counter and continues counting. The timer outputs the underflow pulses periodically. Set the timer in this mode when generating periodical interrupts with a given interval or generating the serial transfer clock.

One-shot mode (TRMD = 1)

The timer is set in one-shot mode when TRMD is set to 1.
In this mode, the timer automatically stops counting when the counter underflows, so only one interrupt can be generated after starting the timer. When an underflow occurs, the counter is preset with the reload data register value before the timer operation stops. Set the timer in this mode when a certain waiting time must be generated.

Note: When setting the count mode, make sure the timer counter is stopped.

II.4.3.3 Count Clock

The 8-bit timer with fine mode uses a prescaler output clock as the count clock. The prescaler divides the PCLK clock by 1 to 16K to generate 15 clocks. Select one of the prescaler output clocks using the DF[3:0] bits (D[3:0]/CLG_T8FU0_CLK register).

∗ **DF[3:0]**: Timer Input Clock Select Bits in the CLG_T8FU0 Input Clock Select (CLG_T8FU0_CLK) Register (D[3:0]/0x4220)

Table II.4.3.3.1 Selecting the Count Clock

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

(Default: 0x0)

Notes:
- Before the timer can start counting, the prescaler must be run.
- When setting the count clock, make sure the timer counter is stopped.

For controlling the prescaler, see Section II.3, "Prescaler (PSC)."

II.4.3.4 Reload Register and Underflow Period

The Reload Data (CLG_T8FU0_TR) Register (0x4222) is used to set the initial value to the down counter.

The counter initial value set in the reload data register is preset to the down counter when the timer is reset or when the counter underflows. When starting the timer after resetting, the timer counts down from the reload value. So the reload value and the input clock frequency determine the period of time from starting the timer until an underflow occurs (and between underflows). This makes it possible to obtain a desired wait time, a periodical interrupt interval, or programmable transfer clock for the serial interface.

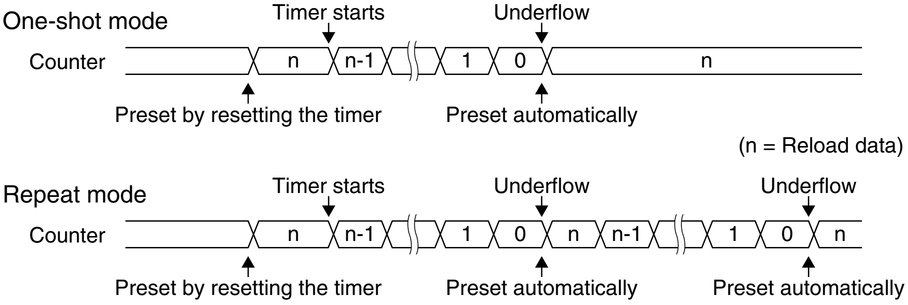


Figure II.4.3.4.1 Preset Timing

The underflow period is calculated by the expression below.

$$\text{Underflow period} = \frac{\text{TR} + 1}{\text{clk\_in}} \text{ [s]} \qquad \text{Underflow cycle} = \frac{\text{clk\_in}}{\text{TR} + 1} \text{ [Hz]}$$

clk_in: Count clock (prescaler output clock) frequency [Hz]
TR: Reload data (0–255)

Note: The UART divides the timer output by 16 to generate the sampling clock. Make sure of the division ratio when setting a transfer rate.

II.4.3.5 Resetting the 8-bit Timer with Fine Mode

To reset the timer, write 1 to the PRESER bit (D1/CLG_T8FU0_CTL register). This initializes the counter by presetting the Reload Data Register value.

∗ **PRESER**: Timer Reset Bit in the CLG_T8FU0 Control (CLG_T8FU0_CTL) Register (D1/0x4226)

II.4.3.6 Run/Stop Control of the 8-bit Timer with Fine Mode

Before starting the timer, set up the conditions as shown below.

(1) Select a count mode (one-shot or repeat). See Section II.4.3.2.
(2) Select the count clock (prescaler output clock). See Section II.4.3.3.
(3) Calculate the counter initial value and set it to the reload data register. See Section II.4.3.4.
(4) Reset the timer to preset the initial value to the counter. See Section II.4.3.5.
(5) Set up the interrupt level and enable the interrupt if the timer interrupt is used. See Section II.4.3.9.

To start the 8-bit timer, write 1 to the PRUN bit (D0/CLG_T8FU0_CTL register).

∗ **PRUN**: Timer Run/Stop Control Bit in the CLG_T8FU0 Control (CLG_T8FU0_CTL) Register (D0/0x4226)

The timer starts counting down from the initial value or the current counter value if the initial value has not been preset. When the counter underflows, the timer outputs an underflow pulse and presets the initial value again. At the same time, an interrupt request is sent to the interrupt controller (ITC).
If the timer is set in one-shot mode, the timer stops counting.
If the timer is set in repeat mode, the timer continues counting from the reloaded initial value.

To stop the timer from the application program, write 0 to the PRUN bit. The counter stops counting and holds the current counter value until the timer is reset or restarted. To restart counting from the initial value, reset the timer before writing 1 to the PRUN bit.

When the timer is reset during running, the timer loads the reload register value to the counter and continues counting.

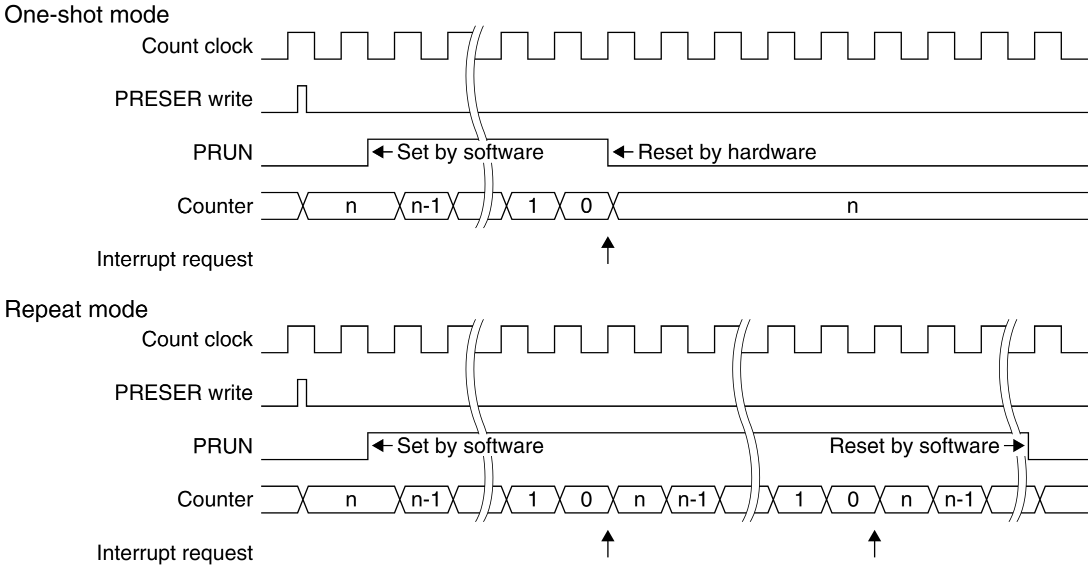


Figure II.4.3.6.1 Count Operation

II.4.3.7 Output Signal of the 8-bit Timer with Fine Mode

The timer outputs an underflow pulse when the counter underflows.
This pulse is used to request a timer interrupt.

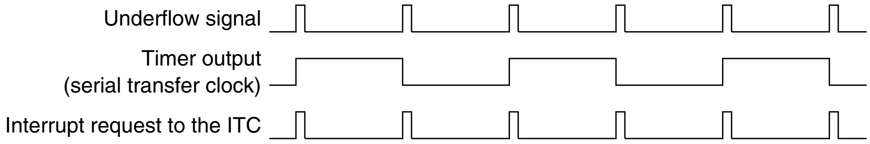


Figure II.4.3.7.1 Timer Output Clock

Also this pulse is used to generate a serial transfer clock and the clock is sent to UART CH.0.
The reload data register value to obtain a desired transfer rate is calculated by the expression below.

$$\text{bps} = \frac{\text{clk\_in}}{\{(\text{TR} + 1) \times 16 + \text{TFMD}\}}$$

$$\text{TR} = \left(\frac{\text{clk\_in}}{\text{bps}} - \text{TFMD} - 16\right) \div 16$$

clk_in: Count clock (prescaler output clock) frequency [Hz]
TR: Reload data (0–255)
bps: Transfer rate (bits/second)
TFMD: Fine mode setting value (0–15)

II.4.3.8 Fine Mode

The fine mode provides a function to minimize transfer rate error.
The 8-bit timer with fine mode can output a programmable clock used as the serial transfer clock for UART CH.0. By selecting an appropriate prescaler output clock and reload data, the timer output clock can be configured with the desired frequency. However, an error may be introduced depending on the transfer rate. In fine mode, the counter delays outputting the underflow pulse to prolong the output clock period. The amount of delay can be specified using the TFMD[3:0] bits (D[11:8]/CLG_T8FU0_CTL register).

* **TFMD[3:0]**: Fine Mode Setup Bits in the CLG_T8FU0 Control (CLG_T8FU0_CTL) Register (D[11:8]/0x4226)

The TFMD[3:0] bits specify a pattern of delays to be inserted in a 16-underflow period. The output clock period will be prolonged for one count clock period per one delay inserted. Also this setting will delay interrupt timings.

Table II.4.3.8.1 Delay Patterns Specified with TFMD[3:0]

TFMD[3:0]	Underflow number															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0x0	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
0x1	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	D
0x2	–	–	–	–	–	–	–	D	–	–	–	–	–	–	–	D
0x3	–	–	–	–	–	–	–	D	–	–	–	D	–	–	–	D
0x4	–	–	–	D	–	–	–	D	–	–	–	D	–	–	–	D
0x5	–	–	–	D	–	–	–	D	–	–	–	D	–	D	–	D
0x6	–	–	–	D	–	D	–	D	–	–	–	D	–	D	–	D
0x7	–	–	–	D	–	D	–	D	–	D	–	D	–	D	–	D
0x8	–	D	–	D	–	D	–	D	–	D	–	D	–	D	–	D
0x9	–	D	–	D	–	D	–	D	–	D	–	D	–	D	D	D
0xa	–	D	–	D	–	D	D	D	–	D	–	D	–	D	D	D
0xb	–	D	–	D	–	D	D	D	–	D	D	D	–	D	D	D
0xc	–	D	D	D	–	D	D	D	–	D	D	D	–	D	D	D
0xd	–	D	D	D	–	D	D	D	–	D	D	D	D	D	D	D
0xe	–	D	D	D	D	D	D	D	–	D	D	D	D	D	D	D
0xf	–	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D

D: Indicates that a delay is inserted.

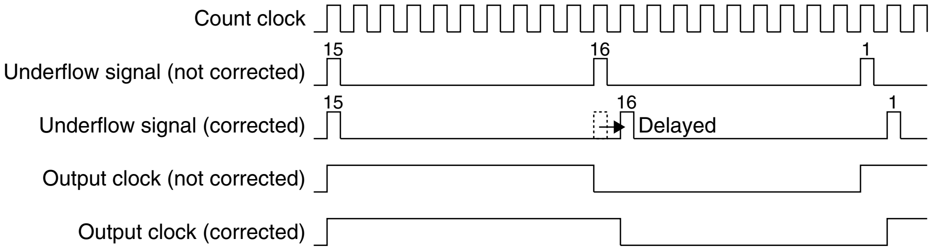

Count clock
Underflow signal (not corrected)
15
16
1
Underflow signal (corrected)
15
16
1
Delayed
Output clock (not corrected)
Output clock (corrected)

Figure II.4.3.8.1 Delay Cycle Insertion in Fine Mode

At initial reset, TFMD[3:0] is set to 0x0. No delay will be inserted in this setting.

II.4.3.9 Interrupt of the 8-bit Timer with Fine Mode

The 8-bit timer with fine mode outputs an interrupt request signal to the interrupt controller (ITC) when the counter underflows.

To generate a timer underflow interrupt, set up the interrupt level and enable the interrupt using the ITC registers.

ITC registers for timer interrupts

The following shows the control bits of the ITC provided for the 8-bit timer with fine mode:

Interrupt flag IIFT1

∗ **IIFT1**: CLG_T8FU0 Timer Interrupt Flag Bit in the Interrupt Flag (ITC_IFLG) Register (D9/0x4300)

Interrupt enable bit IIEN1

∗ **IIEN1**: CLG_T8FU0 Timer Interrupt Enable Bit in the Interrupt Enable (ITC_EN) Register (D9/0x4302)

Interrupt level setup bits IILV1

∗ **IILV1[2:0]**: CLG_T8FU0 Timer Interrupt Level Bits in the Internal Interrupt Level Setup (ITC_ILV0) Register 0 (D[10:8]/0x430e)

When an underflow occurs in the timer, the corresponding interrupt flag is set to 1.

If the interrupt enable bit corresponding to that interrupt flag has been set to 1, the ITC sends an interrupt request to the S1C17 Core. To disable the timer interrupt, set the interrupt enable bit to 0.

The interrupt flag is always set to 1 by the timer underflow pulse, regardless of how the interrupt enable bit is set (even when set to 0).

The interrupt level setup bits set the interrupt level (0 to 7) of the timer interrupt.

An interrupt request to the S1C17 Core is accepted only when all the conditions described below are met.

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

For details on these interrupt control registers, as well as the device operation when an interrupt has occurred, see Section III.1, “Interrupt Controller (ITC).”

Interrupt vector

The following shows the vector number and vector address for the timer interrupt:

Vector number: 13 (0x0d)
Vector address: TTBR + 0x34

II.4.3.10 Details of Control Registers

Table II.4.3.10.1 Register List of the 8-bit Timer with Fine Mode

Address	Register name		Function
0x4220	CLG_T8FU0_CLK	CLG_T8FU0 Input Clock Select Register	Selects a prescaler output clock.
0x4222	CLG_T8FU0_TR	CLG_T8FU0 Reload Data Register	Sets reload data.
0x4224	CLG_T8FU0_TC	CLG_T8FU0 Counter Data Register	Counter data
0x4226	CLG_T8FU0_CTL	CLG_T8FU0 Control Register	Sets the timer mode and starts/stops the timer.

The following describes the register of the 8-bit timer with fine mode. These are all 16-bit registers.

Note: When setting the registers, be sure to write a 0, and not a 1, for all “reserved bits.”

0x4220: CLG_T8FU0 Input Clock Select Register (CLG_T8FU0_CLK)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
CLG_T8FU0 Input Clock Select Register (CLG_T8FU0_ CLK)	**0x4220** (16 bits)	D15–4	–	reserved	–		–	–	0 when being read.
		D3–0	**DF[3:0]**	Timer input clock select (Prescaler output clock)	DF[3:0]	Clock	0x0	R/W	
					0xf	reserved			
					0xe	PCLK•1/16384			
					0xd	PCLK•1/8192			
					0xc	PCLK•1/4096			
					0xb	PCLK•1/2048			
					0xa	PCLK•1/1024			
					0x9	PCLK•1/512			
					0x8	PCLK•1/256			
					0x7	PCLK•1/128			
					0x6	PCLK•1/64			
					0x5	PCLK•1/32			
					0x4	PCLK•1/16			
					0x3	PCLK•1/8			
					0x2	PCLK•1/4			
					0x1	PCLK•1/2			
					0x0	PCLK•1/1			

D[15:4] Reserved

D[3:0] DF[3:0]: Timer Input Clock Select Bits

These bits select the count clock of the timer from 15 prescaler output clocks.

Table II.4.3.10.2 Selecting the Count Clock

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

(Default: 0x0)

Note: When setting the count clock, make sure the timer counter is stopped.

0x4222: CLG_T8FU0 Reload Data Register (CLG_T8FU0_TR)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
CLG_T8FU0 Reload Data Register (CLG_T8FU0_ TR)	**0x4222** (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	**TR[7:0]**	CLG_T8FU0 reload data TR7 = MSB TR0 = LSB	0x0 to 0xff	0x0	R/W	

D[15:8] Reserved

D[7:0] TR[7:0]: CLG_T8FU0 Reload Data Bits

Set the initial value for the counter. (Default: 0x0)

The reload data written in this register is preset to the respective counter when the timer is reset or when the counter underflows.

When starting the timer after resetting, the timer counts down from the reload value. So the reload value and the input clock frequency determine the period of time from starting the timer until an underflow occurs (and between underflows). This makes it possible to obtain a desired wait time, a periodical interrupt interval, or programmable transfer clock for the serial interface.

0x4224: CLG_T8FU0 Counter Data Register (CLG_T8FU0_TC)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
CLG_T8FU0 Counter Data Register (CLG_T8FU0_ TC)	**0x4224** (16 bits)	D15–8	**–**	reserved	–	–	–	0 when being read.
		D7–0	**TC[7:0]**	CLG_T8FU0 counter data TC7 = MSB TC0 = LSB	0x0 to 0xff	0xff	R	

D[15:8] Reserved

D[7:0] TC[7:0]: CLG_T8FU0 Counter Data Bits

The counter data can be read from this register. (Default: 0xff)

This is a read-only register, so the writing operation is invalid.

0x4226: CLG_T8FU0 Control Register (CLG_T8FU0_CTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
CLG_T8FU0 Control Register (CLG_T8FU0_ CTL)	**0x4226** (16 bits)	D15–12	–	reserved	–	–	–	0 when being read.
		D11–8	**TFMD[3:0]**	Fine mode setup	0x0 to 0xf	0x0	R/W	Set a number of times to insert delay into a 16-underflow period.
		D7–5	–	reserved	–	–	–	0 when being read.
		D4	**TRMD**	Count mode select	1 One shot 0 Repeat	0	R/W	
		D3–2	–	reserved	–	–	–	0 when being read.
		D1	**PRESER**	Timer reset	1 Reset 0 Ignored	0	W	
		D0	**PRUN**	Timer run/stop control	1 Run 0 Stop	0	R/W	

D[15:12] Reserved

D[11:8] TFMD[3:0]: Fine Mode Setup Bits

Corrects transfer rate error. (Default: 0x0)

The TFMD[3:0] bits specify a pattern of delays to be inserted in a 16-underflow period. The output clock period will be prolonged for one count clock period per one delay inserted.

Table II.4.3.10.3 Delay Patterns Specified with TFMD[3:0]

TFMD[3:0]	Underflow number															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0x0	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
0x1	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	D
0x2	–	–	–	–	–	–	–	D	–	–	–	–	–	–	–	D
0x3	–	–	–	–	–	–	–	D	–	–	–	D	–	–	–	D
0x4	–	–	–	D	–	–	–	D	–	–	–	D	–	–	–	D
0x5	–	–	–	D	–	–	–	D	–	–	–	D	–	D	–	D
0x6	–	–	–	D	–	D	–	D	–	–	–	D	–	D	–	D
0x7	–	–	–	D	–	D	–	D	–	D	–	D	–	D	–	D
0x8	–	D	–	D	–	D	–	D	–	D	–	D	–	D	–	D
0x9	–	D	–	D	–	D	–	D	–	D	–	D	–	D	D	D
0xa	–	D	–	D	–	D	D	D	–	D	–	D	–	D	D	D
0xb	–	D	–	D	–	D	D	D	–	D	D	D	–	D	D	D
0xc	–	D	D	D	–	D	D	D	–	D	D	D	–	D	D	D
0xd	–	D	D	D	–	D	D	D	–	D	D	D	D	D	D	D
0xe	–	D	D	D	D	D	D	D	–	D	D	D	D	D	D	D
0xf	–	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D

D: Indicates that a delay is inserted.

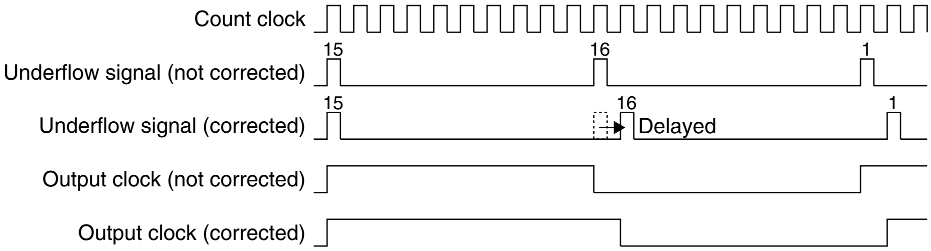

Count clock
15
16
1
Underflow signal (not corrected)
15
16
Delayed
1
Underflow signal (corrected)
Output clock (not corrected)
Output clock (corrected)

Figure II.4.3.10.1 Delay Cycle Inserted in Fine Mode

D[7:5] Reserved

D4 TRMD: Count Mode Select Bit

Selects the count mode of the timer.
1 (R/W): One-shot mode
0 (R/W): Repeat mode (default)

The timer is set in repeat mode when TRMD is set to 0. In this mode, the timer does not stop after it starts counting until the application program stops the timer. When the counter underflows, the timer presets the reload data register value to the counter and continues counting. The timer outputs the underflow pulses periodically. Set the timer in this mode when generating periodical interrupts with a given interval or generating the serial transfer clock.

The timer is set in one-shot mode when TRMD is set to 1. In this mode, the timer automatically stops counting when the counter underflows, so only one interrupt can be generated after starting the timer. When an underflow occurs, the counter is preset with the reload data register value before the timer operation stops. Set the timer in this mode when a certain waiting time must be generated.

Note: When setting the count mode, make sure the timer counter is stopped.

D[3:2] Reserved

D1 PRESER: Timer Reset Bit

Resets the timer.
1 (W): Reset
0 (W): Has no effect
0 (R): Always 0 when read (default)

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

D0 PRUN: Timer Run/Stop Control Bit

Controls the timer's Run/Stop state.
1 (R/W): Run
0 (R/W): Stop (default)

The timer starts counting by writing 1 to PRUN and stops counting by writing 0.
In the stop state, the counter data is retained until the timer is reset or placed in a run state.

II.4.3.11 Precautions

- Before the 8-bit timer with fine mode can start counting, the prescaler must be run.
- When setting the count clock or count mode, make sure the timer is turned off.

II.4.4 8-bit Timers (CLG_T8S and CLG_T8I)

II.4.4.1 Outline of the 8-bit Timers

The S1C17002 CLG incorporates two channels of 8-bit timers (CLG_T8S and CLG_T8I).

The 8-bit timer includes an 8-bit presettable down counter and an 8-bit reload data register for setting 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 period can be programmed by selecting a prescaler clock and setting reload data. This allows the application program to get any desired time intervals and programmable serial transfer rates.

Normally, CLG_T8S is used to generate the SPI CH.0 operating clock and CLG_T8I is used to generate the I^{2}C master operating clock.

Figure II.4.4.1.1 shows the structure of the 8-bit timers.

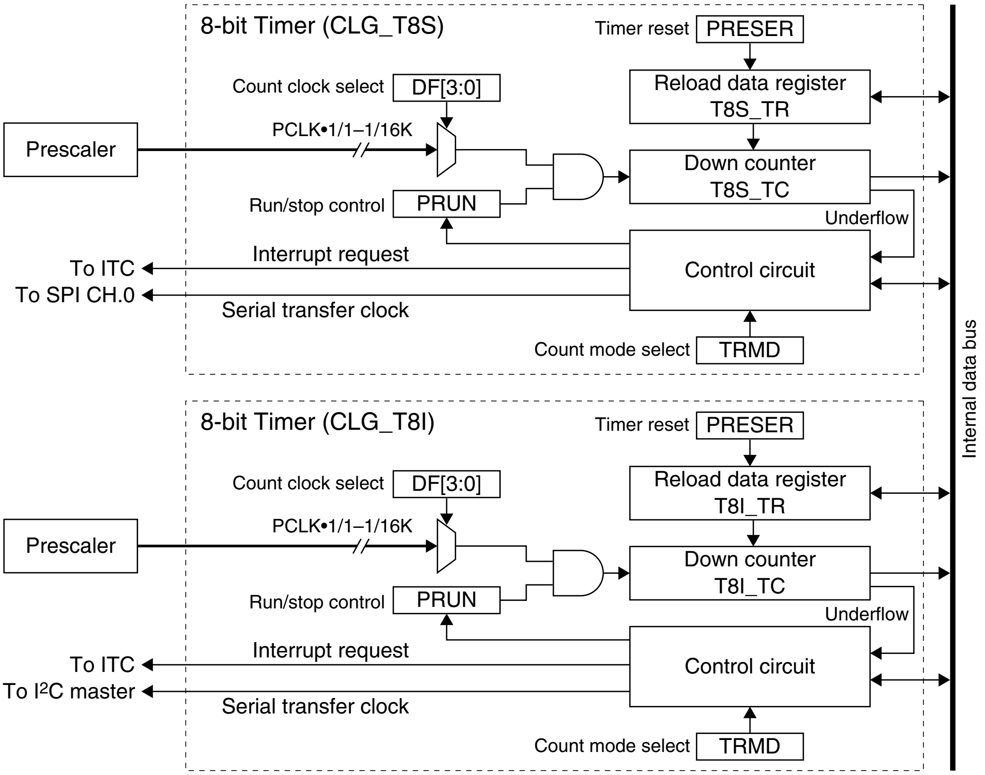


Figure II.4.4.1.1 Structure of 8-bit Timers

Note: The descriptions in this section apply to all 8-bit timers because the they have the same functions except for the control register addresses. The '*x*' in the register names denotes a timer channel (S or I) and the register addresses are described as (CLG_T8S/CLG_T8I).
Example: CLG_T8*x*_CTL register (0x4246/0x4266)
x = S: CLG_T8S_CTL register (0x4246)
x = I: CLG_T8I_CTL register (0x4266)

II.4.4.2 Count Mode

The 8-bit timer has two count modes: repeat mode and one-shot mode. It can be selected using TRMD (D4/CLG_T8*x*_CTL register).

∗ **TRMD**: Count Mode Select Bit in the CLG_T8*x* Control (CLG_T8*x*_CTL) Registers (D4/0x4246/0x4266)

Repeat mode (TRMD = 0, default)

The 8-bit timer is set in repeat mode when TRMD is set to 0.

In this mode, the 8-bit timer does not stop after it starts counting until the application program stops the timer. When the counter underflows, the timer presets the reload data register value to the counter and continues counting. The timer outputs the underflow pulses periodically. Set the 8-bit timer in this mode when generating periodical interrupts with a given interval or generating the serial transfer clock.

One-shot mode (TRMD = 1)

The 8-bit timer is set in one-shot mode when TRMD is set to 1.

In this mode, the 8-bit timer automatically stops counting when the counter underflows, so only one interrupt can be generated after starting the timer. When an underflow occurs, the counter is preset with the reload data register value before the timer operation stops. Set the 8-bit timer in this mode when a certain waiting time must be generated.

Note: When setting the count mode, make sure the 8-bit timer counter is stopped.

II.4.4.3 Count Clock

The 8-bit timer uses a prescaler output clock as the count clock. The prescaler divides PCLK by 1 to 16K to generate 15 clocks. Select one of the prescaler output clocks using DF[3:0] (D[3:0]/CLG_T8*x*_CLK register).

∗ **DF[3:0]**: Timer Input Clock Select Bits in the CLG_T8*x* Input Clock Select (CLG_T8*x*_CLK) Registers (D[3:0]/0x4240/0x4260)

Table II.4.4.3.1 Selecting the Count Clock

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

(Default: 0x0)

Notes:
- Before the 8-bit timer can start counting, the prescaler must be run.
- When setting the count clock, make sure the 8-bit timer counter is stopped.

For controlling the prescaler, refer to Section II.3, “Prescaler (PSC).”

II.4.4.4 8-bit Timer Reload Register and Underflow Period

The Reload Data (CLG_T8*x*_TR) Register (0x4242/0x4262) is used to set the initial value to the down counter. The counter initial value set in the reload data register is preset to the down counter when the 8-bit timer is reset or when the counter underflows. When starting the 8-bit timer after resetting, the timer counts down from the reload value. So the reload value and the input clock frequency determine the period of time from starting the timer until an underflow occurs (and between underflows). This makes it possible to obtain a desired wait time, a periodical interrupt interval, or programmable transfer clock for the serial interface.

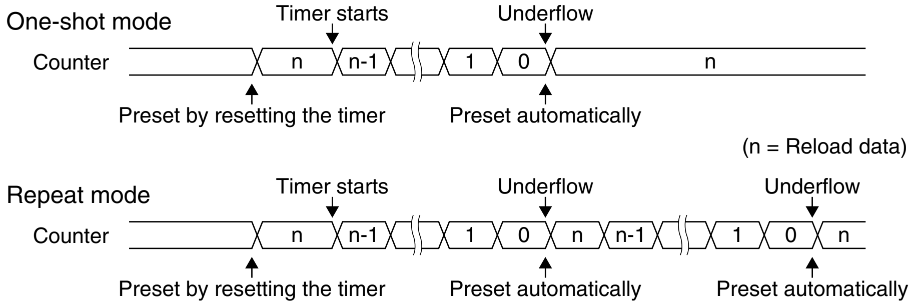


Figure II.4.4.4.1 Preset Timing

The underflow period is calculated by the expression below.

$$\text{Underflow period} = \frac{\text{TR} + 1}{\text{clk\_in}}\ [\text{s}] \qquad \text{Underflow cycle} = \frac{\text{clk\_in}}{\text{TR} + 1}\ [\text{Hz}]$$

clk_in: Count clock (prescaler output clock) frequency [Hz]
TR: Reload data (0–255)

II.4.4.5 Resetting the 8-bit Timer

To reset the 8-bit timer, write 1 to PRESER (D1/CLG_T8*x*_CTL register). This initializes the counter by presetting the Reload Data Register value.

∗ **PRESER**: Timer Reset Bit in the CLG_T8*x* Control (CLG_T8*x*_CTL) Registers (D1/0x4246/0x4266)

II.4.4.6 8-bit Timer Run/Stop Control

Before starting the 8-bit timer, set up the conditions as shown below.

(1) Select a count mode (one-shot or repeat). See Section II.4.4.2.
(2) Select the count clock (prescaler output clock). See Section II.4.4.3.
(3) Calculate the counter initial value and set it to the reload data register. See Section II.4.4.4
(4) Reset the timer to preset the initial value to the counter. See Section II.4.4.5.
(5) Set up the interrupt level and enable the interrupt of the timer channel if the timer interrupt is used. See Section II.4.4.8.

To start the 8-bit timer, write 1 to PRUN (D0/CLG_T8*x*_CTL register).

∗ **PRUN**: Timer Run/Stop Control Bit in the CLG_T8*x* Control (CLG_T8*x*_CTL) Registers (D0/0x4246/0x4266)

The timer starts counting down from the initial value or the current counter value if the initial value has not been preset. When the counter underflows, the timer outputs an underflow pulse and presets the initial value again. At the same time, an interrupt request is sent to the interrupt controller (ITC).
If the timer is set in one-shot mode, the timer stops counting.
If the timer is set in repeat mode, the timer continues counting from the reloaded initial value.

To stop the 8-bit timer from the application program, write 0 to the PRUN bit. The counter stops counting and holds the current counter value until the timer is reset or restarted. To restart counting from the initial value, reset the timer before writing 1 to the PRUN bit.

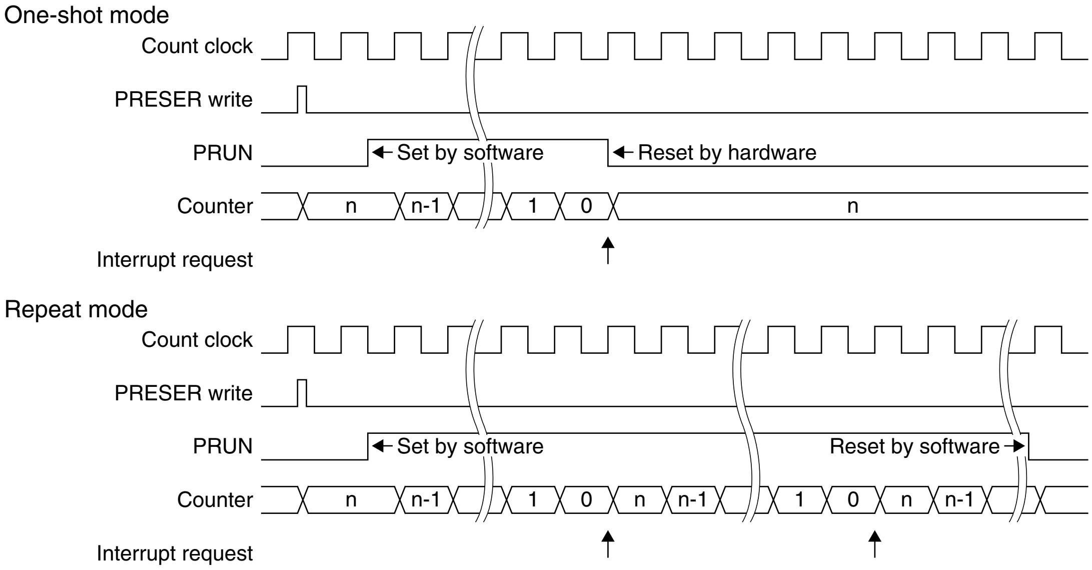


Figure II.4.4.6.1 Count Operation

II.4.4.7 8-bit Timer Output Signal

The 8-bit timer outputs an underflow pulse when the counter underflows.
This pulse is used to request a timer interrupt.
Also this pulse is used to generate a serial transfer clock for the internal serial interface.

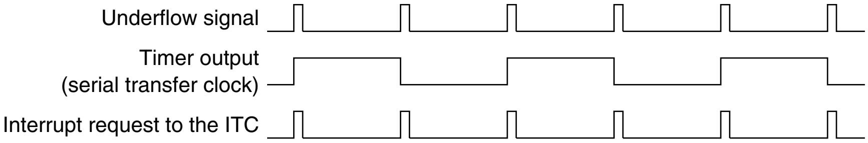


Figure II.4.4.7.1 Timer Output Clock

The generated clocks are sent to the internal serial interfaces as below.
CLG_T8S output clock → SPI
CLG_T8I output clock → I^2C master

The reload data register value to obtain a desired transfer rate is calculated by the expression below.

$$TR = \frac{clk\_in}{bps \times 2} - 1$$

clk_in: Count clock (prescaler output clock) frequency [Hz]
TR: Reload data (0–255)
bps: Transfer rate (bits/second)

II.4.4.8 8-bit Timer Interrupt

The 8-bit timer outputs an interrupt request signal to the interrupt controller (ITC) when the counter underflows. To generate a timer underflow interrupt, set up the interrupt level and enable the interrupt using the ITC registers.

ITC registers for timer interrupts

Table II.4.4.8.1 shows the control registers of the ITC provided for each timer channel.

Table II.4.4.8.1 ITC Registers

Timer	Interrupt flag	Interrupt enable bit	Interrupt level setup bits
CLG_T8S	IIFT2 (D10/ITC_IFLG)	IIEN2 (D10/ITC_EN)	IILV2[2:0] (D[2:0]/ITC_ILV1)
CLG_T8I	IIFT3 (D11/ITC_IFLG)	IIEN3 (D11/ITC_EN)	IILV3[2:0] (D[10:8]/ITC_ILV1)

ITC_IFLG register (0x4300)
ITC_EN register (0x4302)
ITC_ILV1 register (0x4310)

When an underflow occurs in the timer, the corresponding interrupt flag is set to 1.

If the interrupt enable bit corresponding to that interrupt flag has been set to 1, the ITC sends an interrupt request to the S1C17 Core. To disable the timer interrupt, set the interrupt enable bit to 0.

The interrupt flag is always set to 1 by the timer underflow pulse, regardless of how the interrupt enable bit is set (even when set to 0).

The interrupt level setup bits set the interrupt level (0 to 7) of the timer interrupt. If the same interrupt level is set, timer Ch.0 has highest priority and timer Ch.2 has lowest priority.

An interrupt request to the S1C17 Core is accepted only when all the conditions described below are met.

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

For details on these interrupt control registers, as well as the device operation when an interrupt has occurred, see Section III.1, “Interrupt Controller (ITC).”

Interrupt vectors

The following shows the vector numbers and vector addresses for the timer interrupt:

Table II.4.4.8.2 Timer Interrupt Vectors

Timer	Vector number	Vector address
CLG_T8S	14 (0x0e)	TTBR + 0x38
CLG_T8I	15 (0x0f)	TTBR + 0x3c

II.4.4.9 Details of Control Registers

Table II.4.4.9.1 List of 8-bit Timer Registers

Address	Register name		Function
0x4240	CLG_T8S_CLK	CLG_T8S Input Clock Select Register	Selects a prescaler output clock.
0x4242	CLG_T8S_TR	CLG_T8S Reload Data Register	Sets reload data.
0x4244	CLG_T8S_TC	CLG_T8S Counter Data Register	Counter data
0x4246	CLG_T8S_CTL	CLG_T8S Control Register	Sets the timer mode and starts/stops the timer.
0x4260	CLG_T8I_CLK	CLG_T8I Input Clock Select Register	Selects a prescaler output clock.
0x4262	CLG_T8I_TR	CLG_T8I Reload Data Register	Sets reload data.
0x4264	CLG_T8I_TC	CLG_T8I Counter Data Register	Counter data
0x4266	CLG_T8I_CTL	CLG_T8I Control Register	Sets the timer mode and starts/stops the timer.

The following describes each 8-bit timer register. These are all 16-bit registers.

Note: When setting the registers, be sure to write a 0, and not a 1, for all “reserved bits.”

0x4240/0x4260: CLG_T8*x* Input Clock Select Registers (CLG_T8*x*_CLK)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
CLG_T8*x* Input Clock Select Register (CLG_T8*x*_CLK)	**0x4240 0x4260** (16 bits)	D15–4	–	reserved	–		–	–	0 when being read.
		D3–0	**DF[3:0]**	Timer input clock select (Prescaler output clock)	DF[3:0]	Clock	0x0	R/W	
					0xf	reserved			
					0xe	PCLK•1/16384			
					0xd	PCLK•1/8192			
					0xc	PCLK•1/4096			
					0xb	PCLK•1/2048			
					0xa	PCLK•1/1024			
					0x9	PCLK•1/512			
					0x8	PCLK•1/256			
					0x7	PCLK•1/128			
					0x6	PCLK•1/64			
					0x5	PCLK•1/32			
					0x4	PCLK•1/16			
					0x3	PCLK•1/8			
					0x2	PCLK•1/4			
					0x1	PCLK•1/2			
					0x0	PCLK•1/1			

Note: The letter '*x*' in register names, etc., denotes a timer channel (S or I).

0x4240: CLG_T8S Input Clock Select Register (CLG_T8S_CLK)
0x4260: CLG_T8I Input Clock Select Register (CLG_T8I_CLK)

D[15:4] Reserved

D[3:0] DF[3:0]: Timer Input Clock Select Bits

These bits select the count clock of the 8-bit timer from 15 prescaler output clocks.

Table II.4.4.9.2 Selecting the Count Clock

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

(Default: 0x0)

Note: When setting the count clock, make sure the 8-bit timer counter is stopped.

0x4242/0x4262: CLG_T8*x* Reload Data Registers (CLG_T8*x*_TR)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
CLG_T8*x* Reload Data Register (CLG_T8*x*_TR)	**0x4242 0x4262** (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	**TR[7:0]**	8-bit timer reload data TR7 = MSB TR0 = LSB	0x0 to 0xff	0x0	R/W	

Note: The letter '*x*' in register names, etc., denotes a timer channel (S or I).

0x4242: CLG_T8S Reload Data Register (CLG_T8S_TR)
0x4262: CLG_T8I Reload Data Register (CLG_T8I_TR)

D[15:8] Reserved

D[7:0] TR[7:0]: 8-bit Timer Reload Data Bits

Set the initial value for the counter. (Default: 0x0)

The reload data written in this register is preset to the respective counter when the timer is reset or when the counter underflows.

When starting the 8-bit timer after resetting, the timer counts down from the reload value. So the reload value and the input clock frequency determine the period of time from starting the timer until an underflow occurs (and between underflows). This makes it possible to obtain a desired wait time, a periodical interrupt interval, or programmable transfer clock for the serial interface.

0x4244/0x4264: CLG_T8*x* Counter Data Registers (CLG_T8*x*_TC)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
CLG_T8*x* Counter Data Register (CLG_T8*x*_TC)	**0x4244 0x4264** (16 bits)	D15–8	**–**	reserved	–	–	–	0 when being read.
		D7–0	**TC[7:0]**	8-bit timer counter data TC7 = MSB TC0 = LSB	0x0 to 0xff	0xff	R	

Note: The letter '*x*' in register names, etc., denotes a timer channel (S or I).

0x4244: CLG_T8S Counter Data Register (CLG_T8S_TC)
0x4264: CLG_T8I Counter Data Register (CLG_T8I_TC)

D[15:8] Reserved

D[7:0] TC[7:0]: 8-bit Timer Counter Data Bits

The counter data can be read from this register. (Default: 0xff)
This is a read-only register, so the writing operation is invalid.

0x4246/0x4266: CLG_T8*x* Control Registers (CLG_T8*x*_CTL)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
CLG_T8*x* Control Register (CLG_T8*x*_CTL)	**0x4246 0x4266** (16 bits)	D15–5	–	reserved	–				–	–	0 when being read.
		D4	**TRMD**	Count mode select	1	One shot	0	Repeat	0	R/W	
		D3–2	–	reserved	–				–	–	0 when being read.
		D1	**PRESER**	Timer reset	1	Reset	0	Ignored	0	W	
		D0	**PRUN**	Timer run/stop control	1	Run	0	Stop	0	R/W	

Note: The letter '*x*' in register names, etc., denotes a timer channel (S or I).

0x4246: CLG_T8S Control Register (CLG_T8S_CTL)
0x4266: CLG_T8I Control Register (CLG_T8I_CTL)

D[15:5] Reserved

D4 TRMD: Count Mode Select Bit

Selects the count mode of the 8-bit timer.

1 (R/W): One-shot mode
0 (R/W): Repeat mode (default)

The 8-bit timer is set in repeat mode when TRMD is set to 0. In this mode, the 8-bit timer does not stop after it starts counting until the application program stops the timer. When the counter underflows, the timer presets the reload data register value to the counter and continues counting. The timer outputs the underflow pulses periodically. Set the 8-bit timer in this mode when generating periodical interrupts with a given interval or generating the serial transfer clock.

The 8-bit timer is set in one-shot mode when TRMD is set to 1. In this mode, the 8-bit timer automatically stops counting when the counter underflows, so only one interrupt can be generated after starting the timer. When an underflow occurs, the counter is preset with the reload data register value before the timer operation stops. Set the 8-bit timer in this mode when a certain waiting time must be generated.

Note: When setting the count mode, make sure the 8-bit timer counter is stopped.

D[3:2] Reserved

D1 PRESER: Timer Reset Bit

Resets the 8-bit timer.

1 (W): Reset
0 (W): Has no effect
0 (R): Always 0 when read (default)

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

D0 PRUN: Timer Run/Stop Control Bit

Controls the timer's Run/Stop state.

1 (R/W): Run
0 (R/W): Stop (default)

The timer starts counting by writing 1 to PRUN and stops counting by writing 0.
In the stop state, the counter data is retained until the timer is reset or placed in a run state.

II.4.4.10 Precautions

- Before the 8-bit timer can start counting, the prescaler must be run.
- When setting the count clock or count mode, make sure the 8-bit timer is turned off.

THIS PAGE IS BLANK.

II.5 Real-Time Clock (RTC)

II.5.1 Overview of the RTC

The S1C17002 incorporates a real-time clock (RTC) with a perpetual calendar, and an OSC1 oscillator circuit to generate the operating clock for the RTC.

The RTC and OSC1 oscillator circuit operate in SLEEP mode. Moreover, the RTC can periodically generate interrupt requests to the CPU.

The main features of the RTC are outlined below.

- Contains time counters (seconds, minutes, and hours) and calendar counters (days, days of the week, months, and year).
- BCD data can be read from and written to both counters.
- Capable of controlling the starting and stopping of time clocks.
- 24-hour or 12-hour mode can be selected.
- A 30-second correction function can be implemented in software.
- Periodic interrupts are possible.
- Interrupt period can be selected from 1/64 second, 1 second, 1 minute, or 1 hour, with selectable level/edge interrupts.
- A built-in OSC1 oscillator circuit (crystal oscillator or external clock input) that generates a 32.768-kHz (typ.) operating clock.

Figure II.5.1.1 shows a block diagram of the RTC.

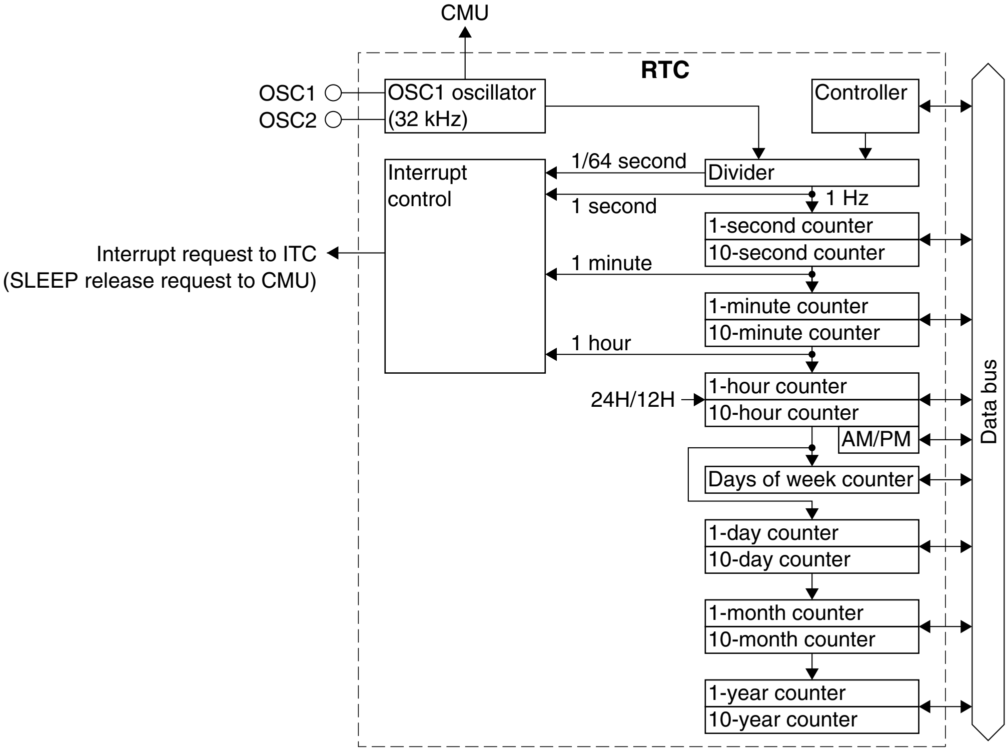


Figure II.5.1.1 RTC Block Diagram

II.5.2 RTC Counters

The RTC contains the following 13 counters, whose count values can be read out as BCD data from the respective registers. Each counter can also be set to any desired date and time by writing data to the respective register.

1-second counter

This 4-bit BCD counter counts in units of seconds. It counts from 0 to 9 synchronously with a 1-second signal derived from the 32.768-kHz OSC1 clock by dividing the clock into smaller frequencies. This counter is reset to 0 after 9 and outputs a carry over of 1 to the 10-second counter. The count data is read out and written using RTCSL[3:0] (D[3:0]/RTC_SEC register).

* **RTCSL[3:0]**: RTC 1-second Counter Bits in the RTC Second (RTC_SEC) Register (D[3:0]/0x4614)

10-second counter

This 3-bit BCD counter counts tens of seconds. It counts from 0 to 5 with 1 carried over from the 1-second counter. This counter is reset to 0 after 5 and outputs a carry over of 1 to the 1-minute counter. The count data is read out and written using RTCSH[2:0] (D[6:4]/RTC_SEC register).

* **RTCSH[2:0]**: RTC 10-second Counter Bits in the RTC Second (RTC_SEC) Register (D[6:4]/0x4614)

1-minute counter

This 4-bit BCD counter counts in units of minutes. It counts from 0 to 9 with 1 carried over from the 10-second counter. This counter is reset to 0 after 9 and outputs a carry over of 1 to the 10-minute counter. The count data is read out and written using RTCMIL[3:0] (D[3:0]/RTC_MIN register).

* **RTCMIL[3:0]**: RTC 1-minute Counter Bits in the RTC Minute (RTC_MIN) Register (D[3:0]/0x4615)

10-minute counter

This 3-bit BCD counter counts tens of minutes. It counts from 0 to 5 with 1 carried over from the 1-minute counter. This counter is reset to 0 after 5 and outputs a carry over of 1 to the 1-hour counter. The count data is read out and written using RTCMIH[2:0] (D[6:4]/RTC_MIN register).

* **RTCMIH[2:0]**: RTC 10-minute Counter Bits in the RTC Minute (RTC_MIN) Register (D[6:4]/0x4615)

1-hour counter

This 4-bit BCD counter counts in units of hours. It counts from 0 to 9 with 1 carried over from the 10-minute counter. This counter is reset to 0 after 9 and outputs a carry over of 1 to the 10-hour counter. Depending whether 12-hour or 24-hour mode is selected, the counter is reset at 12 o’clock or 24 o’clock. The count data is read out and written using RTCHL[3:0] (D[3:0]/RTC_HOUR register).

* **RTCHL[3:0]**: RTC 1-hour Counter Bits in the RTC Hour (RTC_HOUR) Register (D[3:0]/0x4616)

10-hour counter

This 2-bit BCD counter counts tens of hours. With a carry over of 1 from the 1-hour counter, this counter counts from 0 to 1 (when 12-hour mode is selected) or from 0 to 2 (when 24-hour mode is selected). The counter is reset at 12 o’clock or 24 o’clock, and outputs a carry over of 1 to the 1-day counter. The count data is read out and written using RTCHH[1:0] (D[5:4]/RTC_HOUR register).

* **RTCHH[1:0]**: RTC 10-hour Counter Bits in the RTC Hour (RTC_HOUR) Register (D[5:4]/0x4616)

When 12-hour mode is selected, RTCAP (D6/RTC_HOUR register) that indicates A.M. or P.M. is enabled, with A.M. and P.M. represented by 0 and 1, respectively. For 24-hour mode, RTCAP is fixed to 0.

* **RTCAP**: AM/PM Indicator Bit in the RTC Hour (RTC_HOUR) Register (D6/0x4616)

1-day counter

This 4-bit BCD counter counts in units of days. It counts from 0 to 9 with 1 carried over from the hour counter. This counter is reset to 0 after 9 and outputs a carry over of 1 to the 10-day counter. The number of days in each month and leap years are taken into account, so that the counter is reset to 1 when months change. The count data is read out and written using RTCDL[3:0] (D[3:0]/RTC_DAY register).

∗ **RTCDL[3:0]**: RTC 1-day Counter Bits in the RTC Day (RTC_DAY) Register (D[3:0]/0x4617)

10-day counter

This 2-bit BCD counter counts tens of days. It counts from 0 to 2 or 3 with 1 carried over from the 1-day counter. The number of days in each month and leap years are taken into account, so that when months change the counter is reset to 0 along with the 1-day counter, and outputs a carry over of 1 to the 1-month counter. The count data is read out and written using RTCDH[1:0] (D[5:4]/RTC_DAY register).

∗ **RTCDH[1:0]**: RTC 10-day Counter Bits in the RTC Day (RTC_DAY) Register (D[5:4]/0x4617)

1-month counter

This 4-bit BCD counter counts in units of months. It counts from 0 to 9 with 1 carried over from the day counter. This counter is reset to 0 after 9 and outputs a carry over of 1 to the 10-month counter. The counter is reset to 1 when years change. The count data is read out and written using RTCMOL[3:0] (D[3:0]/RTC_MONTH register).

∗ **RTCMOL[3:0]**: RTC 1-month Counter Bits in the RTC Month (RTC_MONTH) Register (D[3:0]/0x4628)

10-month counter

This counter counts in units of 10 months, and is set to 1 with 1 carried over from the 1-month counter. When years change, this counter is reset to 0 along with the 1-month counter, and outputs a carry over of 1 to the 1-year counter. The count data is read out and written using RTCMOH (D4/RTC_MONTH register).

∗ **RTCMOH**: RTC 10-month Counter Bit in the RTC Month (RTC_MONTH) Register (D4/0x4628)

1-year counter

This 4-bit BCD counter counts in units of years. It counts from 0 to 9 with 1 carried over from the month counter. This counter is reset to 0 after 9 and outputs a carry over of 1 to the 10-year counter. The count data is read out and written using RTCYL[3:0] (D[3:0]/RTC_YEAR register).

∗ **RTCYL[3:0]**: RTC 1-year Counter Bits in the RTC Year (RTC_YEAR) Register (D[3:0]/0x4629)

10-year counter

This 4-bit BCD counter counts tens of years. It counts from 0 to 9 with 1 carried over from the 1-year counter. The count data is read out and written using RTCYH[3:0] (D[7:4]/RTC_YEAR register).

∗ **RTCYH[3:0]**: RTC 10-year Counter Bits in the RTC Year (RTC_YEAR) Register (D[7:4]/0x4629)

Days of week counter

This is a septenary counter (that counts from 0 to 6) representing the days of the week. It counts with the same timing as the 1-day counter. The count data is read out and written using RTCWK[2:0] (D[2:0]/RTC_WEEK register).

∗ **RTCWK[2:0]**: RTC Days of Week Counter Bits in the RTC Days of Week (RTC_WEEK) Register (D[2:0]/0x462a)

The correspondence between the counter values and days of the week can be set in a program as desired. Table II.5.2.1 lists the basic correspondence.

Table II.5.2.1 Correspondence between Counter Values and Days of the Week

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

Initial counter values

When initially reset, the counter values are not initialized. After power-on, the counter values are indeterminate. Be sure to initialize the counters by following the procedure described in Section II.5.3.2, “Initial Sequence of the RTC.”

About detection of leap years

The algorithm used in the RTC to detect leap years is for Anno Domini (A.D.) only, and can automatically identify leap years up to the year 2399.

Years (0 to 99) without a remainder when divided by 4 are considered leap years. When the 1-year and 10-year counters both are 0, a common year is assumed.

II.5.3 Control of the RTC

II.5.3.1 Controlling the Operating Clock and Access Wait Cycle

Counter clock

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

Register clock

The RTC_SAPB_CLK clock is used to operate the RTC control registers. To setup the registers, this clock is required. After the registers are set up, the clock supply can be stopped to reduce current consumption by setting RTC_SAPB_CLK_EN (D0/CMU_GATEDCLK2 register) to 0.

∗ **RTC_SAPB_CLK_EN**: RTC SAPB I/F Clock Control Bit in the Gated Clock Control 2 (CMU_GATEDCLK2) Register (D0/0x4908)

Setting the wait cycles for accessing the RTC module

In order to access the RTC registers properly even if the system operates with a high-speed clock, the SRAMC can insert a wait cycle in the RTC access cycle. The number of system clock cycles to be inserted as a wait cycle can be specified using RTC_WAIT[2:0] (D[2:0]/RTC_WAIT register).

∗ **RTC_WAIT[2:0]**: RTC Access Wait Cycle Setup Bits in the RTC Wait Control (RTC_WAIT) Register (D[2:0]/0x5018)

Table II.5.3.1.1 Number of Wait Cycles during RTC Access

RTC_WAIT[2:0]	Number of wait cycles
0x7	7 cycles
0x6	6 cycles
0x5	5 cycles
0x4	4 cycles
0x3	3 cycles
0x2	2 cycles
0x1	1 cycle
0x0	0 cycles

(Default: 0x7)

The S1C17002 is able to operate with RTC_WAIT[2:0] ≥ 1.

II.5.3.2 Initial Sequence of the RTC

Immediately after power-on, the contents of RTC registers are indeterminate. After powering on, follow the procedure below to let the RTC start ticking the time. Later sections detail the contents of each control.

1. Power-on
2. System initialization processing and waiting for OSC1 stabilization
   Although the OSC1 oscillator circuit starts oscillating immediately after power is switched on, a finite time of up to 3 seconds is required before the output clock stabilizes.
3. Disabling RTC interrupts
   To prevent the occurrence of unwanted RTC interrupts, the following register settings are required:
   Write 0x0 to the RTC Interrupt Mode Register (0x4601) to disable RTC interrupts.
   Write 0x1 to the RTC Interrupt Status Register (0x4600) to clear the RTC interrupt status.
   For details, see Section II.5.4, “RTC Interrupts.”
4. Starting the count
   Write 0x2 (for 12-hour mode) or 0x12 (for 24-hour mode) to the RTC Control 0 Register (0x4602) to start counting by the RTC. This operation initializes the contents of 12-hour/24-hour mode, etc. that affect count data when settings are changed, and is not the standard operation to start counting.
   For details, see Section II.5.3.3, “Selecting 12/24-hour Mode and Setting the Counters,” and Section II.5.3.4, “Starting, Stopping, and Resetting Counters.”
5. Confirming accessibility status of the RTC
   Use the RTC Control 1 Register (0x4603) to retain the counters intact and read out the busy flag to confirm that the RTC can now be accessed.
   For details, see Section II.5.3.5, “Counter Hold and Busy Flag.”
6. Stopping and resetting the count
   Write 0x1 to the RTC Control 0 Register (0x4602) to stop the count, then reset the divide-by stage of the count clock.
   For details, see Section II.5.3.4, “Starting, Stopping, and Resetting Counters.”
7. Setting the date and time
   Use the respective count registers to initialize all counters to the current date and time.
   For details, see Section II.5.3.3, “Selecting 12/24-hour Mode and Setting the Counters.”
8. Restarting count
   Release the counters from the hold state (set in step 5) and repeat step 4 to restart counting by the RTC.
   For details, see Section II.5.3.5, “Counter Hold and Busy Flag,” and Section II.5.3.4, “Starting, Stopping, and Resetting Counters.”

II.5.3.3 Selecting 12/24-hour Mode and Setting the Counters

Selecting 12-hour/24-hour mode

Whether to use the time clock in 12-hour or 24-hour mode can be selected using RTC24H (D4/RTC_CNTL0 register).
RTC24H = 1: 24-hour mode
RTC24H = 0: 12-hour mode
The count range of hour counters changes with this selection.

∗ **RTC24H**: 24H/12H Mode Select Bit in the RTC Control 0 (RTC_CNTL0) Register (D4/0x4602)

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

Note: Rewriting RTC24H may corrupt count data for the hours, days, months, years or days of the week. Therefore, once RTC24H settings are changed, be sure to set data back in these counters again.

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

When 12-hour mode is selected, RTCAP (D6/RTC_HOUR register) that indicates A.M. or P.M. is enabled.
RTCAP = 0: A.M.
RTCAP = 1: P.M.
For 24-hour mode, RTCAP is fixed to 0.

∗ **RTCAP**: AM/PM Indicator Bit in the RTC Hour (RTC_HOUR) Register (D6/0x4616)

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

Setting the counters

Idle counters can be accessed for read or write at any time.
However, settings like those shown below should be avoided, since such settings may cause timekeeping errors.

- Settings exceeding the effective range
  Do not set count data exceeding 60 seconds, 60 minutes, 12 or 24 hours, 31 days, 12 months, or 99 years.
- Settings nonexistent in the calendar
  Do not set such nonexistent dates as April 31 or February 29, 2006. Even if such settings are made, the counters operate normally, so that when 1 is carried over from the hour counter to the 1-day counter, the day counter counts up to the first day of the next month. (For April 31, the day counter counts up to May 1; for February 29, 2006, the day counter counts up to March 1, 2006.)

If any counter must be rewritten while operating, there is a procedure that must be followed to ensure that the counter is rewritten correctly. For details, see Section II.5.3.6, “Reading from and Writing to Counters in Operation.”

II.5.3.4 Starting, Stopping, and Resetting Counters

Starting and stopping counters

The RTC starts counting when RTCSTP (D1/RTC_CNTL0 register) is set to 0, and stops counting when this bit is set to 1.

∗ **RTCSTP**: Counter Run/Stop Control Bit in the RTC Control 0 (RTC_CNTL0) Register (D1/0x4602)

The RTC is stopped by writing 1 to RTCSTP at the 32-kHz input clock divide-by stage of 8,192 Hz or those stages that follow. The RTC does not stop at up to the input clock divide-by-2 stage (16,384 Hz).
If the RTC stops counting when 1 is carried over to the next-digit counter, the count value may be corrupted. Therefore, see the next section to ensure that 1 is not carried over when counters are made to stop. This is unnecessary, however, when the contents of all counters are newly set again.

Resetting the counters

RTCRST (D0/RTC_CNTL0 register) is the bit used to reset the 32 kHz to 2 Hz counters.

∗ **RTCRST**: Software Reset Bit in the RTC Control 0 (RTC_CNTL0) Register (D0/0x4602)

Setting RTCRST to 1 resets the counters above (cleared to 0), and writing 0 to this bit negates the reset.

II.5.3.5 Counter Hold and Busy Flag

If 1 is carried over when reading the counters, the correct counter value may not be read out. Moreover, if a write or stop operation is attempted, the counter value may be corrupted. Therefore, whether counters are in a carry (busy) state should be checked before reading or writing data from or to the count registers. For this purpose, control bits RTCBSY (D1/RTC_CNTL1 register) and RTCHLD (D0/RTC_CNTL1 register) are provided.

∗ **RTCBSY**: Counter Busy Flag Bit in the RTC Control 1 (RTC_CNTL1) Register (D1/0x4603)
∗ **RTCHLD**: Counter Hold Control Bit in the RTC Control 1 (RTC_CNTL1) Register (D0/0x4603)

RTCBSY is a read-only flag indicating that 1 is being carried over. RTCBSY is set to 1 when 1 is being carried over; otherwise, it is 0. RTCBSY should be confirmed as being 0 before accessing the counters to ensure that the correct value will be read or set.
Note, however, that RTCBSY is fixed to 1 while counting is in progress. To reflect the current state in the count value, RTCHLD should be set to 1.

RTCBSY = 0 (RTC accessible)

If the value of RTCBSY is 0 when this bit is read out after writing 1 to RTCHLD, it means that 1 is not being carried over. In this case, the counter hold function is actuated, with a carry over of 1 to the 1-second counter disabled in hardware. Counters that count less than seconds continue operating.
Data can be read from or written to the count registers in this state.
After reading or writing data, reset RTCHLD to 0.

When 1 must be carried over while data is being read or written with counters in the hold state, 1 second is automatically added at the time, with RTCHLD reset to 0 to correct the count value. This correction is effective for only 1 second, and the time to carry over 1 on subsequent occasions is ignored. In this case, timekeeping data gets out of order. Therefore, be sure to reset RTCHLD to 0 as soon as possible after completing the necessary read or write operation.

RTCBSY = 1 (RTC is busy)

If the value of RTCBSY is 1 when this bit is read after writing 1 to RTCHLD, it means that 1 is being carried over. The period needed for the counters to carry over 1 is 4 ms per second. In this case, reset RTCHLD to 0 as soon as possible and [A] recheck RTCBSY by following the same procedure or [B] wait 4 ms before checking RTCBSY.
If RTCBSY is found to be 1, be sure to immediately reset RTCHLD to 0. If RTCHLD is left at 1, the time of day may become incorrect.

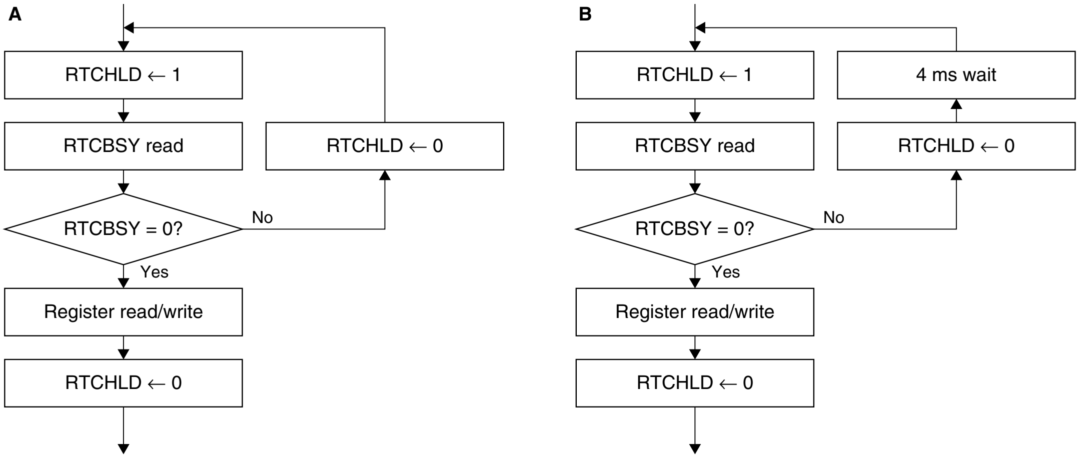


Figure II.5.3.5.1 Procedure for Checking whether the RTC is Busy

There is also a method of reading out data without using RTCHLD and RTCBSY. (See the next section.)

II.5.3.6 Reading from and Writing to Counters in Operation

As described in the previous section, the counters must be accessed for read/write when 1 is not being carried over. Follow the procedure shown in the flowchart in Figure II.5.3.5.1 to read from or write to the counters.

The counters can be read without using RTCHLD and RTCBSY, as shown in Figure II.5.3.6.1.

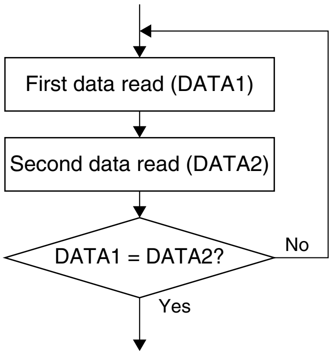


Figure II.5.3.6.1 Procedure for Reading Counters not in the Hold State

II.5.3.7 30-second Correction

The description “30-second correction” means resetting the seconds to 0 and adding 1 to the minutes when seconds of the time clock are in the range of 30 to 59 seconds. When in the range of 0 to 29 seconds, the RTC resets the seconds to 0 but it does not change the minutes. This function may be used to round up seconds to minutes when resetting seconds in an application.
This function can be executed by writing 1 to RTCADJ (D2/RTC_CNTL0 register).

∗ **RTCADJ**: 30-second Adjustment Bit in the RTC Control 0 (RTC_CNTL0) Register (D2/0x4602)

Writing 1 to RTCADJ causes the RTC to operate as follows:

- When the 10-second counter is 3 or more, the RTC generates a carry over of 1 to start counting by the 1-minute counter.
- When the 10-second counter is 2 or less, the RTC does not generate a carry over of 1.

After RTCADJ is set to 1, it remains set for the 4-ms period required for this processing, then automatically returns to 0.
Accessing the counters while RTCADJ = 1 is prohibited. Writing 0 to RTCADJ is also prohibited, because it would cause the RTC to operate erratically.

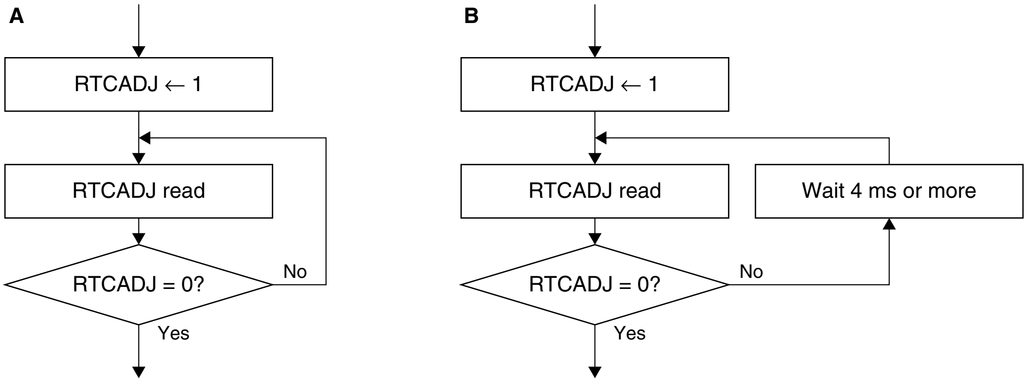


Figure II.5.3.7.1 Procedure for Executing 30-second Correction

II.5.4 RTC Interrupts

The RTC has a function to generate interrupts at given intervals.

Since the RTC is active even in standby mode, interrupts may be used to turn off SLEEP mode.

This section describes the internal interrupt control function of the RTC. To generate interrupts to the CPU, the interrupt controller (ITC) must also be set up. For details on how to control the ITC, see Section III.1, "Interrupt Controller (ITC)." For details on how to turn off SLEEP mode using an interrupt, see Section II.2, "Clock Management Unit (CMU)."

Setting the interrupt cycle

The interrupt cycle (in which the RTC outputs interrupt requests at specific intervals) can be selected from four choices listed in Table II.5.4.1 by using RTCT[1:0] (D[3:2]/RTC_INTMODE register).

∗ **RTCT[1:0]**: RTC Interrupt Cycle Setup Bits in the RTC Interrupt Mode (RTC_INTMODE) Register (D[3:2]/0x4601)

Table II.5.4.1 Interrupt Cycle Settings

RTCT[1:0]	Interrupt cycle
0x3	1 hour
0x2	1 minute
0x1	1 second
0x0	1/64 second

RTCT[1:0] should be set while RTC interrupts are disabled. (See the procedure for enabling and disabling interrupts described below.)

Setting interrupt conditions

The interrupt requests sent to the ITC can be selected as edge-triggered or level-sensed interrupts by setting a register bit. RTCIMD (D1/RTC_INTMODE register) is the bit provided for this purpose.

∗ **RTCIMD**: RTC Interrupt Mode Select Bit in the RTC Interrupt Mode (RTC_INTMODE) Register (D1/0x4601)

Setting RTCIMD to 1 selects a level-sensed interrupt; setting it to 0 selects an edge-triggered interrupt.

When an edge-triggered interrupt has been selected, the RTC outputs an interrupt pulse to the ITC using the bus clock supplied from the CMU. If a cause of interrupt occurs when the bus clock has not been supplied such as in SLEEP mode, the RTC switches the interrupt mode to level-sensed and sets the interrupt signal to the active level from occurrence of the interrupt cause until the bus clock supply is started.

Enabling and disabling interrupts

The RTC interrupt requests output to the ITC are enabled by setting RTCIEN (D0/RTC_INTMODE register) to 1 and disabled by setting it to 0.

∗ **RTCIEN**: RTC Interrupt Enable Bit in the RTC Interrupt Mode (RTC_INTMODE) Register (D0/0x4601)

Interrupt status

When the RTC is up and running, RTCIRQ (D0/RTC_INTSTAT register) is set at the cyclic interrupt intervals set up by RTCT[1:0]. When RTC interrupts are enabled by RTCIEN, interrupt requests are sent to the ITC.

∗ **RTCIRQ**: Interrupt Status Bit in the RTC Interrupt Status (RTC_INTSTAT) Register (D0/0x4600)

Writing 1 to this status bit clears the bit. Because this bit is not cleared in hardware, be sure to clear it in software after an interrupt is generated. If this bit remains set while interrupts are re-enabled or control is returned from the interrupt handler routine by the reti instruction, the same interrupt may be generated again.

Precautions

All RTC interrupt control bits described above are indeterminate when power is turned on. Moreover, these bits are not initialized to specific values by an initial reset.

After power-on, be sure to set RTCIEN to 0 (interrupt disabled) to prevent the occurrence of unwanted RTC interrupts. Also be sure to write 1 to RTCIRQ to reset it.

II.5.5 OSC1 Oscillator Circuit

The S1C17002 contains an oscillator circuit (OSC1) used to generate a 32.768 kHz (typ.) clock as the clock source for timekeeping operation of the RTC and counting operation of the T8OSC1 CH.0/CH.1.
The OSC1 clock can also be used as a power-saving operating clock for the core system or peripheral circuits. For details, see Section II.2, “Clock Management Unit (CMU).”

II.5.5.1 Input/Output Pins of the OSC1 Oscillator Circuit

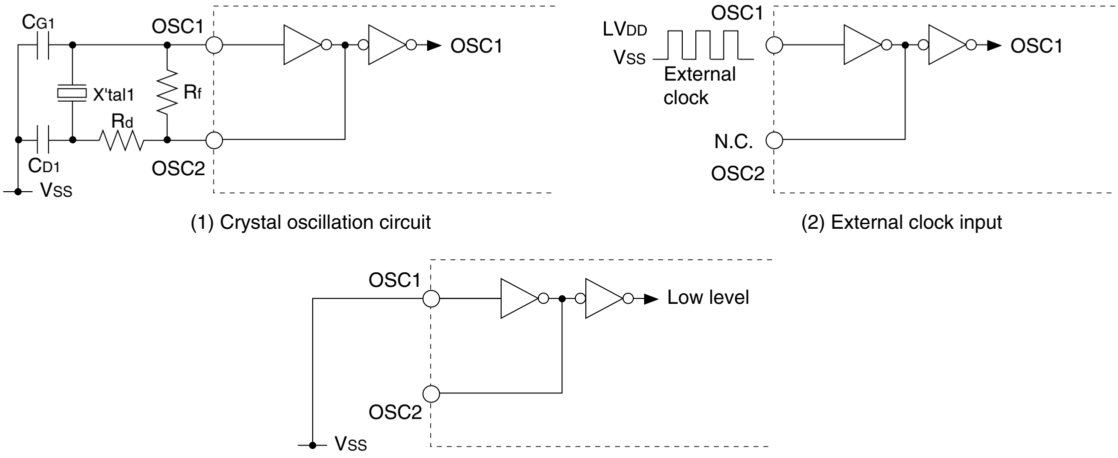
Table II.5.5.1.1 lists the input/output pins of the OSC1 oscillator circuit.

Table II.5.5.1.1 Input/Output Pins of the OSC1 Oscillator Circuit

Pin name	I/O	Function
OSC1	I	OSC1 oscillator input pin: Crystal oscillator or external clock input
OSC2	O	OSC1 oscillator output pin: Crystal oscillator (left open when using external clock input)

II.5.5.2 Structure of the OSC1 Oscillator Circuit

The OSC1 oscillator circuit accommodates a crystal oscillator and external clock input. As for the RTC, LVDD is used to supply power to this circuit.
Figure II.5.5.2.1 shows the structure of the OSC1 oscillator circuit.



(1) Crystal oscillation circuit

(2) External clock input

(3) When not used

Figure II.5.5.2.1 OSC1 Oscillator Circuit

For use as a crystal oscillator circuit, connect a crystal resonator X’tal1 (32.768 kHz, typ.), feedback resistor (Rf), two capacitors (CG1, CD1), and, if necessary, a drain resistor (Rd) to the OSC1 and OSC2 pins and VSS, as shown in Figure II.5.5.2.1 (1).
To use an external clock, leave the OSC2 pin open and input a LVDD level clock (whose duty cycle is 50%) to the OSC1 pin.
The oscillator frequency/input clock frequency is 32.768 kHz (typ.). Make sure the crystal resonator or external clock used in the RTC has this clock frequency. With any other clock frequencies, the RTC cannot be used for timekeeping purposes.
For details of oscillation characteristics and the input characteristics of external clock, see “Electrical Characteristics.”
When not using the OSC1 oscillator circuit, connect the OSC1 pin to VSS and leave the OSC2 pin open.

The OSC1 oscillator always operates without controlling using a register.

Note: When the oscillator is made to start oscillating at power-on, a finite time (of up to 3 seconds) is required until oscillation stabilizes. To prevent system malfunction, do not use the oscillator-derived clock until this oscillation stabilization time elapses.

II.5.6 Details of Control Registers

Table II.5.6.1 RTC Register List

Address	Register name		Function
0x4600	RTC_INTSTAT	RTC Interrupt Status Register	Indicates RTC interrupt status.
0x4601	RTC_INTMODE	RTC Interrupt Mode Register	Sets up RTC interrupt modes.
0x4602	RTC_CNTL0	RTC Control 0 Register	Controls the RTC.
0x4603	RTC_CNTL1	RTC Control 1 Register	
0x4614	RTC_SEC	RTC Second Register	Second counter data
0x4615	RTC_MIN	RTC Minute Register	Minute counter data
0x4616	RTC_HOUR	RTC Hour Register	Hour counter data
0x4617	RTC_DAY	RTC Day Register	Day counter data
0x4628	RTC_MONTH	RTC Month Register	Month counter data
0x4629	RTC_YEAR	RTC Year Register	Year counter data
0x462a	RTC_WEEK	RTC Days of Week Register	Days of week counter data

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

Notes:
- When setting the registers, be sure to write a 0, and not a 1, for all “reserved bits.”
- The contents of all RTC control registers are indeterminate when power is turned on, and are not initialized to specific values by initial reset. These registers should be initialized in software.
- If 1 is being carried over when the counters are accessed for read, the correct counter value may not be read out. Moreover, attempting to write to a counter or other control register may corrupt the counter value. Therefore, do not write to counters while 1 is being carried over. For the correct method of operation, see Section II.5.3.5, “Counter Hold and Busy Flag,” and Section II.5.3.6, “Reading from and Writing to Counters in Operation.”

0x4600: RTC Interrupt Status Register (RTC_INTSTAT)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
RTC Interrupt Status Register (RTC_INTSTAT)	**0x4600** (8 bits)	D7–1	–	reserved	–				–	–	0 when being read.
		D0	**RTCIRQ**	Interrupt status	1	Occurred	0	Not occurred	X	R/W	Reset by writing 1.

D[7:1] Reserved

D0 RTCIRQ: Interrupt Status Bit

This bit indicates whether a cause of RTC interrupt occurred as follows:

1 (R):	Cause of interrupt occurred
0 (R):	No cause of interrupt occurred
1 (W):	Resets this bit to 0
0 (W):	Has no effect

This bit is set at cyclic interrupt intervals set up by RTCT[1:0] (D[3:2]/RTC_INTMODE register). When RTC interrupts have been enabled by RTCIEN (D0/RTC_INTMODE register) at this time, an interrupt request is sent to the ITC. This bit is always set, even when RTC interrupts are disabled.

Note: Writing 1 to this status bit clears it. Because this bit is not cleared in hardware, be sure to clear it in software after an interrupt is generated. If this bit remains set while interrupts are re-enabled or control is returned from the interrupt handler routine by the reti instruction, the same interrupt may be generated again.
Moreover, the value of this bit is indeterminate after power-on, and is not initialized to 0 by initial reset. To prevent the occurrence of unwanted RTC interrupts, be sure to reset this bit in software after power-on and initial reset.

0x4601: RTC Interrupt Mode Register (RTC_INTMODE)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
RTC Interrupt Mode Register (RTC_INTMODE)	**0x4601** (8 bits)	D7–4	**–**	reserved	–				–	–	0 when being read.
		D3–2	**RTCT[1:0]**	RTC interrupt cycle setup	RTCT[1:0]		Cycle		X	R/W	
					0x3		1 hour				
					0x2		1 minute				
					0x1		1 second				
					0x0		1/64 second				
		D1	**RTCIMD**	RTC interrupt mode select	1	Level sense	0	Edge trigger	X	R/W	
		D0	**RTCIEN**	RTC interrupt enable	1	Enable	0	Disable	X	R/W	

D[7:4] Reserved

D[3:2] RTCT[1:0]: RTC Interrupt Cycle Setup Bits

These bits select the RTC interrupt cycle.

Table II.5.6.2 Interrupt Cycle Settings

RTCT[1:0]	Interrupt cycle
0x3	1 hour
0x2	1 minute
0x1	1 second
0x0	1/64 second

(Default: indeterminate)

RTCIRQ (D0/RTC_INTSTAT register) is set by a count-up pulse of the interrupt cycle counter selected. When RTC interrupts are enabled by RTCIEN (D0), an interrupt request is sent to the ITC.
RTCT[1:0] should be set while RTC interrupts are disabled. (These bits may also be set simultaneously when RTC interrupts are enabled.)

D1 RTCIMD: RTC Interrupt Mode Select Bit

This bit specifies whether RTC interrupts are to be generated by an edge or level of the interrupt request signal.
1 (R/W): Level sensed
0 (R/W): Edge triggered

When an edge-triggered interrupt is selected and used to turn off SLEEP mode via the CMU, note that no interrupts will be generated because the ITC is inactive. When an RTC interrupt handler routine must be executed after exiting SLEEP mode, select a level-sensed interrupt.

D0 RTCIEN: RTC Interrupt Enable Bit

This bit enables or disables RTC interrupt request output to the ITC.
1 (R/W): Enable interrupts
0 (R/W): Disable interrupts

To generate an RTC interrupt or use an RTC interrupt request signal to turn off SLEEP mode, set this bit to 1. When this bit is 0, no interrupts are generated even when RTCIRQ (D0/RTC_INTSTAT register) is set and SLEEP mode cannot be turned off.

Note: The value of RTCIEN is indeterminate after power-on, and not initialized to 0 by initial reset. To prevent the occurrence of unwanted RTC interrupts, be sure to clear this bit in software after power-on and initial reset.

0x4602: RTC Control 0 Register (RTC_CNTL0)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
RTC Control 0 Register (RTC_CNTL0)	**0x4602** (8 bits)	D7–5	–	reserved	–				–	–	0 when being read.
		D4	**RTC24H**	24H/12H mode select	1	24H	0	12H	X	R/W	
		D3	–	reserved	–				–	–	0 when being read.
		D2	**RTCADJ**	30-second adjustment	1	Adjust	0	–	X	R/W	
		D1	**RTCSTP**	Counter run/stop control	1	Stop	0	Run	X	R/W	
		D0	**RTCRST**	Software reset	1	Reset	0	–	X	R/W	

D[7:5] Reserved

D4 RTC24H: 24H/12H Mode Select Bit

This bit selects whether to use the hour counter in 24-hour or 12-hour mode.

1 (R/W): 24-hour mode

0 (R/W): 12-hour mode

The count range of hour counters changes with this selection.

Basically, this setting should be changed while the counters are idle. Since this register is assigned a control bit (D1) to start the counters, 12-hour or 24-hour mode may be selected when starting the counters.

Note: Rewriting RTC24H may corrupt the count data for hours, days, months, years, or days of the week. Therefore, after changing the RTC24H setting, be sure to set data back in these counters again.

D3 Reserved

D2 RTCADJ: 30-second Adjustment Bit

This bit executes 30-second correction.

1 (W): Execute 30-second correction

0 (W): Has no effect

1 (R): 30-second correction being executed

0 (R): 30-second correction completed (not being executed)

The description “30-second correction” means adding 1 to the minutes when seconds of the time clock are in the 30-to-59 second range, and doing nothing in the 0-to-29 second range. This function may be used to round up seconds to minutes when resetting seconds in an application.

Writing 1 to this bit causes the RTC to operate as follows:

- When the 10-second counter is 3 or more, the RTC generates a carry over of 1 to start counting by the 1-minute counter.
- When the 10-second counter is 2 or less, the RTC does not generate a carry over of 1.

After being set to 1, this bit remains set for the 4-ms period needed for the processing above, then is automatically reset to 0.

Note: Accessing the counters while RTCADJ = 1 is prohibited. Writing 0 to this bit during such time is also prohibited, because it would cause the RTC to operate erratically.

D1 RTCSTP: Counter Run/Stop Control Bit

This bit starts or stops the counters. It also indicates counter operating status.
1 (R/W): Stops counters/Counters idle
0 (R/W): Starts counters/Counters operating

Setting this bit to 0 starts the counters; setting it to 1 stops the counters.
The value read from this bit is 0 when the counters are operating, and 1 when the counters are idle.
Writing 1 to this bit stops the counters at the 32-kHz input clock divide-by stage of 8,192 Hz or stages that follow. The counters do not stop at up to the input clock divide-by-2 stage (16,384 Hz).
If the counters stop while 1 is being carried over, the count value may be corrupted. Therefore, see Section II.5.3.5 to ensure that 1 is not being carried over when the counters are stopped. This is unnecessary when, for example, the contents of all counters are newly set again.

D0 RTCRST: Software Reset Bit

This bit resets the counters currently at divide-by stages.
1 (R/W): Reset counters
0 (R/W): Negate reset

Setting this bit to 1 resets the 32 kHz to 2 Hz counters (cleared to 0). Writing 0 to this bit negates the reset.

0x4603: RTC Control 1 Register (RTC_CNTL1)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
RTC Control 1 Register (RTC_CNTL1)	**0x4603** (8 bits)	D7–2	–	reserved	–				–	–	0 when being read.
		D1	**RTCBSY**	Counter busy flag	1	Busy	0	R/W possible	X	R	
		D0	**RTCHLD**	Counter hold control	1	Hold	0	Running	X	R/W	

D[7:2] Reserved

D1 RTCBSY: Counter Busy Flag Bit

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

1 (R): Busy (while 1 is being carried over)
0 (R): Accessible for read/write
1/0 (W): Has no effect

If 1 is being carried over while the counters are being read, correct counter values may not be read. Moreover, attempting a write or stop operation may corrupt the counter values. Therefore, this bit should be checked to confirm that the counters are not in a carry (busy) state before reading or writing data from or to the count registers.
However, because this bit is fixed to 1 while the counters are operating, RTCHLD (D0) should be set to 1 so that the count value reflects the current state.

When a value of 0 is read from this bit after writing 1 to RTCHLD (D0), it means that 1 is not now being carried over. In this case, the counter hold function is also actuated, with a carry over of 1 to the 1-second counter disabled in hardware. Counters for less than seconds continue operating. In this state, data can be read from or written to the count registers. After reading or writing data, reset RTCHLD (D0) to 0.
If 1 is being carried over when data is being read from or written to counters in the hold state, 1 second is automatically added at that time, with RTCHLD (D0) reset to 0 for correcting the count value. This correction is only effective for 1 second, thus ignoring the time needed to carry over 1 on subsequent occasions. In this case, the timekeeping data gets out of order. Therefore, be sure to reset RTCHLD (D0) to 0 as soon as possible after completing the required read or write operation.

When a value of 1 is read from this bit after writing 1 to RTCHLD (D0), it means that 1 is now being carried over. A period of 4 ms per second is required for a carry over of 1 to the counters. In this case, reset RTCHLD (D0) to 0 as soon as possible and check this bit again by following the same procedure, or wait 4 ms before checking this bit. If this bit is set to 1, always reset RTCHLD (D0) to 0 immediately. Leaving RTCHLD (D0) set to 1 may result in an incorrect time of day.

D0 RTCHLD: Counter Hold Control Bit

This bit allows the busy state of counters to be checked and the counters held intact.

1 (R/W): Checks for busy state/Holds counters
0 (R/W): Normal operation

For the operation of this bit, see the description of RTCBSY (D1) above.

0x4614: RTC Second Register (RTC_SEC)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RTC Second Register (RTC_SEC)	**0x4614** (8 bits)	D7	–	reserved	–	–	–	0 when being read.
		D6–4	**RTCSH[2:0]**	RTC 10-second counter	0 to 5	X	R/W	
		D3–0	**RTCSL[3:0]**	RTC 1-second counter	0 to 9	X	R/W	

Note: Data should not be read from or written to the counters while 1 is being carried over. (See Section II.5.3.5, "Counter Hold and Busy Flag," and Section II.5.3.6, "Reading from and Writing to Counters in Operation.")

D7 Reserved

D[6:4] RTCSH[2:0]: RTC 10-second Counter Bits

These bits comprise a 3-bit BCD counter used to count tens of seconds.

The counter counts from 0 to 5 with a carry over of 1 from the 1-second counter. This counter is reset to 0 after 5 and outputs a carry over of 1 to the 1-minute counter.

D[3:0] RTCSL[3:0]: RTC 1-second Counter Bits

These bits comprise a 4-bit BCD counter used to count units of seconds.

The counter counts from 0 to 9 synchronously with a 1-second signal derived from the 32.768-kHz OSC1 clock. This counter is reset to 0 after 9 and outputs a carry over of 1 to the 10-second counter.

0x4615: RTC Minute Register (RTC_MIN)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RTC Minute Register (RTC_MIN)	**0x4615** (8 bits)	D7	–	reserved	–	–	–	0 when being read.
		D6–4	**RTCMIH[2:0]**	RTC 10-minute counter	0 to 5	X	R/W	
		D3–0	**RTCMIL[3:0]**	RTC 1-minute counter	0 to 9	X	R/W	

Note: Data should not be read from or written to the counters while 1 is being carried over. (See Section II.5.3.5, “Counter Hold and Busy Flag,” and Section II.5.3.6, “Reading from and Writing to Counters in Operation.”)

D7 Reserved

D[6:4] RTCMIH[2:0]: RTC 10-minute Counter Bits

These bits comprise a 3-bit BCD counter used to count tens of minutes.

The counter counts from 0 to 5 with a carry over of 1 from the 1-minute counter. This counter is reset to 0 after 5 and outputs a carry over of 1 to the 1-hour counter.

D[3:0] RTCMIL[3:0]: RTC 1-minute Counter Bits

These bits comprise a 4-bit BCD counter used to count units of minutes.

The counter counts from 0 to 9 with a carry over of 1 from the 10-second counter. This counter is reset to 0 after 9 and outputs a carry over of 1 to the 10-minute counter.

0x4616: RTC Hour Register (RTC_HOUR)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
RTC Hour Register (RTC_HOUR)	**0x4616** (8 bits)	D7	**–**	reserved	–				–	–	0 when being read.
		D6	**RTCAP**	AM/PM indicator	1	PM	0	AM	X	R/W	
		D5–4	**RTCHH[1:0]**	RTC 10-hour counter	0 to 2 or 0 to 1				X	R/W	
		D3–0	**RTCHL[3:0]**	RTC 1-hour counter	0–9				X	R/W	

Notes:
- Data should not be read from or written to the counters while 1 is being carried over. (See Section II.5.3.5, "Counter Hold and Busy Flag," and Section II.5.3.6, "Reading from and Writing to Counters in Operation.")
- Rewriting RTC24H (D4/RTC_CNTL0 register) may corrupt the count data in this register. Therefore, after changing the RTC24H setting, be sure to set up this register again.

D7 Reserved

D6 RTCAP: AM/PM Indicator Bit

When 12-hour mode is selected, this bit indicates A.M. or P.M.

1 (R/W): P.M.
0 (R/W): A.M.

This bit is only effective when RTC24H (D4/RTC_CNTL0 register) is set to 0 (12-hour mode).
When 24-hour mode is selected, this bit is fixed to 0. In this case, do not write 1 to RTCAP.

Note: The RTCAP bit keeps the current set value even if RTC24H (D4/RTC_CNTL0 register) is changed from 12-hour mode to 24-hour mode, and will be fixed at 0 after the hour counter is updated (or reset in software).

D[5:4] RTCHH[1:0]: RTC 10-hour Counter Bits

These bits comprise a 2-bit BCD counter used to count tens of hours.
With a carry over of 1 from the 1-hour counter, the counter counts from 0 to 1 when 12-hour mode is selected, or from 0 to 2 when 24-hour mode is selected. The counter is reset at 12 o'clock or 24 o'clock, and outputs a carry over of 1 to the 1-day counter.

D[3:0] RTCHL[3:0]: RTC 1-hour Counter Bits

These bits comprise a 4-bit BCD counter used to count units of hours.
The counter counts from 0 to 9 with a carry over of 1 from the 10-minute counter. This counter is reset to 0 after 9 and outputs a carry over of 1 to the 10-hour counter. Depending on whether 12-hour mode or 24-hour mode is selected, the counter is reset at 12 o'clock or 24 o'clock.

0x4617: RTC Day Register (RTC_DAY)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RTC Day Register (RTC_DAY)	**0x4617** (8 bits)	D7–6	**–**	reserved	–	–	–	0 when being read.
		D5–4	**RTCDH[1:0]**	RTC 10-day counter	0 to 3	X	R/W	
		D3–0	**RTCDL[3:0]**	RTC 1-day counter	0 to 9	X	R/W	

Notes:
- Data should not be read from or written to the counters while 1 is being carried over. (See Section II.5.3.5, “Counter Hold and Busy Flag,” and Section II.5.3.6, “Reading from and Writing to Counters in Operation.”)
- Rewriting RTC24H (D4/RTC_CNTL0 register) may corrupt the count data in this register. Therefore, after changing the RTC24H setting, be sure to set up this register again.

D[7:6] Reserved

D[5:4] RTCDH[1:0]: RTC 10-day Counter Bits

These bits comprise a 2-bit BCD counter used to count tens of days. The counter counts from 0 to 2 or 3 with a carry over of 1 from the 1-day counter. The number of days in each month and leap years are taken into account, so that when months change the counter is reset to 0 along with the 1-day counter, and a carry over of 1 is output to the 1-month counter.

D[3:0] RTCDL[3:0]: RTC 1-day Counter Bits

These bits comprise a 4-bit BCD counter used to count units of days.

The counter counts from 0 to 9 with a carry over of 1 from the hour counter. This counter is reset to 0 after 9 and outputs a carry over of 1 to the 10-day counter. The number of days in each month and leap years are taken into account, so that the counter is reset to 1 when months change.

0x4628: RTC Month Register (RTC_MONTH)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RTC Month Register (RTC_MONTH)	**0x4628** (8 bits)	D7–5	**–**	reserved	–	–	–	0 when being read.
		D4	**RTCMOH**	RTC 10-month counter	0 to 1	X	R/W	
		D3–0	**RTCMOL[3:0]**	RTC 1-month counter	0 to 9	X	R/W	

Notes:
- Data should not be read from or written to the counters while 1 is being carried over. (See Section II.5.3.5, "Counter Hold and Busy Flag," and Section II.5.3.6, "Reading from and Writing to Counters in Operation.")
- Rewriting RTC24H (D4/RTC_CNTL0 register) may corrupt the count data in this register. Therefore, after changing the RTC24H setting, be sure to set up this register again.

D[7:5] Reserved

D4 RTCMOH: RTC 10-month Counter Bit

This is a tens of months count bit.

This bit is set to 1 with a carry over of 1 from the 1-month counter. When years change, this bit is reset to 0 along with the 1-month counter, and a carry over of 1 is output to the 1-year counter.

D[3:0] RTCMOL[3:0]: RTC 1-month Counter Bits

These bits comprise a 4-bit BCD counter used to count units of months.

The counter counts from 0 to 9 with a carry over of 1 from the day counter. This counter is reset to 0 after 9 and outputs a carry over of 1 to the 10-month counter. The counter is reset to 1 when years change.

0x4629: RTC Year Register (RTC_YEAR)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RTC Year Register (RTC_YEAR)	**0x4629** (8 bits)	D7–4	**RTCYH[3:0]**	RTC 10-year counter	0 to 9	X	R/W	
		D3–0	**RTCYL[3:0]**	RTC 1-year counter	0 to 9	X	R/W	

Notes:
- Data should not be read from or written to the counters while 1 is being carried over. (See Section II.5.3.5, "Counter Hold and Busy Flag," and Section II.5.3.6, "Reading from and Writing to Counters in Operation.")
- Rewriting RTC24H (D4/RTC_CNTL0 register) may corrupt the count data in this register. Therefore, after changing the RTC24H setting, be sure to set up this register again.

D[7:4] RTCYH[3:0]: RTC 10-year Counter Bits

These bits comprise a 4-bit BCD counter used to count tens of years. The counter counts from 0 to 9 with a carry over of 1 from the 1-year counter.

D[3:0] RTCYL[3:0]: RTC 1-year Counter Bits

These bits comprise a 4-bit BCD counter used to count units of years.

The counter counts from 0 to 9 with a carry over of 1 from the month counter. This counter is reset to 0 after 9 and outputs a carry over of 1 to the 10-year counter.

0x462a: RTC Days of Week Register (RTC_WEEK)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
RTC Days of Week Register (RTC_WEEK)	**0x462a** (8 bits)	D7–3	–	reserved	–		–	–	0 when being read.
		D2–0	**RTCWK[2:0]**	RTC days of week counter	RTCWK[2:0]	Days of week	X	R/W	
					0x7	–			
					0x6	Saturday			
					0x5	Friday			
					0x4	Thursday			
					0x3	Wednesday			
					0x2	Tuesday			
					0x1	Monday			
					0x0	Sunday			

Notes:
- Data should not be read from or written to the counters while 1 is being carried over. (See Section II.5.3.5, “Counter Hold and Busy Flag,” and Section II.5.3.6, “Reading from and Writing to Counters in Operation.”)
- Rewriting RTC24H (D4/RTC_CNTL0 register) may corrupt the count data in this register. Therefore, after changing the RTC24H setting, be sure to set up this register again.

D[7:3] Reserved

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

This is a septenary counter (that counts from 0 to 6) representing days of the week. This counter counts at the same timing as the 1-day counter.

The correspondence between the counter values and days of the week can be set in a program as desired. Table II.5.6.3 lists the basic correspondence.

Table II.5.6.3 Correspondence between Counter Values and Days of the Week

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

(Default: indeterminate)

II.5.7 Precautions

- The contents of all RTC control registers are indeterminate when power is turned on and are not initialized to specific values by initial reset. Be sure to initialize these registers in software.
- While 1 is being carried over to the next-digit counter, the correct counter value may not be read out. Moreover, attempting to write to the counters or other control registers may corrupt the counter value. Therefore, do not write to the counters while 1 is being carried over. For the correct method of operation, see Section II.5.3.5, "Counter Hold and Busy Flag," and Section II.5.3.6, "Reading from and Writing to Counters in Operation."
- Note that rewriting RTC24H (D4/RTC_CNTL0 register) to switch between 12-hour mode and 24-hour mode may corrupt the count data for hours, days, months, years, or days of the week. Therefore, after changing the RTC24H setting, be sure to set data in these counters back again.
- Avoid the settings below that may cause timekeeping errors.

  - Settings exceeding the effective range
    Do not set count data exceeding 60 seconds, 60 minutes, 12 or 24 hours, 31 days, 12 months, or 99 years.

  - Settings nonexistent in the calendar
    Do not set nonexistent dates such as April 31 or February 29, 2006. Even if such settings are made, the counters operate normally, so that when 1 is carried over from the hour counter to the 1-day counter, the day counter counts up to the first day of the next month. (For April 31, the day counter counts up to May 1; for February 29, 2006, the day counter counts up to March 1, 2006.)

- The contents of all RTC interrupt control bits are indeterminate when power is turned on, and are not initialized to specific values by initial reset.
  After power-on, be sure to set RTCIEN (D0/RTC_INTMODE register) to 0 (interrupt disabled) for preventing the occurrence of unwanted RTC interrupts. Also be sure to write 1 to RTCIRQ (D0/RTC_INTSTAT register) to reset it.
- Immediately after the OSC1 oscillator circuit is activated (as at power-on), a finite time (of about 3 seconds) is required for OSC1 oscillation to stabilize. Do not let the RTC start counting until this time elapses.

THIS PAGE IS BLANK.

S1C17002 Technical Manual

III S1C17002 INTERRUPT/ INTERNAL BUS CONTROLLERS

III.1 Interrupt Controller (ITC)

III.1.1 Configuration of ITC

The S1C17002 provides 29 interrupt systems listed below.

1. Port input interrupts (8 types)
2. Multi-function timer interrupt (1 type)
3. A/D converter interrupts (2 types)
4. 16-bit clock generator timer interrupt (1 type)
5. 8-bit clock generator timer interrupts (3 types)
6. UART interrupt (1 type)
7. SPI (SPI CH.0) interrupt (1 type)
8. I^{2}C master interrupts (1 type)
9. I^{2}C slave interrupts (2 types)
10. RTC interrupt (1 type)
11. 8-bit programmable timer interrupts (4 types)
12. 8-bit OSC1 timer interrupts (2 types)
13. Extended SPI (SPI CH.1) interrupt (1 type)
14. Remote controller interrupt (1 type)

Each interrupt system provides an interrupt flag that indicates the occurrence of an interrupt request from the peripheral module and an interrupt enable bit that enables/disables interrupts. In addition, the ITC allows the application program to set the interrupt level (priority) of each interrupt system that determines the order of handling when two or more interrupts occur at the same time.

( ) in the list above represents the number of interrupt causes supported in each interrupt system. Use the control register in the peripheral module to select the interrupt causes for generating an interrupt request. For more information on interrupt causes and control, see the description for each peripheral module.

Figure III.1.1.1 shows the structure of the interrupt system.

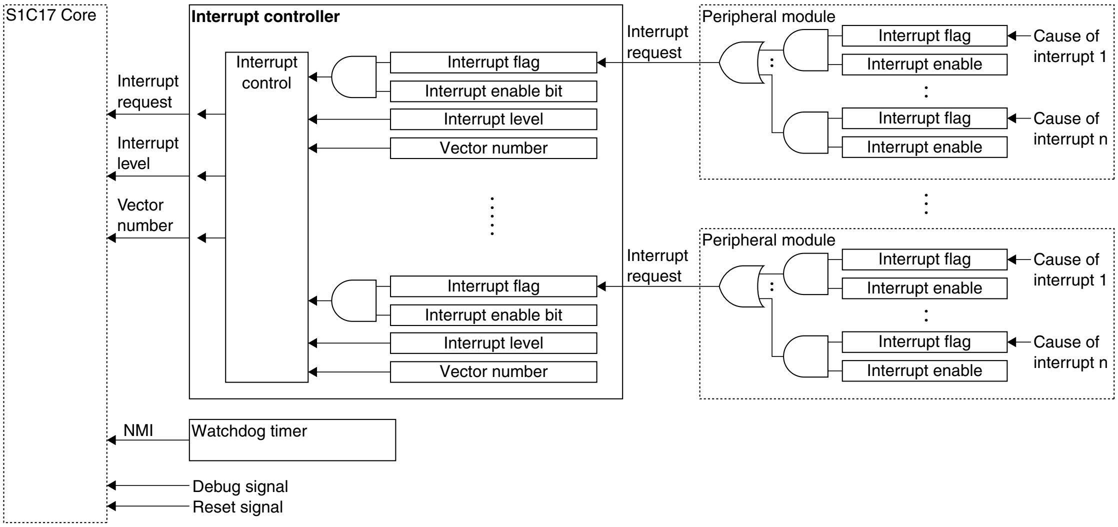


Figure III.1.1.1 Interrupt System

III.1.2 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. The S1C17002 allows the base (starting) address of the vector table to be set using the TTBR_LOW and TTBR_HIGH registers (0x5814, 0x5816). "TTBR" described in Table III.1.2.1 means the value set to these registers. After an initial reset, the TTBR_LOW/HIGH registers are set to 0x20000. Therefore, even when the vector table position is changed, it is necessary that at least the reset vector be written to the above address. Table III.1.2.1 shows the vector table of the S1C17002.

Table III.1.2.1 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	C compiler (reserved)	Used in emulation library for C compiler	5
4 (0x04)	TTBR + 0x10	Port input interrupt 0	P*x*0 input (rising/falling edge or high/low level)	High *1
5 (0x05)	TTBR + 0x14	Port input interrupt 1	P*x*1 input (rising/falling edge or high/low level)	↑
6 (0x06)	TTBR + 0x18	Port input interrupt 2	P*x*2 input (rising/falling edge or high/low level)	
7 (0x07)	TTBR + 0x1c	Port input interrupt 3	P*x*3 input (rising/falling edge or high/low level)	
8 (0x08)	TTBR + 0x20	MFT interrupt	• Compare-match • Period-match • ADC protection input • Port protection input	
9 (0x09)	TTBR + 0x24	reserved	–	
10 (0x0a)	TTBR + 0x28	A/D converter	Out of range results (upper- and lower-limit)	
11 (0x0b)	TTBR + 0x2c		End of conversion	
12 (0x0c)	TTBR + 0x30	CLG_T16U0 timer interrupt	Timer underflow	
		Port input interrupt 4	P*x*4 input (rising/falling edge or high/low level)	
13 (0x0d)	TTBR + 0x34	CLG_T8FU0 timer interrupt	Timer underflow	
		Port input interrupt 5	P*x*5 input (rising/falling edge or high/low level)	
14 (0x0e)	TTBR + 0x38	CLG_T8S timer interrupt	Timer underflow	
		Port input interrupt 6	P*x*6 input (rising/falling edge or high/low level)	
15 (0x0f)	TTBR + 0x3c	CLG_T8I timer interrupt	Timer underflow	
		Port input interrupt 7	P*x*7 input (rising/falling edge or high/low level)	
16 (0x10)	TTBR + 0x40	UART with IrDA CH.0 interrupt	• Transmit buffer empty • Receive buffer full • Receive error	
		Port input interrupt 4	P*x*4 input (rising/falling edge or high/low level)	
17 (0x11)	TTBR + 0x44	Port input interrupt 5	P*x*5 input (rising/falling edge or high/low level)	
18 (0x12)	TTBR + 0x48	SPI CH.0 interrupt	• Transmit buffer empty • Receive buffer full	
		Port input interrupt 6	P*x*6 input (rising/falling edge or high/low level)	
19 (0x13)	TTBR + 0x4c	I^2C master interrupt	• Transmit buffer empty • Receive buffer full	
		Port input interrupt 7	P*x*7 input (rising/falling edge or high/low level)	
20 (0x14)	TTBR + 0x50	RTC interrupt	1/64 second, 1 second, 1 minute, or 1 hour count up	
21 (0x15)	TTBR + 0x54	8-bit timer CH.0 interrupt	Timer 0 underflow	
		8-bit OSC1 timer CH.0 interrupt	Compare match	
22 (0x16)	TTBR + 0x58	8-bit timer CH.1 interrupt	Timer 1 underflow	
		8-bit OSC1 timer CH.1 interrupt	Compare match	
23 (0x17)	TTBR + 0x5c	8-bit timer CH.2 interrupt	Timer 2 underflow	
24 (0x18)	TTBR + 0x60	8-bit timer CH.3 interrupt	Timer 3 underflow	
25 (0x19)	TTBR + 0x64	reserved	–	
26 (0x1a)	TTBR + 0x68	SPI CH.1 interrupt	• Transmit buffer empty • Receive buffer full	
27 (0x1b)	TTBR + 0x6c	reserved	–	
28 (0x1c)	TTBR + 0x70	I^2C slave interrupt	• Transmit buffer empty • Receive buffer full	
29 (0x1d)	TTBR + 0x74		• Bus status	
30 (0x1e)	TTBR + 0x78	REMC interrupt	• Envelope counter underflow • REMC_IN rising edge detection • REMC_IN falling edge detection	↓
31 (0x1f)	TTBR + 0x7c	reserved	–	Low *1

*1 When the same interrupt level is set

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

Interrupts that share an interrupt vector address

The interrupt vector numbers 12–16 and 18–19 are shared with two causes of interrupts. The interrupt that will occur depends on the setting of the interrupt enable bit (see Section III.1.3.3). If both the interrupts assigned to one interrupt vector number have been enabled, the interrupt listed in the upper line in the vector table will occur, and the interrupt listed in the lower line in the vector table will not occur.

Set the interrupt enable bits to configure these interrupt systems according to the interrupt to be used as below.

Table III.1.2.2 Interrupt Vectors 12 and 16 (UART CH.0, CLG_T16U0, and Port 4 Interrupts)

Interrupt enable bit			Interrupt vector 12	Interrupt vector 16
IIEN4 (UART CH.0)	IIEN0 (CLG_T16U0)	EIEN4 (Port 4)		
1	1	1	CLG_T16U0 interrupt	UART CH.0 interrupt
1	1	0	CLG_T16U0 interrupt	UART CH.0 interrupt
1	0	1	Port interrupt 4	UART CH.0 interrupt
1	0	0	–	UART CH.0 interrupt
0	1	1	CLG_T16U0 interrupt	Port interrupt 4
0	1	0	CLG_T16U0 interrupt	–
0	0	1	Port interrupt 4	Port interrupt 4
0	0	0	–	–

(Interrupt enable bit: 1 = enable, 0 = disable)

Table III.1.2.3 Interrupt Vector 13 (CLG_T8FU0 and Port 5 Interrupts)

Interrupt enable bit		Interrupt vector 13
IIEN1 (CLG_T8FU0)	EIEN5 (Port 5)	
1	1	CLG_T8FU0 interrupt
1	0	CLG_T8FU0 interrupt
0	1	Port interrupt 5
0	0	–

Table III.1.2.4 Interrupt Vectors 14 and 18 (SPI CH.0, CLG_T8S, and Port 6 Interrupts)

Interrupt enable bit			Interrupt vector 14	Interrupt vector 18
IIEN6 (SPI CH.0)	IIEN2 (CLG_T8S)	EIEN6 (Port 6)		
1	1	1	CLG_T8S interrupt	SPI CH.0 interrupt
1	1	0	CLG_T8S interrupt	SPI CH.0 interrupt
1	0	1	Port interrupt 6	SPI CH.0 interrupt
1	0	0	–	SPI CH.0 interrupt
0	1	1	CLG_T8S interrupt	Port interrupt 6
0	1	0	CLG_T8S interrupt	–
0	0	1	Port interrupt 6	Port interrupt 6
0	0	0	–	–

Table III.1.2.5 Interrupt Vectors 15 and 19 (I^2C master, CLG_T8I, and Port 7 Interrupts)

Interrupt enable bit			Interrupt vector 15	Interrupt vector 19
IIEN7 (I^2C master)	IIEN3 (CLG_T8I)	EIEN7 (Port 7)		
1	1	1	CLG_T8I interrupt	I^2C master interrupt
1	1	0	CLG_T8I interrupt	I^2C master interrupt
1	0	1	Port interrupt 7	I^2C master interrupt
1	0	0	–	I^2C master interrupt
0	1	1	CLG_T8I interrupt	Port interrupt 7
0	1	0	CLG_T8I interrupt	–
0	0	1	Port interrupt 7	Port interrupt 7
0	0	0	–	–

Also the interrupt vector numbers 21 and 22 are shared with two causes of interrupts.
Interrupt vector 21: 8-bit programmable timer CH.0 and 8-bit OSC1 timer CH.0
Interrupt vector 22: 8-bit programmable timer CH.1 and 8-bit OSC1 timer CH.1

Two interrupts in each vector number use the same interrupt flag and interrupt enable bit. Therefore, use only one of the timers or verify the counter values to determine the interrupt source if both the timer are used.

III.1.3 Control of Maskable Interrupts

III.1.3.1 Enabling ITC

Before the ITC can be used, set the ITEN bit (D0/ITC_CTL register) to 1.

∗ **ITEN**: ITC Enable Bit in the ITC Control (ITC_CTL) Register (D0/0x4304)

III.1.3.2 Interrupt Request from Peripheral Module and Interrupt Flag

When an enabled interrupt cause occurs in a peripheral module, the module sends an interrupt request signal to the ITC. The interrupt request signal sets the interrupt flag in the ITC corresponding to the cause of interrupt to 1. The interrupt flag holds 1 until it is reset to 0 to indicate that an interrupt request has sent from the peripheral module. The flag status can be read from the ITC_IFLG (0x4300) and ITC_AIFLG (0x42e0) registers.
Table III.1.3.2.1 lists the relationship between the causes of interrupt and the interrupt flags.

Note: When ITEN (D0/ITC_CTL register) is set to 0, the interrupt flag will not be set even if an interrupt request is generated from the peripheral module.

Table III.1.3.2.1 Causes of Hardware Interrupt and Interrupt Flags

Cause of hardware interrupt	Interrupt flag
I^{2}C master interrupt: transmit buffer empty/receive buffer full	IIFT7 (D15/ITC_IFLG register)
SPI CH.0 interrupt: transmit buffer empty/receive buffer full	IIFT6 (D14/ITC_IFLG register)
UART interrupt: transmit buffer empty/receive buffer full/receive error	IIFT4 (D12/ITC_IFLG register)
CLG_T8I timer interrupt: timer underflow	IIFT3 (D11/ITC_IFLG register)
CLG_T8S timer interrupt: timer underflow	IIFT2 (D10/ITC_IFLG register)
CLG_T8FU0 timer interrupt: timer underflow	IIFT1 (D9/ITC_IFLG register)
CLG_T16U0 timer interrupt: timer underflow	IIFT0 (D8/ITC_IFLG register)
Port input interrupt 7: P*x*7 rising/falling edge or high/low level input	EIFT7 (D7/ITC_IFLG register)
Port input interrupt 6: P*x*6 rising/falling edge or high/low level input	EIFT6 (D6/ITC_IFLG register)
Port input interrupt 5: P*x*5 rising/falling edge or high/low level input	EIFT5 (D5/ITC_IFLG register)
Port input interrupt 4: P*x*4 rising/falling edge or high/low level input	EIFT4 (D4/ITC_IFLG register)
Port input interrupt 3: P*x*3 rising/falling edge or high/low level input	EIFT3 (D3/ITC_IFLG register)
Port input interrupt 2: P*x*2 rising/falling edge or high/low level input	EIFT2 (D2/ITC_IFLG register)
Port input interrupt 1: P*x*1 rising/falling edge or high/low level input	EIFT1 (D1/ITC_IFLG register)
Port input interrupt 0: P*x*0 rising/falling edge or high/low level input	EIFT0 (D0/ITC_IFLG register)
Remote controller interrupt: envelope counter underflow/input rising edge/input falling edge	AIFT14 (D14/ITC_AIFLG register)
I^{2}C slave interrupt: bus status	AIFT13 (D13/ITC_AIFLG register)
I^{2}C slave interrupt: transmit buffer empty/receive buffer full	AIFT12 (D12/ITC_AIFLG register)
SPI CH.1 interrupt: transmit buffer empty/receive buffer full	AIFT10 (D10/ITC_AIFLG register)
8-bit timer CH.3 interrupt: timer underflow	AIFT8 (D8/ITC_AIFLG register)
8-bit timer CH.2 interrupt: timer underflow	AIFT7 (D7/ITC_AIFLG register)
8-bit timer CH.1 interrupt: timer underflow 8-bit OSC1 timer CH.1 interrupt: compare-match	AIFT6 (D6/ITC_AIFLG register)
8-bit timer CH.0 interrupt: timer underflow 8-bit OSC1 timer CH.1 interrupt: compare-match	AIFT5 (D5/ITC_AIFLG register)
RTC interrupt: 1/64 second, 1 second, 1 minute, or 1 hour count up	AIFT4 (D4/ITC_AIFLG register)
ADC interrupt: end of conversion	AIFT3 (D3/ITC_AIFLG register)
ADC interrupt: out of range	AIFT2 (D2/ITC_AIFLG register)
Multi-function timer interrupt: compare-match/period-match/protection input	AIFT0 (D0/ITC_AIFLG register)

The ITC uses the interrupt flags to generate an interrupt to the S1C17 Core.
When an interrupt flag is set to 1, the ITC sends the interrupt request, interrupt level and vector number signals to the S1C17 Core if the interrupt has been enabled (see the next section).
The interrupt flag that has been set to 1 can be reset by writing 1. Reset the interrupt flag to 0 in the interrupt handler. If the interrupt handler does not reset the interrupt flag, the same interrupt will be generated again when the interrupt handling has finished (interrupts are disabled during interrupt handling and enabled by executing the `reti` instruction placed at the end of the interrupt handler).

Note, however, that the interrupt flags (EIFT0–EIFT7) for the level triggered interrupts (see Section III.1.3.5) cannot be reset by writing 1. Those interrupt flags are reset when the interrupt signal is negated by the interrupt source.

For the occurrence conditions of the causes of interrupt and the module specific settings, refer to the section that describes the interrupt source module.

III.1.3.3 Enabling/Disabling Interrupts

To send an interrupt request to the S1C17 Core, the interrupt must be enabled by the interrupt enable bit in the ITC_EN (0x4302) or ITC_AEN (0x42e2) register corresponding to the interrupt flag. To enable an interrupt, set the interrupt enable bit to 1; to disable an interrupt, set the interrupt enable bit to 0 (default). The interrupt enable bit does not affect the interrupt flag status, so the interrupt flag will be set by an interrupt request from the peripheral module regardless of how the interrupt enable bit is set if ITEN (D0/ITC_CTL register) is set to 1.

Table III.1.3.3.1 lists the correspondence between the interrupt enable bit and the interrupt flag.

Table III.1.3.3.1 List of Interrupt Enable Bits

Hardware interrupt	Interrupt flag	Interrupt enable bit
I^{2}C master interrupt	IIFT7 (D15/ITC_IFLG register)	IIEN7 (D15/ITC_EN register)
SPI CH.0 interrupt	IIFT6 (D14/ITC_IFLG register)	IIEN6 (D14/ITC_EN register)
UART interrupt	IIFT4 (D12/ITC_IFLG register)	IIEN4 (D12/ITC_EN register)
CLG_T8I timer interrupt	IIFT3 (D11/ITC_IFLG register)	IIEN3 (D11/ITC_EN register)
CLG_T8S timer interrupt	IIFT2 (D10/ITC_IFLG register)	IIEN2 (D10/ITC_EN register)
CLG_T8FU0 timer interrupt	IIFT1 (D9/ITC_IFLG register)	IIEN1 (D9/ITC_EN register)
CLG_T16U0 timer interrupt	IIFT0 (D8/ITC_IFLG register)	IIEN0 (D8/ITC_EN register)
Port input interrupt 7	EIFT7 (D7/ITC_IFLG register)	EIEN7 (D7/ITC_EN register)
Port input interrupt 6	EIFT6 (D6/ITC_IFLG register)	EIEN6 (D6/ITC_EN register)
Port input interrupt 5	EIFT5 (D5/ITC_IFLG register)	EIEN5 (D5/ITC_EN register)
Port input interrupt 4	EIFT4 (D4/ITC_IFLG register)	EIEN4 (D4/ITC_EN register)
Port input interrupt 3	EIFT3 (D3/ITC_IFLG register)	EIEN3 (D3/ITC_EN register)
Port input interrupt 2	EIFT2 (D2/ITC_IFLG register)	EIEN2 (D2/ITC_EN register)
Port input interrupt 1	EIFT1 (D1/ITC_IFLG register)	EIEN1 (D1/ITC_EN register)
Port input interrupt 0	EIFT0 (D0/ITC_IFLG register)	EIEN0 (D0/ITC_EN register)
Remote controller interrupt	AIFT14 (D14/ITC_AIFLG register)	AIEN14 (D14/ITC_AEN register)
I^{2}C slave interrupt (bus status)	AIFT13 (D13/ITC_AIFLG register)	AIEN13 (D13/ITC_AEN register)
I^{2}C slave interrupt (transmit/receive)	AIFT12 (D12/ITC_AIFLG register)	AIEN12 (D12/ITC_AEN register)
SPI CH.1 interrupt	AIFT10 (D10/ITC_AIFLG register)	AIEN10 (D10/ITC_AEN register)
PT8 CH.3 interrupt	AIFT8 (D8/ITC_AIFLG register)	AIEN8 (D8/ITC_AEN register)
PT8 CH.2 interrupt	AIFT7 (D7/ITC_AIFLG register)	AIEN7 (D7/ITC_AEN register)
PT8 CH.1/T8OSC1 CH.1 interrupt	AIFT6 (D6/ITC_AIFLG register)	AIEN6 (D6/ITC_AEN register)
PT8 CH.0/T8OSC1 CH.0 interrupt	AIFT5 (D5/ITC_AIFLG register)	AIEN5 (D5/ITC_AEN register)
RTC interrupt	AIFT4 (D4/ITC_AIFLG register)	AIEN4 (D4/ITC_AEN register)
ADC interrupt (end of conversion)	AIFT3 (D3/ITC_AIFLG register)	AIEN3 (D3/ITC_AEN register)
ADC interrupt (out of range)	AIFT2 (D2/ITC_AIFLG register)	AIEN2 (D2/ITC_AEN register)
Multi-function timer interrupt	AIFT0 (D0/ITC_AIFLG register)	AIEN0 (D0/ITC_AEN register)

Notes:
- To avoid unexpected interrupts being generated, always be sure to reset the interrupt flag before enabling the interrupt by writing 1 to the interrupt enable bit.
- In addition to the interrupt enable bit, the IE bit of the Processor Status Register (PSR) in the S1C17 Core must be set to 1 to actually generate an interrupt. If the IE bit has been set to 0, the S1C17 Core cannot accept a maskable interrupt request. In this case, the interrupt request sent from the ITC is held and it will be accepted after the IE bit is set to 1.

III.1.3.4 Processing when Multiple Interrupts Occur

The ITC provides the ITC_ELV*x* (0x4306 to 0x430c), ITC_ILV*x* (0x430e to 0x4314), ITC_AILV*x* (0x42e6 to 0x42f4) registers to set an interrupt level (zero to seven) for each cause of interrupt.

Table III.1.3.4.1 Interrupt Level Setup Bits

Hardware interrupt	Interrupt level setup bits	Register address
I^{2}C master interrupt	IILV7[2:0] (D[10:8]/ITC_ILV3 register)	0x4314
SPI CH.0 interrupt	IILV6[2:0] (D[2:0]/ITC_ILV3 register)	0x4314
UART interrupt	IILV4[2:0] (D[2:0]/ITC_ILV2 register)	0x4312
CLG_T8I timer interrupt	IILV3[2:0] (D[10:8]/ITC_ILV1 register)	0x4310
CLG_T8S timer interrupt	IILV2[2:0] (D[2:0]/ITC_ILV1 register)	0x4310
CLG_T8FU0 timer interrupt	IILV1[2:0] (D[10:8]/ITC_ILV0 register)	0x430e
CLG_T16U0 timer interrupt	IILV0[2:0] (D[2:0]/ITC_ILV0 register)	0x430e
Port input interrupt 7	EILV7[2:0] (D[10:8]/ITC_ELV3 register)	0x430c
Port input interrupt 6	EILV6[2:0] (D[2:0]/ITC_ELV3 register)	0x430c
Port input interrupt 5	EILV5[2:0] (D[10:8]/ITC_ELV2 register)	0x430a
Port input interrupt 4	EILV4[2:0] (D[2:0]/ITC_ELV2 register)	0x430a
Port input interrupt 3	EILV3[2:0] (D[10:8]/ITC_ELV1 register)	0x4308
Port input interrupt 2	EILV2[2:0] (D[2:0]/ITC_ELV1 register)	0x4308
Port input interrupt 1	EILV1[2:0] (D[10:8]/ITC_ELV0 register)	0x4306
Port input interrupt 0	EILV0[2:0] (D[2:0]/ITC_ELV0 register)	0x4306
Remote controller interrupt	AILV14[2:0] (D[2:0]/ITC_AILV7 register)	0x42f4
I^{2}C slave interrupt (bus status)	AILV13[2:0] (D[10:8]/ITC_AILV6 register)	0x42f2
I^{2}C slave interrupt (transmit/receive)	AILV12[2:0] (D[2:0]/ITC_AILV6 register)	0x42f2
SPI CH.1 interrupt	AILV10[2:0] (D[2:0]/ITC_AILV5 register)	0x42f0
PT8 CH.3 interrupt	AILV8[2:0] (D[2:0]/ITC_AILV4 register)	0x42ee
PT8 CH.2 interrupt	AILV7[2:0] (D[10:8]/ITC_AILV3 register)	0x42ec
PT8 CH.1/T8OSC1 CH.1 interrupt	AILV6[2:0] (D[2:0]/ITC_AILV3 register)	0x42ec
PT8 CH.0/T8OSC1 CH.0 interrupt	AILV5[2:0] (D[10:8]/ITC_AILV2 register)	0x42ea
RTC interrupt	AILV4[2:0] (D[2:0]/ITC_AILV2 register)	0x42ea
ADC interrupt (end of conversion)	AILV3[2:0] (D[10:8]/ITC_AILV1 register)	0x42e8
ADC interrupt (out of range)	AILV2[2:0] (D[2:0]/ITC_AILV1 register)	0x42e8
Multi-function timer interrupt	AILV0[2:0] (D[2:0]/ITC_AILV0 register)	0x42e6

The highest interrupt level is seven and the lowest is zero.
The set interrupt level is sent to the S1C17 Core at the same time the ITC sends an interrupt request and is used by the S1C17 Core to disable subsequent interrupts that have the same or a lower interrupt level. (See Section III.1.3.6 for more information.)
At initial reset, the interrupt levels are all set to 0. The S1C17 Core does not accept an interrupt request whose interrupt level is set to 0.

In the ITC, the interrupt level is used when two or more causes of interrupt occur simultaneously.
If two or more causes of interrupt that have been enabled by the interrupt enable bits occur simultaneously, the cause of interrupt whose ITC_ELV*x*, ITC_ILV*x*, or ITC_AILV*x* register contains the highest value is allowed by the ITC to send an interrupt request to the S1C17 Core.
If two or more causes of interrupt that have the same interrupt level occur, the interrupt with the smallest vector number is processed first.
Other causes of interrupt are kept pending until all interrupts of higher priority are accepted by the S1C17 Core.
If another cause of interrupt of higher priority occurs during outputting an interrupt request signal, the ITC changes the vector number and interrupt level to that of the new cause of interrupt. The first interrupt request is left pending.

III.1.3.5 Interrupt Trigger Mode

The ITC provides two trigger modes for the port interrupts, the pulse trigger mode and the level trigger mode, to accept either a pulse signal or a level signal as interrupt requests.
The trigger mode can be selected using the EITG*x* bit in the ITC_ELV*x* registers (0x4306 to 0x430c). When EITG*x* is set to 1, level trigger mode is selected; when EITG*x* is set to 0 (default), pulse trigger mode is selected.

The ITC allows these interrupt sources to select the polarity of the interrupt request signal to be sent to the ITC. The signal polarity can be selected using the EITP*x* bit in the ITC_ELV*x* registers (0x4306 to 0x430c). When EITP*x* is set to 1, positive pulse/rising edge (in pulse trigger mode) or active high (in level mode) is selected; when EIPT*x* is set to 0 (default), negative pulse/falling edge (in pulse trigger mode) or active low (in level mode) is selected.

Table III.1.3.5.1 Trigger Mode/Polarity Select Bits

Hardware interrupt	Trigger mode select bit	Trigger polarity select bit	Register address
Port interrupt 0	EITG0 (D4/ITC_ELV0 register)	EITP0 (D5/ITC_ELV0 register)	0x4306
Port interrupt 1	EITG1 (D12/ITC_ELV0 register)	EITP1 (D13/ITC_ELV0 register)	0x4306
Port interrupt 2	EITG2 (D4/ITC_ELV1 register)	EITP2 (D5/ITC_ELV1 register)	0x4308
Port interrupt 3	EITG3 (D12/ITC_ELV1 register)	EITP3 (D13/ITC_ELV1 register)	0x4308
Port interrupt 4	EITG4 (D4/ITC_ELV2 register)	EITP4 (D5/ITC_ELV2 register)	0x430a
Port interrupt 5	EITG5 (D12/ITC_ELV2 register)	EITP5 (D13/ITC_ELV2 register)	0x430a
Port interrupt 6	EITG6 (D4/ITC_ELV3 register)	EITP6 (D5/ITC_ELV3 register)	0x430c
Port interrupt 7	EITG7 (D12/ITC_ELV3 register)	EITP7 (D13/ITC_ELV3 register)	0x430c

The interrupt source modules other than ports and RTC output only a pulse signal (positive pulse) to the ITC to request an interrupt, therefore, no trigger mode selection bit and trigger polarity selection bit are provided. The RTC interrupt can be configured to either pulse or level trigger mode using a control bit in the RTC module.

Pulse trigger mode

In pulse trigger mode, the ITC samples interrupt signals at the rising edge of the system clock. When the active edge (rising edge or falling edge which can be configured by the trigger polarity bit EITP*x*) of the interrupt signal is sampled, the interrupt detector generates an interrupt pulse. This interrupt pulse sets the interrupt flag EIFT*x* to 1 and trigger the ITC. After the interrupt flag is reset to 0 with software, the interrupt flag is enabled to be set again by the following interrupt pulse.

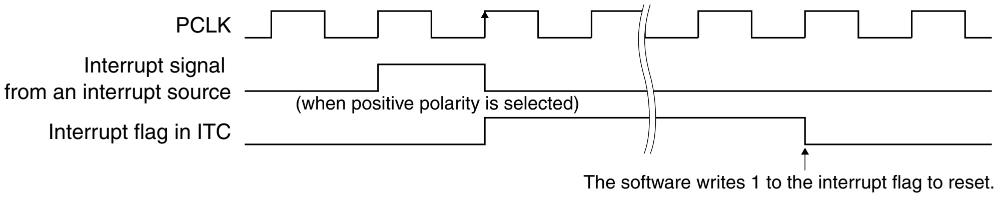

PCLK
Interrupt signal
from an interrupt source
(when positive polarity is selected)
Interrupt flag in ITC
The software writes 1 to the interrupt flag to reset.

Figure III.1.3.5.1 Pulse Trigger Mode

Level trigger mode

In level trigger mode, the ITC samples interrupt at any time. When the active level (high level or low level which can be configured by the trigger polarity bit EITP*x*) of the interrupt signal is sampled, the interrupt detector asserts the interrupt signal. This interrupt signal level sets the interrupt flag EIFT*x* to 1 and trigger the ITC. After the interrupt flag is reset to 0 with software, the interrupt flag will be set again immediately and the ITC will be triggered, if the interrupt source holds the interrupt signal at active level.

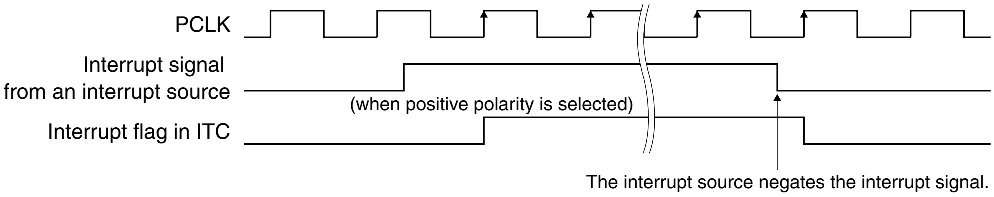

PCLK
Interrupt signal
from an interrupt source
(when positive polarity is selected)
Interrupt flag in ITC
The interrupt source negates the interrupt signal.

Figure III.1.3.5.2 Level Trigger Mode

III.1.3.6 Interrupt Processing by the S1C17 Core

A maskable interrupt to the S1C17 Core occurs when all of the conditions described below are met.

- The ITEN bit (D0/ITC_CTL register) is set to 1.

  * **ITEN**: ITC Enable Bit in the ITC Control (ITC_CTL) Register (D0/0x4304)

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

When a cause of interrupt occurs, the corresponding interrupt flag is set to 1 and the flag remains set until it is reset in the software program or by the hardware for a level triggered interrupt. Therefore, in no cases can the generated cause of interrupt be inadvertently cleared even if the above conditions are not met when the cause of interrupt has occurred. The interrupt will occur when the above conditions are met.
If two or more maskable causes of interrupt occur simultaneously, the cause of interrupt that has the highest priority is allowed to signal an interrupt request to the S1C17 Core. The other interrupts with lower priorities are kept pending until the above conditions are met.

The S1C17 Core keeps sampling interrupt requests every cycle. When the S1C17 Core accepts an interrupt request, it enters interrupt processing after completing execution of the instruction that was being executed.
The following lists the contents executed in interrupt processing.

(1) The PSR and the current program counter (PC) value are saved to the stack.
(2) The IE bit of the PSR is reset to 0 (following maskable interrupts are disabled).
(3) The IL of the PSR is set to the interrupt level of the accepted interrupt (NMI does not change the interrupt level).
(4) The vector of the interrupt occurred is loaded into the PC, thus executing the interrupt handler routine.

Thus, once an interrupt is accepted, all maskable interrupts that may follow are disabled in (2). Multiple interrupts can also be handled by setting the IE bit to 1 in the interrupt handler routine. In this case, since the IL has been changed in (3), only an interrupt that has a higher level than that of the currently processed interrupt is accepted.
When the interrupt handler routine is terminated by the `reti` instruction, the PSR is restored to its previous status before the interrupt has occurred. The program restarts processing after branching to the instruction next to the one that was being executed when the interrupt occurred.

III.1.4 NMI

In the S1C17002, the watchdog timer generates a non-maskable interrupt (NMI). The vector number of NMI is 2, with the vector address set to the vector table's starting address + 8 bytes.
This interrupt is prioritized over other interrupts and is unconditionally accepted by the S1C17 Core.

For how to generate NMI, see Section IV.3, “Watchdog Timer (WDT).”

III.1.5 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 handling as that of the hardware interrupt.

III.1.6 Clearing Standby Mode by Interrupts

The standby mode (HALT and SLEEP) can be cleared by NMI and normal interrupts.

HALT mode can be cleared by an NMI and a normal interrupt from the modules that are operating in HALT mode with the clock supplied. Note, however, that normal interrupts cannot clear HALT mode when the clock supply to the interrupt source module is stopped in HALT mode or when ITEN (D0/ITC_CTL register) has been set to 0 (interrupt controller is disabled).

∗ **ITEN**: ITC Enable Bit in the ITC Control (ITC_CTL) Register (D0/0x4304)

In SLEEP mode, the system clock is not supply to the interrupt controller (ITC). Therefore, the S1C17 Core can only be released from SLEEP mode by initial reset and a normal interrupt from the modules which support level trigger mode such as the port and RTC interrupts.

After the S1C17 Core is released from the standby mode, the instruction next to `halt` or `slp` will be executed.

Notes:
- Normal interrupts are effective to wake up the S1C17 Core even if the IE bit in PSR is set to 0 (interrupt disabled).
- The interrupts that are set to pulse trigger mode cannot be used to clear SLEEP mode.
- When a cause of interrupt is used to clear HALT or SLEEP mode, the interrupt enable bit corresponding to the cause of interrupt must be set to 1 (interrupt enabled).

III.1.7 Details of Control Registers

Table III.1.7.1 List of ITC Registers

Address	Register name		Function
0x42e0	ITC_AIFLG	Additional Interrupt Flag Register	Indicates/resets interrupt occurrence status.
0x42e2	ITC_AEN	Additional Interrupt Enable Register	Enables/disables each maskable interrupt.
0x42e6	ITC_AILV0	Additional Interrupt Level Setup Register 0	Sets the MFT interrupt level.
0x42e8	ITC_AILV1	Additional Interrupt Level Setup Register 1	Sets the ADC interrupt level.
0x42ea	ITC_AILV2	Additional Interrupt Level Setup Register 2	Sets the RTC and PT8 CH.0/T8OSC1 CH.0 interrupt levels.
0x42ec	ITC_AILV3	Additional Interrupt Level Setup Register 3	Sets the PT8 CH.1/T8OSC1 CH.1 and PT8 CH.2 interrupt levels.
0x42ee	ITC_AILV4	Additional Interrupt Level Setup Register 4	Sets the PT8 CH.3 interrupt level.
0x42f0	ITC_AILV5	Additional Interrupt Level Setup Register 5	Sets the SPI CH.1 interrupt level.
0x42f2	ITC_AILV6	Additional Interrupt Level Setup Register 6	Sets the I^{2}C slave interrupt level.
0x42f4	ITC_AILV7	Additional Interrupt Level Setup Register 7	Sets the REMC interrupt level.
0x4300	ITC_IFLG	Interrupt Flag Register	Indicates/resets interrupt occurrence status.
0x4302	ITC_EN	Interrupt Enable Register	Enables/disables each maskable interrupt.
0x4304	ITC_CTL	ITC Control Register	Enables/disables the ITC.
0x4306	ITC_ELV0	External Interrupt Level Setup Register 0	Sets the port 0 and port 1 interrupt levels and trigger modes.
0x4308	ITC_ELV1	External Interrupt Level Setup Register 1	Sets the port 2 and port 3 interrupt levels and trigger modes.
0x430a	ITC_ELV2	External Interrupt Level Setup Register 2	Sets the port 4 and port 5 interrupt levels and trigger modes.
0x430c	ITC_ELV3	External Interrupt Level Setup Register 3	Sets the port 6 and port 7 interrupt levels and trigger modes.
0x430e	ITC_ILV0	Internal Interrupt Level Setup Register 0	Sets the CLG_T16U0 and CLG_T8FU0 timer interrupt levels.
0x4310	ITC_ILV1	Internal Interrupt Level Setup Register 1	Sets the CLG_T8S and CLG_T8I timer interrupt levels.
0x4312	ITC_ILV2	Internal Interrupt Level Setup Register 2	Sets the UART interrupt level.
0x4314	ITC_ILV3	Internal Interrupt Level Setup Register 3	Sets the SPI CH.0 and I^{2}C master interrupt levels.

The following describes each ITC register. These are all 16-bit registers.

Note: When setting the registers, be sure to write a 0, and not a 1, for all “reserved bits.”

0x42e0: Additional Interrupt Flag Register (ITC_AIFLG)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Additional Interrupt Flag Register (ITC_AIFLG)	**0x42e0** (16 bits)	D15	–	reserved	–				–	–	0 when being read.
		D14	**AIFT14**	REMC interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D13	**AIFT13**	I^{2}C slave bus status interrupt flag					0	R/W	
		D12	**AIFT12**	I^{2}C slave transmit/receive interrupt flag					0	R/W	
		D11	–	reserved	–				–	–	0 when being read.
		D10	**AIFT10**	SPI CH.1 interrupt flag	1	Occurred	0	Not occurred	0	R/W	Reset by writing 1.
		D9	–	reserved	–				–	–	0 when being read.
		D8	**AIFT8**	PT8 CH.3 interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D7	**AIFT7**	PT8 CH.2 interrupt flag					0	R/W	
		D6	**AIFT6**	PT8 CH.1/T8OSC1 CH.1 interrupt flag					0	R/W	
		D5	**AIFT5**	PT8 CH.0/T8OSC1 CH.0 interrupt flag					0	R/W	
		D4	**AIFT4**	RTC interrupt flag					0	R/W	
		D3	**AIFT3**	ADC end-of-conversion interrupt flag					0	R/W	
		D2	**AIFT2**	ADC out-of-range interrupt flag					0	R/W	
		D1	–	reserved	–				–	–	0 when being read.
		D0	**AIFT0**	MFT interrupt flag	1	Occurred	0	Not occurred	0	R/W	Reset by writing 1.

D15, D11, D9, D1 Reserved

D[14:12], D10, D[8:2], D0 AIFT[14:12], AIFT10, AIFT[8:2], AIFT0: Interrupt Flag Bits

These bits are interrupt flags to indicate the 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):	Has no effect

The interrupt flag is set to 1 if a cause of interrupt occurs in each peripheral module when ITEN (D0/ ITC_CTL register) in the ITC is set to 1.

If the following conditions are met at this time, an interrupt is generated to the S1C17 Core:

1. The corresponding bit of the Interrupt Enable Register is set to 1.
2. No other interrupt request of higher priority has occurred.
3. The IE bit of the PSR is set to 1 (interrupt enabled).
4. The corresponding interrupt level setup bits are set to a level higher than the S1C17 Core's interrupt level (IL).

The interrupt flag is always set to 1 when a cause of interrupt occurs regardless of how the interrupt enable and interrupt level setup bits are set.

In order for the next interrupt to be accepted after interrupt generation, the interrupt flag must be reset and the PSR must be set up again (by setting the IE bit to 1 or executing the `reti` instruction).

The flag that has been set to 1 can be reset by writing 1.

Table III.1.7.2 Causes of Hardware Interrupt and Interrupt Flags

Interrupt flag	Cause of hardware interrupt
AIFT14 (D14)	REMC interrupt: envelope counter underflow/input rising edge/input falling edge
AIFT13 (D13)	I^{2}C slave interrupt: bus status
AIFT12 (D12)	I^{2}C slave interrupt: transmit buffer empty/receive buffer full
AIFT10 (D10)	SPI CH.1 interrupt: transmit buffer empty/receive buffer full
AIFT8 (D8)	8-bit timer CH.3 interrupt: timer underflow
AIFT7 (D7)	8-bit timer CH.2 interrupt: timer underflow
AIFT6 (D6)	8-bit timer CH.1 interrupt: timer underflow 8-bit OSC1 timer CH.1 interrupt: compare-match
AIFT5 (D5)	8-bit timer CH.0 interrupt: timer underflow 8-bit OSC1 timer CH.0 interrupt: compare-match
AIFT4 (D4)	RTC interrupt: 1/64 second, 1 second, 1 minute, or 1 hour count up
AIFT3 (D3)	ADC interrupt: end of conversion
AIFT2 (D2)	ADC interrupt: out of range
AIFT0 (D0)	Multi-function timer interrupt: compare-match/period-match/protection input

0x42e2: Additional Interrupt Enable Register (ITC_AEN)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Additional Interrupt Enable Register (ITC_AEN)	**0x42e2** (16 bits)	D15	–	reserved	–				–	–	0 when being read.
		D14	**AIEN14**	REMC interrupt enable	1	Enable	0	Disable	0	R/W	
		D13	**AIEN13**	I^{2}C slave bus status interrupt enable					0	R/W	
		D12	**AIEN12**	I^{2}C slave transmit/receive interrupt enable					0	R/W	
		D11	–	reserved	–				–	–	0 when being read.
		D10	**AIEN10**	SPI CH.1 interrupt enable	1	Enable	0	Disable	0	R/W	
		D9	–	reserved	–				–	–	0 when being read.
		D8	**AIEN8**	PT8 CH.3 interrupt enable	1	Enable	0	Disable	0	R/W	
		D7	**AIEN7**	PT8 CH.2 interrupt enable					0	R/W	
		D6	**AIEN6**	PT8 CH.1/T8OSC1 CH.1 interrupt enable					0	R/W	
		D5	**AIEN5**	PT8 CH.0/T8OSC1 CH.0 interrupt enable					0	R/W	
		D4	**AIEN4**	RTC interrupt enable					0	R/W	
		D3	**AIEN3**	ADC end-of-conversion interrupt enable					0	R/W	
		D2	**AIEN2**	ADC out-of-range interrupt enable					0	R/W	
		D1	–	reserved	–				–	–	0 when being read.
		D0	**AIEN0**	MFT interrupt enable	1	Enable	0	Disable	0	R/W	

D15, D11, D9, D1 Reserved

D[14:12], D10, D[8:2], D0 AIEN[14:12], AIEN10, AIEN[8:2], AIEN0: Interrupt Enable Bits

These bits enable or disable interrupt generation.

1 (R/W): Interrupt enabled

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

Interrupts are enabled when the corresponding interrupt enable bit is set to 1 and are disabled when the bit is set to 0.

When using a cause of interrupt to clear standby mode, the corresponding interrupt enable bit must be set to 1.

Table III.1.7.3 Causes of Hardware Interrupt and Interrupt Enable Bits

Interrupt enable bits	Cause of hardware interrupt
AIEN14 (D14)	REMC interrupt: envelope counter underflow/input rising edge/input falling edge
AIEN13 (D13)	I^{2}C slave interrupt: bus status
AIEN12 (D12)	I^{2}C slave interrupt: transmit buffer empty/receive buffer full
AIEN10 (D10)	SPI CH.1 interrupt: transmit buffer empty/receive buffer full
AIEN8 (D8)	8-bit timer CH.3 interrupt: timer underflow
AIEN7 (D7)	8-bit timer CH.2 interrupt: timer underflow
AIEN6 (D6)	8-bit timer CH.1 interrupt: timer underflow 8-bit OSC1 timer CH.1 interrupt: compare-match
AIEN5 (D5)	8-bit timer CH.0 interrupt: timer underflow 8-bit OSC1 timer CH.0 interrupt: compare-match
AIEN4 (D4)	RTC interrupt: 1/64 second, 1 second, 1 minute, or 1 hour count up
AIEN3 (D3)	ADC interrupt: end of conversion
AIEN2 (D2)	ADC interrupt: out of range
AIEN0 (D0)	Multi-function timer interrupt: compare-match/period-match/protection input

0x42e6: Additional Interrupt Level Setup Register 0 (ITC_AILV0)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Additional Interrupt Level Setup Register 0 (ITC_AILV0)	**0x42e6** (16 bits)	D15–3	–	reserved	–	–	–	0 when being read.
		D2–0	**AILV0[2:0]**	MFT interrupt level	0 to 7	0x0	R/W	

D[15:3] Reserved

D[2:0] AILV0[2:0]: MFT Interrupt Level Bits

Sets the interrupt level (0 to 7) of the multi-function timer interrupt. (Default: 0x0)

If the level is set below the IL value of the PSR, the S1C17 Core does not accept the interrupt request.

In the ITC, the interrupt level is used when two or more causes of interrupt occur simultaneously.

If two or more causes of interrupt that have been enabled by the interrupt enable register occur simultaneously, the cause of interrupt whose Interrupt Level Setup Register contains the highest value is allowed by the ITC to send an interrupt request to the S1C17 Core. If two or more causes of interrupt that have the same interrupt level occur, the interrupt with the smallest vector number is processed first. Other causes of interrupt are kept pending until all interrupts of higher priority are accepted by the S1C17 Core. If another cause of interrupt of higher priority occurs during outputting an interrupt request signal, the ITC changes the vector number and interrupt level to those of the new cause of interrupt. The first interrupt request is left pending.

0x42e8: Additional Interrupt Level Setup Register 1 (ITC_AILV1)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Additional Interrupt Level Setup Register 1 (ITC_AILV1)	**0x42e8** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**AILV3[2:0]**	ADC end-of-conversion interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**AILV2[2:0]**	ADC out-of-range interrupt level	0 to 7	0x0	R/W	

D[15:11] Reserved

D[10:8] AILV3[2:0]: ADC End-of-Conversion Interrupt Level Bits

Sets the interrupt level (0 to 7) of the end-of-conversion interrupt of the A/D converter. (Default: 0x0)

If the level is set below the IL value of the PSR, the S1C17 Core does not accept the interrupt request.

In the ITC, the interrupt level is used when two or more causes of interrupt occur simultaneously.

If two or more causes of interrupt that have been enabled by the interrupt enable register occur simultaneously, the cause of interrupt whose Interrupt Level Setup Register contains the highest value is allowed by the ITC to send an interrupt request to the S1C17 Core. If two or more causes of interrupt that have the same interrupt level occur, the interrupt with the smallest vector number is processed first. Other causes of interrupt are kept pending until all interrupts of higher priority are accepted by the S1C17 Core. If another cause of interrupt of higher priority occurs during outputting an interrupt request signal, the ITC changes the vector number and interrupt level to those of the new cause of interrupt. The first interrupt request is left pending.

D[7:3] Reserved

D[2:0] AILV2[2:0]: ADC Out-of-Range Interrupt Level Bits

Sets the interrupt level (0 to 7) of the out-of-range interrupt of the A/D converter. (Default: 0x0)

See the description of AILV3[2:0] (D[10:8]).

0x42ea: Additional Interrupt Level Setup Register 2 (ITC_AILV2)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Additional Interrupt Level Setup Register 2 (ITC_AILV2)	**0x42ea** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**AILV5[2:0]**	PT8 CH.0/T8OSC1 CH.0 interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**AILV4[2:0]**	RTC interrupt level	0 to 7	0x0	R/W	

D[15:11] Reserved

D[10:8] AILV5[2:0]: PT8 CH.0/T8OSC1 CH.0 Interrupt Level Bits

Sets the interrupt level (0 to 7) of the 8-bit programmable timer CH.0 and 8-bit OSC1 timer CH.0 interrupts. (Default: 0x0)

If the level is set below the IL value of the PSR, the S1C17 Core does not accept the interrupt request.

In the ITC, the interrupt level is used when two or more causes of interrupt occur simultaneously.

If two or more causes of interrupt that have been enabled by the interrupt enable register occur simultaneously, the cause of interrupt whose Interrupt Level Setup Register contains the highest value is allowed by the ITC to send an interrupt request to the S1C17 Core. If two or more causes of interrupt that have the same interrupt level occur, the interrupt with the smallest vector number is processed first. Other causes of interrupt are kept pending until all interrupts of higher priority are accepted by the S1C17 Core. If another cause of interrupt of higher priority occurs during outputting an interrupt request signal, the ITC changes the vector number and interrupt level to those of the new cause of interrupt. The first interrupt request is left pending.

D[7:3] Reserved

D[2:0] AILV4[2:0]: RTC Interrupt Level Bits

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

See the description of AILV5[2:0] (D[10:8]).

0x42ec: Additional Interrupt Level Setup Register 3 (ITC_AILV3)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Additional Interrupt Level Setup Register 3 (ITC_AILV3)	**0x42ec** (16 bits)	D15–11	–	reserved	–	–	–	0 when being read.
		D10–8	**AILV7[2:0]**	PT8 CH.2 interrupt level	0 to 7	0x0	R/W	
		D7–3	–	reserved	–	–	–	0 when being read.
		D2–0	**AILV6[2:0]**	PT8 CH.1/T8OSC1 CH.1 interrupt level	0 to 7	0x0	R/W	

D[15:11] Reserved

D[10:8] AILV7[2:0]: PT8 CH.2 Interrupt Level Bits

Sets the interrupt level (0 to 7) of the 8-bit programmable timer CH.2 interrupt. (Default: 0x0)

If the level is set below the IL value of the PSR, the S1C17 Core does not accept the interrupt request.

In the ITC, the interrupt level is used when two or more causes of interrupt occur simultaneously.

If two or more causes of interrupt that have been enabled by the interrupt enable register occur simultaneously, the cause of interrupt whose Interrupt Level Setup Register contains the highest value is allowed by the ITC to send an interrupt request to the S1C17 Core. If two or more causes of interrupt that have the same interrupt level occur, the interrupt with the smallest vector number is processed first. Other causes of interrupt are kept pending until all interrupts of higher priority are accepted by the S1C17 Core. If another cause of interrupt of higher priority occurs during outputting an interrupt request signal, the ITC changes the vector number and interrupt level to those of the new cause of interrupt. The first interrupt request is left pending.

D[7:3] Reserved

D[2:0] AILV6[2:0]: PT8 CH.1/T8OSC1 CH.1 Interrupt Level Bits

Sets the interrupt level (0 to 7) of the 8-bit programmable timer CH.1 and 8-bit OSC1 timer CH.1 interrupts. (Default: 0x0)

See the description of AILV7[2:0] (D[10:8]).

0x42ee: Additional Interrupt Level Setup Register 4 (ITC_AILV4)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Additional Interrupt Level Setup Register 4 (ITC_AILV4)	**0x42ee** (16 bits)	D15–3	–	reserved	–	–	–	0 when being read.
		D2–0	**AILV8[2:0]**	PT8 CH.3 interrupt level	0 to 7	0x0	R/W	

D[15:3] Reserved

D[2:0] AILV8[2:0]: PT8 CH.3 Interrupt Level Bits

Sets the interrupt level (0 to 7) of the 8-bit programmable timer CH.3 interrupt. (Default: 0x0)

If the level is set below the IL value of the PSR, the S1C17 Core does not accept the interrupt request.

In the ITC, the interrupt level is used when two or more causes of interrupt occur simultaneously.

If two or more causes of interrupt that have been enabled by the interrupt enable register occur simultaneously, the cause of interrupt whose Interrupt Level Setup Register contains the highest value is allowed by the ITC to send an interrupt request to the S1C17 Core. If two or more causes of interrupt that have the same interrupt level occur, the interrupt with the smallest vector number is processed first. Other causes of interrupt are kept pending until all interrupts of higher priority are accepted by the S1C17 Core. If another cause of interrupt of higher priority occurs during outputting an interrupt request signal, the ITC changes the vector number and interrupt level to those of the new cause of interrupt. The first interrupt request is left pending.

0x42f0: Additional Interrupt Level Setup Register 5 (ITC_AILV5)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Additional Interrupt Level Setup Register 5 (ITC_AILV5)	**0x42f0** (16 bits)	D15–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**AILV10[2:0]**	SPI CH.1 interrupt level	0 to 7	0x0	R/W	

D[15:3] Reserved

D[2:0] AILV10[2:0]: SPI CH.1 Interrupt Level Bits

Sets the interrupt level (0 to 7) of the SPI CH.1 interrupt. (Default: 0x0)

If the level is set below the IL value of the PSR, the S1C17 Core does not accept the interrupt request.

In the ITC, the interrupt level is used when two or more causes of interrupt occur simultaneously.

If two or more causes of interrupt that have been enabled by the interrupt enable register occur simultaneously, the cause of interrupt whose Interrupt Level Setup Register contains the highest value is allowed by the ITC to send an interrupt request to the S1C17 Core. If two or more causes of interrupt that have the same interrupt level occur, the interrupt with the smallest vector number is processed first. Other causes of interrupt are kept pending until all interrupts of higher priority are accepted by the S1C17 Core. If another cause of interrupt of higher priority occurs during outputting an interrupt request signal, the ITC changes the vector number and interrupt level to those of the new cause of interrupt. The first interrupt request is left pending.

0x42f2: Additional Interrupt Level Setup Register 6 (ITC_AILV6)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Additional Interrupt Level Setup Register 6 (ITC_AILV6)	**0x42f2** (16 bits)	D15–11	–	reserved	–	–	–	0 when being read.
		D10–8	**AILV13[2:0]**	I^{2}C slave bus status interrupt level	0 to 7	0x0	R/W	
		D7–3	–	reserved	–	–	–	0 when being read.
		D2–0	**AILV12[2:0]**	I^{2}C slave transmit/receive interrupt level	0 to 7	0x0	R/W	

D[15:11] Reserved

D[10:8] AILV13[2:0]: I^{2}C Slave Bus Status Interrupt Level Bits

Sets the interrupt level (0 to 7) of the I^{2}C slave bus status interrupt. (Default: 0x0)

If the level is set below the IL value of the PSR, the S1C17 Core does not accept the interrupt request.

In the ITC, the interrupt level is used when two or more causes of interrupt occur simultaneously.

If two or more causes of interrupt that have been enabled by the interrupt enable register occur simultaneously, the cause of interrupt whose Interrupt Level Setup Register contains the highest value is allowed by the ITC to send an interrupt request to the S1C17 Core. If two or more causes of interrupt that have the same interrupt level occur, the interrupt with the smallest vector number is processed first. Other causes of interrupt are kept pending until all interrupts of higher priority are accepted by the S1C17 Core. If another cause of interrupt of higher priority occurs during outputting an interrupt request signal, the ITC changes the vector number and interrupt level to those of the new cause of interrupt. The first interrupt request is left pending.

D[7:3] Reserved

D[2:0] AILV12[2:0]: I^{2}C Slave Transmit/Receive Interrupt Level Bits

Sets the interrupt level (0 to 7) of the I^{2}C slave transmit/receive interrupt. (Default: 0x0)

See the description of AILV13[2:0] (D[10:8]).

0x42f4: Additional Interrupt Level Setup Register 7 (ITC_AILV7)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Additional Interrupt Level Setup Register 7 (ITC_AILV7)	**0x42f4** (16 bits)	D15–3	–	reserved	–	–	–	0 when being read.
		D2–0	**AILV14[2:0]**	REMC interrupt level	0 to 7	0x0	R/W	

D[15:3] Reserved

D[2:0] AILV14[2:0]: REMC Interrupt Level Bits

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

If the level is set below the IL value of the PSR, the S1C17 Core does not accept the interrupt request.

In the ITC, the interrupt level is used when two or more causes of interrupt occur simultaneously.

If two or more causes of interrupt that have been enabled by the interrupt enable register occur simultaneously, the cause of interrupt whose Interrupt Level Setup Register contains the highest value is allowed by the ITC to send an interrupt request to the S1C17 Core. If two or more causes of interrupt that have the same interrupt level occur, the interrupt with the smallest vector number is processed first. Other causes of interrupt are kept pending until all interrupts of higher priority are accepted by the S1C17 Core. If another cause of interrupt of higher priority occurs during outputting an interrupt request signal, the ITC changes the vector number and interrupt level to those of the new cause of interrupt. The first interrupt request is left pending.

0x4300: Interrupt Flag Register (ITC_IFLG)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Interrupt Flag Register (ITC_IFLG)	**0x4300** (16 bits)	D15	**IIFT7**	I^2C master interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D14	**IIFT6**	SPI CH.0 interrupt flag					0	R/W	
		D13	–	reserved	–				–	–	0 when being read.
		D12	**IIFT4**	UART interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D11	**IIFT3**	CLG_T8I timer interrupt flag					0	R/W	
		D10	**IIFT2**	CLG_T8S timer interrupt flag					0	R/W	
		D9	**IIFT1**	CLG_T8FU0 timer interrupt flag					0	R/W	
		D8	**IIFT0**	CLG_T16U0 timer interrupt flag					0	R/W	
		D7	**EIFT7**	Port interrupt 7 flag					0	R/W	Reset by writing 1 in pulse trigger mode. Cannot be reset by software in level trigger mode.
		D6	**EIFT6**	Port interrupt 6 flag					0	R/W	
		D5	**EIFT5**	Port interrupt 5 flag					0	R/W	
		D4	**EIFT4**	Port interrupt 4 flag					0	R/W	
		D3	**EIFT3**	Port interrupt 3 flag					0	R/W	
		D2	**EIFT2**	Port interrupt 2 flag					0	R/W	
		D1	**EIFT1**	Port interrupt 1 flag					0	R/W	
		D0	**EIFT0**	Port interrupt 0 flag					0	R/W	

D13 Reserved

D[15:14], D[12:8] IIFT[7:6], IIFT[4:0]: Interrupt Flag Bits

These bits are interrupt flags to indicate the 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): Has no effect

The interrupt flag is set to 1 if a cause of interrupt occurs in each peripheral module when ITEN (D0/ ITC_CTL register) in the ITC is set to 1.
If the following conditions are met at this time, an interrupt is generated to the S1C17 Core:

1. The corresponding bit of the Interrupt Enable Register is set to 1.
2. No other interrupt request of higher priority has occurred.
3. The IE bit of the PSR is set to 1 (interrupt enabled).
4. The corresponding interrupt level setup bits are set to a level higher than the S1C17 Core's interrupt level (IL).

The interrupt flag is always set to 1 when a cause of interrupt occurs regardless of how the interrupt enable and interrupt level setup bits are set.
In order for the next interrupt to be accepted after interrupt generation, the interrupt flag must be reset and the PSR must be set up again (by setting the IE bit to 1 or executing the `reti` instruction).
The flag that has been set to 1 can be reset by writing 1.

Table III.1.7.4 Causes of Hardware Interrupt and Interrupt Flags

Interrupt flag	Cause of hardware interrupt
IIFT7 (D15)	I^2C master interrupt: transmit buffer empty/receive buffer full
IIFT6 (D14)	SPI CH.0 interrupt: transmit buffer empty/receive buffer full
IIFT4 (D12)	UART interrupt: transmit buffer empty/receive buffer full/receive error
IIFT3 (D11)	CLG_T8I timer interrupt: timer underflow
IIFT2 (D10)	CLG_T8S timer interrupt: timer underflow
IIFT1 (D9)	CLG_T8FU0 timer interrupt: timer underflow
IIFT0 (D8)	CLG_T16U0 timer interrupt: timer underflow

D[7:0] EIFT[7:0]: Interrupt Flag Bits

These bits are interrupt flags to indicate the interrupt cause occurrence status.

1 (R): Cause of interrupt has occurred
0 (R): No cause of interrupt has occurred (default)
1 (W): Has no effect
0 (W): Has no effect

See the description for IIFT*x*.
However, these interrupts allows selection of interrupt trigger conditions using the ITC_ELV*x* register (0x4306 to 0x430c).

Table III.1.7.5 Causes of Hardware Interrupt and Interrupt Flags

Interrupt flag	Cause of hardware interrupt
EIFT7 (D7)	Port input interrupt 7: P*x*7 rising/falling edge or high/low level input
EIFT6 (D6)	Port input interrupt 6: P*x*6 rising/falling edge or high/low level input
EIFT5 (D5)	Port input interrupt 5: P*x*5 rising/falling edge or high/low level input
EIFT4 (D4)	Port input interrupt 4: P*x*4 rising/falling edge or high/low level input
EIFT3 (D3)	Port input interrupt 3: P*x*3 rising/falling edge or high/low level input
EIFT2 (D2)	Port input interrupt 2: P*x*2 rising/falling edge or high/low level input
EIFT1 (D1)	Port input interrupt 1: P*x*1 rising/falling edge or high/low level input
EIFT0 (D0)	Port input interrupt 0: P*x*0 rising/falling edge or high/low level input

0x4302: Interrupt Enable Register (ITC_EN)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Interrupt Enable Register (ITC_EN)	**0x4302** (16 bits)	D15	**IIEN7**	I^{2}C master interrupt enable	1	Enable	0	Disable	0	R/W	
		D14	**IIEN6**	SPI CH.0 interrupt enable					0	R/W	
		D13	**–**	reserved	–				–	–	0 when being read.
		D12	**IIEN4**	UART interrupt enable	1	Enable	0	Disable	0	R/W	
		D11	**IIEN3**	CLG_T8I timer interrupt enable					0	R/W	
		D10	**IIEN2**	CLG_T8S timer interrupt enable					0	R/W	
		D9	**IIEN1**	CLG_T8FU0 timer interrupt enable					0	R/W	
		D8	**IIEN0**	CLG_T16U0 timer interrupt enable					0	R/W	
		D7	**EIEN7**	Port interrupt 7 enable					0	R/W	
		D6	**EIEN6**	Port interrupt 6 enable					0	R/W	
		D5	**EIEN5**	Port interrupt 5 enable					0	R/W	
		D4	**EIEN4**	Port interrupt 4 enable					0	R/W	
		D3	**EIEN3**	Port interrupt 3 enable					0	R/W	
		D2	**EIEN2**	Port interrupt 2 enable					0	R/W	
		D1	**EIEN1**	Port interrupt 1 enable					0	R/W	
		D0	**EIEN0**	Port interrupt 0 enable					0	R/W	

D13 Reserved

D[15:14], D[12:8], D[5:0] IIEN[7:6], IIEN[4:0], EIEN[7:0]: Interrupt Enable Bits

These bits enable or disable interrupt generation.

1 (R/W): Interrupt enabled

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

Interrupts are enabled when the corresponding interrupt enable bit is set to 1 and are disabled when the bit is set to 0.

When using a cause of interrupt to clear standby mode, the corresponding interrupt enable bit must be set to 1.

Table III.1.7.6 Causes of Hardware Interrupt and Interrupt Enable Bits

Interrupt enable bits	Cause of hardware interrupt
IIEN7 (D15)	I^{2}C master interrupt: transmit buffer empty/receive buffer full
IIEN6 (D14)	SPI CH.0 interrupt: transmit buffer empty/receive buffer full
IIEN4 (D12)	UART interrupt: transmit buffer empty/receive buffer full/receive error
IIEN3 (D11)	CLG_T8I timer interrupt: timer underflow
IIEN2 (D10)	CLG_T8S timer interrupt: timer underflow
IIEN1 (D9)	CLG_T8FU0 timer interrupt: timer underflow
IIEN0 (D8)	CLG_T16U0 timer interrupt: timer underflow
EIEN7 (D7)	Port input interrupt 7: P*x*7 rising/falling edge or high/low level input
EIEN6 (D6)	Port input interrupt 6: P*x*6 rising/falling edge or high/low level input
EIEN5 (D5)	Port input interrupt 5: P*x*5 rising/falling edge or high/low level input
EIEN4 (D4)	Port input interrupt 4: P*x*4 rising/falling edge or high/low level input
EIEN3 (D3)	Port input interrupt 3: P*x*3 rising/falling edge or high/low level input
EIEN2 (D2)	Port input interrupt 2: P*x*2 rising/falling edge or high/low level input
EIEN1 (D1)	Port input interrupt 1: P*x*1 rising/falling edge or high/low level input
EIEN0 (D0)	Port input interrupt 0: P*x*0 rising/falling edge or high/low level input

0x4304: ITC Control Register (ITC_CTL)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
ITC Control Register (ITC_CTL)	**0x4304** (16 bits)	D15–1	–	reserved	–				–	–	0 when being read.
		D0	**ITEN**	ITC enable	1	Enable	0	Disable	0	R/W	

D[15:1] Reserved

D0 ITEN: ITC Enable Bit

Enables the ITC to control interrupt generation.

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

Before the ITC can be used, this bit must be set to 1.

0x4306: External Interrupt Level Setup Register 0 (ITC_ELV0)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
External Interrupt Level Setup Register 0 (ITC_ELV0)	**0x4306** (16 bits)	D15–14	**–**	reserved	–				–	–	0 when being read.
		D13	**EITP1**	Port interrupt 1 trigger polarity	1	Positive	0	Negative	0	R/W	
		D12	**EITG1**	Port interrupt 1 trigger mode	1	Level	0	Pulse	0	R/W	
		D11	**–**	reserved	–				–	–	0 when being read.
		D10–8	**EILV1[2:0]**	Port interrupt 1 level	0 to 7				0x0	R/W	
		D7–6	**–**	reserved	–				–	–	0 when being read.
		D5	**EITP0**	Port interrupt 0 trigger polarity	1	Positive	0	Negative	0	R/W	
		D4	**EITG0**	Port interrupt 0 trigger mode	1	Level	0	Pulse	0	R/W	
		D3	**–**	reserved	–				–	–	0 when being read.
		D2–0	**EILV0[2:0]**	Port interrupt 0 level	0 to 7				0x0	R/W	

D[15:14] Reserved

D13 EITP1: Port Interrupt 1 Trigger Polarity Bit

Selects the polarity of the port interrupt 1 signal.

1 (R/W): Positive/active high

0 (R/W): Negative/active low (default)

In pulse trigger mode, the port outputs a positive pulse for an interrupt request to the ITC when this bit is set to 1 or a negative pulse when this bit is set to 0.

In level trigger mode, the port outputs an active high signal for an interrupt request to the ITC when this bit is set to 1 or an active low signal when this bit is set to 0.

D12 EITG1: Port Interrupt 1 Trigger Mode Bit

Selects the trigger mode of the port interrupt 1.

1 (R/W): Level trigger mode

0 (R/W): Pulse trigger mode (default)

In pulse trigger mode, the ITC samples interrupt signals at the rising edge of the system clock. When a pulse with the specified polarity is sampled, the ITC sets the interrupt flag (EIFT*x*) to 1 and stops sampling of that interrupt signal. The ITC resumes the sampling operation for the interrupt signal after the interrupt flag (EIFT*x*) is reset to 0 in the application program (interrupt handler).

In level trigger mode, the ITC continuously samples interrupt signals at every rising edge of the system clock. The interrupt flag (EIFT*x*) is set to 1 when the specified active level is sampled and is reset to 0 when the inactive level is sampled. In this mode, writing 1 cannot reset the interrupt flag (EIFT*x*). Therefore, the interrupt source module must hold the interrupt signal to high until the S1C17 Core accepts the interrupt request and must reset the interrupt signal after that.

D11 Reserved

D[10:8] EILV1[2:0]: Port Interrupt 1 Level Bits

Sets the interrupt level (0 to 7) of the port interrupt 1. (Default: 0x0)

If the level is set below the IL value of the PSR, the S1C17 Core does not accept the interrupt request.

In the ITC, the interrupt level is used when two or more causes of interrupt occur simultaneously.

If two or more causes of interrupt that have been enabled by the interrupt enable register occur simultaneously, the cause of interrupt whose Interrupt Level Setup Register contains the highest value is allowed by the ITC to send an interrupt request to the S1C17 Core. If two or more causes of interrupt that have the same interrupt level occur, the interrupt with the smallest vector number is processed first. Other causes of interrupt are kept pending until all interrupts of higher priority are accepted by the S1C17 Core. If another cause of interrupt of higher priority occurs during outputting an interrupt request signal, the ITC changes the vector number and interrupt level to those of the new cause of interrupt. The first interrupt request is left pending.

D[7:6] Reserved

D5 EITP0: Port Interrupt 0 Trigger Polarity Bit

Selects the polarity of the port interrupt 0 signal.
1 (R/W): Positive/active high
0 (R/W): Negative/active low (default)

See the description of EITP1 (D13).

D4 EITG0: Port Interrupt 0 Trigger Mode Bit

Selects the trigger mode of the port interrupt 0.
1 (R/W): Level trigger mode
0 (R/W): Pulse trigger mode (default)

See the description of EITG1 (D12).

D3 Reserved

D[2:0] EILV0[2:0]: Port Interrupt 0 Level Bits

Sets the interrupt level (0 to 7) of the port interrupt 0. (Default: 0x0)
See the description of EILV1[2:0] (D[10:8]).

0x4308: External Interrupt Level Setup Register 1 (ITC_ELV1)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
External Interrupt Level Setup Register 1 (ITC_ELV1)	**0x4308** (16 bits)	D15–14	–	reserved	–				–	–	0 when being read.
		D13	**EITP3**	Port interrupt 3 trigger polarity	1	Positive	0	Negative	0	R/W	
		D12	**EITG3**	Port interrupt 3 trigger mode	1	Level	0	Pulse	0	R/W	
		D11	–	reserved	–				–	–	0 when being read.
		D10–8	**EILV3[2:0]**	Port interrupt 3 level	0 to 7				0x0	R/W	
		D7–6	–	reserved	–				–	–	0 when being read.
		D5	**EITP2**	Port interrupt 2 trigger polarity	1	Positive	0	Negative	0	R/W	
		D4	**EITG2**	Port interrupt 2 trigger mode	1	Level	0	Pulse	0	R/W	
		D3	–	reserved	–				–	–	0 when being read.
		D2–0	**EILV2[2:0]**	Port interrupt 2 level	0 to 7				0x0	R/W	

D[15:14] Reserved

D13 EITP3: Port Interrupt 3 Trigger Polarity Bit

Selects the polarity of the port interrupt 3 signal.
1 (R/W): Positive/active high
0 (R/W): Negative/active low (default)

See the description of EITP1 (D13) in the ITC_ELV0 register (0x4306).

D12 EITG3: Port Interrupt 3 Trigger Mode Bit

Selects the trigger mode of the port interrupt 3.
1 (R/W): Level trigger mode
0 (R/W): Pulse trigger mode (default)

See the description of EITG1 (D12) in the ITC_ELV0 register (0x4306).

D11 Reserved

D[10:8] EILV3[2:0]: Port Interrupt 3 Level Bits

Sets the interrupt level (0 to 7) of the port interrupt 3. (Default: 0x0)
See the description of EILV1[2:0] (D[10:8]) in the ITC_ELV0 register (0x4306).

D[7:6] Reserved

D5 EITP2: Port Interrupt 2 Trigger Polarity Bit

Selects the polarity of the port interrupt 2 signal.
1 (R/W): Positive/active high
0 (R/W): Negative/active low (default)

See the description of EITP1 (D13) in the ITC_ELV0 register (0x4306).

D4 EITG2: Port Interrupt 2 Trigger Mode Bit

Selects the trigger mode of the port interrupt 2.
1 (R/W): Level trigger mode
0 (R/W): Pulse trigger mode (default)

See the description of EITG1 (D12) in the ITC_ELV0 register (0x4306).

D3 Reserved

D[2:0] EILV2[2:0]: Port Interrupt 2 Level Bits

Sets the interrupt level (0 to 7) of the port interrupt 2. (Default: 0x0)
See the description of EILV1[2:0] (D[10:8]) in the ITC_ELV0 register (0x4306).

0x430a: External Interrupt Level Setup Register 2 (ITC_ELV2)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
External Interrupt Level Setup Register 2 (ITC_ELV2)	**0x430a** (16 bits)	D15–14	–	reserved	–				–	–	0 when being read.
		D13	**EITP5**	Port interrupt 5 trigger polarity	1	Positive	0	Negative	0	R/W	
		D12	**EITG5**	Port interrupt 5 trigger mode	1	Level	0	Pulse	0	R/W	
		D11	–	reserved	–				–	–	0 when being read.
		D10–8	**EILV5[2:0]**	Port interrupt 5 level	0 to 7				0x0	R/W	
		D7–6	–	reserved	–				–	–	0 when being read.
		D5	**EITP4**	Port interrupt 4 trigger polarity	1	Positive	0	Negative	0	R/W	
		D4	**EITG4**	Port interrupt 4 trigger mode	1	Level	0	Pulse	0	R/W	
		D3	–	reserved	–				–	–	0 when being read.
		D2–0	**EILV4[2:0]**	Port interrupt 4 level	0 to 7				0x0	R/W	

D[15:14] Reserved

D13 EITP5: Port Interrupt 5 Trigger Polarity Bit

Selects the polarity of the port interrupt 5 signal.

1 (R/W): Positive/active high

0 (R/W): Negative/active low (default)

See the description of EITP1 (D13) in the ITC_ELV0 register (0x4306).

D12 EITG5: Port Interrupt 5 Trigger Mode Bit

Selects the trigger mode of the port interrupt 5.

1 (R/W): Level trigger mode

0 (R/W): Pulse trigger mode (default)

See the description of EITG1 (D12) in the ITC_ELV0 register (0x4306).

D11 Reserved

D[10:8] EILV5[2:0]: Port Interrupt 5 Level Bits

Sets the interrupt level (0 to 7) of the port interrupt 5. (Default: 0x0)

See the description of EILV1[2:0] (D[10:8]) in the ITC_ELV0 register (0x4306).

D[7:6] Reserved

D5 EITP4: Port Interrupt 4 Trigger Polarity Bit

Selects the polarity of the port interrupt 4 signal.

1 (R/W): Positive/active high

0 (R/W): Negative/active low (default)

See the description of EITP1 (D13) in the ITC_ELV0 register (0x4306).

D4 EITG4: Port Interrupt 4 Trigger Mode Bit

Selects the trigger mode of the port interrupt 4.

1 (R/W): Level trigger mode

0 (R/W): Pulse trigger mode (default)

See the description of EITG1 (D12) in the ITC_ELV0 register (0x4306).

D3 Reserved

D[2:0] EILV4[2:0]: Port Interrupt 4 Level Bits

Sets the interrupt level (0 to 7) of the port interrupt 4. (Default: 0x0)

See the description of EILV1[2:0] (D[10:8]) in the ITC_ELV0 register (0x4306).

0x430c: External Interrupt Level Setup Register 3 (ITC_ELV3)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
External Interrupt Level Setup Register 3 (ITC_ELV3)	**0x430c** (16 bits)	D15–14	–	reserved	–				–	–	0 when being read.
		D13	**EITP7**	Port interrupt 7 trigger polarity	1	Positive	0	Negative	0	R/W	
		D12	**EITG7**	Port interrupt 7 trigger mode	1	Level	0	Pulse	0	R/W	
		D11	–	reserved	–				–	–	0 when being read.
		D10–8	**EILV7[2:0]**	Port interrupt 7 level	0 to 7				0x0	R/W	
		D7–6	–	reserved	–				–	–	0 when being read.
		D5	**EITP6**	Port interrupt 6 trigger polarity	1	Positive	0	Negative	0	R/W	
		D4	**EITG6**	Port interrupt 6 trigger mode	1	Level	0	Pulse	0	R/W	
		D3	–	reserved	–				–	–	0 when being read.
		D2–0	**EILV6[2:0]**	Port interrupt 6 level	0 to 7				0x0	R/W	

D[15:14] Reserved

D13 EITP7: Port Interrupt 7 Trigger Polarity Bit

Selects the polarity of the port interrupt 7 signal.
1 (R/W): Positive/active high
0 (R/W): Negative/active low (default)

See the description of EITP1 (D13) in the ITC_ELV0 register (0x4306).

D12 EITG7: Port Interrupt 7 Trigger Mode Bit

Selects the trigger mode of the port interrupt 7.
1 (R/W): Level trigger mode
0 (R/W): Pulse trigger mode (default)

See the description of EITG1 (D12) in the ITC_ELV0 register (0x4306).

D11 Reserved

D[10:8] EILV7[2:0]: Port Interrupt 7 Level Bits

Sets the interrupt level (0 to 7) of the port interrupt 7. (Default: 0x0)
See the description of EILV1[2:0] (D[10:8]) in the ITC_ELV0 register (0x4306).

D[7:6] Reserved

D5 EITP6: Port Interrupt 6 Trigger Polarity Bit

Selects the polarity of the port interrupt 6 signal.
1 (R/W): Positive/active high
0 (R/W): Negative/active low (default)

See the description of EITP1 (D13) in the ITC_ELV0 register (0x4306).

D4 EITG6: Port Interrupt 6 Trigger Mode Bit

Selects the trigger mode of the port interrupt 6.
1 (R/W): Level trigger mode
0 (R/W): Pulse trigger mode (default)

See the description of EITG1 (D12) in the ITC_ELV0 register (0x4306).

D3 Reserved

D[2:0] EILV6[2:0]: Port Interrupt 6 Level Bits

Sets the interrupt level (0 to 7) of the port interrupt 6. (Default: 0x0)
See the description of EILV1[2:0] (D[10:8]) in the ITC_ELV0 register (0x4306).

0x430e: Internal Interrupt Level Setup Register 0 (ITC_ILV0)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Internal Interrupt Level Setup Register 0 (ITC_ILV0)	**0x430e** (16 bits)	D15–11	–	reserved	–	–	–	0 when being read.
		D10–8	**IILV1[2:0]**	CLG_T8FU0 timer interrupt level	0 to 7	0x0	R/W	
		D7–3	–	reserved	–	–	–	0 when being read.
		D2–0	**IILV0[2:0]**	CLG_T16U0 timer interrupt level	0 to 7	0x0	R/W	

D[15:11] Reserved

D[10:8] IILV1[2:0]: CLG_T8FU0 Timer Interrupt Level Bits

Sets the interrupt level (0 to 7) of the CLG_T8FU0 timer interrupt. (Default: 0x0)

If the level is set below the IL value of the PSR, the S1C17 Core does not accept the interrupt request.

In the ITC, the interrupt level is used when two or more causes of interrupt occur simultaneously.

If two or more causes of interrupt that have been enabled by the interrupt enable register occur simultaneously, the cause of interrupt whose Interrupt Level Setup Register contains the highest value is allowed by the ITC to send an interrupt request to the S1C17 Core. If two or more causes of interrupt that have the same interrupt level occur, the interrupt with the smallest vector number is processed first. Other causes of interrupt are kept pending until all interrupts of higher priority are accepted by the S1C17 Core. If another cause of interrupt of higher priority occurs during outputting an interrupt request signal, the ITC changes the vector number and interrupt level to those of the new cause of interrupt. The first interrupt request is left pending.

D[7:3] Reserved

D[2:0] IILV0[2:0]: CLG_T16U0 Timer Interrupt Level Bits

Sets the interrupt level (0 to 7) of the CLG_T16U0 timer interrupt. (Default: 0x0)

See the description of IILV1[2:0] (D[10:8]).

0x4310: Internal Interrupt Level Setup Register 1 (ITC_ILV1)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Internal Interrupt Level Setup Register 1 (ITC_ILV1)	**0x4310** (16 bits)	D15–11	–	reserved	–	–	–	0 when being read.
		D10–8	**IILV3[2:0]**	CLG_T8I timer interrupt level	0 to 7	0x0	R/W	
		D7–3	–	reserved	–	–	–	0 when being read.
		D2–0	**IILV2[2:0]**	CLG_T8S timer interrupt level	0 to 7	0x0	R/W	

D[15:11] Reserved

D[10:8] IILV3[2:0]: CLG_T8I Timer Interrupt Level Bits

Sets the interrupt level (0 to 7) of the CLG_T8I timer interrupt. (Default: 0x0)

See the description of IILV1[2:0] (D[10:8]) in the ITC_ILV0 register (0x430e).

D[7:3] Reserved

D[2:0] IILV2[2:0]: CLG_T8S Timer Interrupt Level Bits

Sets the interrupt level (0 to 7) of the CLG_T8S timer interrupt. (Default: 0x0)

See the description of IILV1[2:0] (D[10:8]) in the ITC_ILV0 register (0x430e).

0x4312: Internal Interrupt Level Setup Register 2 (ITC_ILV2)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Internal Interrupt Level Setup Register 2 (ITC_ILV2)	**0x4312** (16 bits)	D15–3	–	reserved	–	–	–	0 when being read.
		D2–0	**IILV4[2:0]**	UART interrupt level	0 to 7	0x0	R/W	

D[15:3] Reserved

D[2:0] IILV4[2:0]: UART Interrupt Level Bits

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

See the description of IILV1[2:0] (D[10:8]) in the ITC_ILV0 register (0x430e).

0x4314: Internal Interrupt Level Setup Register 3 (ITC_ILV3)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Internal Interrupt Level Setup Register 3 (ITC_ILV3)	**0x4314** (16 bits)	D15–11	–	reserved	–	–	–	0 when being read.
		D10–8	**IILV7[2:0]**	I^{2}C master interrupt level	0 to 7	0x0	R/W	
		D7–3	–	reserved	–	–	–	0 when being read.
		D2–0	**IILV6[2:0]**	SPI CH.0 interrupt level	0 to 7	0x0	R/W	

D[15:11] Reserved

D[10:8] IILV7[2:0]: I^{2}C Master Interrupt Level Bits

Sets the interrupt level (0 to 7) of the I^{2}C master interrupt. (Default: 0x0)

See the description of IILV1[2:0] (D[10:8]) in the ITC_ILV0 register (0x430e).

D[7:3] Reserved

D[2:0] IILV6[2:0]: SPI CH.0 Interrupt Level Bits

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

See the description of IILV1[2:0] (D[10:8]) in the ITC_ILV0 register (0x430e).

III.1.8 Precautions

- To prevent another interrupt from being generated for the same cause again after generation of an interrupt, be sure to reset the interrupt flag before enabling interrupts and setting the PSR again or executing the reti instruction.
- When an interrupt is used to cancel standby mode, the S1C17 Core always executes the instruction that follows the slp or halt instruction before the interrupt is accepted. Therefore, place one nop instruction following the slp or halt instruction to ensure that the interrupt can be generated. If a memory access instruction that makes the S1C17 Core into a wait status is placed, level interrupt signals may not be accepted and the interrupt handling may not be executed.
- To ensure that the interrupt handler routine will be executed when a port input interrupt (level interrupt) is used to cancel standby mode, the port input signal must be asserted for more than the time shown below.

  (1) When the clock is stopped during SLEEP mode
  OSC3 oscillation start time + OSC3 oscillation stabilization wait time (set by the user) + 10 system clock cycle time

  (2) When the clock is not stopped during SLEEP mode, or in HALT mode
  10 system clock cycle time
- To generate an interrupt or to clear standby mode, the CMU must be configured to supply the clock required for the interrupt along with the interrupt control register settings.
  Table III.1.8.1 lists the clock settings required for generating interrupts.

Table III.1.8.1 Clock Settings Required for Generating Interrupts

Clock system	Mode in which the CMU register settings are effective	Clock supply status			Clock settings required for generating interrupts																			
					Normal/HALT mode																SLEEP mode			
		Normal mode	HALT mode	SLEEP mode	AMI	DBG	NMI	MFT	ADC	T16	T8	UART	SPI	I^2C	PORT	RTC	PT8	SPI	REMC	T8	DBG	NMI	PORT	RTC
PCLK	Normal/HALT	REG	REG	Stop				○	○	○	○	○	○	○	○	○	○	○	○	○			○	○
ROMC	HALT	Cannot be stopped	REG	Stop																				
MFT_CLK	Normal/HALT	REG	REG	Stop				○																
PT8_CLK	Normal/HALT	REG	REG	Stop													○							
SRAMC_CLK	HALT	Cannot be stopped	REG	Stop																				
RTC_SAPB_CLK	Normal/HALT	REG	REG	Stop												○				○				○
PORT_CLK	Normal/HALT	REG	REG	Stop											∗4								∗5	
WDT_CLK	Normal/HALT	REG	REG	Stop			∗3																	
ADC_CLK	Normal/HALT	REG	REG	Stop					○															
REMC_CLK	Normal/HALT	REG	REG	Stop															○					
SPI_CLK	Normal/HALT	REG	REG	Stop														○						
T8OSC1_CLK	Normal/HALT	REG	REG	Stop																○				
CPU	–	Cannot be stopped	Stop	Stop	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/
RTC	–	Cannot be stopped ∗1	Cannot be stopped ∗1	Cannot be stopped ∗1												○								○
OSC1 oscillator	–	Cannot be stopped ∗2	Cannot be stopped ∗2	Cannot be stopped ∗2	∗6	∗6	∗6	∗6	∗6	∗6	∗6	∗6	∗6	∗6	∗6	○	∗6	∗6	∗6	∗6	∗6	∗6	∗6	○
OSC3 oscillator	Normal/HALT	REG	REG	REG	∗6	∗6	∗6	∗6	∗6	∗6	∗6	∗6	∗6	∗6	∗6	∗6	∗6	∗6	∗6	∗6	∗6	∗6	∗6	∗6

REG: Configurable with the register
○: Clock absolutely required for generating interrupts
AMI: Address misaligned interrupt
∗1: Stops when the function is disabled in the peripheral module.
∗2: Cannot be oscillated if the external input signal level is fixed.
∗3: The clock must be supplied to generate NMI from WDT.
∗4: Required only when pulse trigger interrupt is configured.
∗5: Not required after the registers have been configured. (Level trigger interrupt only)
∗6: The OSC1 or OSC3 oscillator circuit, which is used as the system clock source, must always be active except in SLEEP mode.

THIS PAGE IS BLANK.

III.2 ROM Controller (ROMC)

III.2.1 Overview of the ROMC

The S1C17002 contains a mask ROM in the 128K-byte area from address 0x20000 to address 0x3ffff for storing application programs and data. Address 0x20000 is defined as the vector table base address by default, therefore the reset vector must be placed on this address.

The ROM controller (ROMC) provides the registers to configure the vector table base address and a ROM read access condition.

III.2.2 Boot Address and TTBR

After an initial reset, the vector table is located at the beginning of the ROM (address 0x20000). Therefore, the boot vector must be written to address 0x20000. See Section I.5.4 for the vector table.

The S1C17002 allows the base (starting) address of the vector table to be set using the TTBR_LOW and TTBR_HIGH registers (0x5814, 0x5816) in the ROMC module. This makes it possible to change the vector table location after the system has booted from address 0x20000 once.

The TTBR_LOW register specifies the low-order 16 bits of the vector table address and the TTBR_HIGH register specifies the high-order 8 bits. Bits 7 to 0 in the TTBR_LOW register are fixed at 0, so the trap table starting address always begins with a 256-byte boundary address.

Note: The Vector Table Base Registers are write-protected. Before these registers can be rewritten, write protection must be removed by writing data 0x96 to the ROMC Protect Register (0x5810). Note that since unnecessary rewrites to the Vector Table Base Registers could lead to erratic system operation, the ROMC Protect Register (0x5810) should be set to other than 0x96 unless the Vector Table Base Registers must be rewritten.

III.2.3 Read Access Control

The ROM can be read in a minimum of one cycle.

The ROM controller can insert a wait cycle in the ROM read cycle. The number of system clock cycles to be inserted as a wait cycle can be specified using ROM_WAIT[2:0] (D[2:0]/ROMC_WAIT register).

∗ **ROM_WAIT[2:0]**: ROM Read Access Wait Cycle Setup Bits in the ROMC Wait (ROMC_WAIT) Register (D[2:0]/0x5804)

Table III.2.3.1 Setting ROM Read Access Wait Cycle

ROM_WAIT[2:0]	Number of wait cycles	Number of read access cycles	System clock frequency
0x7	7 cycles	8 cycles	Less than 20 MHz
0x6	6 cycles	7 cycles	
0x5	5 cycles	6 cycles	
0x4	4 cycles	5 cycles	
0x3	3 cycles	4 cycles	
0x2	2 cycles	3 cycles	
0x1	1 cycle	2 cycles	
0x0	0 cycles	1 cycle	

(Default: 0x7)

The S1C17002 can operate with 0 wait cycles by setting ROM_WAIT[2:0] to 0x0.

III.2.4 Details of Control Registers

Table III.2.4.1 List of ROMC Registers

Address	Register name		Function
0x5804	ROMC_WAIT	ROMC Wait Register	Sets the wait cycle for ROM read.
0x5810	ROMC_PRT	ROMC Protect Register	Enables ROMC registers for writing.
0x5814	TTBR_LOW	Trap Table Base Register 0	Sets the vector table address.
0x5816	TTBR_HIGH	Trap Table Base Register 1	

The following describes each ROMC control register. These are all 16-bit registers.

Note: When setting the registers, be sure to write 0, and not 1, for all “reserved bits.”

0x5804: ROMC Wait Register (ROMC_WAIT)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
ROMC Wait Register (ROMC_WAIT)	**0x5804** (16 bits)	D15–3	–	reserved	–		–	–	0 when being read.
		D2–0	**ROM_WAIT [2:0]**	ROM read access wait cycle setup	ROM_WAIT[2:0]	Wait cycle	0x7	R/W	
					0x7 : 0x0	7 cycles : 0 cycles			

D[15:3] Reserved

D[2:0] ROM_WAIT[2:0]: ROM Read Access Wait Cycle Setup Bits

These bits set the number of wait cycles to be inserted when the ROM is read.

Table III.2.4.2 Setting Read Access Wait Cycle

ROM_WAIT[2:0]	Number of wait cycles	Number of read access cycles	System clock frequency
0x7	7 cycles	8 cycles	Less than 20 MHz
0x6	6 cycles	7 cycles	
0x5	5 cycles	6 cycles	
0x4	4 cycles	5 cycles	
0x3	3 cycles	4 cycles	
0x2	2 cycles	3 cycles	
0x1	1 cycle	2 cycles	
0x0	0 cycles	1 cycle	

(Default: 0x7)

The S1C17002 can operate with 0 wait cycles by setting ROM_WAIT[2:0] to 0x0.

0x5810: ROMC Protect Register (ROMC_PRT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
ROMC Protect Register (ROMC_PRT)	**0x5810** (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	**ROMC_PRT [7:0]**	ROMC register protect flag	Writing 10010110 (0x96) removes the write protection of the Trap Table Base Registers (0x5814–0x5816). Writing another value set the write protection.	0x0	R/W	

D[15:8] Reserved

D[7:0] ROMC_PRT[7:0]: ROMC Register Protect Flag Bits

Enables/disables write protection of the ROMC registers (0x5814–0x5816).

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

Before altering the Trap Table Base Registers (0x5814–0x5816), write data 0x96 to the ROMC Protect Register to disable write protection. If this register is set to other than 0x96, even if an attempt is made to alter the Trap Table Base Registers by executing a write instruction, the content of said register will not be altered even though the instruction may have been executed without a problem. Once this register is set to 0x96, the Trap Table Base Registers can be rewritten any number of times until being reset to other than 0x96. When rewriting the Trap Table Base Registers has finished, this register should be set to other than 0x96 to prevent accidental writing to the Trap Table Base Registers.

0x5814–0x5816: Trap Table Base Registers (TTBR_LOW, TTBR_HIGH)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Trap Table Base Register 0 (TTBR_LOW)	**0x5814** (16 bits)	D15–8	**TTBR[15:8]**	Trap table base address A[15:8]	0x0–0xff	0x0	R/W	
		D7–0	**TTBR[7:0]**	Trap table base address A[7:0] (fixed at 0)	0x0	0x0	R	
Trap Table Base Register 1 (TTBR_HIGH)	**0x5816** (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	**TTBR[23:16]**	Trap table base address A[23:16]	0x0–0xff	0x2	R/W	

Note: The Trap Table Base Registers are write-protected. Before these registers can be rewritten, write protection must be removed by writing data 0x96 to the ROMC Protect Register (0x5810). Note that since unnecessary rewrites to the Trap Table Base Registers could lead to erratic system operation, the ROMC Protect Register (0x5810) should be set to other than 0x96 unless the Trap Table Base Registers must be rewritten.

D[7:0]/0x5816, D[15:0]/0x5814 TTBR[23:0]: Trap Table Base Address Bits

These registers are used to set the starting address of the vector table.

After an initial reset, the TTBR_LOW/HIGH registers are set to 0x20000. Therefore, even when the trap table position is changed, it is necessary that at least the reset vector be written to the above address. Bits 7 to 0 in the TTBR_LOW register are fixed at 0, so the trap table starting address always begins with a 256-byte boundary address.

III.2.5 Precautions

The Trap Table Base Registers (0x5814–0x5816) are write-protected. Before these registers can be rewritten, their write protection must be removed by writing data 0x96 to the ROMC Protect Register (0x5810). Note that since unnecessary rewrites to addresses 0x5814–0x5816 could lead to erratic system operation, the ROMC Protect Register (0x5810) should be set to other than 0x96 unless the Trap Table Base Registers must be rewritten.

III.3 SRAM Controller (SRAMC)

III.3.1 Overview of the SRAMC

The SRAM Controller (SRAMC) is a bus controller module for accessing internal extended peripheral modules. The SRAMC functions and features are outlined below.

Extended internal peripheral interface

- 8-bit/16-bit selectable data bus
- Supports 8-bit, 16-bit, or 24-bit access.
- 0 wait access (4-cycle bus access)
- Little endian (fixed)

III.3.2 Data Configuration in Memory

The S1C17002 SRAMC handles 8-bit, 16-bit, and 24/32-bit data. To access data in memory, addresses aligned to the boundary of the data size must be specified. Specifying other addresses generates address misaligned interrupts. Instructions (e.g., stack manipulating and branch instructions) that rewrite the content of the Stack Pointer (SP) or Program Counter (PC) forcibly alter the address specified to a boundary address to prevent address misaligned interrupts. For details of address misaligned exceptions, refer to the S1C17 Core Manual.
Table III.3.2.1 shows where each type of data is located in memory.

Table III.3.2.1 Data Locations in Memory

Data type	Location
8-bit data	8-bit boundary (all addresses)
16-bit data	16-bit boundary (A[0] = 0)
24/32-bit data	32-bit boundary (A[1:0] = 0b00)

All 16-bit and 24/32-bit data in memory are accessed in little endian mode. To increase memory efficiency, try locating the same type of data at contiguous addresses to reduce blank areas created by positioning at boundary addresses as much as possible.

III.3.3 SRAMC Operating Clock

The SRAMC is clocked by the SRAMC_CLK clock (= system clock) generated by the CMU.
The bus control signals are generated synchronously with SRAMC_CLK.
The SRAMC provides SRAMC_CLK_EN (D7/CMU_GATEDCLK1 register) for controlling the SRAMC clock (SRAMC_CLK). However, the SRAMC controls the internal peripheral bus (SAPB), so SRAMC_CLK cannot be stopped while the IC is running. In other words, SRAMC_CLK does not stop in normal operation mode (except when the halt or slp instruction is executed) even if SRAMC_CLK_EN is set to 0. However, SRAMC_CLK can be automatically turned off in HALT mode (after the halt instruction is executed) by setting SRAMC_CLK_EN to 0 (default: on).

∗ **SRAMC_CLK_EN**: SRAMC Clock Control (in HALT mode) Bit in the Gated Clock Control 1 (CMU_GATEDCLK1) Register (D7/0x4907)

When initially reset, SRAMC_CLK_EN is set to 1 (on) to enable the SRAMC_CLK supply. If SRAMC_CLK is unused in HALT mode, set SRAMC_CLK_EN to 0 (off). The SRAMC_CLK supply will be stopped after the CPU executes the halt instruction. After that, the SRAMC_CLK supply will be enabled when HALT mode is cancelled by an interrupt or an other cause. SRAMC_CLK will stop in SLEEP mode regardless of how the CMU registers are set (see Section II.2.8.2 for more information).
For details on how to set and control clocks, see Section II.2, “Clock Management Unit (CMU).”

III.3.4 Setting the wait cycles for accessing the RTC module

In order to access the RTC registers properly even if the system operates with a high-speed clock, the SRAMC can insert a wait cycle in the RTC access cycle. The number of system clock cycles to be inserted as a wait cycle can be specified using RTC_WAIT[2:0] (D[2:0]/RTC_WAIT register).

∗ **RTC_WAIT[2:0]**: RTC Access Wait Cycle Setup Bits in the RTC Wait Control (RTC_WAIT) Register (D[2:0]/0x5018)

Table III.3.4.1 Number of Wait Cycles during RTC Access

RTC_WAIT[2:0]	Number of wait cycles
0x7	7 cycles
0x6	6 cycles
0x5	5 cycles
0x4	4 cycles
0x3	3 cycles
0x2	2 cycles
0x1	1 cycle
0x0	0 cycles

(Default: 0x7)

The S1C17002 is able to operate with RTC_WAIT[2:0] ≥ 1.

III.3.5 Control Register Details

Table III.3.5.1 SRAMC Register List

Address	Register name		Function
0x5018	RTC_WAIT	RTC Wait Control Register	Sets up RTC access cycle.

The following describes the SRAMC control register. This is a 16-bit register.

Note: When setting the registers, be sure to write a 0, and not a 1, for all "reserved bits."

0x5018: RTC Wait Control Register (RTC_WAIT)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
RTC Wait Control Register (RTC_WAIT)	**0x5018** (16 bits)	D15–3	**–**	reserved	–		–	–	0 when being read.
		D2–0	**RTC_WAIT [2:0]**	RTC access wait cycle setup	RTC_WAIT[2:0]	Wait cycle	0x7	R/W	
					0x7 : 0x0	7 cycles : 0 cycles			

D[15:3] Reserved

D[2:0] RTC_WAIT[2:0]: RTC Access Wait Cycle Setup Bits

These bits set the number of wait cycles to be inserted when an RTC register is accessed.

Table III.3.5.2 Number of Wait Cycles during RTC Access

RTC_WAIT[2:0]	Number of wait cycles
0x7	7 cycles
0x6	6 cycles
0x5	5 cycles
0x4	4 cycles
0x3	3 cycles
0x2	2 cycles
0x1	1 cycle
0x0	0 cycles

(Default: 0x7)

The S1C17002 is able to operate with RTC_WAIT[2:0] ≥ 1.

THIS PAGE IS BLANK.

S1C17002 Technical Manual

IV S1C17002 TIMER MODULES

IV.1 8-bit Programmable Timers (PT8)

IV.1.1 Outline of the 8-bit Programmable Timers

The S1C17002 incorporates four channels of 8-bit programmable timers (PT8 CH.0–CH.3).

The 8-bit programmable timer includes an 8-bit presettable down counter and an 8-bit reload data register for setting 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. The CH.0 underflow signal can also be used as the A/D converter trigger signal to start A/D conversion. The underflow period can be programmed by selecting a prescaler clock and setting reload data. This allows the application program to get any desired time intervals. The prescaler is built into the PT8 module and it generates 13 count clocks from PT8_CLK•1/1 to PT8_CLK•1/4096.

Figure IV.1.1.1 shows the structure of the 8-bit programmable timers.

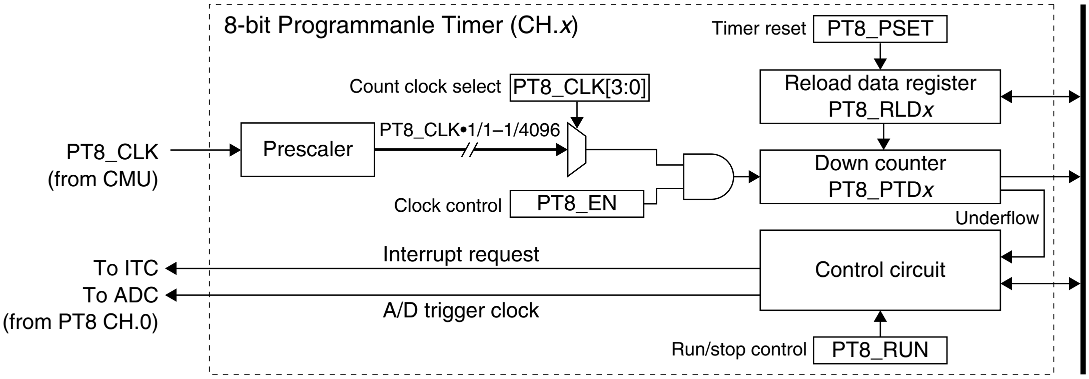


Figure IV.1.1.1 Structure of 8-bit Programmable Timer (one channel)

Notes:
- The descriptions in this section apply to all 8-bit programmable timer channels because they have the same functions except for the control register addresses. The '*x*' in the register names denotes a channel number (0 to 3) and the register addresses are described as (*CH.0*/*CH.1*/*CH.2*/*CH.3*).

  Example: PT8_CTL*x* register (0x4803/0x4807/0x480b/0x480f)
  CH.0: PT8_CTL0 register (0x4803)
  CH.1: PT8_CTL1 register (0x4807)
  CH.2: PT8_CTL2 register (0x480b)
  CH.3: PT8_CTL3 register (0x480f)

- The prescaler in the PT8 module is also be used by the SPI CH.1.

IV.1.2 Count Clock

The 8-bit programmable timer uses a prescaler output clock as the count clock. The prescaler divides PT8_CLK (with the same frequency as the system clock) by 1 to 4096 to generate 13 clocks. Select one of the prescaler output clocks using PT8_CLK[3:0] (D[3:0]/PT8_CLK*x* register).

∗ **PT8_CLK[3:0]**: PT8 Clock Division Ratio Selection Bits in the PT8 CH. *x* Input Clock Select (PT8_CLK*x*) Registers (D[3:0]/0x4800/0x4804/0x4808/0x480c)

Table IV.1.2.1 Selecting the Count Clock

PT8_CLK[3:0]	Prescaler output clock	PT8_CLK[3:0]	Prescaler output clock
0xf	Reserved	0x7	PT8_CLK•1/128
0xe	Reserved	0x6	PT8_CLK•1/64
0xd	Reserved	0x5	PT8_CLK•1/32
0xc	PT8_CLK•1/4096	0x4	PT8_CLK•1/16
0xb	PT8_CLK•1/2048	0x3	PT8_CLK•1/8
0xa	PT8_CLK•1/1024	0x2	PT8_CLK•1/4
0x9	PT8_CLK•1/512	0x1	PT8_CLK•1/2
0x8	PT8_CLK•1/256	0x0	PT8_CLK•1/1

(Default: 0x0)

The selected clock is input to the counter by setting PT8_EN (D4/PT8_CLK*x* register) to 1. When PT8_EN is set to 0, the timer channel does not operate.

∗ **PT8_EN**: PT8 Clock Enable Bit in the PT8 CH.*x* Input Clock Select (PT8_CLK*x*) Registers (D4/0x4800/0x4804/0x4808/0x480c)

Note: When setting the count clock, make sure the 8-bit programmable timer counter is stopped.

IV.1.3 Reload Register and Underflow Period

The Reload Data (PT8_RLD*x*) Register (0x4801/0x4805/0x4809/0x480d) is used to set the initial value to the down counter.

The counter initial value set in the reload data register is preset to the down counter when the 8-bit programmable timer is reset or when the counter underflows. When starting the 8-bit programmable timer after resetting, the timer counts down from the reload value. So the reload value and the input clock frequency determine the period of time from starting the timer until an underflow occurs (and between underflows). This makes it possible to obtain a desired wait time or a periodical interrupt interval.

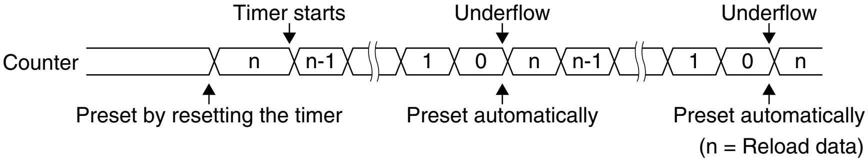


Figure IV.1.3.1 Preset Timing

The underflow period is calculated by the expression below.

$$\text{Underflow period} = \frac{\text{RLD} + 1}{\text{clk\_in}} \text{ [s]} \qquad \text{Underflow cycle} = \frac{\text{clk\_in}}{\text{RLD} + 1} \text{ [Hz]}$$

clk_in: Count clock (prescaler output clock) frequency [Hz]
RLD: Reload data (0–255)

IV.1.4 Resetting the Timer

To reset the 8-bit programmable timer, write 1 to PT8_PSET (D1/PT8_CTL*x* register). This initializes the counter by presetting the Reload Data Register value.

∗ **PT8_PSET**: Timer Reset Bit in the PT8 CH. *x* Control (PT8_CTL*x*) Registers (D1/0x4803/0x4807/0x480b/0x480f)

IV.1.5 Timer Run/Stop Control

Before starting the 8-bit programmable timer, set up the conditions as shown below.

(1) Select the count clock (prescaler output clock). See Section IV.1.2.
(2) Calculate the counter initial value and set it to the reload data register. See Section IV.1.3.
(3) Reset the timer to preset the initial value to the counter. See Section IV.1.4.
(4) Set up the interrupt level and enable the interrupt of the timer channel if the timer interrupt is used. See Section IV.1.7.

To start the 8-bit programmable timer, write 1 to PT8_RUN (D0/PT8_CTL*x* register).

∗ **PT8_RUN**: Timer Run/Stop Control Bit in the PT8 CH.*x* Control (PT8_CTL*x*) Registers (D0/0x4803/0x4807/0x480b/0x480f)

The timer starts counting down from the initial value or the current counter value if the initial value has not been preset. When the counter underflows, the timer outputs an underflow pulse and presets the initial value again. At the same time, an interrupt request is sent to the interrupt controller (ITC).
The timer continues counting from the reloaded initial value.

To stop the 8-bit programmable timer from the application program, write 0 to PT8_RUN. The counter stops counting and holds the current counter value until the timer is reset or restarted. To restart counting from the initial value, reset the timer before writing 1 to PT8_RUN.

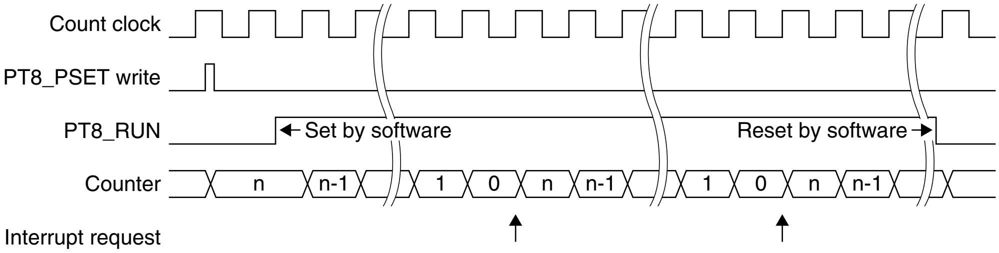


Figure IV.1.5.1 Count Operation

IV.1.6 Timer Output Signal

The 8-bit programmable timer outputs an underflow pulse when the counter underflows.
This pulse is used to request a timer interrupt. The CH.0 underflow pulse can also be used as the A/D converter trigger signal to start A/D conversion.

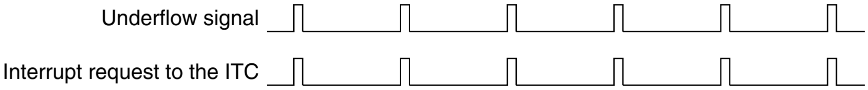


Figure IV.1.6.1 Timer Output Clock

IV.1.7 8-bit Programmable Timer Interrupt

The 8-bit programmable timer outputs an interrupt request signal to the interrupt controller (ITC) when the counter underflows.

To generate a timer underflow interrupt, set up the interrupt level and enable the interrupt using the ITC registers.

ITC registers for timer interrupts

Table IV.1.7.1 shows the control registers of the ITC provided for each timer channel.

Table IV.1.7.1 ITC Registers

Channel	Interrupt flag	Interrupt enable bit	Interrupt level setup bits
CH.0	AIFT5 (D5/ITC_AIFLG)	AIEN5 (D5/ITC_AEN)	AILV5[2:0] (D[10:8]/ITC_AILV2)
CH.1	AIFT6 (D6/ITC_AIFLG)	AIEN6 (D6/ITC_AEN)	AILV6[2:0] (D[2:0]/ITC_AILV3)
CH.2	AIFT7 (D7/ITC_AIFLG)	AIEN7 (D7/ITC_AEN)	AILV7[2:0] (D[10:8]/ITC_AILV3)
CH.3	AIFT8 (D8/ITC_AIFLG)	AIEN8 (D8/ITC_AEN)	AILV8[2:0] (D[2:0]/ITC_AILV4)

ITC_AIFLG register (0x42e0)
ITC_AEN register (0x42e2)
ITC_AILV2, ITC_AILV3, ITC_AILV4 registers (0x42ea, 0x42ec, 0x42ee)

When an underflow occurs in the timer, the corresponding interrupt flag is set to 1.

If the interrupt enable bit corresponding to that interrupt flag has been set to 1, the ITC sends an interrupt request to the S1C17 Core. To disable the timer interrupt, set the interrupt enable bit to 0.

The interrupt flag is always set to 1 by the timer underflow pulse, regardless of how the interrupt enable bit is set (even when set to 0).

The interrupt level setup bits set the interrupt level (0 to 7) of the timer interrupt. If the same interrupt level is set, timer CH.0 has highest priority and timer CH.3 has lowest priority.

An interrupt request to the S1C17 Core is accepted only when all the conditions described below are met.

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

For details on these interrupt control registers, as well as the device operation when an interrupt has occurred, see Section III.1, “Interrupt Controller (ITC).”

Interrupt vectors

The following shows the vector numbers and vector addresses for the timer interrupt:

Table IV.1.7.2 Timer Interrupt Vectors

Channel	Vector number	Vector address
CH.0	21 (0x15)	TTBR + 0x54
CH.1	22 (0x16)	TTBR + 0x58
CH.2	23 (0x17)	TTBR + 0x5c
CH.3	24 (0x18)	TTBR + 0x60

IV.1.8 Details of Control Registers

Table IV.1.8.1 List of 8-bit Programmable Timer Registers

Address	Register name		Function
0x4800	PT8_CLK0	PT8 CH.0 Input Clock Select Register	Selects the count clock.
0x4801	PT8_RLD0	PT8 CH.0 Reload Data Register	Sets reload data.
0x4802	PT8_PTD0	PT8 CH.0 Counter Data Register	Counter data
0x4803	PT8_CTL0	PT8 CH.0 Control Register	Sets the timer mode and starts/stops the timer.
0x4804	PT8_CLK1	PT8 CH.1 Input Clock Select Register	Selects the count clock.
0x4805	PT8_RLD1	PT8 CH.1 Reload Data Register	Sets reload data.
0x4806	PT8_PTD1	PT8 CH.1 Counter Data Register	Counter data
0x4807	PT8_CTL1	PT8 CH.1 Control Register	Sets the timer mode and starts/stops the timer.
0x4808	PT8_CLK2	PT8 CH.2 Input Clock Select Register	Selects the count clock.
0x4809	PT8_RLD2	PT8 CH.2 Reload Data Register	Sets reload data.
0x480a	PT8_PTD2	PT8 CH.2 Counter Data Register	Counter data
0x480b	PT8_CTL2	PT8 CH.2 Control Register	Sets the timer mode and starts/stops the timer.
0x480c	PT8_CLK3	PT8 CH.3 Input Clock Select Register	Selects the count clock.
0x480d	PT8_RLD3	PT8 CH.3 Reload Data Register	Sets reload data.
0x480e	PT8_PTD3	PT8 CH.3 Counter Data Register	Counter data
0x480f	PT8_CTL3	PT8 CH.3 Control Register	Sets the timer mode and starts/stops the timer.

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

Note: When setting the registers, be sure to write a 0, and not a 1, for all “reserved bits.”

0x4800/0x4804/0x4808/0x480c: PT8 CH.*x* Input Clock Select Registers (PT8_CLK*x*)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
PT8 CH.*x* Input Clock Select Register (PT8_CLK*x*)	**0x4800** **0x4804** **0x4808** **0x480c** (8 bits)	D7–5	**–**	reserved	–		–	–	0 when being read.
		D4	**PT8_EN**	PT8 Clock Enable	1 Enable	0 Disable	0	R/W	
		D3–0	**PT8_CLK[3:0]**	PT8 clock division ratio selection (Prescaler output clock)	PT8_CLK[3:0]	Clock	0x0	R/W	
					0xf–0xd	reserved			
					0xc	PT8_CLK•1/4096			
					0xb	PT8_CLK•1/2048			
					0xa	PT8_CLK•1/1024			
					0x9	PT8_CLK•1/512			
					0x8	PT8_CLK•1/256			
					0x7	PT8_CLK•1/128			
					0x6	PT8_CLK•1/64			
					0x5	PT8_CLK•1/32			
					0x4	PT8_CLK•1/16			
					0x3	PT8_CLK•1/8			
					0x2	PT8_CLK•1/4			
					0x1	PT8_CLK•1/2			
					0x0	PT8_CLK•1/1			

Note: The letter '*x*' in register names, etc., denotes a channel number from 0 to 3.

0x4800: PT8 CH.0 Input Clock Select Register (PT8_CLK0)
0x4804: PT8 CH.1 Input Clock Select Register (PT8_CLK1)
0x4808: PT8 CH.2 Input Clock Select Register (PT8_CLK2)
0x480c: PT8 CH.3 Input Clock Select Register (PT8_CLK3)

D[7:5] Reserved

D4 PT8_EN: PT8 Clock Enable Bit

Enables the count clock input to the counter.

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

Write 1 to this bit before the timer channel can start counting.

D[3:0] PT8_CLK[3:0]: PT8 Clock Division Ratio Selection Bits

These bits select the count clock of the 8-bit programmable timer from 13 prescaler output clocks.

Table IV.1.8.2 Selecting the Count Clock

PT8_CLK[3:0]	Prescaler output clock	PT8_CLK[3:0]	Prescaler output clock
0xf	Reserved	0x7	PT8_CLK•1/128
0xe	Reserved	0x6	PT8_CLK•1/64
0xd	Reserved	0x5	PT8_CLK•1/32
0xc	PT8_CLK•1/4096	0x4	PT8_CLK•1/16
0xb	PT8_CLK•1/2048	0x3	PT8_CLK•1/8
0xa	PT8_CLK•1/1024	0x2	PT8_CLK•1/4
0x9	PT8_CLK•1/512	0x1	PT8_CLK•1/2
0x8	PT8_CLK•1/256	0x0	PT8_CLK•1/1

(Default: 0x0)

Note: When setting the count clock, make sure the 8-bit programmable timer counter is stopped.

0x4801/0x4805/0x4809/0x480d: PT8 CH.*x* Reload Data Registers (PT8_RLD*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
PT8 CH.*x* Reload Data Register (PT8_RLD*x*)	**0x4801** **0x4805** **0x4809** **0x480d** (8 bits)	D7–0	**PT8_RLD [7:0]**	8-bit programmable timer reload data PT8_RLD7 = MSB PT8_RLD0 = LSB	0 to 255	X	R/W	

Note: The letter '*x*' in register names, etc., denotes a channel number from 0 to 3.

0x4801: PT8 CH.0 Reload Data Register (PT8_RLD0)
0x4805: PT8 CH.1 Reload Data Register (PT8_RLD1)
0x4809: PT8 CH.2 Reload Data Register (PT8_RLD2)
0x480d: PT8 CH.3 Reload Data Register (PT8_RLD3)

D[7:0] PT8_RLD[7:0]: 8-bit Programmable Timer Reload Data Bits

Set the initial value for the counter. (Default: undefined)

The reload data written in this register is preset to the respective counter when the timer is reset or when the counter underflows.

When starting the 8-bit programmable timer after resetting, the timer counts down from the reload value. So the reload value and the input clock frequency determine the period of time from starting the timer until an underflow occurs (and between underflows). This makes it possible to obtain a desired wait time or a periodical interrupt interval.

0x4802/0x4806/0x480a/0x480e: PT8 CH.*x* Counter Data Registers (PT8_PTD*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
PT8 CH.*x* Counter Data Register (PT8_PTD*x*)	**0x4802** **0x4806** **0x480a** **0x480e** (8 bits)	D7–0	**PT8_PTD [7:0]**	8-bit programmable timer counter data PT8_PTD7 = MSB PT8_PTD0 = LSB	0 to 255	X	R	

Note: The letter '*x*' in register names, etc., denotes a channel number from 0 to 3.

0x4802: PT8 CH.0 Counter Data Register (PT8_PTD0)
0x4806: PT8 CH.1 Counter Data Register (PT8_PTD1)
0x480a: PT8 CH.2 Counter Data Register (PT8_PTD2)
0x480e: PT8 CH.3 Counter Data Register (PT8_PTD3)

D[7:0] PT8_PTD[7:0]: 8-bit Programmable Timer Counter Data Bits

The counter data can be read from this register. (Default: undefined)
This is a read-only register, so the writing operation is invalid.

0x4803/0x4807/0x480b/0x480f: PT8 CH.*x* Control Registers (PT8_CTL*x*)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
PT8 CH.*x* Control Register (PT8_CTL*x*)	**0x4803 0x4807 0x480b**	D7–2	–	reserved	–				–	–	0 when being read.
	0x480f	D1	**PT8_PSET**	Timer reset	1	Reset	0	Ignored	0	W	
	(8 bits)	D0	**PT8_RUN**	Timer run/stop control	1	Run	0	Stop	0	R/W	

Note: The letter '*x*' in register names, etc., denotes a channel number from 0 to 3.

0x4803: PT8 CH.0 Control Register (PT8_CTL0)
0x4807: PT8 CH.1 Control Register (PT8_CTL1)
0x480b: PT8 CH.2 Control Register (PT8_CTL2)
0x480f: PT8 CH.3 Control Register (PT8_CTL3)

D[7:2] Reserved

D1 PT8_PSET: Timer Reset Bit

Resets the 8-bit programmable timer.

1 (W): Reset
0 (W): Has no effect
0 (R): Always 0 when read (default)

Writing 1 to this bit presets the reload data in the counter. If the counter is reset when in a run state, the counter starts counting immediately after the reload data is preset.

D0 PT8_RUN: Timer Run/Stop Control Bit

Controls the timer's Run/Stop state.

1 (R/W): Run
0 (R/W): Stop (default)

The timer starts counting by writing 1 to PT8_RUN and stops counting by writing 0.
In the stop state, the counter data is retained until the timer is reset or placed in a run state.

IV.1.9 Precaution

When setting the count clock, make sure the 8-bit programmable timer is turned off.

THIS PAGE IS BLANK.

IV.2 16-bit Multi-Function Timer (MFT)

IV.2.1 Configuration of 16-bit Multi-Function Timer

The S1C17002 contains a 16-bit multi-function timer (MFT). The following lists the main functions of the MFT.

- 16-bit presettable up-counter
- Programmable count clocks using the prescaler embedded in the MFT module
- Event counter function using an external clock
- Interrupt generation function with programmable interrupt cycles using the period and compare data registers
- PWM output with an IGBT control function

Figure IV.2.1.1 shows the structure of one channel of the MFT.

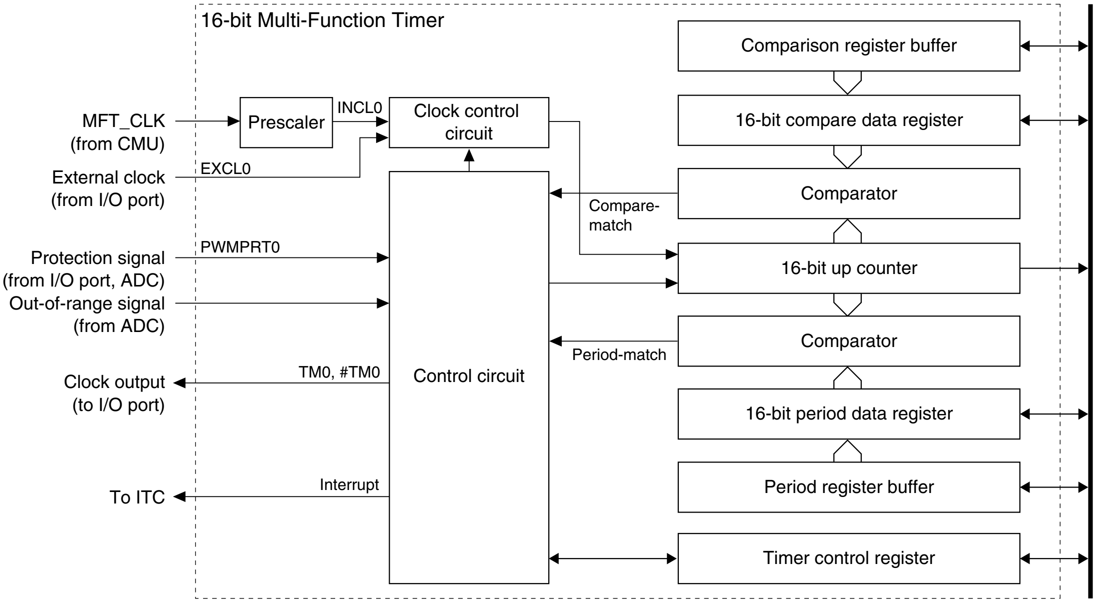


Figure IV.2.1.1 Structure of 16-bit Multi-Function Timer

The MFT consists of a 16-bit up-counter (MFT_TC register), 16-bit period data and comparison data registers (MFT_PRD register, MFT_CMP register) and their buffers.

∗ **TC[15:0]**: Counter Data Bits in the MFT Counter Data (MFT_TC) Register (D[15:0]/0x5200)
∗ **PRD[15:0]**: Period Data Bits in the MFT Period Data (MFT_PRD) Register (D[15:0]/0x5202)
∗ **CR[15:0]**: Compare Data Bits in the MFT Compare Data (MFT_CMP) Register (D[15:0]/0x5204)

The 16-bit counter can be reset to 0 by software and counts up using the prescaler output clock or an external signal input from the I/O port. The counter value can be read by software.

The period and comparison data registers are used to store the data to be compared with the content of the up-counter. These registers can be directly read and written. Furthermore, period and comparison data can be set via the buffers. In this case, the set value is loaded to the period and comparison data register when the counter is reset by the period-match signal or software (by writing 1 to PRST (D1/MFT_CTL register)). The software can select whether period/comparison data is written to the register or the buffer.

∗ **PRST**: Timer Reset Bit in the MFT Control (MFT_CTL) Register (D1/0x5206)

When the counter value matches to the content of the period/comparison data register, the comparator outputs a signal that controls the interrupt and the output signal. Thus the registers allow interrupt generating intervals and the timer's output clock frequency and duty ratio to be programmed.

IV.2.2 I/O Pins of 16-bit Multi-Function Timer

Table IV.2.2.1 shows the input/output pins used for the MFT.

Table IV.2.2.1 MFT I/O Pins

Pin name	I/O	Function
EXCL0	I	External clock input for the event counter function
TM0	O	MFT output (positive)
#TM0	O	MFT output (negative)
PWMPRT0	I	PWM channel protection input

TM0, #TM0 (output pins of the MFT)

These pins output the clock or PWM signal generated by the MFT. The TM0 pin outputs the positive signal and the #TM0 pin outputs the negative signal.

EXCL0 (event counter input pin)

When using the MFT as an event counter, input count pulses from an external source to this pin.

PWMPRT0 (PWM channel protection input pin)

This pin is used to input an output protect signal for disabling the MFT PWM output forcibly. This function can be used to control the external IGBT for protecting the chip from over-voltage, over-current, and excessive temperature. The out-of-range signal from the A/D converter is also used as the protect signal.

Note: The MFT input/output pins are shared with general-purpose I/O ports or other peripheral circuit inputs/outputs, so that functionality in the initial state is set to other than the MFT input/output. Before the MFT input/output signals assigned to these pins can be used, the function of these pins must be switched for the MFT input/output by setting the corresponding Port Function Select Registers. For details of pin functions and how to switch over, see Section I.3.3, "Switching Over the Multiplexed Pin Functions."

IV.2.3 Uses of 16-bit Multi-Function Timer

The comparators of the MFT cyclically output a compare-match signal and a period-match signal in accordance with the comparison data and period data that are set in the software. These signals are used to generate an interrupt request to the CPU or control the internal peripheral circuits. A clock generated from the signals can also be output to external devices.

CPU interrupt request

A matching of the counter data and comparison or period data can be used as a cause of interrupt to generate an interrupt request to the CPU. Therefore, an interrupt can be generated at an interval that is set in the software.

Clock output to external devices

The positive (TM0) and negative (#TM0) clocks (PWM signals) generated from the compare-match and period-match signals can be output from the chip to the outside. The clock cycle is determined by period data, and the duty ratio is determined by comparison data. These outputs can be used to control external devices. The output pins are described in the preceding section.

The outputs are also be controlled by the protection signal input from the I/O port or A/D converter. If the protection signal is asserted, the outputs will be disabled (forcibly set to the initial levels). This function can be used to control the external IGBT for power protection.

A/D converter start trigger

The A/D converter allows a trigger to start the A/D conversion to be selected from among four available types. One is the period-match of the MFT. This makes it possible to perform the A/D conversion at programmable intervals.

To use this function, write 0b01 to the A/D converter control bit TS[1:0] (D[4:3]/AD_TRIG_CH register) to select the MFT as the trigger.

∗ **TS[1:0]**: A/D Conversion Trigger Selection Bits in the A/D Trigger/Channel Select (AD_TRIG_CH) Register (D[4:3]/0x5542)

IV.2.4 MFT Operating Clock

The MFT use the MFT_CLK clock (= system clock) generated by the CMU as the operating clock. The count clock is generated from the MFT_CLK by the prescaler embedded in the MFT module.

Controlling the supply of the operating clock

The MFT_CLK clock is supplied to the MFT with default settings. It can be turned off using MFT_CLK_EN (D0/CMU_GATEDCLK1 register) to reduce the amount of power consumed on the chip if MFT is not used.

* **MFT_CLK_EN**: Multi-Function Timer Clock Control Bit in the Gated Clock Control 1 (CMU_GATEDCLK1) Register (D0/0x4907)

Setting MFT_CLK_EN to 0 (1 by default) turns off the MFT_CLK clock supply to the MFT. When the clock supply is turned off, the MFT control registers cannot be accessed and the count operation stops.
For details on how to set and control the operating clock, refer to Section II.2, "Clock Management Unit (CMU)."

Clock state in standby mode

The clock supply to the MFT stops depending on type of standby mode.
HALT mode: The operating clock is supplied the same way as in normal mode.
SLEEP mode: The operating clock supply stops.

Therefore, the MFT also stops operating in SLEEP mode.

IV.2.5 Control and Operation of the MFT

The following settings must first be made before the 16-bit multi-function timer starts counting:

1. Setting pins for input/output (only when necessary) ... See Sections IV.2.2 and I.3.3.
2. Setting count clock
3. Selecting comparison/period data register/buffer
4. Setting clock output conditions (signal active level, initial signal level, protect input) ... See Section IV.2.6.
5. Setting comparison/period data
6. Setting interrupt ... See Section IV.2.7.

Setting the count clock

The count clock can be selected from between an internal clock and an external clock. Select the input clock using TCKS (D5/MFT_CTL register).

∗ **TCKS**: Input Clock Selection Bit in the MFT Control (MFT_CTL) Register (D5/0x5206)

An external clock is selected by writing 1 to TCKS, and the internal clock is selected by writing 0. At initial reset, TCKS is set for the internal clock.
An external clock can be used for the timer for which the pin is set for input.

- **Internal clock**

When the internal clock is selected, the count clock is generated from the MFT_CLK by the prescaler embedded in the MFT module.
The prescaler's division ratio can be selected from among eight ratios using TPS[2:0] (D[10:8]/MFT_CTL register). The divided clock is output from the prescaler by writing 1 to TPSON (D11/MFT_CTL register).

∗ **TPS[2:0]**: Clock Division Ratio Selection Bits in the MFT Control (MFT_CTL) Register (D[10:8]/0x5206)
∗ **TPSON**: Clock Control Bit in the MFT Control (MFT_CTL) Register (D11/0x5206)

Table IV.2.5.1 Division Ratio

TPS[2:0]	Count clock
0x7	MFT_CLK•1/128
0x6	MFT_CLK•1/64
0x5	MFT_CLK•1/32
0x4	MFT_CLK•1/16
0x3	MFT_CLK•1/8
0x2	MFT_CLK•1/4
0x1	MFT_CLK•1/2
0x0	MFT_CLK•1/1

(Default: 0x0)

Notes: • When setting a count clock, make sure the MFT is turned off.

• TPSON (D11/MFT_CTL register) should be set to 0 to reduce current consumption when the MFT is not used.

- **External clock**

When using the MFT as an event counter by supplying clock pulses from an external source, make sure the event cycle is at least two MFT_CLK cycles.

Selecting comparison/period data register/buffer

The comparison data and period data registers are used to store the data to be compared with the content of the up-counter. These registers can be directly read and written. Furthermore, comparison/period data can be set via the comparison/period register buffer. In this case, the set value is loaded to the comparison/period data register when the counter is reset by the period-match signal or software (by writing 1 to PRST (D1/MFT_CTL register)).

Select whether comparison/period data is written to the comparison/period data register or the buffer using BUFEN (D7/MFT_CTL register).

∗ **BUFEN**: Comparison/Period Buffer Enable Bit in the MFT Control (MFT_CTL) Register (D7/5206)

When 1 is written to BUFEN, the comparison/period register buffer is selected and when 0 is written, the comparison/period data register is selected.
At initial reset, the comparison/period data register is selected.

Setting comparison and period data

The timer contains two data comparators that allow the count data to be compared with given values. CR[15:0] (D[15:0]/MFT_CMP register) and PRD[15:0] (D[15:0]/MFT_PRD register) are used to set these values.

∗ **CR[15:0]**: Compare Data Bits in the MFT Compare Data (MFT_CMP) Register (D[15:0]/0x5204)
∗ **PRD[15:0]**: Period Data Bits in the MFT Period Data (MFT_PRD) Register (D[15:0]/0x5202)

When BUFEN is set to 0, these registers allow direct reading/writing from/to the comparison and period data registers.
When BUFEN is set to 1, these registers are used to read/write from/to the comparison and period register buffers. The content of the buffer is loaded to the comparison and period data registers when the counter is reset.
At initial reset, the comparison and period data registers/buffers are initialized to 0.

The MFT compares the comparison data register and count data and, when the two values are equal, generates a compare-match signal. Also the period data register is compared with count data and, when the two values are equal, the MFT generates a period-match signal. The compare-match and period-match signals control the clock outputs (TM0 and #TM0 signals) to external devices, in addition to generating an interrupt.
The period-match signal is also used to reset the counter.

Resetting the counter

The MFT provides PRST (D1/MFT_CTL register) to reset the counter.

∗ **PRST**: Timer Reset Bit in the MFT Control (MFT_CTL) Register (D1/0x5206)

Normally, reset the counter before starting count-up by writing 1 to this control bit.
After the counter starts counting, it will be reset when a period-match occurs.

Timer RUN/STOP control

The MFT provides TMEN (D0/MFT_CTL register) to control RUN/STOP.

∗ **TMEN**: Timer Run/Stop Control Bit in the MFT Control (MFT_CTL) Register (D0/0x5206)

The timer starts counting when 1 is written to TMEN. The clock input is disabled and the timer stops counting when 0 is written to TMEN.
This RUN/STOP control does not affect the counter data. Even when the timer has stopped counting, the counter retains its count so that the timer can start counting again from that point.
If the count of the counter matches the set value of the comparison data register during count-up, the timer generates a compare-match interrupt.
When the counter matches period data, an interrupt is generated and the counter is reset. At the same time, the values set in the compare and period register buffers are loaded to the compare and period data registers if BUFEN is set to 1.
The counter continues counting up regardless of which interrupt has occurred. In the case of a period-match interrupt, the counter starts counting beginning with 0.
When both PRST and TMEN are set to 1 at the same time, the timer starts counting after resetting the counter.

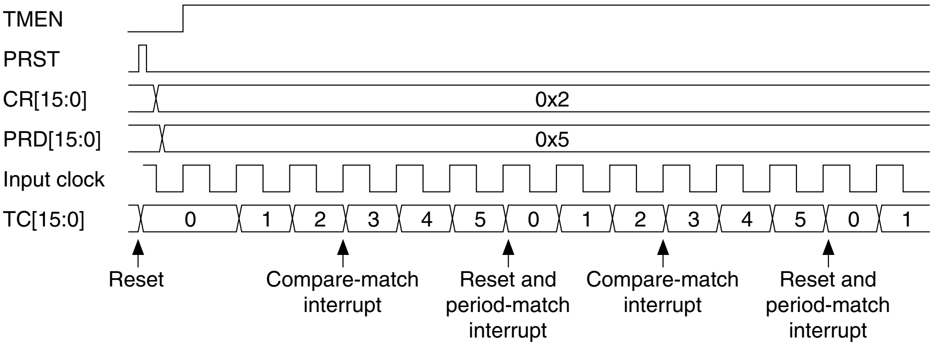


Figure IV.2.5.1 Basic Operation Timing of Counter

Reading counter data

The counter data can be read out from TC[15:0] (D[15:0]/MFT_TC register) at any time.

∗ **TC[15:0]**: Counter Data Bits in the MFT Counter Data (MFT_TC) Register (D[15:0]/0x5200)

Writing counter data

Counter data can be written to TC[15:0] at any time. This makes it possible to change the interrupt and/or clock output cycles temporarily.

IV.2.6 Controlling Clock Output

The timers can generate the TM0 and #TM0 signals using the compare-match and period-match signals from the counter.

Figure IV.2.6.1 shows the MFT clock output circuit.

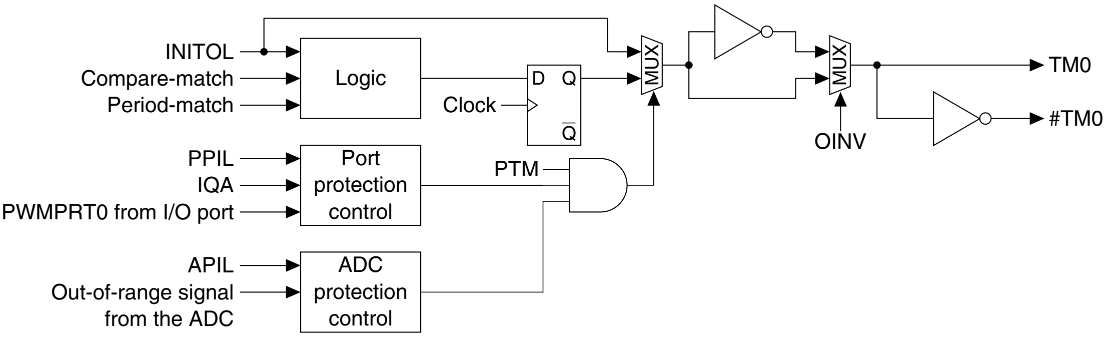


Figure IV.2.6.1 MFT Clock Output Circuit

Setting the initial output level

The default output level while the clock output is turned off is 0 (low level). This level can be changed to 1 (high level) using INITOL (D1/MFT_IOCTL register).

∗ **INITOL**: Initial Output Level Bit in the MFT Input/Output Control (MFT_IOCTL) Register (D1/0x521e)

When INITOL is 0 (default), the initial output level is low. When INITOL is set to 1, the initial output level is set to high.

The timer output goes to the initial output level when the timer output is turned off.

Setting the signal active level

By default, an active high signal (normal low) is generated. This logic can be inverted using OINV (D0/MFT_IOCTL register). When 1 is written to OINV, the timer generates an active low (normal high) signal.

∗ **OINV**: Inverse Output Bit in the MFT Input/Output Control (MFT_IOCTL) Register (D0/0x521e)

Note that the initial output level set by INITOL is inverted when OINV is set to 1.

See Figure IV.2.6.2 for the waveforms.

Setting the output port

The TM0 (#TM0) signal generated here can be output from the clock output pins (see Table IV.2.2.1), enabling a programmable clock to be supplied to external devices.

After an initial reset, the output pins are set for the I/O ports and set in input mode. The pins go into high-impedance status.

When the pin function is switched to the timer output, the pin outputs the level according to the set values of INITOL and OINIV. The output pin holds this level until the output level changes due to the counter value after the timer output is enabled.

Table IV.2.6.1 Initial Output Level

INITOL	OINV	Initial output level
1	1	Low
1	0	High
0	1	High
0	0	Low

Starting clock output

To output the TM0 (#TM0) clock, write 1 to the clock output control bit PTM (D2/MFT_IOCTL register). Clock output is stopped by writing 0 to PTM and goes to the initial output level according to the set values of INITOL and OINIV.

* **PTM**: Clock Output Enable Bit in the MFT Input/Output Control (MFT_IOCTL) Register (D2/0x521e)

Figure IV.2.6.2 shows the waveform of the output signal.

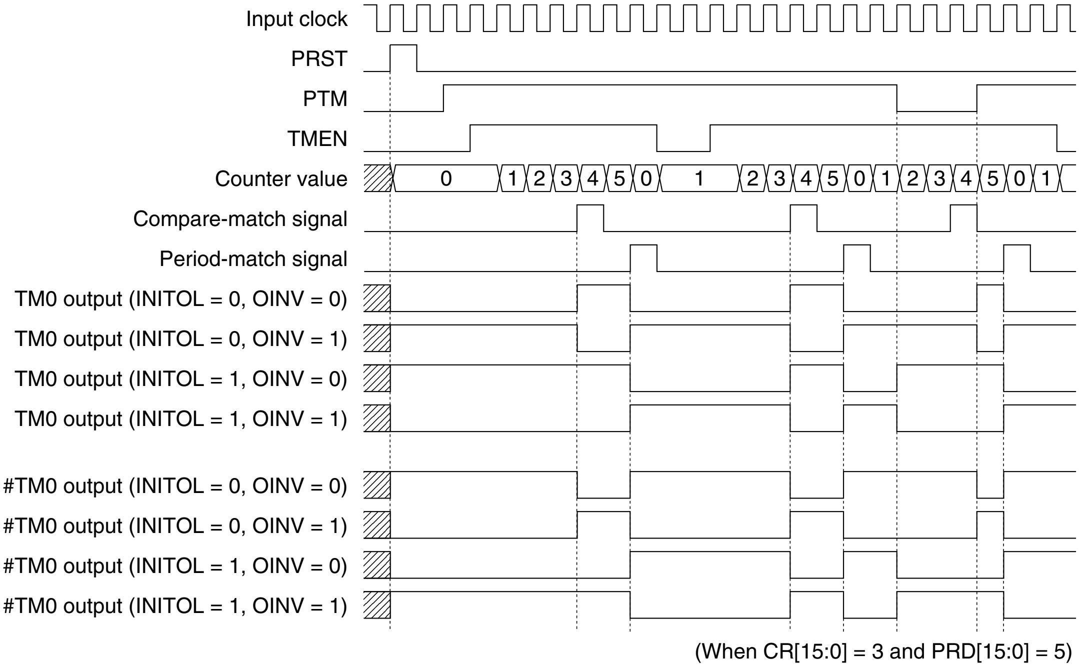

Input clock
PRST
PTM
TMEN
Counter value
0
1
2
3
4
5
0
1
2
3
4
5
0
1
2
3
4
5
0
1
Compare-match signal
Period-match signal
TM0 output (INITOL = 0, OINV = 0)
TM0 output (INITOL = 0, OINV = 1)
TM0 output (INITOL = 1, OINV = 0)
TM0 output (INITOL = 1, OINV = 1)
#TM0 output (INITOL = 0, OINV = 0)
#TM0 output (INITOL = 0, OINV = 1)
#TM0 output (INITOL = 1, OINV = 0)
#TM0 output (INITOL = 1, OINV = 1)
(When CR[15:0] = 3 and PRD[15:0] = 5)

Figure IV.2.6.2 Waveform of MFT Output

TM0 output when OINV = 0 (active high):

The timer outputs a low level (initial output level when output is started) until the counter becomes equal to the comparison data set in CR[15:0] (D[15:0]/MFT_CMP register). When the counter is incremented to the next value from the comparison data, the TM0 output pin goes high and a compare-match interrupt occurs. When the counter becomes equal to the period data set in PRD[15:0] (D[15:0]/MFT_PRD register), the counter is reset and the TM0 output pin goes low. At the same time a period-match interrupt occurs.

* **CR[15:0]**: Compare Data Bits in the MFT Compare Data (MFT_CMP) Register (D[15:0]/0x5204)
* **PRD[15:0]**: Period Data Bits in the MFT Period Data (MFT_PRD) Register (D[15:0]/0x5202)

TM0 output when OINV = 1 (active low):

The timer outputs a high level (inverted initial output level when output is started) until the counter becomes equal to the comparison data set in CR[15:0] (D[15:0]/MFT_CMP register). When the counter is incremented to the next value from the comparison data, the output pin goes low and a compare-match interrupt occurs. When the counter becomes equal to the period data set in PRD[15:0] (D[15:0]/MFT_PRD register), the counter is reset and the output pin goes high. At the same time a period-match interrupt occurs.

Output protection for IGBT control

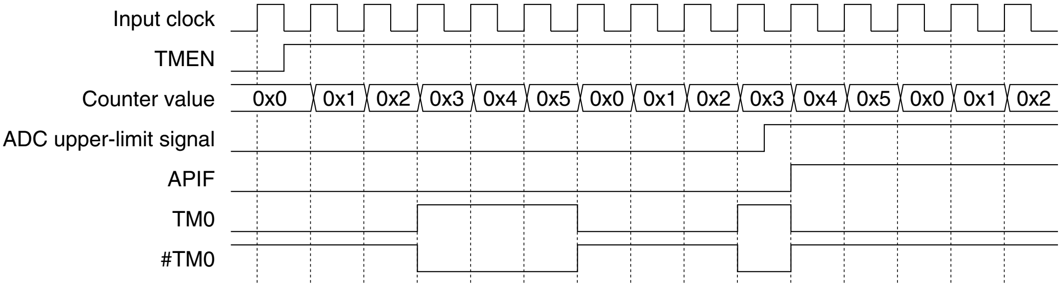
The MFT provides a power protection feature with the interrupt for safe operation of the external IGBT output. If the power protection is enabled and activated, the PWM output pins will be put in its initial status immediately. At the same time, an interrupt will also be generated. The interrupt can be used to inform the monitoring program of IGBT drive abnormalities such as over-voltage, over-current, and excessive temperature. The ADC out-of-range signal or a port input can be used to control the power protection.

ADC protection

Setting APIE (D1/MFT_IE register) to 1 enables the ADC protection that uses ADC CH.0 as the protection detection input. The ADC allows the application to check whether the conversion results of the specified channel are within the specified range or not. If the conversion result is higher than the upper-limit value set with software or is lower than the lower-limit value set with software, the ADC outputs the out-of-range signal and generates an interrupt. The MFT uses the out-of-range signal generated by the ADC CH.0 to control the power protection. Use APIL (D6/MFT_IOCTL register) to select upper-limit protection or lower-limit protection mode. Set APIL to 0 (default) to disable the PWM output (setting the output to the initial level) when the A/D conversion result is higher than the upper-limit value, or set APIL to 1 to disable the PWM output when the A/D conversion result is lower than the lower-limit value. The upper- and lower-limit values can be set to the AD_UPPER (0x5558) and AD_LOWER (0x555a) registers in the ADC module. When the designated out-of-range signal is asserted by the ADC, APIF (D1/MFT_IF register) is set to 1. At the same time, the MFT forcibly sets the PWM output pins to the initial status (PWM outputs are disabled) if APIE has been set to 1.

* **APIE**: ADC Protection Interrupt Enable Bit in the MFT Interrupt Enable (MFT_IE) Register (D1/0x5230)
* **APIF**: ADC Protection Interrupt Flag Bit in the MFT Interrupt Flag (MFT_IF) Register (D1/0x5238)
* **APIL**: ADC Protection Input Selection Bit in the MFT Input/Output Control (MFT_IOCTL) Register (D6/0x521e)

Figure IV.2.6.3 shows the ADC protection timing (upper limit protection).



(When CR[15:0] = 2, PRD[15:0] = 5, and INITOL = OINV = 0)

Figure IV.2.6.3 ADC Protection for IGBT Output

Port protection

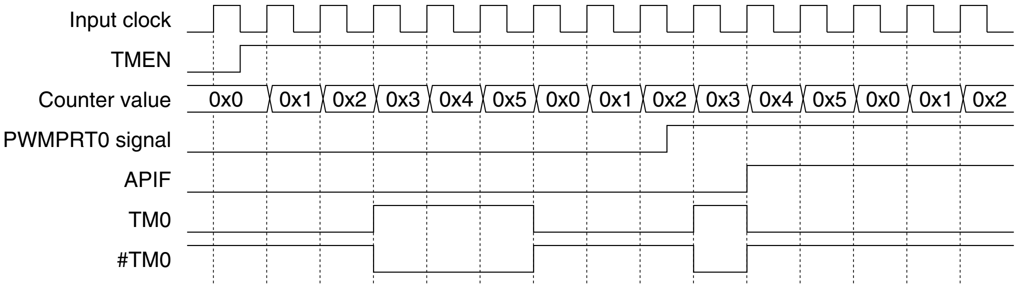
The MFT also supports a port protection that uses the dedicated input port PWMPRT0. Setting PPIE (D0/ MFT_IE register) to 1 enables the port protection. Either high or low level can be selected as the protection input signal level using PPIL (D5/MFT_IOCTL register). When PPIL is set to 1, the PWM output is disabled and PPIF (D0/MFT_IF register) is set to 1 to generate a port protection interrupt when the PWMPRT0 input signal goes low. When PPIL is set to 0 (default), the PWM output is disabled and PPIF is set to 1 when the PWMPRT0 input signal goes high.

* **PPIE**: Port Protection Interrupt Enable Bit in the MFT Interrupt Enable (MFT_IE) Register (D0/0x5230)
* **PPIF**: Port Protection Interrupt Flag Bit in the MFT Interrupt Flag (MFT_IF) Register (D0/0x5238)
* **PPIL**: Port Protection Input Level Selection Bit in the MFT Input/Output Control (MFT_IOCTL) Register (D5/0x521e)

To avoid the port protection to be activated due to noise, the MFT provides a noise filter at the input of the PWMPRT0 pin. The pulse width to be removed as noise can be selected using IQA (D4/MFT_IOCTL register). When IQA is set to 1, a pulse shorter than 12 system clocks are regarded as noise and it is not accepted as the valid input. When IQA is set to 0 (default), a pulse shorter than 6 system clocks are regarded as noise.

* **IQA**: Port Protection Input Noise Filter Bit in the MFT Input/Output Control (MFT_IOCTL) Register (D4/0x521e)

Figure IV.2.6.4 shows the port protection timing (high level protection).



(When CR[15:0] = 2, PRD[15:0] = 5, and INITOL = OINV = 0)

Figure IV.2.6.4 Port Protection for IGBT Output

Precautions

(1) If a same value is set to the comparison data and period data registers, a hazard may be generated in the output signal. Therefore, do not set the data registers as MFT_CMP = MFT_PRD.
There is no problem when the interrupt function only is used.

(2) When using the output clock, set the comparison and period data registers as MFT_CMP ≥ 0 and MFT_PRD ≥ 1. The minimum settings are MFT_CMP = 0 and MFT_PRD = 1. In this case, the timer output clock cycle is the input clock × 1/2.

(3) When the comparison and period data registers are set as MFT_CMP > MFT_PRD, no compare-match signal is generated. In this case, the output signal is fixed at the off level.

IV.2.7 MFT Interrupts

The MFT module can generate the following four types of interrupts:
- Compare-match interrupt
- Period-match interrupt
- ADC protection interrupt
- Port protection interrupt

The MFT module has one interrupt signal to be output to the interrupt controller (ITC) and it is shared with the four causes of interrupt. To determine the cause of interrupt that has occurred, read the interrupt flags in the MFT module.

Compare-match interrupt

This interrupt request occurs when the count of the counter matches the set value of the compare data register during count-up, and it sets the interrupt flag CMPIF (D3/MFT_IF register) in the MFT module to 1.

* **CMPIF**: Compare-Match Interrupt Flag Bit in the MFT Interrupt Flag (MFT_IF) Register (D3/0x5238)

Set CMPIE (D3/MFT_IE register) to 1 when using this interrupt. If CMPIE is set to 0 (default), CMPIF will not be set to 1 and an interrupt request by this cause will not be sent to the ITC.

* **CMPIE**: Compare-Match Interrupt Enable Bit in the MFT Interrupt Enable (MFT_IE) Register (D3/0x5230)

If CMPIF is set to 1, the MFT module outputs the interrupt request signal to the ITC. The interrupt request signal sets the MFT interrupt flag in the ITC to 1 and an interrupt occurs if other interrupt conditions meet the ITC and S1C17 Core settings.
The MFT interrupt handler routine should read the CMPIF flag to check if the interrupt has occurred due to a compare-match or another cause.
Furthermore, the interrupt handler routine must reset (write 1 to) CMPIF in the MFT module, not the MFT interrupt flag in the ITC, to clear the cause of interrupt.

Period-match interrupt

This interrupt request occurs when the count of the counter matches the set value of the period data register during count-up, and it sets the interrupt flag PRDIF (D2/MFT_IF register) in the MFT module to 1.

* **PRDIF**: Period-Match Interrupt Flag Bit in the MFT Interrupt Flag (MFT_IF) Register (D2/0x5238)

Set PRDIE (D2/MFT_IE register) to 1 when using this interrupt. If PRDIE is set to 0 (default), PRDIF will not be set to 1 and an interrupt request by this cause will not be sent to the ITC.

* **PRDIE**: Period-Match Interrupt Enable Bit in the MFT Interrupt Enable (MFT_IE) Register (D2/0x5230)

If PRDIF is set to 1, the MFT module outputs the interrupt request signal to the ITC. The interrupt request signal sets the MFT interrupt flag in the ITC to 1 and an interrupt occurs if other interrupt conditions meet the ITC and S1C17 Core settings.
The MFT interrupt handler routine should read the PRDIF flag to check if the interrupt has occurred due to a period-match or another cause.
Furthermore, the interrupt handler routine must reset (write 1 to) PRDIF in the MFT module, not the MFT interrupt flag in the ITC, to clear the cause of interrupt.

ADC protect interrupt

This interrupt request occurs when the A/D converter CH.0 asserts the out-of-range signal (upper-limit or lower-limit mode selectable) input to the MFT after an A/D conversion. This signal indicates that the A/D conversion result falls outside the range specified with software. It sets the interrupt flag APIF (D1/MFT_IF register) in the MFT module to 1.

* **APIF**: ADC Protection Interrupt Flag Bit in the MFT Interrupt Flag (MFT_IF) Register (D1/0x5238)

Set APIE (D1/MFT_IE register) to 1 when using this interrupt and disabling the PWM output. Although APIF will be set to 1 by the input signal even if APIE is set to 0 (default), an interrupt request by this cause will not be sent to the ITC and the PWM output will not be changed to the initial value.

* **APIE**: ADC Protection Interrupt Enable Bit in the MFT Interrupt Enable (MFT_IE) Register (D1/0x5230)

If APIF is set to 1 when APIE has been set to 1, the MFT module outputs the interrupt request signal to the ITC. The interrupt request signal sets the MFT interrupt flag in the ITC to 1 and an interrupt occurs if other interrupt conditions meet the ITC and S1C17 Core settings.
The MFT interrupt handler routine should read the APIF flag to check if the interrupt has occurred by the ADC protect input or another cause.
Furthermore, the interrupt handler routine must reset (write 1 to) APIF in the MFT module, not the MFT interrupt flag in the ITC, to clear the cause of interrupt.

Port protect interrupt

This interrupt request occurs when the specified level (high or low) of a signal is input to the PWMPRT0 pin. This signal sets the interrupt flag PPIF (D0/MFT_IF register) in the MFT module to 1.

* **PPIF**: Port Protection Interrupt Flag Bit in the MFT Interrupt Flag (MFT_IF) Register (D0/0x5238)

Set PPIE (D0/MFT_IE register) to 1 when using this interrupt and disabling the PWM output. Although PPIF will be set to 1 by the input signal even if PPIE is set to 0 (default), an interrupt request by this cause will not be sent to the ITC and the PWM output will not be changed to the initial value.

* **PPIE**: Port Protection Interrupt Enable Bit in the MFT Interrupt Enable (MFT_IE) Register (D0/0x5230)

If PPIF is set to 1 when PPIE has been set to 1, the MFT module outputs the interrupt request signal to the ITC. The interrupt request signal sets the MFT interrupt flag in the ITC to 1 and an interrupt occurs if other interrupt conditions meet the ITC and S1C17 Core settings.
The MFT interrupt handler routine should read the PPIF flag to check if the interrupt has occurred by the port protect input or another cause.
Furthermore, the interrupt handler routine must reset (write 1 to) PPIF in the MFT module, not the MFT interrupt flag in the ITC, to clear the cause of interrupt.

Note: To avoid occurrence of unnecessary interrupts, be sure to reset the CMPIF, PRDIF, APIF, or PPIF flag before the MFT interrupt is enabled using CMPIE, PRDIE, APIE, or PPIE.

ITC registers for MFT interrupt

When an enabled cause of MFT interrupt occurs according to the interrupt condition settings shown above, the MFT module asserts the interrupt signal sent to the ITC.
To generate an MFT interrupt, set the interrupt level and enable the interrupt using the ITC registers.
The following shows the control bits for the MFT interrupt in the ITC.

Interrupt flag in the ITC

∗ **AIFT0**: MFT Interrupt Flag Bit in the Additional Interrupt Flag (ITC_AIFLG) Register (D0/0x42e0)

Interrupt enable bit in the ITC

∗ **AIEN0**: MFT Interrupt Enable Bit in the Additional Interrupt Enable (ITC_AEN) Register (D0/0x42e2)

Interrupt level setup bits in the ITC

∗ **AILV0[2:0]**: MFT Interrupt Level Bits in the Additional Interrupt Level Setup (ITC_AILV0) Register 0 (D[2:0]/0x42e6)

When an enabled cause of MFT interrupt occurs, AIFT0 is set to 1. If AIEN0 has been set to 1, the ITC sends an interrupt request to the S1C17 Core. To disable the MFT interrupt, set AIEN0 to 0. AIFT0 is always set to 1 by the interrupt signal sent from the MFT module, regardless of how AIEN0 is set (even when set to 0).
AILV0[2:0] sets the interrupt level (0 to 7) of the MFT interrupt.

An interrupt request to the S1C17 Core is accepted only when all the conditions described below are met.

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

For details on these interrupt control registers, as well as the device operation when an interrupt has occurred, refer to Section III.1 “Interrupt Controller (ITC).”

Note: After an MFT interrupt occurs, reset the CMPIF, PRDIF, APIF, or PPIF interrupt flag of the MFT module in the interrupt handler routine (this also resets the interrupt flag in the ITC).

Interrupt vector

The following shows the vector number and vector address for the MFT interrupt:

Vector number: 8 (0x8)
Vector address: TTBR + 0x20

IV.2.8 Details of Control Registers

Table IV.2.8.1 List of MFT Registers

Address	Register name		Function
0x5200	MFT_TC	MFT Counter Data Register	Counter data
0x5202	MFT_PRD	MFT Period Data Register	Sets period data.
0x5204	MFT_CMP	MFT Compare Data Register	Sets compare data.
0x5206	MFT_CTL	MFT Control Register	Sets the timer mode and starts/stops the timer.
0x521e	MFT_IOCTL	MFT Input/Output Control Register	Controls the clock input/output.
0x5230	MFT_IE	MFT Interrupt Enable Register	Enables the MFT interrupt.
0x5238	MFT_IF	MFT Interrupt Flag Register	Indicates the MFT interrupt status.
0x527e	MFT_TST	MFT Test Register	Controls the MFT test.

The following describes each MFT register. These are all 16-bit registers.

Note: When setting the registers, be sure to write a 0, and not a 1, for all “reserved bits.”

0x5200: MFT Counter Data Register (MFT_TC)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
MFT Counter Data Register (MFT_TC)	**0x5200** (16 bits)	D15–0	**TC[15:0]**	Counter data TC15 = MSB TC0 = LSB	0x0 to 0xffff	0x0	R/W	

D[15:0] TC[15:0]: Counter Data Bits

The counter data can be read from this register. (Default: 0x0)

Furthermore, data can be set to the counter by writing it to this register.

0x5202: MFT Period Data Register (MFT_PRD)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
MFT Period Data Register (MFT_PRD)	**0x5202** (16 bits)	D15–0	**PRD[15:0]**	Period data PRD15 = MSB PRD0 = LSB	0x0 to 0xffff	0x0	R/W	

D[15:0] PRD[15:0]: Period Data Bits

Sets the period data for the MFT. (Default: 0x0)

When BUFEN (D7/MFT_CTL register) is set to 0, period data is directly read or written from/to the period data register.

When BUFEN is set to 1, period data is read or written from/to the period data buffer through this address. The content of the buffer is loaded to the period data register when the counter is reset.

The data set in this register is compared with the counter data. When the contents match, a cause of period-match interrupt is generated and the output signal rises (OINV (D0/MFT_IOCTL register) = 0) or falls (OINV = 1). Furthermore, the counter is reset to 0.

0x5204: MFT Compare Data Register (MFT_CMP)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
MFT Compare Data Register (MFT_CMP)	**0x5204** (16 bits)	D15–0	**CR[15:0]**	Compare data CR15 = MSB CR0 = LSB	0x0 to 0xffff	0x0	R/W	

D[15:0] CR[15:0]: Compare Data Bits

Sets the compare data for the MFT. (Default: 0x0)

When BUFEN (D7/MFT_CTL register) is set to 0, compare data is directly read or written from/to the compare data register.

When BUFEN is set to 1, compare data is read or written from/to the compare data buffer through this address. The content of the buffer is loaded to the compare data register when the counter is reset.

The data set in this register is compared with the counter data. When the contents match, a cause of compare-match interrupt is generated and the output signal rises (OINV (D0/MFT_IOCTL register) = 0) or falls (OINV = 1). This does not affect the counter value and count-up operation.

0x5206: MFT Control Register (MFT_CTL)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
MFT Control Register (MFT_CTL)	**0x5206** (16 bits)	D15–12	–	reserved	–				–	–	0 when being read.
		D11	**TPSON**	Clock control	1	On	0	Off	0	R/W	
		D10–8	**TPS[2:0]**	Clock division ratio selection (Prescaler output clock)	TPS[2:0]		Count clock		0x0	R/W	
					0x7		MFT_CLK•1/128				
					0x6		MFT_CLK•1/64				
					0x5		MFT_CLK•1/32				
					0x4		MFT_CLK•1/16				
					0x3		MFT_CLK•1/8				
					0x2		MFT_CLK•1/4				
					0x1		MFT_CLK•1/2				
					0x0		MFT_CLK•1/1				
		D7	**BUFEN**	Comparison/period buffer enable	1	Enable	0	Disable	0	R/W	
		D6	–	reserved	–				–	–	0 when being read.
		D5	**TCKS**	Input clock selection	1	External	0	Internal	0	R/W	
		D4–2	–	reserved	–				–	–	0 when being read.
		D1	**PRST**	Timer reset	1	Reset	0	Ignored	0	W	
		D0	**TMEN**	Timer run/stop control	1	Run	0	Stop	0	R/W	

D[15:12] Reserved

D11 TPSON: MFT Clock Control Bit

Enables the MFT prescaler to output the count clock.

1 (R/W): On

0 (R/W): Off (default)

Write 1 to this bit before the MFT can start counting.

D[10:8] TPS[2:0]: MFT Clock Division Ratio Selection Bits

These bits select the count clock of the MFT from 8 prescaler output clocks.

Table IV.2.8.2 Selecting the Count Clock

TPS[2:0]	Count clock
0x7	MFT_CLK•1/128
0x6	MFT_CLK•1/64
0x5	MFT_CLK•1/32
0x4	MFT_CLK•1/16
0x3	MFT_CLK•1/8
0x2	MFT_CLK•1/4
0x1	MFT_CLK•1/2
0x0	MFT_CLK•1/1

(Default: 0x0)

Note: When setting the count clock, make sure the MFT counter is stopped.

D7 BUFEN: Comparison/Period Data Buffer Enable Bit

Enables or disables writing to the compare/period data buffer.

1 (R/W): Enabled

0 (R/W): Disabled (default)

When BUFEN is set to 1, compare and period data are read and written from/to the compare and period data buffers. The contents of the buffers are loaded to the compare and period data registers when the counter is reset by the software or the period-match signal.

When BUFEN is set to 0, compare and period data are read and written from/to the compare and period data registers.

D6 Reserved

D5 TCKS: Input Clock Selection Bit

Selects the input clock for the MFT.
1 (R/W): External clock
0 (R/W): Internal clock (default)

The internal clock (prescaler output) is selected for the input clock of the timer by writing 0 to TCKS. An external clock (one that is fed from the EXCL0 pin) is selected by writing 1, and the timer functions as an event counter. In this case, the clock input pin must be configured using the corresponding port function select register before an external clock is selected here.

D[4:2] Reserved

D1 PRST: Timer Reset Bit

Resets the counter.
1 (W): Reset
0 (W): Has no effect
0 (R): Always 0 when read (default)

Writing 1 to PRST resets the counter in the MFT.

D0 TMEN: Timer Run/Stop Control Bit

Controls the timer's Run/Stop state.
1 (R/W): Run
0 (R/W): Stop (default)

The MFT starts counting up by writing 1 to TMEN and stops by writing 0.
In stop state, the counter data is retained until the timer is reset or placed in a run state. By changing states from stop to run, the timer can restart counting beginning at the retained count.

0x521e: MFT Input/Output Control Register (MFT_IOCTL)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
MFT Input/ Output Control Register (MFT_IOCTL)	**0x521e** (16 bits)	D15–7	–	reserved	–				–	–	0 when being read.
		D6	**APIL**	ADC protection input selection	1	Lower limit	0	Upper limit	0	R/W	
		D5	**PPIL**	Port protection input level selection	1	Low level	0	High level	0	R/W	
		D4	**IQA**	Port protection input noise filter	1	12 clocks	0	6 clocks	0	R/W	
		D3	–	reserved	–				–	–	0 when being read.
		D2	**PTM**	Clock output enable	1	Enable	0	Disable	0	R/W	
		D1	**INITOL**	Initial output level	1	High	0	Low	0	R/W	
		D0	**OINV**	Inverse output	1	Invert	0	Normal	0	R/W	

D[15:7] Reserved

D6 APIL: ADC Protection Input Selection Bit

Selects the ADC out-of-range signal for power protection.

1 (R/W): Lower limit signal

0 (R/W): Upper limit signal (default)

When using the ADC protection function that disables the PWM output when an A/D conversion result is out-of-range, select the ADC output signal to activate the protection from the lower-limit and upper-limit signals using APIL.

The ADC protection function is enabled by setting APIE (D1/MFT_IE register) to 1.

D5 PPIL: Port Protection Input Level Selection Bit

Selects the port input signal level for power protection.

1 (R/W): Low level

0 (R/W): High level (default)

When using the port protection function that disables the PWM output by the PWMPRT0 input signal, select the input signal level to activate the protection.

The port protection function is enabled by setting PPIE (D0/MFT_IE register) to 1.

D4 IQA: Port Protection Input Noise Filter Bit

Configures the noise filter for the port protection input signal.

1 (R/W): 12 clocks

0 (R/W): 6 clocks (default)

To avoid the port protection to be activated due to noise, the MFT provides a noise filter at the input of the PWMPRT0 pin. The pulse width to be removed as noise can be selected using IQA. When IQA is set to 1, a pulse shorter than 12 system clocks are regarded as noise and it is not accepted as the valid input. When IQA is set to 0, a pulse shorter than 6 system clocks are regarded as noise.

D3 Reserved

D2 PTM: Clock Output Enable Bit

Controls the output of the TM0 and #TM0 signals (timer output clocks).

1 (R/W): Enable

0 (R/W): Disable (default)

The TM0 and #TM0 signal outputs are enabled by writing 1 to PTM. The clock outputs are stopped by writing 0 to PTM and go to the off level according to the set values of OINV (D0) and INITOL (D1). In this case, the clock output pins must be configured using the corresponding port function select register before outputting the TM0 and #TM0 signals here.

D1 INITOL: Initial Output Level Bit

Selects an initial output level for timer output.
1 (R/W): High
0 (R/W): Low (default)

The timer output pin goes to the initial output level set using this bit when the timer output is turned off by writing 0 to PTM (D2) or when the timer is reset by writing 1 to PRST (D1/MFT_CTL register). However, this level is inverted if OINV (D0) is set to 1.

D0 OINV: Inverse Output Bit

Selects a logic of the output signal.
1 (R/W): Inverted (active low)
0 (R/W): Normal (active high) (default)

By writing 1 to OINV, an active-low signal (off level = high) is generated for the TM0 output (active-high for #TM0). When OINV is set to 0, an active-high signal (off level = low) is generated for the TM0 output (active-low for #TM0).
Writing 1 to this bit inverts the initial output level set using INITOL (D1) as well.

0x5230: MFT Interrupt Enable Register (MFT_IE)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
MFT Interrupt Enable Register (MFT_IE)	**0x5230** (16 bits)	D15–4	–	reserved	–				–	–	0 when being read.
		D3	**CMPIE**	Compare-match interrupt enable	1	Enable	0	Disable	0	R/W	
		D2	**PRDIE**	Period-match interrupt enable	1	Enable	0	Disable	0	R/W	
		D1	**APIE**	ADC protection interrupt enable	1	Enable	0	Disable	0	R/W	
		D0	**PPIE**	Port protection interrupt enable	1	Enable	0	Disable	0	R/W	

D[15:4] Reserved

D3 CMPIE: Compare-Match Interrupt Enable Bit

Enables/disables the compare-match interrupt.

1 (R/W): Enable interrupt

0 (R/W): Disable interrupt (default)

Setting CMPIE to 1 enables the compare-match interrupt; setting to 0 disables the interrupt.

In addition, it is necessary to set the MFT interrupt enable bits in the ITC to interrupt enabled to actually generate an interrupt.

D2 PRDIE: Period-Match Interrupt Enable Bit

Enables/disables the period-match interrupt.

1 (R/W): Enable interrupt

0 (R/W): Disable interrupt (default)

Setting PRDIE to 1 enables the period-match interrupt; setting to 0 disables the interrupt.

In addition, it is necessary to set the MFT interrupt enable bits in the ITC to interrupt enabled to actually generate an interrupt.

D1 APIE: ADC Protection Interrupt Enable Bit

Enables/disables the ADC protection interrupt.

1 (R/W): Enable interrupt

0 (R/W): Disable interrupt (default)

Setting APIE to 1 enables the ADC protection function and its interrupt; setting to 0 disables the ADC protection function and the interrupt.

In addition, it is necessary to set the MFT interrupt enable bits in the ITC to interrupt enabled to actually generate an interrupt.

D0 PPIE: Port Protection Interrupt Enable Bit

Enables/disables the port protection interrupt.

1 (R/W): Enable interrupt

0 (R/W): Disable interrupt (default)

Setting PPIE to 1 enables the port protection function and its interrupt; setting to 0 disables the port protection function and the interrupt.

In addition, it is necessary to set the MFT interrupt enable bits in the ITC to interrupt enabled to actually generate an interrupt.

0x5238: MFT Interrupt Flag Register (MFT_IF)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
MFT Interrupt Flag Register (MFT_IF)	**0x5238** (16 bits)	D15–4	**–**	reserved	–				–	–	0 when being read.
		D3	**CMPIF**	Compare-match interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D2	**PRDIF**	Period-match interrupt flag					0	R/W	
		D1	**APIF**	ADC protection interrupt flag					0	R/W	
		D0	**PPIF**	Port protection interrupt flag					0	R/W	

D[15:4] Reserved

D3 CMPIF: Compare-Match Interrupt Flag Bit

This is the interrupt flag to indicate the compare-match 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): Has no effect

CMPIF is the interrupt flag for the compare-match interrupt. The interrupt flag is set to 1 when the count of the counter matches the set value of the compare data register during count-up if CMPIE (D3/MFT_IE register) has been set to 1. At the same time, the MFT interrupt request signal is output to the ITC. The interrupt request signal sets the MFT interrupt flag in the ITC to 1 and an interrupt occurs if other interrupt conditions meet the ITC and S1C17 Core settings.
CMPIF is reset by writing 1.

D2 PRDIF: Period-Match Interrupt Flag Bit

This is the interrupt flag to indicate the period-match 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): Has no effect

PRDIF is the interrupt flag for the period-match interrupt. The interrupt flag is set to 1 when the count of the counter matches the set value of the period data register during count-up if PRDIE (D2/MFT_IE register) has been set to 1. At the same time, the MFT interrupt request signal is output to the ITC. The interrupt request signal sets the MFT interrupt flag in the ITC to 1 and an interrupt occurs if other interrupt conditions meet the ITC and S1C17 Core settings.
PRDIF is reset by writing 1.

D1 APIF: ADC Protection Interrupt Flag Bit

This is the interrupt flag to indicate the ADC protection 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): Has no effect

APIF is the interrupt flag for the ADC protection interrupt. The interrupt flag is set to 1 when the A/D converter CH.0 asserts the out-of-range signal (upper-limit or lower-limit mode selectable) input to the MFT after an A/D conversion if APIE (D1/MFT_IE register) has been set to 1. At the same time, the MFT interrupt request signal is output to the ITC. The interrupt request signal sets the MFT interrupt flag in the ITC to 1 and an interrupt occurs if other interrupt conditions meet the ITC and S1C17 Core settings.
APIF is reset by writing 1.

D0 PPIF: Port Protection Interrupt Flag Bit

This is the interrupt flag to indicate the port protection 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):	Has no effect

PPIF is the interrupt flag for the port protection interrupt. The interrupt flag is set to 1 when the specified level (high or low) of a signal is input to the PWMPRT0 pin if PPIE (D0/MFT_IE register) has been set to 1. At the same time, the MFT interrupt request signal is output to the ITC. The interrupt request signal sets the MFT interrupt flag in the ITC to 1 and an interrupt occurs if other interrupt conditions meet the ITC and S1C17 Core settings.
PPIF is reset by writing 1.

Notes:
- After an MFT interrupt occurs, reset the PRDIF, CMPIF, APIF, or PPIF interrupt flag of the MFT module in the interrupt handler routine (this also resets the interrupt flag in the ITC).
- To avoid occurrence of unnecessary interrupts, be sure to reset the CMPIF, PRDIF, APIF, or PPIF flag before the MFT interrupt is enabled using CMPIE, PRDIE, APIE, or PPIE (D3, D2, D1, D0/MFT_IE register).

0x527e: MFT Test Register (MFT_TST)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
MFT Test Register (MFT_TST)	**0x527e** (16 bits)	D15–7	–	reserved	–				–	–	0 when being read.
		D6	**DBGMD**	MFT operation in debug mode	1	Stop	0	Run	0	R/W	
		D5–3	–	reserved	–				–	–	0 when being read.
		D2–0	–	reserved	0x7				0x7	–	Fix at 0x7.

D[15:7] Reserved

D6 DBGMD: MFT Operation in Debug Mode Bit

Selects the MFT operation in debug mode.

1 (R/W): Stop
0 (R/W): Run (default)

If DBGMD is set to 0, the MFT operates in debug mode. If DBGMD is set to 1, the MFT stops operating when the S1C17 Core enters debug mode.

D[5:3] Reserved

D[2:0] Reserved (Always set these bits to 0x7.)

IV.2.9 Precautions

- When setting the count clock, make sure the MFT is turned off.
- If a same value is set to the comparison and period data registers, a hazard may be generated in the output signal. Therefore, do not set the registers as MFT_CMP = MFT_PRD.
  There is no problem when the interrupt function only is used.
- When using the output clock, set the comparison and period data registers as MFT_CMP ≥ 0 and MFT_PRD ≥ 1. The minimum settings are MFT_CMP = 0 and MFT_PRD = 1. In this case, the timer output clock cycle is the input clock × 1/2.
- When the comparison and period data registers are set as MFT_CMP > MFT_PRD, no compare-match interrupt is generated. In this case, the output signal is fixed at the off level.
- To prevent another interrupt from being generated by the same cause of interrupt after an interrupt has occurred, be sure to reset the interrupt flag before setting the PSR again or executing the reti instruction.

THIS PAGE IS BLANK.

IV.3 Watchdog Timer (WDT)

IV.3.1 Configuration of the Watchdog Timer

The S1C17002 incorporates a watchdog timer to detect the CPU running uncontrollably. The watchdog timer consists of a 30-bit up counter and comparison data register for generating an NMI or internal reset signal at programmable cycles.

By resetting the watchdog timer within such a cycle in software so as not to generate NMI or internal reset signals, it is possible to detect a program running uncontrollably that does not execute that processing routine.

The WDT clock (= system clock) or external clock input for the MFT (EXCL0) can be selected as the count clock for the watchdog timer.

Moreover, a clock can be generated synchronously with NMI/reset generation cycles (set by the comparison data register) and output from the watchdog timer to external devices.

Figure IV.3.1.1 shows a block diagram of the watchdog timer.

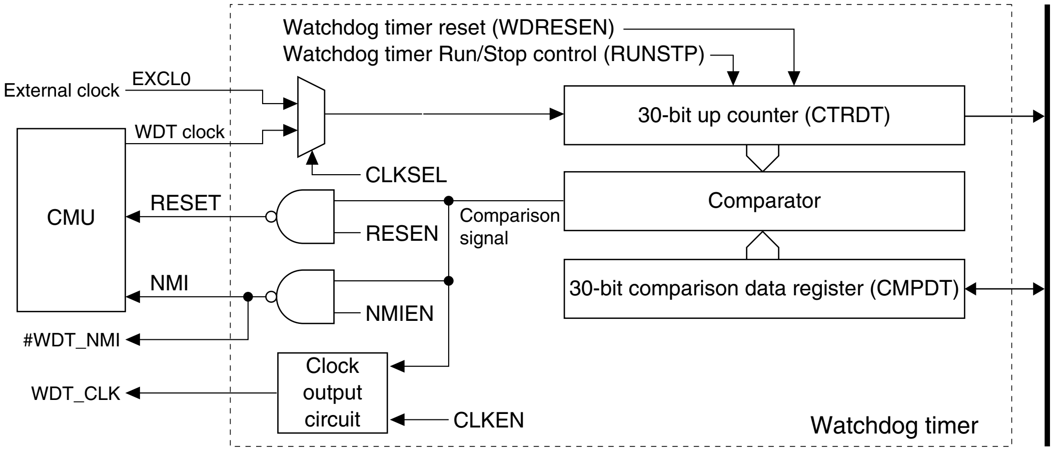


Figure IV.3.1.1 Block Diagram of Watchdog Timer

IV.3.2 Input/Output Pins of the Watchdog Timer

Table IV.3.2.1 Input/Output Pins of Watchdog Timer

Pin name	I/O	Function
EXCL0	I	External clock input pin (external clock input for MFT)
WDT_CLK	O	Watchdog timer clock output pin
#WDT_NMI	O	Watchdog timer NMI output pin

The EXCL0 pin is used to clock the counter of the watchdog timer with an external clock.
The WDT_CLK pin is used to output the clock generated in the watchdog timer to external devices.
The #WDT_NMI pin is used to output the NMI signal generated in the watchdog timer to external devices.

Note: These pins are shared with general-purpose input/output ports or other peripheral circuit input/output pins, and set for other than the watchdog timer function by default. Therefore, before these pins can be used as input/output ports for the watchdog timer clock, the corresponding Port Function Select Register must be set to switch over the pin functions.

For details about pin functions and how to switch over, see Section I.3.3, "Switching Over the Multiplexed Pin Functions."

IV.3.3 Operating Clock of the Watchdog Timer

The watchdog timer module is clocked by the WDT clock (= system clock) supplied from the CMU. At initial reset, this clock is selected as the operating clock for the watchdog timer. While the watchdog timer remains idle or is not being used, the clock supplied from the CMU can be turned off to reduce the amount of current consumed on the chip. Use WDT_CLK_EN (D2/CMU_GATEDCLK2 register) for this control.

∗ **WDT_CLK_EN**: WDT Module Clock Control Bit in the Gated Clock Control 2 (CMU_GATEDCLK2) Register (D2/0x4908)

Setting WDT_CLK_EN to 0 turns off the clock supplied from the CMU to the watchdog timer.
For details about clock generation and control, see Section II.2, "Clock Management Unit (CMU)."

Notes:
- Even when using an external clock as the count clock for the watchdog timer, the WDT clock is required for watchdog timer operation and access to its control register.
- The Gated Clock Control 2 Register (0x4908) is write-protected. To rewrite this register and other CMU control registers at addresses 0x4900 to 0x4908, write protection must be removed by writing 0x96 to the CMU Write Protect Register (0x4920). Since unnecessary rewrites to addresses 0x4900 to 0x4908 may cause the system to operate erratically, make sure that data set in the CMU Write Protect Register (0x4920) is other than 0x96 unless rewriting said registers.

IV.3.4 Control of the Watchdog Timer

IV.3.4.1 Setting Up the Watchdog Timer

Selecting the count clock

The internal clock or external clock (EXCL0) can be selected as the count clock for the 30-bit up-counter by using CLKSEL (D6/WD_EN register).

∗ **CLKSEL**: WDT Input Clock Select Bit in the WDT Enable and Setup (WD_EN) Register (D6/0x5662)

Setting CLKSEL to 0 (default) selects the internal clock; setting it to 1 selects the external clock (EXCL0). Therefore, before an external clock can be used, the function of the pin set as an I/O port by default must be switched to EXCL0 (external clock input for MFT) by using the Port Function Select Register. For details about pin functions and how to switch over, see Section I.3.3, "Switching Over the Multiplexed Pin Functions."
For details about WDT clock supply control, see Section II.2, "Clock Management Unit (CMU)."

Setting the NMI/reset generation cycle

The watchdog timer has a 30-bit comparison data register that can be used to set a cycle in which to generate an NMI or reset signal.

∗ **CMPDT[15:0]**: WDT Comparison Data Bits in the WDT Comparison Data L (WD_CMP_L) Register (D[15:0]/0x5664)
∗ **CMPDT[29:16]**: WDT Comparison Data Bits in the WDT Comparison Data H (WD_CMP_H) Register (D[13:0]/0x5666)

The data set in these register bits is compared with the up-counter value. When both match, a specified NMI or reset signal is output. The up-counter is reset to 0 at this time.
The NMI/reset generation cycle can be calculated from the equation below.

$$\text{NMI generating cycle} = \frac{\text{CMPDT} + 1}{f_{\text{WDTIN}}} \text{ [sec]}$$

where

CMPDT = value set in CMPDT[29:0] (D[13:0]/WD_CMP_H register, D[15:0]/WD_CMP_L register)
f_{WDTIN} = Input clock frequency [Hz]

For example, the specifiable maximum NMI/reset generation cycle is about 21.47 seconds at 50-MHz clock input.

Note: Do not set a value equal to or less than 0x1f in the comparison data register.

Selecting the NMI/reset generation function

To output an NMI signal when the watchdog timer is not reset within a specified cycle, set NMIEN (D1/WD_EN register) to 1. To output a reset signal instead, set RESEN (D0/WD_EN register) to 1.

∗ **NMIEN**: WDT NMI Enable Bit in the WDT Enable and Setup (WD_EN) Register (D1/0x5662)
∗ **RESEN**: WDT RESET Enable Bit in the WDT Enable and Setup (WD_EN) Register (D0/0x5662)

Setting both bits to 0 (default) generates neither an NMI signal nor a reset signal, although the up-counter remains active and can output a clock.
Setting both bits to 1 outputs both an NMI signal and a reset signal. In this case, however, reset handling is executed since it has priority over the NMI handling.
The NMI and reset signals are both output as pulses of 32 system clocks in width.

Note: Depending on the counter and comparison register values, an NMI or reset signal may be generated after the NMI or reset function is enabled here (or even when the watchdog timer has not yet been started). Always be sure to set comparison data and reset the watchdog timer before writing 1 to NMIEN or RESEN.

Write protection of watchdog timer registers

The WDT Enable and Setup Register (0x5662) and WDT Comparison Data Registers (0x5664, 0x5666) are write-protected to prevent NMI or reset signals from being inadvertently generated by unnecessary write operations. To rewrite these registers, write protection must be removed by writing 0x96 to the WDT Write Protect Register (0x5660). Once the registers are rewritten, be sure to write other than 0x96 to the WDT Write Protect Register (0x5660) to reapply write protection.

IV.3.4.2 Starting/Stopping the Watchdog Timer

Writing 1 to RUNSTP (D4/WD_EN register) starts counting by the watchdog timer; writing 0 stops the watchdog timer.

* **RUNSTP**: WDT Run/Stop Control Bit in the WDT Enable and Setup (WD_EN) Register (D4/0x5662)

Since RUNSTP exists in the write-protected WDT Enable and Setup Register, write protection must be removed by writing 0x96 to the WDT Write Protect Register (0x5660) before the content of RUNSTP can be altered.

IV.3.4.3 Resetting the Watchdog Timer

Before the NMI/reset generation function of the watchdog timer can be used, a routine to reset the watchdog timer before NMI or reset generation must be prepared in a location for periodic processing. Make sure that this routine is processed within the NMI/reset generation cycle described earlier.
Writing 1 to WDRESEN (D0/WD_CTL register) resets the watchdog timer. The up-counter is reset to 0 at this time, then starts counting NMI/reset generation cycles all over again.

* **WDRESEN**: WDT Reset Bit in the WDT Control (WD_CTL) Register (D0/0x566c)

If the watchdog timer is not reset within the set cycle for some reason, the CPU is placed into trap handling by an NMI or reset signal to execute the processing routine.
The reset and NMI vector addresses are set by default to 0x20000 and 0x20008, respectively. The vector table base address can be altered by using TTBR.
The count value of the up-counter can be read out from the WDT Count Data Registers (0x5668, 0x566a) at any time.

* **CTRDT[15:0]**: WDT Counter Data Bits in the WDT Count Data L (WD_CNT_L) Register (D[15:0]/0x5668)
* **CTRDT[29:16]**: WDT Counter Data Bits in the WDT Count Data H (WD_CNT_H) Register (D[13:0]/0x566a)

IV.3.4.4 Operation in Standby Mode

In HALT mode

In HALT mode, the watchdog timer remains active as it is supplied with a clock. Therefore, if HALT mode remains active beyond the NMI/reset generation cycle, an NMI or reset signal deactivates HALT mode.
To disable the watchdog timer in HALT mode, set NMIEN (D1/WD_EN register) or RESEN (D0/WD_EN register) to 0. Otherwise, write 0 to RUNSTP (D4/WD_EN register) to stop the watchdog timer before executing the halt instruction.
When NMIEN (D1/WD_EN register) or RESEN (D0/WD_EN register) disables NMI or reset generation, the watchdog timer continues counting even in HALT mode. To reenable NMI or reset generation after exiting HALT mode, be sure to reset the watchdog timer beforehand.
When HALT mode is entered after stopping the watchdog timer, be sure to reset the watchdog timer before restarting it.

In SLEEP mode

The supply of the WDT clock from the CMU stops in SLEEP mode. Therefore, the watchdog timer also stops operating. To prevent an unnecessary NMI or reset signal from being generated after exiting SLEEP mode, be sure to reset the watchdog timer before executing the slp instruction. Moreover, disable NMI/reset generation by setting NMIEN (D1/WD_EN register) or RESEN (D0/WD_EN register) as required.

IV.3.4.5 Clock Output of the Watchdog Timer

The watchdog timer can output an NMI/reset generation cycle-synchronous clock from the IC to external devices. For this clock output, set CLKEN (D5/WD_EN register) to 1 after setting up the WDT_CLK pin.

∗ **CLKEN**: WDT Clock Output Control Bit in the WDT Enable and Setup (WD_EN) Register (D5/0x5662)

Since CLKEN also exists in the write-protected WDT Enable and Setup Register, write protection must be removed by writing 0x96 to the WDT Write Protect Register before the content of CLKEN can be altered.

If the watchdog timer is not reset in software, the level of clock output from the IC is reversed synchronously with the NMI generation cycles. (This applies when reset generation is disabled.)
When the watchdog timer is reset in software, clock output from the IC goes low at that time and remains low.

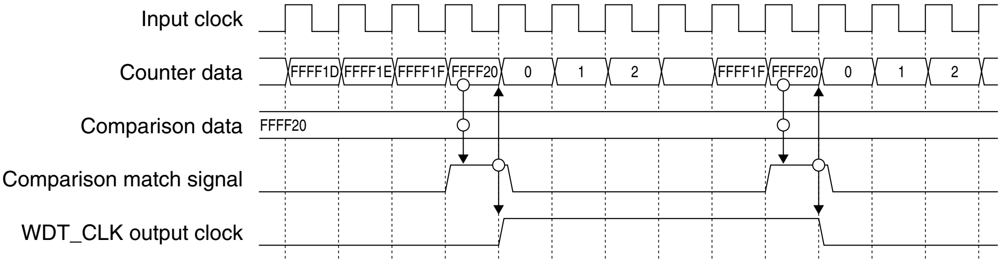


Figure IV.3.4.5.1 Clock Output of Watchdog Timer

IV.3.4.6 External NMI Output

The watchdog timer can output the NMI signal generated to external devices. The watchdog timer uses the #WDT_NMI pin for this output. This pin is configured as a general-purpose I/O pin at initial reset, therefore, the pin function must be set as #WDT_NMI (see Section I.3.3).

Setting NMIEN (D1/WD_EN register) to 1 enables the external NMI signal output as well as the internal NMI signal output.
When the watchdog timer counter reaches the comparison data, the #WDT_NMI pin outputs a low pulse with 32 system clock cycles.

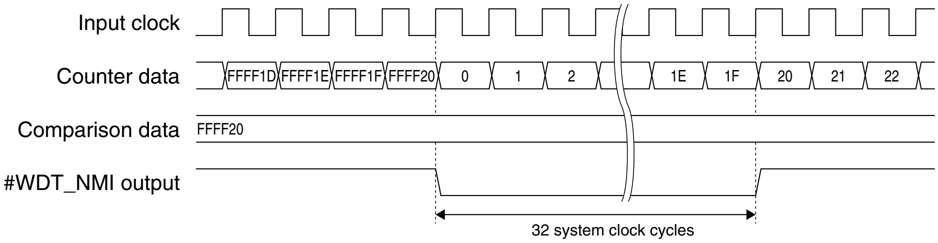


Figure IV.3.4.6.1 External NMI Output

IV.3.5 Details of Control Registers

Table IV.3.5.1 List of WDT Control Registers

Address	Register name		Function
0x5660	WD_WP	WDT Write Protect Register	Enables WDT control registers for writing.
0x5662	WD_EN	WDT Enable and Setup Register	Configures and starts watchdog timer.
0x5664	WD_CMP_L	WDT Comparison Data L Register	Comparison data
0x5666	WD_CMP_H	WDT Comparison Data H Register	
0x5668	WD_CNT_L	WDT Count Data L Register	Watchdog timer counter data
0x566a	WD_CNT_H	WDT Count Data H Register	
0x566c	WD_CTL	WDT Control Register	Resets watchdog timer.

The following describes each WDT register. These are all 16-bit registers.

Note: When setting the registers, be sure to write a 0, and not a 1, for all “reserved bits.”

0x5660: WDT Write Protect Register (WD_WP)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
WDT Write Protect Register (WD_WP)	**0x5660** (16 bits)	D15–0	**WDPTC [15:0]**	WDT register write protect flag	Writing 0x96 removes the write protection of the WD_EN, WD_CMP_L, and WD_CMP_H registers (0x5662–0x5666). Writing another value set the write protection.	X	W	0 when being read.

D[15:0] WDPTC[15:0]: WDT Register Write Protect Flag Bits

These bits set or clear write protection at addresses 0x5662 to 0x5666.

0x96 (W):	Clears write protection
Other than 0x96 (W):	Applies write protection (default, indeterminate value)
0x0 (R):	Always 0x0 when read

Before altering the WDT Enable and Setup Register (0x5662) or WDT Comparison Data Registers (0x5664, 0x5666), write 0x96 to this register to remove write protection. Setting this register to other than 0x96 will result in the contents of the registers above not being altered even when executing the write instruction without any problem. Once write protection is removed by writing 0x96 to this register, said registers can be rewritten any number of times until this register is set to other than 0x96. When the WDT Enable and Setup Register (0x5662) or WDT Comparison Data Registers (0x5664, 0x5666) have been rewritten, be sure to write other than 0x96 to this register to prevent erroneous writing to the registers.

0x5662: WDT Enable and Setup Register (WD_EN)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
WDT Enable and Setup Register (WD_EN)	**0x5662** (16 bits)	D15–7	–	reserved	–				–	–	0 when being read.
		D6	**CLKSEL**	WDT input clock select	1	External clk	0	Internal clk	0	R/W	
		D5	**CLKEN**	WDT clock output control	1	On	0	Off	0	R/W	
		D4	**RUNSTP**	WDT Run/Stop control	1	Run	0	Stop	0	R/W	
		D3–2	–	reserved	–				–	–	0 when being read.
		D1	**NMIEN**	WDT NMI enable	1	Enable	0	Disable	0	R/W	
		D0	**RESEN**	WDT RESET enable	1	Enable	0	Disable	0	R/W	

Note: This register is write-protected to prevent NMI or reset signals from being inadvertently generated by unnecessary write operations. To rewrite this register, write protection must be removed by writing 0x96 to the WDT Write Protect Register (0x5660). Once the register has been rewritten, be sure to write other than 0x96 to the WDT Write Protect Register (0x5660) to reapply write protection.

D[15:7] Reserved

D6 CLKSEL: WDT Input Clock Select Bit

This bit selects the count clock for the watchdog timer.

1 (R/W): External clock (EXCL0)
0 (R/W): Internal clock (default)

Setting this bit to 0 (default) selects the internal clock; setting it to 1 selects the external clock (EXCL0). Before an external clock can be used, the function of the pin set by default as an I/O port must be switched to EXCL0 (external clock input for MFT) by using the Port Function Select Register. For details about pin functions and how to switch over, see Section I.3.3, “Switching Over the Multiplexed Pin Functions.”

D5 CLKEN: WDT Clock Output Control Bit

This bit controls the clock output of the watchdog timer.

1 (R/W): On
0 (R/W): Off (default)

Setting this bit to 1 outputs an NMI/reset generation cycle-synchronous clock from the IC. Before this clock output can be used, however, the function of the pin set by default as an I/O port must be switched to WDT_CLK (watchdog timer clock output) by using the Port Function Select Register. For details about pin functions and how to switch over, see Section I.3.3, “Switching Over the Multiplexed Pin Functions.”

D4 RUNSTP: WDT Run/Stop Control Bit

This bit starts or stops the watchdog timer.

1 (R/W): Start
0 (R/W): Stop (default)

When the NMI or reset generation function is enabled, be sure to set comparison data and reset the watchdog timer before starting the watchdog timer, thus preventing the generation of unnecessary NMI or reset signals.

D[3:2] Reserved

D1 NMIEN: WDT NMI Enable Bit

This bit enables NMI signal output by the watchdog timer.
1 (R/W): Enable
0 (R/W): Disable (default)

Setting this bit to 1 outputs an NMI signal (a pulse 32 system clocks in width) to the CMU and the #WDT_NMI pin when the count of the up-counter matches the value set in the comparison data register. Setting this bit to 0 outputs no NMI signals.
Regardless of how this bit is set, the up-counter is reset to 0 when the up-counter and set value of the comparison data register match, then starts counting all over again.

D0 RESEN: WDT RESET Enable Bit

This bit enables internal reset signal output by the watchdog timer.
1 (R/W): Enable
0 (R/W): Disable (default)

Setting this bit to 1 outputs a reset signal (a pulse 32 system clocks in width) to the CMU when the count of the up-counter matches the value set in the comparison data register. Setting this bit to 0 outputs no reset signals.

0x5664: WDT Comparison Data L Register (WD_CMP_L)
0x5666: WDT Comparison Data H Register (WD_CMP_H)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
WDT Comparison Data L Register (WD_CMP_L)	**0x5664** (16 bits)	D15–0	**CMPDT [15:0]**	WDT comparison data CMPDT0 = LSB	0x0 to 0x3fffffff (low-order 16 bits)	0x0	R/W	
WDT Comparison Data H Register (WD_CMP_H)	**0x5666** (16 bits)	D15–14	–	reserved	–	–	–	0 when being read.
		D13-0	**CMPDT [29:16]**	WDT comparison data CMPDT29 = MSB	0x0 to 0x3fffffff (high-order 14 bits)	0x0	R/W	

Note: These registers are write-protected to prevent NMI or reset signals from being inadvertently generated by unnecessary write operations. To rewrite these registers, write protection must be removed by writing 0x96 to the WDT Write Protect Register (0x5660). Once the registers have been rewritten, be sure to write other than 0x96 to the WDT Write Protect Register (0x5660) to reapply write protection.

Use these registers to set the NMI/reset generation cycle.
With NMI or reset generation enabled, an NMI or reset signal is output when the up-counter matches the comparison data set in these registers.
When a clock is output from the watchdog timer, these registers also set the output clock cycle.

D[15:0]/0x5664 CMPDT[15:0]: WDT Comparison Data Bits (16 low-order bits)

The 16 low-order bits of comparison data are set in these bits. (Default: 0x0)

D[13:0]/0x5666 CMPDT[29:16]: WDT Comparison Data Bits (14 high-order bits)

The 14 high-order bits of comparison data are set in these bits. (Default: 0x0)

Note: Do not set a value equal to or less than 0x1f as comparison data.

0x5668: WDT Count Data L Register (WD_CNT_L)
0x566a: WDT Count Data H Register (WD_CNT_H)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
WDT Count Data L Register (WD_CNT_L)	**0x5668** (16 bits)	D15–0	**CTRDT [15:0]**	WDT counter data CTRDT0 = LSB	0x0 to 0x3fffffff (low-order 16 bits)	X	R	
WDT Count Data H Register (WD_CNT_H)	**0x566a** (16 bits)	D15–14	**–**	reserved	–	–	–	0 when being read.
		D13-0	**CTRDT [29:16]**	WDT counter data CTRDT29 = MSB	0x0 to 0x3fffffff (high-order 14 bits)	X	R	

The current count value of the up-counter can be read out from these registers.

D[15:0]/0x5668 CTRDT[15:0]: WDT Counter Data Bits (16 low-order bits)

The 16 low-order bits of the 30-bit up-counter are read out from these bits. (Default: indeterminate)

D[13:0]/0x566a CTRDT[29:16]: WDT Counter Data Bits (14 high-order bits)

The 14 high-order bits of the 30-bit up-counter are read out from these bits. (Default: indeterminate)

0x566c: WDT Control Register (WD_CTL)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
WDT Control Register (WD_CTL)	**0x566c** (16 bits)	D15–1	–	reserved	–				–	–	0 when being read.
		D0	**WDRESEN**	WDT reset	1	Reset	0	ignored	0	W	

D[15:1] Reserved

D0 WDRESEN: WDT Reset Bit

This bit resets the watchdog timer.

1 (W): Reset

0 (W): Has no effect

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

With NMI or reset signal output enabled, the watchdog timer must be reset by writing 1 to this bit within the set NMI/reset generation cycle. The up-counter is thereby reset to 0, then starts counting NMI/reset generation cycles all over again.

IV.3.6 Precautions

- When NMI or reset signal output by the watchdog timer is enabled, the watchdog timer must be reset within the set NMI/reset generation cycle.
- Do not set a value equal to or less than 0x1f in the comparison data register.
- Depending on the counter and comparison register values, an NMI or reset signal may be generated after the NMI or reset function is enabled, or immediately after the watchdog timer starts. Always be sure to set comparison data and reset the watchdog timer before writing 1 to NMIEN (D1/WD_EN register), RESEN (D0/WD_EN register), or RUNSTP (D4/WD_EN register).

  * **NMIEN**: WDT NMI Enable Bit in the WDT Enable and Setup (WD_EN) Register (D1/0x5662)
  * **RESEN**: WDT RESET Enable Bit in the WDT Enable and Setup (WD_EN) Register (D0/0x5662)
  * **RUNSTP**: WDT Run/Stop Control Bit in the WDT Enable and Setup (WD_EN) Register (D4/0x5662)

IV.4 8-bit OSC1 Timers (T8OSC1)

IV.4.1 Outline of the 8-bit OSC1 Timers

The S1C17002 incorporates two channel of 8-bit OSC1 timer that uses OSC1 as its clock source.
Figure IV.4.1.1 shows the structure of the 8-bit OSC1 timer.

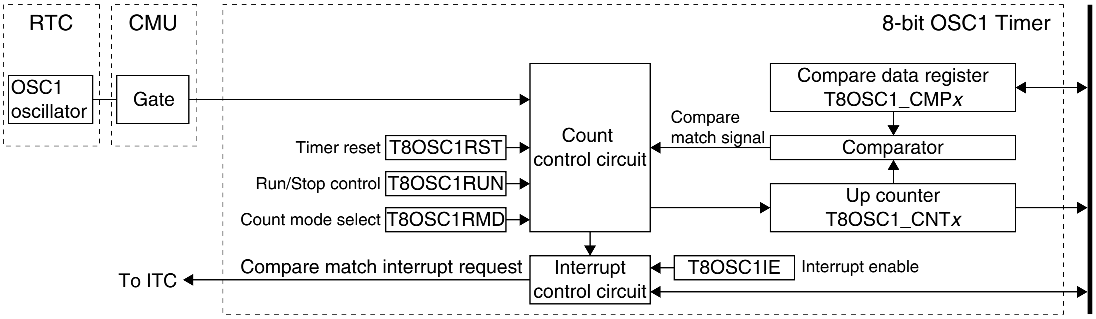


Figure IV.4.1.1 Structure of 8-bit OSC1 Timer (one channel)

In the 8-bit OSC1 timer, an 8-bit up-counter (T8OSC1_CNT*x* register) and an 8-bit compare data register (T8OSC1_CMP*x* register) are provided.

The 8-bit counter can be reset to 0 with software and counts up using the OSC1 clock (32.768 kHz typ.). The counter value can be read by software.

The compare data register is used to store the data to be compared with the content of the up-counter. When the counter value matches to the content of the compare data register, the comparator outputs a signal that controls the interrupt. Thus the register allows interrupt generating interval to be programmed.

Note: The descriptions in this section apply to all 8-bit OSC1 timer channels because they have the same functions except for the control register addresses. The '*x*' in the register names denotes a channel number (0 or 1) and the register addresses are described as (*CH.0*/*CH.1*).
Example: T8OSC1_CTL*x* register (0x4a00/0x4b00)
CH.0: T8OSC1_CTL0 register (0x4a00)
CH.1: T8OSC1_CTL1 register (0x4b00)

IV.4.2 Count Mode of the 8-bit OSC1 Timer

The 8-bit OSC1 timer has two count modes: repeat mode and one-shot mode. It can be selected using the T8OSC1RMD (D1/T8OSC1_CTL*x* register).

∗ **T8OSC1RMD**: Count Mode Select Bit in the 8-bit OSC1 Timer CH.*x* Control (T8OSC1_CTL*x*) Registers (D1/0x4a00/0x4b00)

Repeat mode (T8OSC1RMD = 0, default)

The 8-bit OSC1 timer is set in repeat mode when T8OSC1RMD is set to 0.

In this mode, the 8-bit OSC1 timer does not stop after it starts counting until the application program stops the timer. When the counter value matches to the compare data, the timer resets the counter and continues counting. At the same time, the timer outputs the interrupt signal. Set the 8-bit OSC1 timer in this mode when generating periodical interrupts with a given interval.

One-shot mode (T8OSC1RMD = 1)

The 8-bit OSC1 timer is set in one-shot mode when T8OSC1RMD is set to 1.

In this mode, the 8-bit OSC1 timer automatically stops counting when the counter value matches to the compare data, so only one interrupt can be generated after starting the timer. When a compare match occurs, the counter is reset before the timer operation stops. Set the 8-bit OSC1 timer in this mode when a certain waiting time must be generated.

Note: When setting the count mode, make sure the 8-bit OSC1 timer counter is stopped.

IV.4.3 Count Clock

The 8-bit OSC1 timer uses the OSC1 clock output from the CMU module as the count clock. Use the T8OSC1_CLK_EN (D3/CMU_GATEDCLK1 register) to control the OSC1 clock supply to the 8-bit OSC1 timer. T8OSC1_CLK_EN is set to 1 by default, so the clock is supplied to the 8-bit OSC1 timer. If the 8-bit OSC1 timer does not need to operate, stop the clock supply to reduce the current consumption by setting T8OSC1_CLK_EN to 0.

∗ **T8OSC1_CLK_EN**: 8-bit OSC1 Timer Clock Control Bit in the Gated Clock Control 1 (CMU_GATEDCLK1) Register (D3/0x4907)

Note: When setting the count clock, make sure the 8-bit OSC1 timer counter is stopped.

For control of the clock, see Section II.2, "Clock Management Unit (CMU)."

IV.4.4 Resetting the 8-bit OSC1 Timer

To reset the 8-bit OSC1 timer, write 1 to the T8OSC1RST bit (D4/T8OSC1_CTL*x* register). This initializes the counter to 0.

∗ **T8OSC1RST**: Timer Reset Bit in the 8-bit OSC1 Timer CH.*x* Control (T8OSC1_CTL*x*) Registers (D4/0x4a00/0x4b00)

Normally, reset the counter before starting count-up by writing 1 to this control bit.
After the counter starts counting, it will be reset by the hardware when the counter reaches compare data.

IV.4.5 Setting Compare Data

Write compare data to T8OSC1CMP[7:0] (D[7:0]/T8OSC1_CMP*x* register).

∗ **T8OSC1CMP[7:0]**: Compare Data Bits in the 8-bit OSC1 Timer CH.*x* Compare Data (T8OSC1_CMP*x*) Registers (D[7:0]/0x4a02/0x4b02)

At initial reset, the compare data register is set to 0x0.

The timer compares the compare data register and count data and, when the two values are equal, resets the counter and generates a compare match signal. This compare match signal is used to generate an interrupt.
The compare match period is calculated by the expression below.

$$\text{Compare match period} = \frac{\text{CMP} + 1}{\text{clk\_in}} \text{ [s]}$$

$$\text{Compare match cycle} = \frac{\text{clk\_in}}{\text{CMP} + 1} \text{ [Hz]}$$

CMP: Compare data (T8OSC1_CMP*x* register value)
clk_in: 8-bit OSC1 timer count clock frequency

IV.4.6 8-bit OSC1 Timer Run/Stop Control

Before starting the 8-bit OSC1 timer, set up the conditions as shown below.
(1) Select a count mode (one-shot or repeat). See Section IV.4.2.
(2) Select the operating clock. See Section IV.4.3.
(3) Set up the interrupt level and enable the 8-bit OSC1 timer interrupt if the interrupt is used. See Section IV.4.7.
(4) Reset the timer. See Section IV.4.4.
(5) Set compare data. See Section IV.4.5.

The 8-bit OSC1 timer provides T8OSC1RUN (D0/T8OSC1_CTL*x* register) to run and stop the counter.

∗ **T8OSC1RUN**: Timer Run/Stop Control Bit in the 8-bit OSC1 Timer CH.*x* Control (T8OSC1_CTL*x*) Registers (D0/0x4a00/0x4b00)

The timer starts counting when 1 is written to T8OSC1RUN. The clock input is disabled and the timer stops counting when 0 is written to T8OSC1RUN. This control does not affect the counter data. Even when the timer has stopped counting, the counter retains its count so that the timer can start counting again from that point.
When both T8OSC1RUN and T8OSC1RST are set to 1 at the same time, the timer starts counting after resetting the counter.

If the count of the counter matches the set value of the compare data register during count-up, the timer outputs the compare match signal as a cause of interrupt. At the same time, the counter is reset to 0. If the interrupt has been enabled, an interrupt request is sent to the interrupt controller (ITC).
If the timer is set in one-shot mode, the timer stops counting.
If the timer is set in repeat mode, the timer continues counting from the counter value 0.

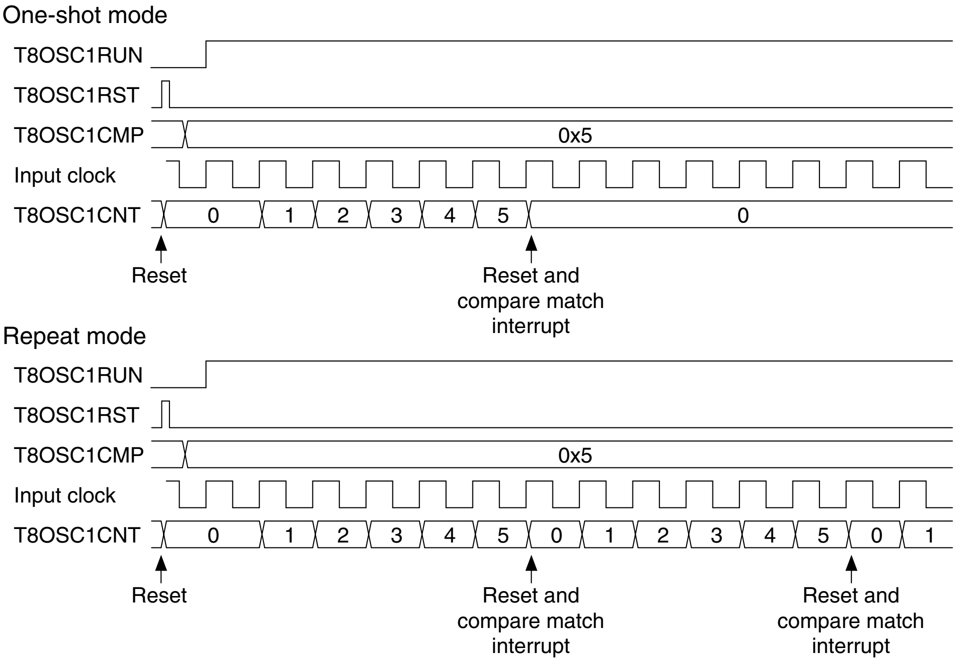


Figure IV.4.6.1 Basic Operation Timing of Counter

IV.4.7 8-bit OSC1 Timer Interrupt

The T8OSC1 module is able to output an interrupt request signal to the interrupt controller (ITC) when a compare match occurs.

Compare match interrupt

This interrupt request occurs when the count of the counter matches the set value of the compare data register during count-up, and it sets the interrupt flag T8OSC1IF (D0/T8OSC1_IFLG*x* register) in the T8OSC1 module to 1.

* **T8OSC1IF**: 8-bit OSC1 Timer Interrupt Flag Bit in the 8-bit OSC1 Timer CH.*x* Interrupt Flag (T8OSC1_IFLG*x*) Registers (D0/0x4a04/0x4b04)

Set the T8OSC1IE (D0/T8OSC1_IMSKx register) to 1 when using this interrupt. If T8OSC1IE is set to 0 (default), T8OSC1IF will not be set to 1 and an interrupt request by this cause will not be sent to the ITC.

* **T8OSC1IE**: 8-bit OSC1 Timer Interrupt Enable Bit in the 8-bit OSC1 Timer CH.*x* Interrupt Mask (T8OSC1_IMSK*x*) Registers (D0/0x4a03/0x4b03)

If T8OSC1IF is set to 1, the T8OSC1 module outputs the interrupt request signal to the ITC. The interrupt request signal sets the 8-bit OSC1 timer interrupt flag in the ITC to 1 and an interrupt occurs if other interrupt conditions meet the ITC and S1C17 Core settings.

The interrupt handler routine must reset (write 1 to) T8OSC1IF in the T8OSC1 module, not the 8-bit OSC1 timer interrupt flag in the ITC, to clear the cause of interrupt.

Note: To avoid occurrence of unnecessary interrupts, be sure to reset the T8OSC1IF flag before the compare match interrupt is enabled using T8OSC1IE.

ITC registers for 8-bit OSC1 timer interrupt

When a compare match occurs according to the interrupt condition settings shown above, the 8-bit OSC1 timer asserts the interrupt signal sent to the ITC. To generate an 8-bit OSC1 timer interrupt, set the interrupt level and enable the interrupt using the ITC registers. Table IV.4.7.1 shows the control bits for the 8-bit OSC1 timer interrupt in the ITC.

Table IV.4.7.1 ITC Registers

Channel	Interrupt flag	Interrupt enable bit	Interrupt level setup bits
CH.0	AIFT5 (D5/ITC_AIFLG)	AIEN5 (D5/ITC_AEN)	AILV5[2:0] (D[10:8]/ITC_AILV2)
CH.1	AIFT6 (D6/ITC_AIFLG)	AIEN6 (D6/ITC_AEN)	AILV6[2:0] (D[2:0]/ITC_AILV3)

ITC_AIFLG register (0x42e0)
ITC_AEN register (0x42e2)
ITC_AILV2, ITC_AILV3 registers (0x42ea, 0x42ec)

When a compare match whose interrupt is enabled in the T8OSC1 module occurs, the corresponding interrupt flag is set to 1.

If the interrupt enable bit corresponding to that interrupt flag has been set to 1, the ITC sends an interrupt request to the S1C17 Core. To disable the timer interrupt, set the interrupt enable bit to 0.

The interrupt flag is always set to 1 by the 8-bit OSC1 timer interrupt signal, regardless of how the interrupt enable bit is set (even when set to 0).

The interrupt level setup bits set the interrupt level (0 to 7) of the timer interrupt. If the same interrupt level is set, timer CH.0 has highest priority and timer CH.1 has lowest priority.

An interrupt request to the S1C17 Core is accepted only when all the conditions described below are met.

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

For details on these interrupt control registers, as well as the device operation when an interrupt has occurred, see Section III.1, “Interrupt Controller (ITC).”

Interrupt vector

The following shows the vector number and vector address for the 8-bit OSC1 timer interrupt:

Table IV.4.7.2 Timer Interrupt Vectors

Channel	Vector number	Vector address
CH.0	21 (0x15)	TTBR + 0x54
CH.1	22 (0x16)	TTBR + 0x58

IV.4.8 Details of Control Registers

Table IV.4.8.1 List of 8-bit OSC1 Timer Registers

Address	Register name		Function
0x4a00	T8OSC1_CTL0	8-bit OSC1 Timer CH.0 Control Register	Sets the timer mode and starts/stops the timer.
0x4a01	T8OSC1_CNT0	8-bit OSC1 Timer CH.0 Counter Data Register	Counter data
0x4a02	T8OSC1_CMP0	8-bit OSC1 Timer CH.0 Compare Data Register	Sets compare data.
0x4a03	T8OSC1_IMSK0	8-bit OSC1 Timer CH.0 Interrupt Mask Register	Enables/disables interrupt.
0x4a04	T8OSC1_IFLG0	8-bit OSC1 Timer CH.0 Interrupt Flag Register	Indicates/resets interrupt occurrence status.
0x4b00	T8OSC1_CTL1	8-bit OSC1 Timer CH.1 Control Register	Sets the timer mode and starts/stops the timer.
0x4b01	T8OSC1_CNT1	8-bit OSC1 Timer CH.1 Counter Data Register	Counter data
0x4b02	T8OSC1_CMP1	8-bit OSC1 Timer CH.1 Compare Data Register	Sets compare data.
0x4b03	T8OSC1_IMSK1	8-bit OSC1 Timer CH.1 Interrupt Mask Register	Enables/disables interrupt.
0x4b04	T8OSC1_IFLG1	8-bit OSC1 Timer CH.1 Interrupt Flag Register	Indicates/resets interrupt occurrence status.

The following describes each 8-bit OSC1 timer register. These are all 8-bit registers.

Note: When setting the registers, be sure to write a 0, and not a 1, for all “reserved bits.”

0x4a00/0x4b00: 8-bit OSC1 Timer CH.*x* Control Registers (T8OSC1_CTL*x*)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
8-bit OSC1 Timer CH.*x* Control Register (T8OSC1_ CTL*x*)	**0x4a00 0x4b00** (8 bits)	D7–5	–	reserved	–				–	–	0 when being read.
		D4	T8OSC1RST	Timer reset	1	Reset	0	Ignored	0	W	
		D3–2	–	reserved	–				–	–	
		D1	T8OSC1RMD	Count mode select	1	One shot	0	Repeat	0	R/W	
		D0	T8OSC1RUN	Timer run/stop control	1	Run	0	Stop	0	R/W	

Note: The letter ‘*x*’ in register names, etc., denotes a channel number (0 or 1).

0x4a00: 8-bit OSC1 Timer CH.0 Control Register (T8OSC1_CTL0)
0x4b00: 8-bit OSC1 Timer CH.1 Control Register (T8OSC1_CTL1)

D[7:5] Reserved

D4 T8OSC1RST: Timer Reset Bit

Resets the 8-bit OSC1 timer.

1 (W): Reset
0 (W): Has no effect
0 (R): Always 0 when read (default)

Writing 1 to this bit resets the counter to 0.

D[3:2] Reserved

D1 T8OSC1RMD: Count Mode Select Bit

Selects the count mode of the 8-bit OSC1 timer.

1 (R/W): One-shot mode
0 (R/W): Repeat mode (default)

The 8-bit OSC1 timer is set in repeat mode when T8OSC1RMD is set to 0. In this mode, the 8-bit OSC1 timer does not stop after it starts counting until the application program stops the timer. When the counter value matches to the compare data, the timer resets the counter and continues counting. At the same time, the timer outputs the interrupt signal. Set the 8-bit OSC1 timer in this mode when generating periodical interrupts with a given interval.

The 8-bit OSC1 timer is set in one-shot mode when T8OSC1RMD is set to 1. In this mode, the 8-bit OSC1 timer automatically stops counting when the counter value matches to the compare data, so only one interrupt can be generated after starting the timer. When a compare match occurs, the counter is reset before the timer operation stops. Set the 8-bit OSC1 timer in this mode when a certain waiting time must be generated.

Note: When setting the count mode, make sure the 8-bit OSC1 timer counter is stopped.

D0 T8OSC1RUN: Timer Run/Stop Control Bit

Controls the timer's Run/Stop state.

1 (R/W): Run
0 (R/W): Stop (default)

The timer starts counting by writing 1 to T8OSC1RUN and stops counting by writing 0.
In the stop state, the counter data is retained until the timer is reset or placed in a run state.

0x4a01/0x4b01: 8-bit OSC1 Timer CH.*x* Counter Data Registers (T8OSC1_CNT*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
8-bit OSC1 Timer CH.*x* Counter Data Register (T8OSC1 _CNT*x*)	**0x4a01** **0x4b01** (8 bits)	D7–0	**T8OSC1CNT [7:0]**	Timer counter data T8OSC1CNT7 = MSB T8OSC1CNT0 = LSB	0x0 to 0xff	0x0	R	

Note: The letter '*x*' in register names, etc., denotes a channel number (0 or 1).

0x4a01: 8-bit OSC1 Timer CH.0 Counter Data Register (T8OSC1_CNT0)
0x4b01: 8-bit OSC1 Timer CH.1 Counter Data Register (T8OSC1_CNT1)

D[7:0] T8OSC1CNT[7:0]: Timer Counter Data Bits

The counter data can be read from this register. (Default: 0x0)

This is a read-only register, so the writing operation is invalid.

Note: If this register is read while the counter is running, the read value may not represent the current counter value (an indefinite value may be read out).

The counter value should be obtained by one of the following procedures:

- Read the counter value after stopping the counter.
- Read the counter value twice to determine that both read results are the same and that the read value is significant.

0x4a02/0x4b02: 8-bit OSC1 Timer CH.*x* Compare Data Registers (T8OSC1_CMP*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
8-bit OSC1 Timer CH.*x* Compare Data Register (T8OSC1 _CMP*x*)	**0x4a02** **0x4b02** (8 bits)	D7–0	**T8OSC1CMP [7:0]**	Compare data T8OSC1CMP7 = MSB T8OSC1CMP0 = LSB	0x0 to 0xff	0x0	R/W	

Note: The letter '*x*' in register names, etc., denotes a channel number (0 or 1).

0x4a02: 8-bit OSC1 Timer CH.0 Compare Data Register (T8OSC1_CMP0)
0x4b02: 8-bit OSC1 Timer CH.1 Compare Data Register (T8OSC1_CMP1)

D[7:0] T8OSC1CMP[7:0]: Compare Data Bits

Sets the compare data for the 8-bit OSC1 timer. (Default: 0x0)

The data set in this register is compared with the counter data. When the contents match, a cause of compare interrupt is generated. At the same time, the counter is reset to 0.

0x4a03/0x4b03: 8-bit OSC1 Timer CH.*x* Interrupt Mask Register (T8OSC1_IMSK*x*)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
8-bit OSC1 Timer CH.*x* Interrupt Mask Register (T8OSC1 _IMSK*x*)	**0x4a03** **0x4b03** (8 bits)	D7–1	–	reserved	–				–	–	0 when being read.
		D0	**T8OSC1IE**	8-bit OSC1 timer interrupt enable	1	Enable	0	Disable	0	R/W	

Note: The letter '*x*' in register names, etc., denotes a channel number (0 or 1).

0x4a03: 8-bit OSC1 Timer CH.0 Interrupt Mask Register (T8OSC1_IMSK0)
0x4b03: 8-bit OSC1 Timer CH.1 Interrupt Mask Register (T8OSC1_IMSK1)

D[7:1] Reserved

D0 T8OSC1IE: 8-bit OSC1 Timer Interrupt Enable Bit

Enables/disables the compare match interrupt.

1 (R/W): Enable interrupt
0 (R/W): Disable interrupt (default)

Setting T8OSC1IE to 1 enables the 8-bit OSC1 timer to request interrupts to the ITC; setting to 0 disables the interrupt.
In addition, it is necessary to set the 8-bit OSC1 timer interrupt enable bits in the ITC to interrupt enabled to actually generate an interrupt.

0x4a04/0x4b04: 8-bit OSC1 Timer CH.*x* Interrupt Flag Registers (T8OSC1_IFLG*x*)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
8-bit OSC1 Timer CH.*x* Interrupt Flag Register (T8OSC1_IFLG*x*)	**0x4a04** **0x4b04** (8 bits)	D7–1	–	reserved	–				–	–	0 when being read.
		D0	**T8OSC1IF**	8-bit OSC1 timer interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.

Note: The letter '*x*' in register names, etc., denotes a channel number (0 or 1).

0x4a04: 8-bit OSC1 Timer CH.0 Interrupt Flsg Register (T8OSC1_IFLG0)
0x4b04: 8-bit OSC1 Timer CH.1 Interrupt Flag Register (T8OSC1_IFLG1)

D[7:1] Reserved

D0 T8OSC1IF: 8-bit OSC1 Timer Interrupt Flag Bit

This is the interrupt flag to indicate the compare match 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):	Has no effect

T8OSC1IF is the interrupt flag for the T8OSC1 module. The interrupt flag is set to 1 when the count of the counter matches the set value of the compare data register during count-up if T8OSC1IE (D0/T8OSC1_IMSK*x* register) has been set to 1. At the same time, the 8-bit OSC1 timer interrupt request signal is output to the ITC. The interrupt request signal sets the 8-bit OSC1 timer interrupt flag in the ITC to 1 and an interrupt occurs if other interrupt conditions meet the ITC and S1C17 Core settings.
The T8OSC1IF flag is reset by writing 1.

Note: To avoid occurrence of unnecessary interrupts, be sure to reset the T8OSC1IF flag before the compare match interrupt is enabled using T8OSC1IE (D0/T8OSC1_IMSK*x* register).

IV.4.9 Precautions

- Before the 8-bit OSC1 timer can start counting, the 8-bit OSC1 timer clock must be supplied from the CMU module.
- When setting the count clock or count mode, make sure the 8-bit OSC1 timer is turned off.
- To avoid occurrence of unnecessary interrupts, be sure to reset T8OSC1IF (D0/T8OSC1_IFLG*x* register) before the compare match interrupt is enabled using T8OSC1IE (D0/T8OSC1_IMSK*x* register).
- If the counter data register is read while the counter is running, the read value may not represent the current counter value (an indefinite value may be read out).
  To obtain the counter value, read the counter data register after stopping the counter. Or read the counter value twice to determine that both read results are the same and that the read value is significant.

THIS PAGE IS BLANK.

V S1C17002 INTERFACE MODULES

V.1 UART

V.1.1 Outline of the UART

The S1C17002 equipped with one channel of UART. The UART performs asynchronous data transfer from/to an external serial device in a 150 to 460800 bps (max. 115200 bps in IrDA mode) transfer rate. The UART contains two-byte receive data buffer and one-byte transmit data buffer allowing full-duplex communication. The transfer clock is internally generated using a timer module or an external clock is input from the #SCLK0 pin. The character length (seven or eight bits), number of stop bits (one or two bits), and parity mode (even, odd, or none) are programmable. The start bit is fixed at one bit. In data receive operation, overrun, framing, and parity errors are detectable. The UART can generate three types of interrupts (transmit buffer empty, receive buffer full, and receive error), this makes it possible to process serial data transfer simply in an interrupt handler.

Furthermore, the UART module contains an RZI modulator/demodulator, allowing an infrared-ray communication circuit to be configured based on IrDA 1.0 simply by adding an external circuit.

Figure V.1.1.1 shows the structure of the UART.

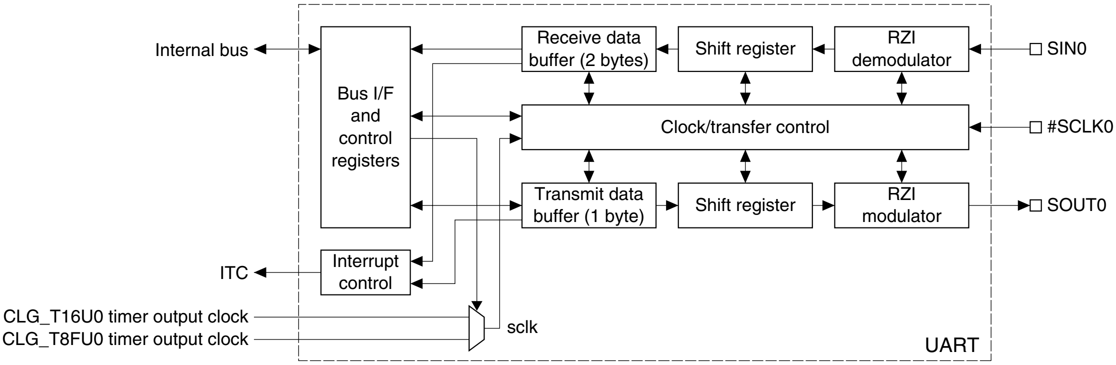

Internal bus
Bus I/F and control registers
Receive data buffer (2 bytes)
Shift register
RZI demodulator
SIN0
Clock/transfer control
#SCLK0
Transmit data buffer (1 byte)
Shift register
RZI modulator
SOUT0
ITC
Interrupt control
CLG_T16U0 timer output clock
CLG_T8FU0 timer output clock
sclk
UART

Figure V.1.1.1 Structure of UART

V.1.2 UART Pins

Table V.1.2.1 lists the I/O pins for the UART.

Table V.1.2.1 List of UART Pins

Pin name	I/O	Size	Function
SIN0	I	1	UART data input pin This pin inputs serial data sent from an external serial device.
SOUT0	O	1	UART data output pin This pin outputs serial data to be sent to an external serial device.
#SCLK0	I	1	UART clock input pin This pin inputs the transfer clock when an external clock is used.

The UART input/output pins (SIN0, SOUT0, #SCLK0) are shared with the I/O ports (P40, P41, P42) and they are initialized as general-purpose I/O port pins by default. Before using these pins for the UART, the pin functions must be switched using the Port Function Select Register.

For details on switching pin function, Section I.3.3, “Switching Over the Multiplexed Pin Functions.”

V.1.3 Transfer Clock

The UART allows the application to select either an internal clock or an external clock as the transfer clock. Use the SSCK bit (D0/UART_MOD register) for this selection.

∗ **SSCK**: Input Clock Select Bit in the UART Mode (UART_MOD) Register (D0/0x4103)

Note: Make sure that the UART is disabled (RXEN/UART_CTL register = 0) when alter the SSCK bit.

∗ **RXEN**: UART Enable Bit in the UART Control (UART_CTL) Register (D0/0x4104)

Internal clock

When SSCK is set to 0 (default), the internal clock is selected. The UART uses the CLG_T16U0 or CLG_T8FU0 timer output clock as the transfer clock.
Use TMSEL (D5/UART_MOD register) to select the timer to be used as the clock source for the UART.

∗ **TMSEL**: Timer Select Bit in the UART Mode (UART_MOD) Register (D5/0x4103)

When TMSEL is 0 (default), the CLG_T16U0 output clock is used. When TMSEL is set to 1, CLG_T8FU0 output clock is used.
Furthermore, it is necessary to program the CLG_T16U0/CLG_T8FU0 timer so that it will output a clock according to the transfer rate.
See Section II.4, "Clock Generator (CLG)," for controlling the CLG_T16U0/CLG_T8FU0 timer.

External clock

When SSCK is set to 1, an external clock is selected. Configure the #SCLK0 pin and input an external clock to the pin.

Notes:
- The UART divides the CLG_T16U0/CLG_T8FU0 timer output clock or external clock by 16 to generate the sampling clock. Make sure of the division ratio when setting a transfer rate.
- The frequency of the external clock input from the #SCLK0 pin must be half of PCLK or lower and the clock duty ratio must be 50%.

V.1.4 Setting Transfer Data Conditions

The following conditions are selectable to configure transfer data format:

• Character length:	7 or 8 bits
• Start bit:	1 bit, fixed
• Stop bit:	1 or 2 bits
• Parity bit:	Even, odd, or none

Note: Make sure that the UART is disabled (RXEN/UART_CTL register = 0) when setting the transfer data conditions.

∗ **RXEN**: UART Enable Bit in the UART Control (UART_CTL) Register (D0/0x4104)

Character length

Use the CHLN bit (D4/UART_MOD register) to select the character length. When CHLN is set to 0 (default), the character length is configured to seven bits; when CHLN is set to 1, the character length is configured to eight bits.

∗ **CHLN**: Character Length Bit in the UART Mode (UART_MOD) Register (D4/0x4103)

Stop bit

Use the STPB bit (D1/UART_MOD register) to select the stop bit length. When STPB is set to 0 (default), the stop bit length is set to one bit; when STPB is set to 1, the stop bit length is set to two bits.

∗ **STPB**: Stop Bit Select Bit in the UART Mode (UART_MOD) Register (D1/0x4103)

Parity bit

Use the PREN bit (D3/UART_MOD register) to select whether the parity function is enabled or not. When PREN is set to 0 (default), parity function is disabled. In this case, a parity bit will not be added to transfer data and the parity check will not be performed when data is received. When PREN is set to 1, parity function is enabled. In this case, a parity bit will be added to transfer data and the parity check will be performed when data is received.

When the parity function is enabled, select a parity mode using the PMD bit (D2/UART_MOD register). When PMD is set to 0 (default), the parity bit is added/checked as even parity; when PMD is set to 1, the parity bit is added/checked as odd parity.

∗ **PREN**: Parity Enable Bit in the UART Mode (UART_MOD) Register (D3/0x4103)
∗ **PMD**: Parity Mode Select Bit in the UART Mode (UART_MOD) Register (D2/0x4103)

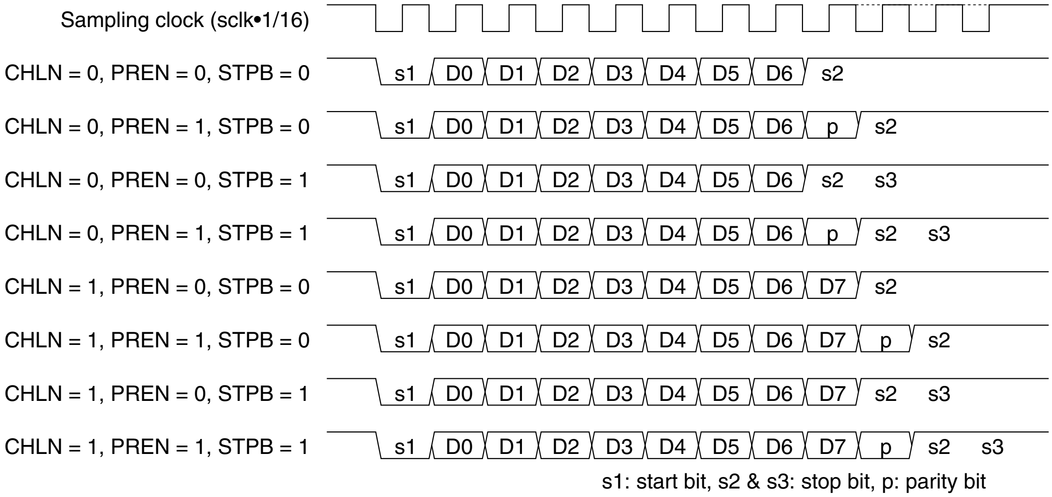

Sampling clock (sclk•1/16)
CHLN = 0, PREN = 0, STPB = 0
s1 D0 D1 D2 D3 D4 D5 D6 s2
CHLN = 0, PREN = 1, STPB = 0
s1 D0 D1 D2 D3 D4 D5 D6 p s2
CHLN = 0, PREN = 0, STPB = 1
s1 D0 D1 D2 D3 D4 D5 D6 s2 s3
CHLN = 0, PREN = 1, STPB = 1
s1 D0 D1 D2 D3 D4 D5 D6 p s2 s3
CHLN = 1, PREN = 0, STPB = 0
s1 D0 D1 D2 D3 D4 D5 D6 D7 s2
CHLN = 1, PREN = 1, STPB = 0
s1 D0 D1 D2 D3 D4 D5 D6 D7 p s2
CHLN = 1, PREN = 0, STPB = 1
s1 D0 D1 D2 D3 D4 D5 D6 D7 s2 s3
CHLN = 1, PREN = 1, STPB = 1
s1 D0 D1 D2 D3 D4 D5 D6 D7 p s2 s3
s1: start bit, s2 & s3: stop bit, p: parity bit

Figure V.1.4.1 Transfer Data Format

V.1.5 Data Transmit/Receive Control

Before starting data transfer, set up the conditions as shown below.

(1) Select an input clock. See Section V.1.3.
Set up the CLG_T16U0 or CLG_T8FU0 timer to output the transfer clock if the internal clock is used as the transfer clock. See Sections II.4 and V.1.3.

(2) Configure the transfer data format. See Section V.1.4.

(3) Set IrDA mode when using the IrDA interface. See Section V.1.8.

(4) Set up the interrupt conditions if the UART interrupt is used. See Section V.1.7.

Note: Make sure that the UART is disabled (RXEN/UART_CTL register = 0) when setting the conditions above.

∗ **RXEN**: UART Enable Bit in the UART Control (UART_CTL) Register (D0/0x4104)

Enabling data transmission/reception

First, set the RXEN bit (D0/UART_CTL register) to 1 to enable data transmission/reception. This puts the transmitter/receiver in ready-to-transmit/receive status.

Note: Do not set the RXEN bit to 0 while the UART is transmitting/receiving data.

Data transmit control

To start transmission, write transmit data to the UART_TXD register (0x4101).

∗ **UART_TXD**: UART Transmit Data Register (0x4101)

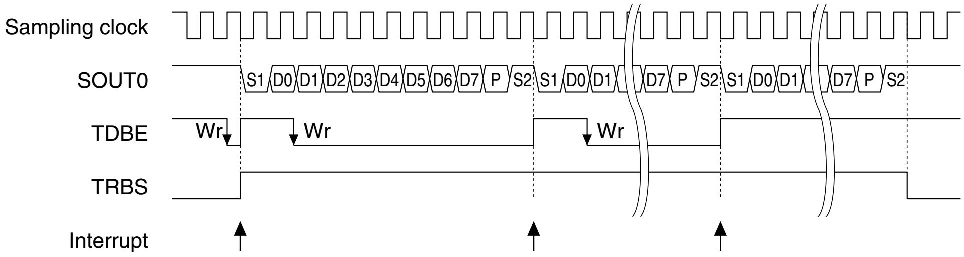
Data is written to the transmit data buffer and the transmitter starts data transmission.
The buffered data is sent to the shift register for transmission and a start bit is output from the SOUT0 pin. Then data in the shift register is output from the LSB. The transmit data bits are shifted in sync with the rising edge of the sampling clock and output from the SOUT0 pin sequentially. After the MSB has been output, a parity bit (if parity is enabled) and a stop bit are output.

The transmitter provides two status flags, TDBE (D0/UART_ST register) and TRBS (D2/UART_ST register).

∗ **TDBE**: Transmit Data Buffer Empty Flag Bit in the UART Status (UART_ST) Register (D0/0x4100)
∗ **TRBS**: Transmit Busy Flag Bit in the UART Status (UART_ST) Register (D2/0x4100)

The TDBE flag indicates the transmit data buffer status; it goes 0 when the application program writes data to the transmit data buffer and returns to 1 when the data in the transmit data buffer is sent to the shift register for transmitting. An interrupt can be generated when this flag goes 1 (see Section V.1.7). Use this interrupt or read the TDBE flag to check that the transmit data buffer is empty before transmitting the next data. Although the transmit data buffer size is one byte, transmit data can be written while the previous data is being transmitted as the shift register is separately provided. However, make sure that the transmit data buffer is empty before writing transmit data. If data is written when the TDBE flag is 0, the previous transmit data in the transmit data buffer is overwritten with the new data.

The TRBS flag indicates the shift register status; it goes 1 when transmit data is loaded from the transmit data buffer and returns to 0 upon completion of a data transmission. Read this flag to check whether the transmitter is busy or idle.



S1: Start bit, S2: Stop bit, P: Parity bit, Wr: Data write to transmit data buffer

Figure V.1.5.1 Data Transmit Timing Chart

Data receive control

The receiver activates by setting the RXEN bit to 1 and is ready to receive data sent from an external serial device.
When an external serial device has sent a start bit, the receiver detects its low level and starts following data bit sampling. The data bits are sampled at the rising edge of the sampling clock and received in the receive shift register assuming that the first data bit is LSB. After the MSB is received in the shift register, the received data is loaded to the receive data buffer. At the same time, the receiver performs a parity check with the parity bit received after the MSB if parity check is enabled.
The receive data buffer is a two-byte FIFO and can receive data until it becomes full.

The received data in the buffer can be read from the UART_RXD register (0x4102). The older data is read out first and cleared by reading.

∗ **UART_RXD**: UART Receive Data Register (0x4102)

The receiver provides two buffer status flags, RDRY (D1/UART_ST register) and RD2B (D3/UART_ST register).

∗ **RDRY**: Receive Data Ready Flag Bit in the UART Status (UART_ST) Register (D1/0x4100)
∗ **RD2B**: Second Byte Receive Flag Bit in the UART Status (UART_ST) Register (D3/0x4100)

The RDRY flag indicates that the receive data buffer contains the received 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 SectionV.1.6.

By reading these flags, the application program can check how many data have been received.

Furthermore, the UART can generate a receive data buffer full interrupt when data is received in the receive data buffer. This interrupt can be used to read the received data. A receive data buffer full interrupt occurs when one data has been received in the receive data buffer (status (2) above) by default. This may be changed by setting the RBFI bit (D1/UART_CTL register) to 1 so that the interrupt will occur when two data have been received in the received data buffer.

∗ **RBFI**: Receive Buffer Full Interrupt Condition Bit in the UART Control (UART_CTL) Register (D1/0x4104)

In addition to the flags above, three receive error flags are provided. Refer to Section V.1.6 for these flags and details of receive errors.

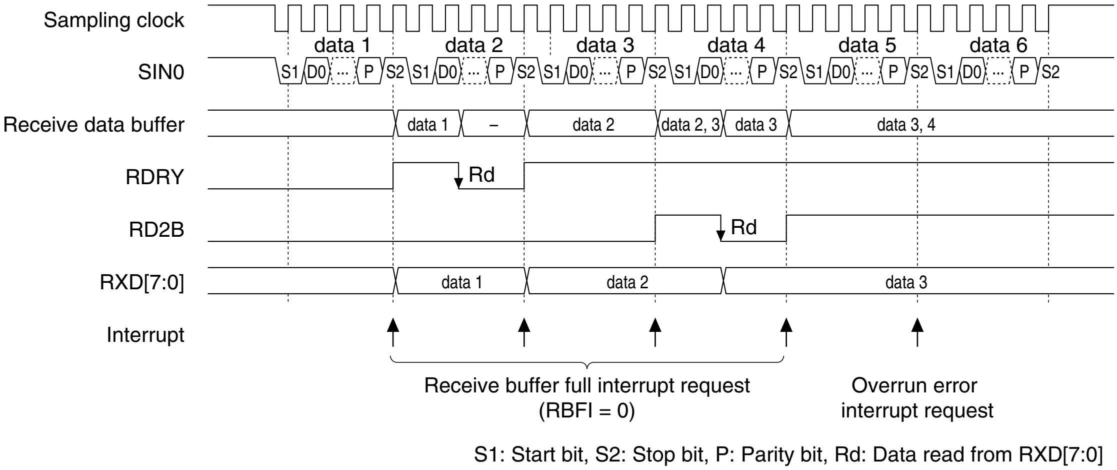


Figure V.1.5.2 Data Receive Timing Chart

Disabling data transmission/reception

After data transfer (both transmission and reception) has finished, write 0 to the RXEN bit to disable data transmission/reception.

Always make sure that the TDBE flag is 1 and TRBS and RDRY flags are 0 before data transmission/reception is disabled.

When the RXEN bit is set to 0, the transmit and receive data buffers are placed in empty status (data is cleared if any remains). Furthermore, the data being transferred cannot be guaranteed if RXEN is set to 0 during transmitting/receiving.

V.1.6 Receive Errors

Three types of receive errors can be detected in data reception.

The receive errors are causes of interrupt, so the error can be processed in the interrupt handler routine. Refer to Section V.1.7 for controlling the UART interrupts.

Parity error

If the PREN bit (D3/UART_MOD register) is set to 1 (parity enabled), the parity bit is checked when data is received.

This parity check is performed when the data received in the shift register is loaded to the receive data buffer in order to check conformity with the PMD bit (D2/UART_MOD register) setting (odd or even parity).

If any nonconformity is found in this check, a parity error is assumed and the parity error flag PER (D5/UART_ST register) is set to 1.

Even when this error occurs, the received data in error is loaded to the receive data buffer and the receive operation is continued. However, the received data in which a parity error has occurred cannot be guaranteed.

The PER flag (D5/UART_ST register) is reset to 0 by writing 1.

∗ **PREN**: Parity Enable Bit in the UART Mode (UART_MOD) Register (D3/0x4103)
∗ **PMD**: Parity Mode Select Bit in the UART Mode (UART_MOD) Register (D2/0x4103)
∗ **PER**: Parity Error Flag Bit in the UART Status (UART_ST) Register (D5/0x4100)

Framing error

If data with a stop bit = 0 is received, the UART assumes that the data is out of sync and generates a framing error.

If two stop bits are used, only the first stop bit is checked.

When this error occurs, the framing-error flag FER (D6/UART_ST register) is set to 1.

Even when this error occurs, the received data in error is loaded to the receive data buffer and the receive operation is continued. However, the received data in which a framing error has occurred cannot be guaranteed, even if no framing error is found in the following data received.

The FER flag (D6/UART_ST register) is reset to 0 by writing 1.

∗ **FER**: Framing Error Flag Bit in the UART Status (UART_ST) Register (D6/0x4100)

Overrun error

Even when the receive data buffer is full (two data have been received), the next (third) data can be received into the shift register. If there is no space in the buffer (data has not been read) when the third data has been received, the third data in the shift register cannot be transferred to the buffer and an overrun error occurs.

When an overrun error occurs, the overrun error flag OER (D4/UART_ST register) is set to 1.

Even when this error occurs, the receive operation is continued.

The OER flag (D4/UART_ST register) is reset to 0 by writing 1.

∗ **OER**: Overrun Error Flag Bit in the UART Status (UART_ST) Register (D4/0x4100)

V.1.7 UART Interrupt

The UART can generate the following three types of interrupts:

- Transmit buffer empty interrupt
- Receive buffer full interrupt
- Receive error interrupt

The UART has one interrupt signal to be output to the interrupt controller (ITC) and it is shared with all three causes of interrupt. To determine the cause of interrupt that has occurred, read the status and error flags.

Transmit buffer empty interrupt

Set the TIEN bit (D4/UART_CTL register) to 1 when using this interrupt. If TIEN is set to 0 (default), an interrupt request by this cause will not be sent to the ITC.

∗ **TIEN**: Transmit Buffer Empty Interrupt Enable Bit in the UART Control (UART_CTL) Register (D4/0x4104)

When the transmit data set in the transmit data buffer is transferred to the shift register, the UART sets the TDBE bit (D0/UART_ST register) to 1 to indicate that the transmit data buffer is empty. At the same time, the UART outputs an interrupt request pulse to the ITC if the transmit buffer empty interrupt has been enabled (TIEN = 1).

∗ **TDBE**: Transmit Data Buffer Empty Flag Bit in the UART Status (UART_ST) Register (D0/0x4100)

If other interrupt conditions are satisfied, an interrupt is generated.
The UART interrupt handler routine should read the TDBE flag to check if the interrupt has occurred due to a transmit buffer empty or another cause. When TDBE = 1, the UART interrupt handler routine can write the next transmit data to the transmit data buffer.

Receive buffer full interrupt

Set the RIEN bit (D5/UART_CTL register) to 1 when using this interrupt. If RIEN is set to 0 (default), an interrupt request by this cause will not be sent to the ITC.

∗ **RIEN**: Receive Buffer Full Interrupt Enable Bit in the UART Control (UART_CTL) Register (D5/0x4104)

When the specified number of received data is loaded to the receive data buffer, the UART outputs an interrupt request pulse to the ITC if the receive buffer full interrupt has been enabled (RIEN = 1). If the RBFI bit (D1/UART_CTL register) is 0, an interrupt request pulse is output when received data is loaded to the receive data buffer (when the RDRY flag (D1/UART_ST register) goes 1). If the RBFI bit (D1/UART_CTL register) is 1, an interrupt request pulse is output when two received data occupy the receive data buffer (when the RD2B flag (D3/UART_ST register) goes 1).

∗ **RBFI**: Receive Buffer Full Interrupt Condition Bit in the UART Control (UART_CTL) Register (D1/0x4104)
∗ **RDRY**: Receive Data Ready Flag Bit in the UART Status (UART_ST) Register (D1/0x4100)
∗ **RD2B**: Second Byte Receive Flag Bit in the UART Status (UART_ST) Register (D3/0x4100)

If other interrupt conditions are satisfied, an interrupt is generated.
The UART interrupt handler routine should read the RDRY and RD2B flags to check if the interrupt has occurred due to a receive buffer full or another cause. When RDRY or RD2B = 1, the UART interrupt handler routine can read the received data from the receive data buffer.

Receive error interrupt

Set the REIEN bit (D6/UART_CTL register) to 1 when using this interrupt. If REIEN is set to 0 (default), an interrupt request by this cause will not be sent to the ITC.

* **REIEN**: Receive Error Interrupt Enable Bit in the UART Control (UART_CTL) Register (D6/0x4104)

When a parity, framing, or overrun error is detected during data reception, the UART sets the error flag listed below to 1 and outputs an interrupt request pulse to the ITC if the receive error interrupt has been enabled (REIEN = 1).

* **PER**: Parity Error Flag Bit in the UART Status (UART_ST) Register (D5/0x4100)
* **FER**: Framing Error Flag Bit in the UART Status (UART_ST) Register (D6/0x4100)
* **OER**: Overrun Error Flag Bit in the UART Status (UART_ST) Register (D4/0x4100)

If other interrupt conditions are satisfied, an interrupt is generated.
The UART interrupt handler routine should read the error flags to check if the interrupt has occurred due to a receive error or another cause. When an error flag has been set to 1, the UART interrupt handler routine should execute an error recovery process.

ITC registers for UART interrupts

The following shows the control bits of the ITC provided for the UART:

Interrupt flag

* **IIFT4**: UART Interrupt Flag Bit in the Interrupt Flag (ITC_IFLG) Register (D12/0x4300)

Interrupt enable bits

* **IIEN4**: UART Interrupt Enable Bit in the Interrupt Enable (ITC_EN) Register (D12/0x4302)

Interrupt level setup bits

* **IILV4[2:0]**: UART Interrupt Level Bits in the Internal Interrupt Level Setup (ITC_ILV2) Register 2 (D[2:0]/0x4312)

When the UART outputs an interrupt request pulse, the corresponding interrupt flag is set to 1.
If the interrupt enable bit corresponding to that interrupt flag has been set to 1, the ITC sends an interrupt request to the S1C17 Core. To disable the UART interrupt, set the interrupt enable bit to 0.
The interrupt flag is always set to 1 by the UART interrupt request pulse, regardless of how the interrupt enable register is set (even when set to 0).
The interrupt level setup bits set the interrupt level (0 to 7) of the UART interrupt.
An interrupt request to the S1C17 Core is accepted only when all the conditions described below are met.

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

For details on these interrupt control registers, as well as the device operation when an interrupt has occurred, see Section III.1, "Interrupt Controller (ITC)."

Interrupt vector

The following shows the vector number and vector address for the UART interrupt:

Vector number: 16 (0x10)
Vector address: TTBR + 0x40

V.1.8 IrDA Interface

The UART module contains an RZI modulator/demodulator, allowing an infrared-ray communication circuit to be configured based on IrDA 1.0 simply by adding an external circuit.

The transmit data output from the shift register of the UART is input to the modulator to convert the low pulse width into 3/16 sclk cycles before it is output from the SOUT0 pin.

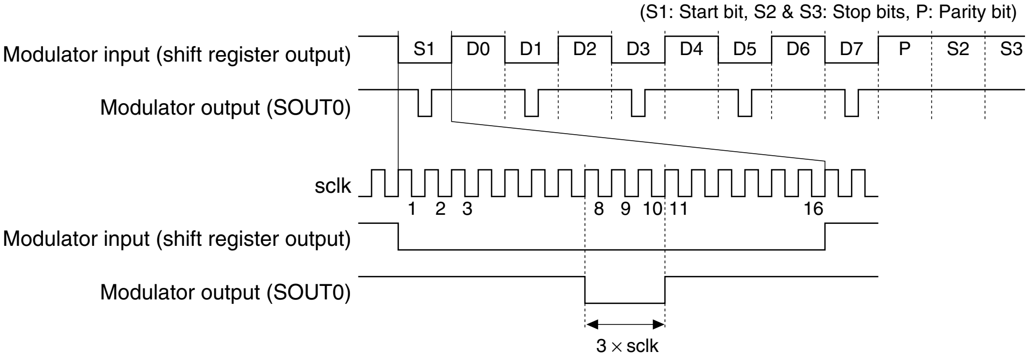

(S1: Start bit, S2 & S3: Stop bits, P: Parity bit)
Modulator input (shift register output)
S1
D0
D1
D2
D3
D4
D5
D6
D7
P
S2
S3
Modulator output (SOUT0)
sclk
1 2 3
8 9 10 11
16
Modulator input (shift register output)
Modulator output (SOUT0)
3 × sclk

Figure V.1.8.1 Transmit Signal Waveform

The received IrDA signal is input to the demodulator to convert the low pulse width into 16 sclk cycles before input to the shift register for receiving. To detect low pulses input to the demodulator (minimum pulse width = 1.41 μs at 115200 bps), the demodulator uses a pulse detection clock selected from the prescaler output clocks separately with the transfer clock sclk.

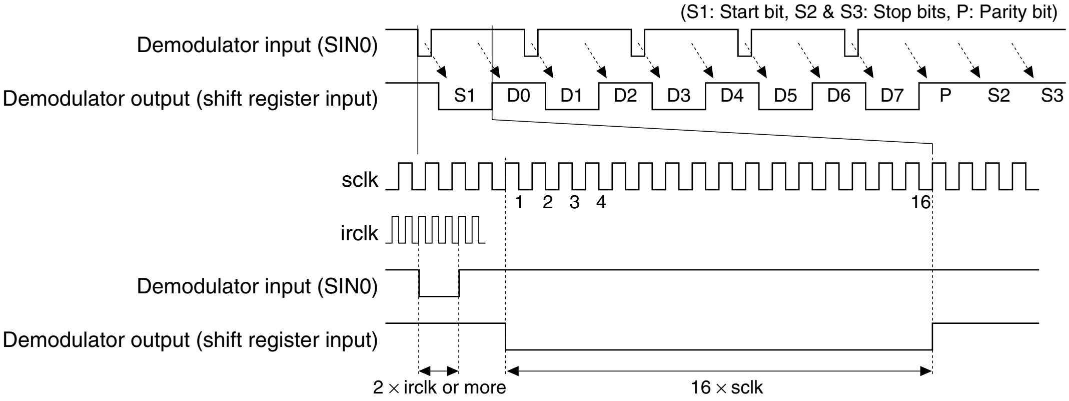

(S1: Start bit, S2 & S3: Stop bits, P: Parity bit)
Demodulator input (SIN0)
Demodulator output (shift register input)
S1
D0
D1
D2
D3
D4
D5
D6
D7
P
S2
S3
sclk
1 2 3 4
16
irclk
Demodulator input (SIN0)
Demodulator output (shift register input)
2 × irclk or more
16 × sclk

Figure V.1.8.2 Receive Signal Waveform

Enabling the IrDA mode

To use the IrDA interface function, set the IRMD bit (D0/UART_EXP register) to 1. This enables the RZI modulator/demodulator.

∗ **IRMD**: IrDA Mode Select Bit in the UART Expansion (UART_EXP) Register (D0/0x4105)

Note: This setting must be performed before setting other UART conditions.

Selecting the IrDA receive detection clock

Select a prescaler output clock within the range from PCLK•1/1 to PCLK•1/128 as the input pulse detection clock using the IRCLK[2:0] bits (D[6:4]/UART_EXP register).

∗ **IRCLK[2:0]:** IrDA Receive Detection Clock Select Bits in the UART Expansion (UART_EXP) Register (D[6:4]/0x4105)

Table V.1.8.1 Selecting the IrDA Receive Detection Clock

IRCLK[2:0]	Prescaler output clock
0x7	PCLK•1/128
0x6	PCLK•1/64
0x5	PCLK•1/32
0x4	PCLK•1/16
0x3	PCLK•1/8
0x2	PCLK•1/4
0x1	PCLK•1/2
0x0	PCLK•1/1

(Default: 0x0)

This clock must be faster than the transfer clock sclk supplied from the CLG_T16U0/CLG_T8FU0 timer or input from the #SCLK0 pin.

The demodulator regards a low pulse of which the width is longer than two cycles of the IrDA receive detection clock as a valid low pulse and converts it to a 16 sclk cycles width of low pulse. Select an appropriate prescaler output clock that can detect a minimum 1.41 µs width of an input pulse.

Controlling serial data transfer

The control method to transmit/receive data in IrDA mode is the same as that of the normal interface. See previous sections for details on how to set and control the data formats, data transfers, and interrupts.

V.1.9 Details of Control Registers

Table V.1.9.1 List of UART Registers

Address	Register name		Function
0x4100	UART_ST	UART Status Register	Indicates transfer, buffer and error statuses.
0x4101	UART_TXD	UART Transmit Data Register	Transmit data
0x4102	UART_RXD	UART Receive Data Register	Receive data
0x4103	UART_MOD	UART Mode Register	Sets transfer data format.
0x4104	UART_CTL	UART Control Register	Controls data transfer.
0x4105	UART_EXP	UART Expansion Register	Sets IrDA mode.

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

Note: When setting the registers, be sure to write a 0, and not a 1, for all “reserved bits.”

0x4100: UART Status Register (UART_ST)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
UART Status Register (UART_ST)	**0x4100** (8 bits)	D7	–	reserved	–				–	–	0 when being read.
		D6	**FER**	Framing error flag	1	Error	0	Normal	0	R/W	Reset by writing 1.
		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 Reserved

D6 FER: Framing Error Flag Bit

Indicates whether a framing error has occurred or not.

1 (R): An error has occurred
0 (R): No error has occurred (default)
1 (W): Reset to 0
0 (W): Has no effect

When a framing error has occurred, FER is set to 1. A framing error occurs when data with a stop bit = 0 is received.

FER is reset by writing 1 or when RXEN (D0/UART_CTL register) is set to 0.

D5 PER: Parity Error Flag Bit

Indicates whether a parity error has occurred or not.

1 (R): An error has occurred
0 (R): No error has occurred (default)
1 (W): Reset to 0
0 (W): Has no effect

When a parity error has occurred, PER is set to 1. The parity check function is effective only when PREN (D3/UART_MOD register) is set to 1. This check is performed when the received data is transferred from the shift register to the receive data buffer.

PER is reset by writing 1 or when RXEN (D0/UART_CTL register) is set to 0.

D4 OER: Overrun Error Flag Bit

Indicates whether an overrun error has occurred or not.

1 (R): An error has occurred
0 (R): No error has occurred (default)
1 (W): Reset to 0
0 (W): Has no effect

When an overrun error has occurred, OER is set to 1. An overrun error will occur if new data is received when the receive data buffer is full and also if the shift register contains received data. When this error occurs, the shift register is overwritten with the new received data. The receive data in the buffer is left unchanged.

OER is reset by writing 1 or when RXEN (D0/UART_CTL register) is set to 0.

D3 RD2B: Second Byte Receive Flag Bit

Indicates that the receive data buffer contains two received data.

1 (R): Second byte is ready to read out
0 (R): Second entry is empty (default)

RD2B is set to 1 when the second data is loaded to the receive data buffer, and is reset to 0 when the first data is read out from the receive data buffer.

D2 TRBS: Transmit Busy Flag Bit

Indicates the transmit shift register status.

1 (R):	Busy
0 (R):	Idle (default)

TRBS goes 1 when transmit data is loaded to the shift register from the transmit data buffer and returns to 0 upon completion of a data transmission. Read this flag to check whether the transmitter is busy or idle.

D1 RDRY: Receive Data Ready Flag Bit

Indicates that the receive data buffer contains valid received data.

1 (R):	Data is ready to read out
0 (R):	Buffer is empty (default)

RDRY is set to 1 when received data is loaded to the receive data buffer, and is reset to 0 when all data are read out from the receive data buffer.

D0 TDBE: Transmit Data Buffer Empty Flag Bit

Indicates the status of the transmit data buffer.

1 (R):	Empty (default)
0 (R):	Not empty

TDBE is reset to 0 when transmit data is written to the transmit data buffer and set to 1 when the transmit data in the buffer is transferred to the shift register.

0x4101: UART Transmit Data Register (UART_TXD)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
UART Transmit Data Register (UART_TXD)	**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 Bits

Write transmit data to be set to the transmit data buffer. (Default: 0x0)

When data is written to this register, the UART starts transmitting. The data written to TXD[7:0] enters the transmit data buffer and waits for transmission. When the data in the transmit data buffer is transferred, a cause of transmit buffer empty interrupt occurs.

In 7-bit mode, TXD7 (MSB) is ignored.

The serial-converted data is output from the SOUT0 pin beginning with the LSB, in which the bits set to 1 are output as high-level signals and those set to 0 output as low-level signals.

This register can be read as well as written.

0x4102: UART Receive Data Register (UART_RXD)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
UART Receive Data Register (UART_RXD)	**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 in the Receive Data Buffer Bits

The data in the receive data buffer can be read from this register beginning with the older data first. The received data enters the receive data buffer. The receive data buffer is a two-byte FIFO and can receive data until it becomes full. When the buffer is full and also if the shift register contains received data, an overrun error will occur if the received data is not read by the time the next data receiving begins. The receive data buffer status flags RDRY (D1/UART_ST register) and RD2B (D3/UART_ST register) are provided to indicate that the receive data buffer contains valid received data and the second data, respectively.

When the receive data buffer has received the number of data specified with RBFI (D1/UART_CTL register), a cause of receive buffer full interrupt occurs.

In 7-bit mode, 0 is stored in RXD7.

The serial data input from the SIN0 pin is converted into parallel data beginning with the LSB, with the high-level signals changed to 1s and the low-level signals changed to 0s. The resulting data is stored in the receive data buffer.

This register is a read-only register, so no data can be written to it. (Default: 0x0)

0x4103: UART Mode Register (UART_MOD)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
UART Mode Register (UART_MOD)	**0x4103** (8 bits)	D7–6	–	reserved	–				–	–	0 when being read.
		D5	**TMSEL**	Timer select	1	CLG_T8FU0	0	CLG_T16U0	0	R/W	
		D4	**CHLN**	Character length	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	**SSCK**	Input clock select	1	External	0	Internal	0	R/W	

D[7:6] Reserved

D5 TMSEL: Timer Select Bit

Selects the timer to be used as the internal clock source.

1 (R/W): CLG_T8FU0 timer
0 (R/W): CLG_T16U0 timer (default)

Select the timer for generating the UART input clock when internal clock is selected with SSCK (D0). The CLG_T8FU0 timer is selected by writing 1 to this bit, and the CLG_T16U0 timer is selected by writing 0.

D4 CHLN: Character Length Bit

Selects the character length of serial transfer data.

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 the parity check for receive data will be performed or not, and whether a parity bit will be added to transmit data. When PREN is set to 1, the received data is checked for parity. A parity bit is automatically added to the transmit data. When PREN is set to 0, parity is not checked and no parity bit is added.

D2 PMD: Parity Mode Select Bit

Selects the parity mode.

1 (R/W): Odd parity
0 (R/W): Even parity (default)

Odd parity is selected by writing 1 to PMD, and even parity is selected by writing 0. Parity check and the addition of a parity bit are effective only when PREN (D3) is set to 1. If PREN (D3) = 0, settings of PMD do not have any effect.

D1 STPB: Stop Bit Select Bit

Selects a stop bit length.

1 (R/W): 2 bits
0 (R/W): 1 bit (default)

Two stop bits are selected by writing 1 to STPB, and one stop bit is selected by writing 0. The start bit is fixed at 1 bit.

D0 SSCK: Input Clock Select Bit

Selects the clock source.

1 (R/W): External clock (#SCLK0 pin)
0 (R/W): Internal clock (default)

This bit is used to select the clock source between the internal clock (CLG_T16U0/CLG_T8FU0 timer output clock) and an external clock (input from the #SCLK0 pin). An external clock is selected by writing 1 to this bit, and an internal clock is selected by writing 0.

0x4104: UART Control Register (UART_CTL)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
UART Control Register (UART_CTL)	**0x4104** (8 bits)	D7	–	reserved	–				–	–	0 when being read.
		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	1	2 bytes	0	1 byte	0	R/W	
		D0	**RXEN**	UART enable	1	Enable	0	Disable	0	R/W	

D7 Reserved

D6 REIEN: Receive Error Interrupt Enable Bit

Enables an interrupt request to be output to the ITC when a receive error has occurred.
1 (R/W): Enable
0 (R/W): Disable (default)

Set this bit to 1 when processing receive errors in the interrupt handler routine.

D5 RIEN: Receive Buffer Full Interrupt Enable Bit

Enables an interrupt request to be output to the ITC when the receive data buffer receives the number of data specified by RBFI (D1).
1 (R/W): Enable
0 (R/W): Disable (default)

Set this bit to 1 when reading the received data in the interrupt handler routine.

D4 TIEN: Transmit Buffer Empty Interrupt Enable Bit

Enables an interrupt request to be output to the ITC when the transmit data written to the transmit data buffer is transferred to the shift register (when data transmission starts).
1 (R/W): Enable
0 (R/W): Disable (default)

Set this bit to 1 when writing transmit data to the transmit data buffer in the interrupt handler routine.

D[3:2] Reserved

D1 RBFI: Receive Buffer Full Interrupt Condition Bit

Sets the number 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)

When the specified number of received data is loaded to the receive data buffer, the UART outputs an interrupt request pulse to the ITC if the receive buffer full interrupt has been enabled (RIEN = 1). If RBFI is 0, an interrupt request pulse is output when a received data is loaded to the receive data buffer (when the RDRY flag (D1/UART_ST register) goes 1). If RBFI is 1, an interrupt request pulse is output when two received data occupy the receive data buffer (when the RD2B flag (D3/UART_ST register) goes 1).

D0 RXEN: UART Enable Bit

Enables the UART to transmit/receive data.
1 (R/W): Enable
0 (R/W): Disable (default)

Before the UART can transmit/receive data, RXEN must be set to 1. When RXEN is set to 0, data transmission/reception is disabled.
Always make sure RXEN = 0 before setting the transfer conditions.
Writing 0 to RXEN also clears the transmit/receive data buffers.

0x4105: UART Expansion Register (UART_EXP)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
UART Expansion Register (UART_EXP)	**0x4105** (8 bits)	D7	–	reserved	–		–	–	0 when being read.
		D6–4	**IRCLK[2:0]**	IrDA receive detection clock select	IRCLK[2:0] 0x7 0x6 0x5 0x4 0x3 0x2 0x1 0x0	Clock PCLK•1/128 PCLK•1/64 PCLK•1/32 PCLK•1/16 PCLK•1/8 PCLK•1/4 PCLK•1/2 PCLK•1/1	0x0	R/W	
		D3–1	–	reserved	–		–	–	0 when being read.
		D0	**IRMD**	IrDA mode select	1 On	0 Off	0	R/W	

D7 Reserved

D[6:4] IRCLK[2:0]: IrDA Receive Detection Clock Select Bits

These bits select a prescaler output clock as the input pulse detection clock.

Table V.1.9.2 Selecting the IrDA Receive Detection Clock

IRCLK[2:0]	Prescaler output clock
0x7	PCLK•1/128
0x6	PCLK•1/64
0x5	PCLK•1/32
0x4	PCLK•1/16
0x3	PCLK•1/8
0x2	PCLK•1/4
0x1	PCLK•1/2
0x0	PCLK•1/1

(Default: 0x0)

This clock must be faster than the transfer clock #SCLK0 supplied from the CLG_T16U0/CLG_T8FU0 timer or input from the #SCLK0 pin.

The demodulator regards a low pulse of which the width is longer than two cycles of the IrDA receive detection clock as a valid low pulse. Select an appropriate prescaler output clock that can detect a minimum 1.41 µs width of an input pulse.

D[3:1] Reserved

D0 IRMD: IrDA Mode Select Bit

Turns the IrDA interface function on and off.

1 (R/W): On

0 (R/W): Off (default)

Set this bit to 1 when using the IrDA interface. When set to 0, the module functions as a standard UART without IrDA.

V.1.10 Precautions

- Before setting the bits listed below, make sure the transmit and receive operations are disabled (RXEN = 0).
  - All bits (SSCK, STPB, PMD, PREN, CHLN, and TMSEL) of the UART_MOD register (0x4103)
  - All bits (RBFI, TIEN, RIEN, and REIEN except RXEN) of the UART_CTL register (0x4104)
  - All bits (IRMD and IRCLK[2:0]) of the UART_EXP register (0x4105)

  ∗ **RXEN**: UART Enable Bit in the UART Control (UART_CTL) Register (D0/0x4104)
- When the UART is transmitting or receiving data, do not set RXEN to 0.
- The maximum transfer rate of the UART is limited to 460800 bps (115200 bps in IrDA mode). Do not set a transfer rate that exceeds the limit.
- When the RXEN bit is set to 0 to disable transmit/receive operations, the transmit/receive data buffers are cleared (initialized). Therefore, make sure that the buffers do not contain any data waiting for transmission or reading before writing 0 to the RXEN bit.
- The IrDA receive detection clock must be faster than the transfer clock sclk supplied from the CLG_T16U0/ CLG_T8FU0 timer or input from the #SCLK0 pin.
- The demodulator regards a low pulse of which the width is longer than two cycles of the IrDA receive detection clock as a valid low pulse. Select an appropriate prescaler output clock as the IrDA receive detection clock so that it will be able to detect a minimum 1.41 µs width of an input pulse.

THIS PAGE IS BLANK.

V.2 I2C Master

V.2.1 Configuration of the I2C Master Module

The S1C17002 equipped with an I2C master module for high-speed synchronous serial communication. This I2C master module operates as an I2C bus master using the clock supplied from the CLG_T8I timer (supports single master mode only). It supports standard (100 kbps) and fast (400 kbps) modes, and 7-bit/10-bit slave addressing. The I2C module includes a noise remove function to secure reliable data transfer.

Also it can generate two types of interrupts (transmit buffer empty and receive buffer full interrupts), this makes it possible to process continuous serial data transfer simply in an interrupt handler.

Figure V.2.1.1 shows the structure of the I2C master module.

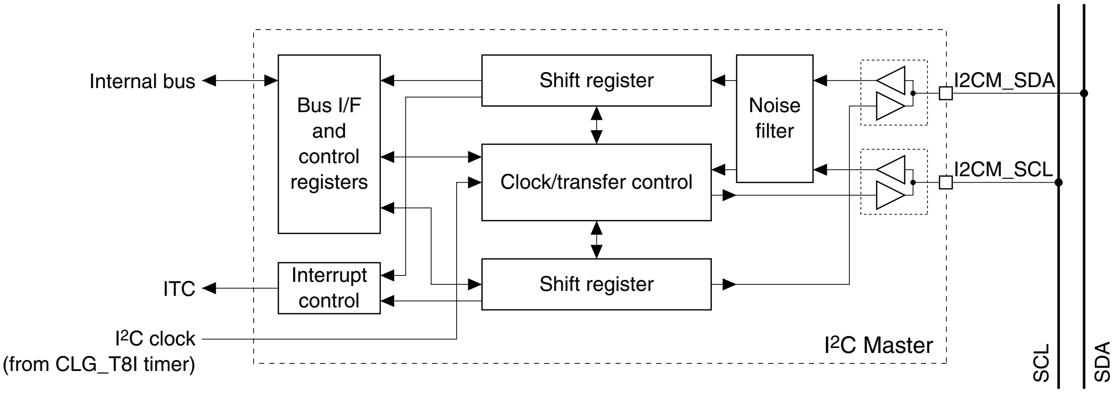


Figure V.2.1.1 Structure of I2C Master Module

Note: The I2C master module does not have a clock stretch function. Therefore, it does not support I2C slave devices that use clock stretch for synchronization of data communication.

V.2.2 I^{2}C Master I/O Pins

Table V.2.2.1 lists the I^{2}C Master pins.

Table V.2.2.1 List of I^{2}C Master Pins

Pin name	I/O	Size	Function
I2CM_SDA	I/O	1	I^{2}C master data input/output pin This pin inputs serial data from the I^{2}C bus and outputs serial data to the I^{2}C bus.
I2CM_SCL	I/O	1	I^{2}C master clock input/output pin This pin inputs the SCL line status and outputs the serial clock to the I^{2}C bus.

The I^{2}C master input/output pins (I2CM_SDA, I2CM_SCL) are shared with the I/O ports and they are initialized as general-purpose I/O port pins by default. Before using these pins for the I^{2}C master, the pin functions must be switched using the Port Function Select Register.
For details on switching pin function, Section I.3.3, “Switching Over the Multiplexed Pin Functions.”

V.2.3 I^{2}C Master Clock

The I^{2}C master module uses the internal clock output from the CLG_T8I timer as the synchronous clock. This clock drives the shift register and is output from the I2CM_SCL pin to the slave I^{2}C device.
Program the CLG_T8I timer so that it will output a clock according to the transfer rate. Refer to Section II.4, "Clock Generator (CLG)," for controlling the CLG_T8I timer.

The I^{2}C master module does not function as a slave device. The I2CM_SCL input is used to check the I2CM_SCL status of the I^{2}C bus but it is not used to input synchronous clock.

V.2.4 Setting before Starting Data Transfer

The I^{2}C master module has a noise remove function selectable in the application program.

Noise remove function

The I^{2}C master module contains a function to remove noise from the I2CM_SDA and I2CM_SCL input signals. This function is enabled by setting NSERM (D4/I2CM_CTL register) to 1.
Note, however, that the I^{2}C master clock (CLG_T8I timer output clock) frequency must be a 1/6 of PCLK or lower to use the noise remove function.

∗ **NSERM**: Noise Remove On/Off Bit in the I^{2}C Master Control (I2CM_CTL) Register (D4/0x4342)

V.2.5 Data Transmit/Receive Control

Before starting data transfer, set up the conditions by the procedure below.

(1) Set up the CLG_T8I timer to output the I^{2}C master clock. See Section II.4.

(2) Select optional functions. See Section V.2.4.

(3) Set up the interrupt conditions if the I^{2}C master interrupt is used. See Section V.2.6.

Note: Make sure that the I^{2}C module is disabled (I2CMEN/I2CM_EN register = 0) before setting the conditions above.

∗ **I2CMEN**: I^{2}C Master Enable Bit in the I^{2}C Master Enable (I2CM_EN) Register (D0/0x4340)

Enabling data transmission/reception

First, set the I2CMEN bit (D0/I2CM_EN register) to 1 to enable I^{2}C master operation. This makes the I^{2}C master in ready-to-transmit/receive status and enables clock output.

Note: Do not set the I2CMEN bit to 0 while the I^{2}C master module is transmitting/receiving data.

Starting data transmission/reception

To start data transmission, the I^{2}C master (this module) must generate a START condition, and then send a slave address to establish the communication.

(1) Register setting procedure

To generate a START condition, set the following registers in the order shown below:

1. Set the slave address to RTDT[7:0] (D[7:0]/I2CM_DAT register).
2. Set TXE (D9/I2CM_DAT register) to 1.
3. Set STRT (D0/I2CM_CTL register) to 1.

∗ **RTDT[7:0]**: Receive/Transmit Data Bits in the I^{2}C Master Data (I2CM_DAT) Register (D[7:0]/0x4344)
∗ **TXE**: Transmit Execution Bit in the I^{2}C Master Data (I2CM_DAT) Register (D9/0x4344)
∗ **STRT**: Start Control Bit in the I^{2}C Master Control (I2CM_CTL) Register (D0/0x4342)

This procedure generates the communication waveforms as shown in Items (2) and (3) below. Be sure to follow the register setting procedure.

(2) Generating a START condition

The START condition is a state in which the I2CM_SDA line is pulled down to low with the I2CM_SCL line held at high.

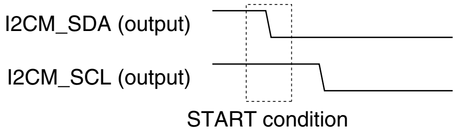


Figure V.2.5.1 START Condition

Set the STRT bit (D0/I2CM_CTL register) to 1 to generate a START condition.

∗ **STRT**: Start Control Bit in the I^{2}C Master Control (I2CM_CTL) Register (D0/0x4342)

After a START condition has been generated, STRT is automatically reset to 0.

(3) Sending a slave address

Once the start condition has been generated, the I^{2}C master (this module) sends a bit indicating the slave address and transfer direction for communications. I^{2}C 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.

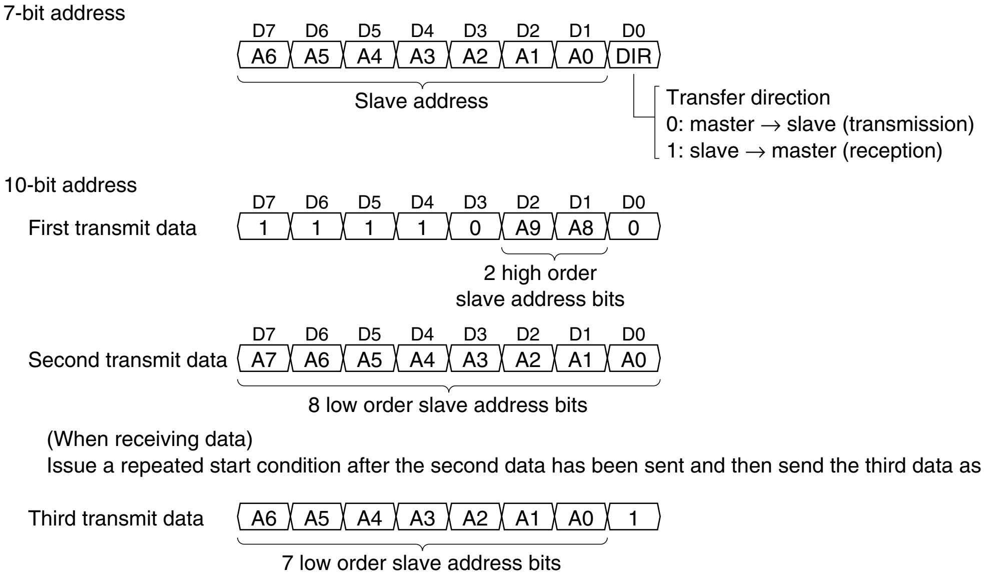


Figure V.2.5.2 Transmit Data to Specify Slave Address and Data Direction

The transfer direction bit specifies the direction for the data transfer that follows the slave address transfer. Set the transfer direction bit to 0 when transmitting data from the master to the slave; set it to 1 when receiving data from the slave.

Configure an 8-bit data as above and set it into the transmit/receive data register. After that control data transmission as described below.

The slave address with a transfer direction bit must be sent once after a START condition has been generated.

After a slave address has been sent, perform data transmission or data reception as many times as necessary. It is necessary to perform data transmission or data reception according to the transfer direction specified with the slave address.

Data transmit control

The following explains how to transmit data. The slave address should be sent in the same way.

To transmit byte data, set the data to the RTDT[7:0] bits (D[7:0]/I2CM_DAT register). At the same time, set the TXE bit (D9/I2CM_DAT register) to 1 to execute one byte data transmission.

∗ **RTDT[7:0]**: Receive/Transmit Data Bits in the I^{2}C Master Data (I2CM_DAT) Register (D[7:0]/0x4344)
∗ **TXE**: Transmit Execution Bit in the I^{2}C Master Data (I2CM_DAT) Register (D9/0x4344)

When the TXE bit is set to 1, the I^{2}C master module starts data transmission in sync with the clock. If a START condition is being generated or the previous data is being transferred, the I^{2}C master module starts data transmission after waiting for completion of the process.

First, the I^{2}C master module transfer the written data to the shift register and starts outputting the clock from the I2CM_SCL pin. At this time, TXE is reset to 0 and a cause of interrupt occurs. This allows the program to set the next transmit data and TXE again.

The data bits in the shift register are shifted one by one at the falling edge of the clock and are output from the I2CM_SDA pin. The MSB is transmitted first.

The I^{2}C master module outputs nine clocks for one data transmission. In the ninth clock cycle, the I^{2}C module sets the I2CM_SDA signal into high-impedance status to input an ACK or NAK bit from the slave.

If the slave could receive byte data, it returns an ACK (0) bit to the master. If the slave could not receive byte data, the I2CM_SDA line is not pulled down. The I^{2}C module regards this status as a NAK (1) returned (transmission fails).

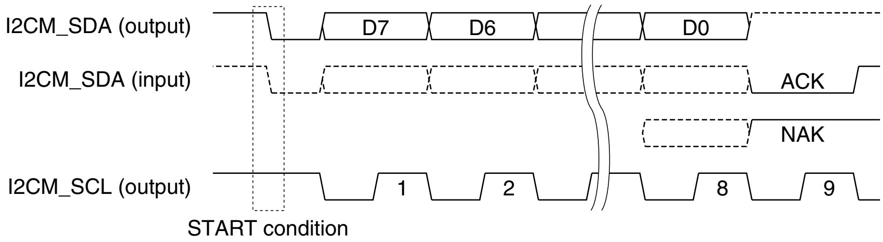


Figure V.2.5.3 ACK and NAK

The I2C master module provides two status bits for data transmit control, TBUSY flag (D8/I2CM_CTL register) and RTACK bit (D8/I2CM_DAT register).

∗ **TBUSY**: Transmit Busy Flag Bit in the I2C Master Control (I2CM_CTL) Register (D8/0x4342)
∗ **RTACK**: Receive/Transmit ACK Bit in the I2C Master Data (I2CM_DAT) Register (D8/0x4344)

The TBUSY flag indicates the data transmit status; it goes 1 when data transmission (including slave address transmission) starts and returns to 0 upon completion of data transmission. Also TBUSY returns to 0 in wait state. Read this flag to check whether the I2C master module is busy or idle.

The RTACK bit indicates whether the slave returned ACK or not in a previous data transmission; it goes 0 when an ACK bit is received or goes 1 if an ACK bit is not received.

Data receive control

The following explains how to receive data. Even in data reception, a START condition must be generated and a slave address with the transfer direction bit set to 1 must be sent before starting data reception.

To receive byte data, set the RXE bit (D10/I2CM_DAT register) to 1 to execute receiving one byte data. RXE can be set to 1 at the same time when TXE (D9/I2CM_DAT register) is set to 1 for sending a slave address. If TXE and RXE are both set to 1, TXE is effective.

∗ **RXE**: Receive Execution Bit in the I2C Master Data (I2CM_DAT) Register (D10/0x4344)

When the RXE bit is set to 1 and the I2C master module is ready to receive, the I2C master module sets the I2CM_SDA line into high-impedance and starts outputting the clock from the I2CM_SCL pin. The data bits are fetched in the shift register one by one at the rising edge of the clock. The MSB is received first.

RXE is reset to 0 during fetching D6.

When eight data bits are received in the shift register, the received data is loaded into RTDT[7:0].

The I2C master module provides two status bits for data receive control, RBRDY (D11/I2CM_DAT register) and RBUSY (D9/I2CM_CTL register).

∗ **RBRDY**: Receive Buffer Ready Bit in the I2C Master Data (I2CM_DAT) Register (D11/0x4344)
∗ **RBUSY**: Receive Busy Flag Bit in the I2C Master Control (I2CM_CTL) Register (D9/0x4342)

The RBRDY flag indicates the received data status; it goes 1 when the received data in the shift register is loaded into RTDT[7:0] and returns to 0 when the received data is read from RTDT[7:0]. An interrupt can be generated when this flag goes 1. Use this interrupt or read the RBRDY flag to check that RTDT[7:0] contains valid data when reading received data. If the next data has been received before the previous received data in RTDT[7:0] is read, the previous received data is overwritten with the new data.

The RBUSY flag indicates the data receive operation status; it goes 1 when data reception starts and returns to 0 upon completion of data reception. Also RBUSY returns to 0 in wait state. Read this flag to check whether the I2C master module is busy or idle.

The I2C master module outputs nine clocks for one data transmission. In the ninth clock cycle, the I2C master module sends a response bit, ACK when RTACK (D8/I2CM_DAT register) is set to 0 or NAK when RTACK is set to 1, from the I2CM_SDA pin to the slave. Be sure to avoid altering RTACK while the I2C master is sending an ACK or NAK bit.

Note: A receive buffer full interrupt occurs during the ACK/NAK response period (ninth clock cycle). If RTACK is rewritten immediately after this interrupt has occurred without waiting the ACK/NAK period to be finished, the ACK/NAK signal may change and this may cause communication to fail.

Terminating data transmission/reception (generating a STOP condition)

To terminate data transfer after all data have been sent/received, the I2C master (this module) must generate a STOP condition. The STOP condition is a state in which the I2CM_SDA line is pulled up from low to high with the I2CM_SCL line held at high.

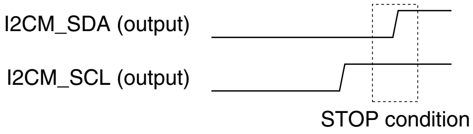


Figure V.2.5.4 STOP Condition

Set the STP bit (D1/I2CM_CTL register) to 1 to generate a STOP condition.

∗ **STP**: Stop Control Bit in the I2C Master Control (I2CM_CTL) Register (D1/0x4342)

When STP is set to 1, the I2C master module pulls up the I2CM_SDA line from low to high with the I2CM_SCL line held at high to generate a STOP condition on the I2C bus. This makes the I2C bus in free status.
Furthermore, the I2C master module allows presetting for generating a STOP condition in advance. To do this, set STP to 1 after checking if the I2C master is operating (TBUSY = 1 or RBUSY = 1). A STOP condition will be generated upon completion of data transmission/reception (including an ACK transfer).
STP is automatically reset to 0 after a STOP condition has been generated.

The I2C master module does not support repeated START condition. A STOP condition cannot be omitted before generating a new START condition for starting the next data transfer.

Wait state by setting TXE, RXE, STRT, and STP

If TXE (D9/I2CM_DAT register), RXE (D10/I2CM_DAT register), STRT (D0/I2CM_CTL register), and STP (D1/I2CM_CTL register) have all been set to 0 when byte data and ACK transfer have finished, the I2C master module fixes the I2CM_SCL output at low and enters wait state. The wait state will be canceled by writing 1 to TXE or RXE to resume data transmission/reception or by writing 1 to STP to generate a STOP condition.

Disabling data transmission/reception

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 I2CM_SCL and I2CM_SDA output levels and transfer data at that point cannot be guaranteed.

Timing charts

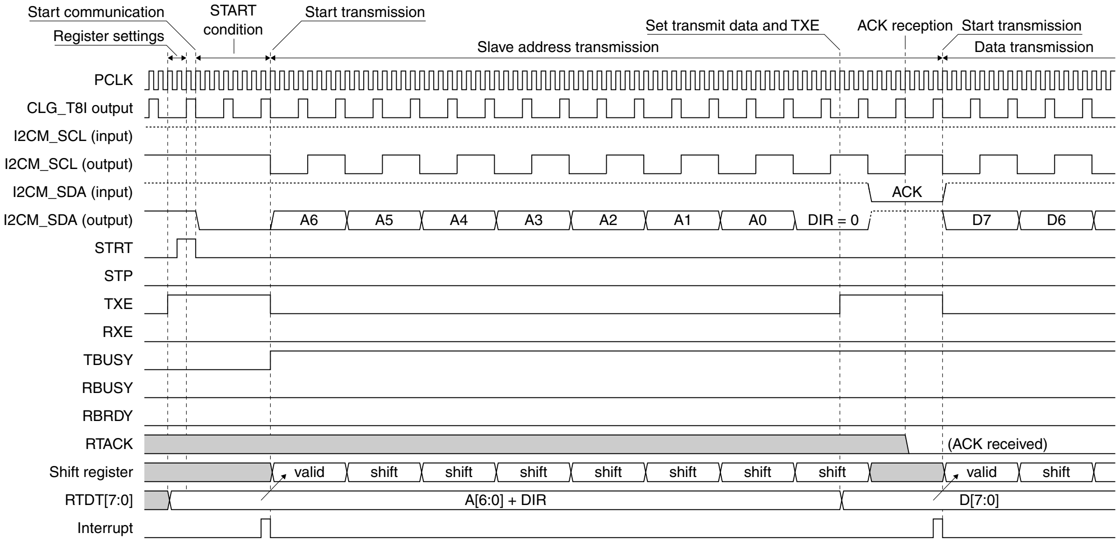


Figure V.2.5.5 I2C Master Timing Chart 1 (START condition → data transmission)

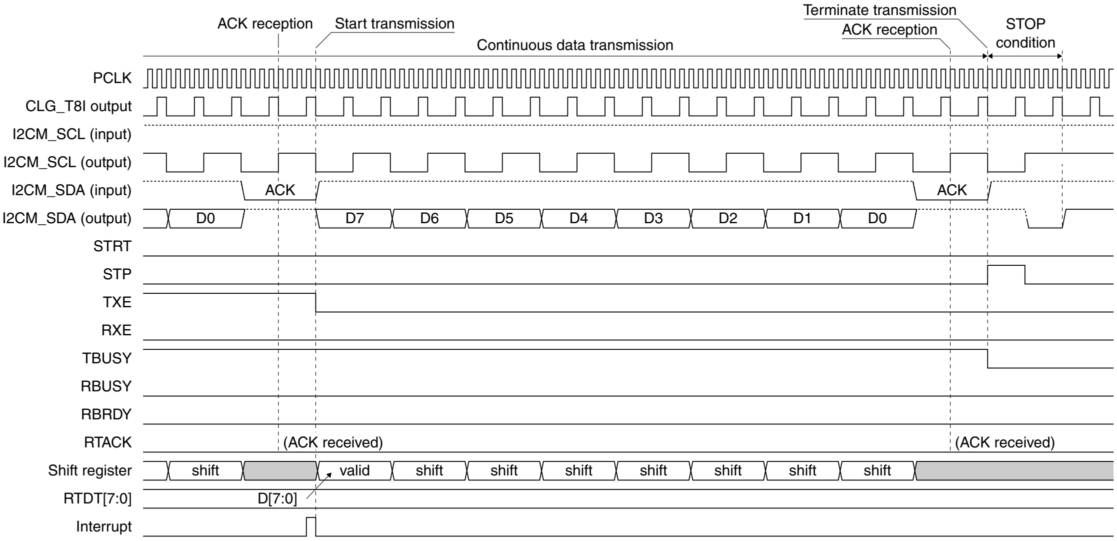


Figure V.2.5.6 I2C Master Timing Chart 2 (data transmission → STOP condition)

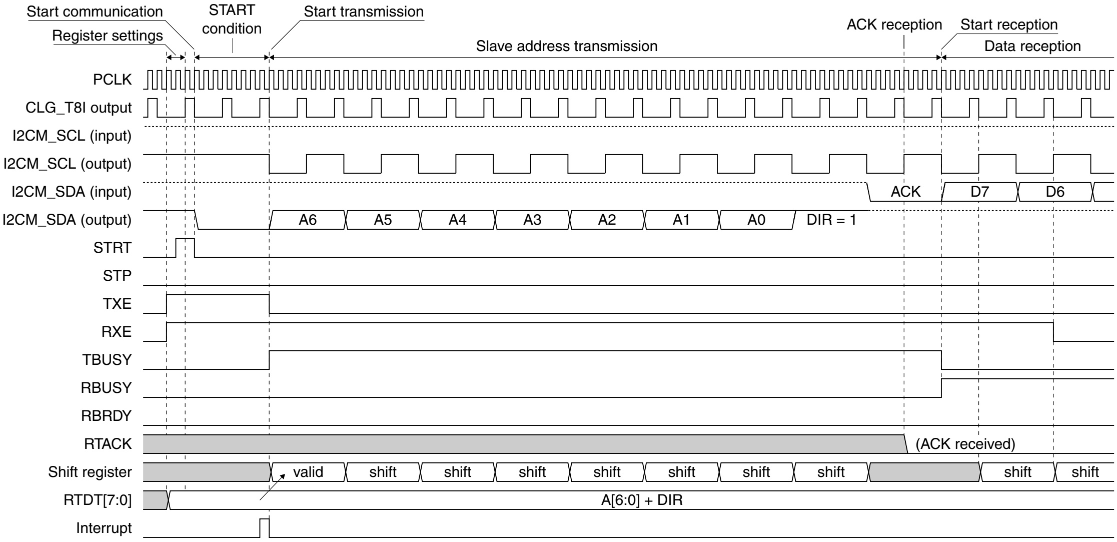


Figure V.2.5.7 I2C Master Timing Chart 3 (START condition → data reception)

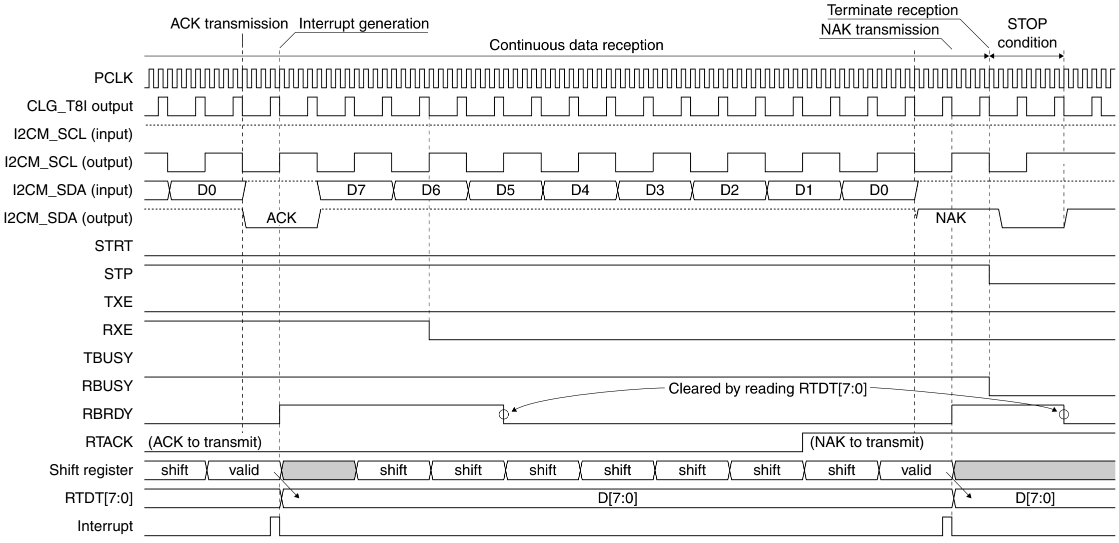


Figure V.2.5.8 I2C Master Timing Chart 4 (data reception → STOP condition)

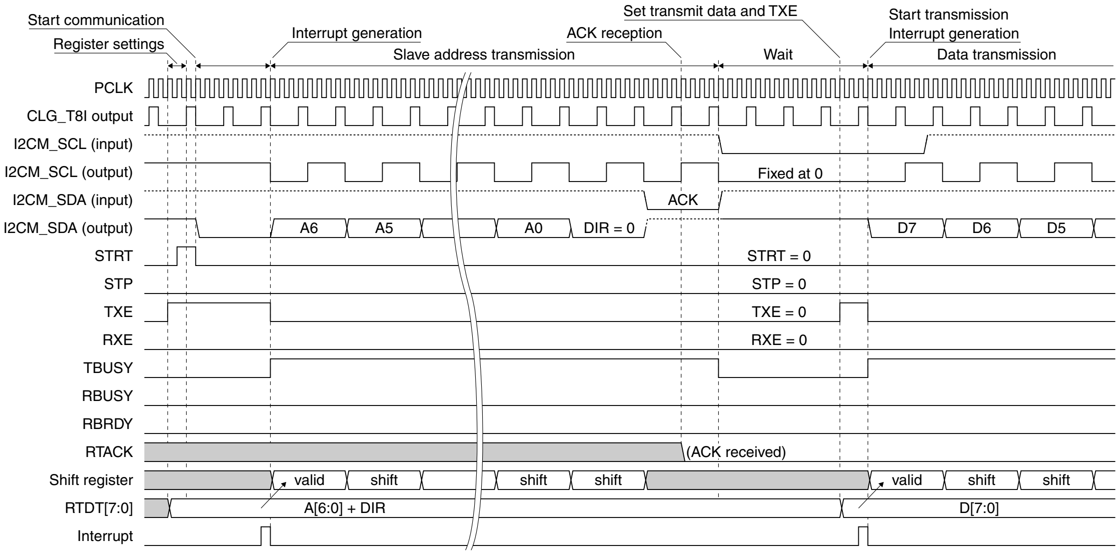


Figure V.2.5.9 I2C Master Timing Chart 5 (wait state)

V.2.6 I^{2}C Master Interrupt

The I^{2}C master module can generate the following two types of interrupts:

- Transmit buffer empty interrupt
- Receive buffer full interrupt

The I^{2}C master module has one interrupt signal to be output to the interrupt controller (ITC) and it is shared with the two causes of interrupt.

Transmit buffer empty interrupt

Set the TINTE bit (D0/I2CM_ICTL register) to 1 when using this interrupt. If TINTE is set to 0 (default), an interrupt request by this cause will not be sent to the ITC.

∗ **TINTE**: Transmit Interrupt Enable Bit in the I^{2}C Master Interrupt Control (I2CM_ICTL) Register (D0/0x4346)

When the transmit data set in RTDT[7:0] (D[7:0]/I2CM_DAT register) is transferred to the shift register, the I^{2}C master module outputs an interrupt request pulse to the ITC if the transmit buffer empty interrupt has been enabled (TINTE = 1).

∗ **RTDT[7:0]**: Receive/Transmit Data Bits in the I^{2}C Master Data (I2CM_DAT) Register (D[7:0]/0x4344)

If other interrupt conditions are satisfied, an interrupt is generated.

Receive buffer full interrupt

Set the RINTE bit (D1/I2CM_ICTL register) to 1 when using this interrupt. If RINTE is set to 0 (default), an interrupt request by this cause will not be sent to the ITC.

∗ **RINTE**: Receive Interrupt Enable Bit in the I^{2}C Master Interrupt Control (I2CM_ICTL) Register (D1/0x4346)

When data received in the shift register is loaded to RTDT[7:0], the I^{2}C master module outputs an interrupt request pulse to the ITC if the receive buffer full interrupt has been enabled (RINTE = 1).

If other interrupt conditions are satisfied, an interrupt is generated.

ITC registers for I^{2}C interrupts

The following shows the control bits of the ITC provided for the I^{2}C master module:

Interrupt flag

∗ **IIFT7**: I^{2}C Master Interrupt Flag Bit in the Interrupt Flag (ITC_IFLG) Register (D15/0x4300)

Interrupt enable bit

∗ **IIEN7**: I^{2}C Master Interrupt Enable Bit in the Interrupt Enable (ITC_EN) Register (D15/0x4302)

Interrupt level setup bits

∗ **IILV7[2:0]**: I^{2}C Master Interrupt Level Bits in the Internal Interrupt Level Setup (ITC_ILV3) Register 3 (D[10:8]/0x4314)

When the I^{2}C master module outputs an interrupt request pulse, the interrupt flag IIFT7 is set to 1.
If the interrupt enable bit IIEN7 has been set to 1, the ITC sends an interrupt request to the S1C17 Core. To disable the I^{2}C master interrupt, set the IIEN7 bit to 0.
The IIFT7 flag is always set to 1 by the I^{2}C master interrupt request pulse, regardless of how the IIEN7 bit is set (even when set to 0).
The interrupt level setup bits IILV7[2:0] set the interrupt level (0 to 7) of the I^{2}C master interrupt.

An interrupt request to the S1C17 Core is accepted only when all the conditions described below are met.

- The interrupt enable bit is set to 1.
- The IE (Interrupt Enable) bit of the PSR (Processor Status Register PSR) in the S1C17 Core is set to 1.
- The I^{2}C master interrupt has a higher interrupt level than the value that is set in the IL field of the PSR.
- No other cause of interrupt having higher priority, such as NMI, has occurred.

For details on these interrupt control registers, as well as the device operation when an interrupt has occurred, see Section III.1, "Interrupt Controller (ITC)."

Interrupt vector

The following shows the vector number and vector address for the I^{2}C master interrupt:

Vector number: 19 (0x13)
Vector address: TTBR + 0x4c

V.2.7 Details of Control Registers

Table V.2.7.1 List of I^{2}C Master Registers

Address	Register name		Function
0x4340	I2CM_EN	I^{2}C Master Enable Register	Enables the I^{2}C master module.
0x4342	I2CM_CTL	I^{2}C Master Control Register	Controls the I^{2}C master operation and indicates transfer status.
0x4344	I2CM_DAT	I^{2}C Master Data Register	Transmit/receive data
0x4346	I2CM_ICTL	I^{2}C Master Interrupt Control Register	Controls the I^{2}C master interrupt.

The following describes each I^{2}C master register. These are all 16-bit registers.

Notes:
- When setting the registers, be sure to write a 0, and not a 1, for all "reserved bits."
- Be sure to use 16-bit access instructions for reading/writing from/to the I^{2}C master registers. The I^{2}C master registers do not allow reading/writing using 32-bit and 8-bit access instructions.

0x4340: I^{2}C Master Enable Register (I2CM_EN)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
I^{2}C Master Enable Register (I2CM_EN)	**0x4340** (16 bits)	D15–1	–	reserved	–				–	–	0 when being read.
		D0	**I2CMEN**	I^{2}C master enable	1	Enable	0	Disable	0	R/W	

D[15:1] Reserved

D0 I2CMEN: I^{2}C Master Enable Bit

Enables/disables operation of the I^{2}C master module.

1 (R/W): Enable

0 (R/W): Disable (default)

When I2CMEN is set to 1, the I^{2}C master module is activated and data transfer is enabled. When I2CMEN is set to 0, the I^{2}C master module goes off.

0x4342: I^{2}C Master Control Register (I2CM_CTL)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
I^{2}C 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 I^{2}C master receive operation status.

1 (R): Busy

0 (R): Idle (default)

RBUSY is set to 1 when the I^{2}C master starts data reception and stays 1 while data reception is in progress. RBUSY is reset to 0 upon completion of receive operation. Also RBUSY returns to 0 in wait state.

D8 TBUSY: Transmit Busy Flag Bit

Indicates the I^{2}C master transmit operation status.

1 (R): Busy

0 (R): Idle (default)

TBUSY is set to 1 when the I^{2}C master starts data transmission and stays 1 while data transmission is in progress. TBUSY is reset to 0 upon completion of transmit operation. Also TBUSY returns to 0 in wait state.

D[7:5] Reserved

D4 NSERM: Noise Remove On/Off Bit

Turns the noise remove function on and off.

1 (R/W): On

0 (R/W): Off (default)

The I^{2}C master module contains a function to remove noise from the I2CM_SDA and I2CM_SCL input signals. This function is enabled by setting NSERM to 1.

Note, however, that the I^{2}C master clock (CLG_T8I timer output clock) frequency must be 1/6 of PCLK or lower to use the noise remove function.

D[3:2] Reserved

D1 STP: Stop Control Bit

Generates a STOP condition.

1 (R/W): Generate STOP condition

0 (R/W): Ignore (default)

When STP is set to 1, the I^{2}C module pulls up the I2CM_SDA line from low to high with the I2CM_SCL line held at high to generate a STOP condition on the I^{2}C bus. This makes the I^{2}C bus in free status.

Furthermore, the I^{2}C master module allows presetting for generating a STOP condition in advance. To do this, set STP to 1 after checking if the I^{2}C master is operating (TBUSY = 1 or RBUSY = 1). A STOP condition will be generated upon completion of data transmission/reception (including an ACK transfer).

STP is automatically reset to 0 after a STOP condition has been generated.

D0 STRT: Start Control Bit

Generates a START condition.

1 (R/W): Generate START condition
0 (R/W): Ignore (default)

When STRT is set to 1, the I^{2}C master module pulls down the I2CM_SDA line to low with the I2CM_SCL line held at high to generate a START condition on the I^{2}C bus. This makes the I^{2}C bus in busy status.

To generate a START condition, set the following registers in the order shown below:

1. Set the slave address to RTDT[7:0] (D[7:0]/I2CM_DAT register).
2. Set TXE (D9/I2CM_DAT register) to 1.
3. Set STRT to 1.

STRT is automatically reset to 0, after a START condition has been generated.

0x4344: I^{2}C Master Data Register (I2CM_DAT)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
I^{2}C Master Data Register (I2CM_DAT)	**0x4344** (16 bits)	D15–12	–	reserved	–				–	–	0 when being read.
		D11	**RBRDY**	Receive buffer ready	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 Bit

Indicates the receive buffer status.

1 (R): Received data is present

0 (R): No received data is present (default)

The RBRDY flag goes 1 when the received data in the shift register is loaded into RTDT[7:0] (D[7:0]) and returns to 0 when the received data is read from RTDT[7:0]. An interrupt can be generated when this flag goes 1. Use this interrupt or read the RBRDY flag to check that RTDT[7:0] contains valid data when reading received data.

D10 RXE: Receive Execution Bit

Execute a data reception for one byte.

1 (R/W): Start data reception

0 (R/W): Ignore (default)

The I^{2}C master module starts data reception for one byte by setting RXE to 1 and TXE (D9) to 0. RXE can be set to 1 for the next data reception even if a slave address is being transmitted or data is being received. RXE is reset to 0 when D6 is input to the shift register.

D9 TXE: Transmit Execution Bit

Execute a data transmission for one byte.

1 (R/W): Start data transmission

0 (R/W): Ignore (default)

Set the transmit data to RTDT[7:0] (D[7:0]) and write 1 to TXE to start data transmission. TXE can be set to 1 for the next data transmission even if a slave address or data is being transmitted. TXE is reset to 0 when the data set in RTDT[7:0] is transferred to the shift register.

D8 RTACK: Receive/Transmit ACK Bit

In data transmission

Indicates the acknowledge bit status.

1 (R/W): Error (NAK)

0 (R/W): ACK (default)

This bit is set to 0 when the slave returned ACK after one-byte data has been transmitted. This indicates that the slave could receive the data normally. If this bit is set to 1, the slave may be inactive or it could not receive the data normally.

In data reception

Set the acknowledge bit to be sent to the slave.

1 (R/W): Error (NAK)

0 (R/W): ACK (default)

To return ACK to the slave after data is received, set RTACK to 0 before the I^{2}C module sends the acknowledge bit.

To return NAK, set RTACK to 1.

D[7:0] RTDT[7:0]: Receive/Transmit Data Bits

In data transmission

Set a transmit data in this register. (Default: 0x0)

Data transmission begins when TXE (D9) is set to 1. If a slave address or data is being transmitted, a new transmission starts after the current transmission has completed. The serial-converted data is output from the I2CM_SDA 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 transferred to the shift register, a cause of transmit buffer empty interrupt occurs. The next transmit data can be written to the register after that.

In data reception

The received data can be read from this register. (Default: 0x0)

Data reception begins when RXE (D10) is set to 1. If a slave address is being transmitted or data is being received, a new reception starts after the current reception has completed. When a receive operation is completed and the data received in the shift register is loaded to this register, the RBRDY flag (D11) is set and a cause of receive buffer full interrupt occurs. Thereafter, the data can be read out at any time before a receive operation for the next data is completed.

If the next data receive operation is completed before this register is read out, the data in it is overwritten with the newly received data.

The serial data input from the I2CM_SDA pin is converted into parallel data beginning with the MSB, with the high-level signals changed to 1s and the low-level signals changed to 0s. The resulting data is stored in this register.

0x4346: I^{2}C Master Interrupt Control Register (I2CM_ICTL)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
I^{2}C 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/disables the I^{2}C master receive buffer full interrupt.

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

When RINTE is set to 1, I^{2}C master receive buffer full interrupt requests to the ITC are enabled. A receive buffer full interrupt request occurs when the data received in the shift register is loaded to RTDT[7:0] (D[7:0]/I2CM_DAT register) (receive operation completed).
When RINTE is set to 0, an I^{2}C master receive buffer full interrupt is not generated.

D0 TINTE: Transmit Interrupt Enable Bit

Enables/disables the I^{2}C master transmit buffer empty interrupt.

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

When TINTE is set to 1, I^{2}C master transmit buffer empty interrupt requests to the ITC are enabled. A transmit buffer empty interrupt request occurs when the data written to RTDT[7:0] (D[7:0]/I2CM_DAT register) is transferred to the shift register.
When TINTE is set to 0, an I^{2}C master transmit buffer empty interrupt is not generated.

THIS PAGE IS BLANK.

V.3 I2C Slave

V.3.1 Configuration of the I2C Slave Module

The S1C17002 equipped with an I2C slave module for high-speed synchronous serial communication. This I2C slave module operates as an I2C slave device using the clock supplied from the I2C master. It supports standard (100 kbps) and fast (400 kbps) modes, 7-bit slave addressing, and a clock stretch function. The I2C slave module includes a noise remove function to secure reliable data transfer.

Also it can generate three types of interrupts (transmit, receive, and bus status interrupts), this makes it possible to process continuous serial data transfer simply in an interrupt handler.

Figure V.3.1.1 shows the structure of the I2C slave module.

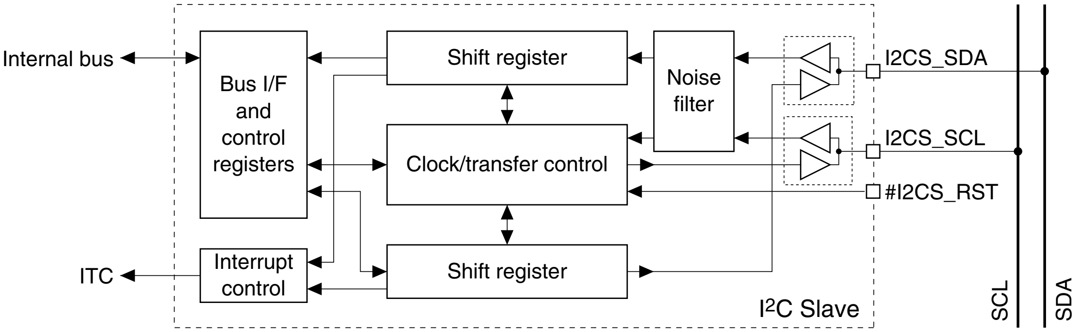


Figure V.3.1.1 Structure of I2C Slave Module

Note: The I2C slave module does not support general call address and 10-bit address mode.

V.3.2 I^{2}C Slave I/O Pins

Table V.3.2.1 lists the I^{2}C slave pins.

Table V.3.2.1 List of I^{2}C Slave Pins

Pin name	I/O	Size	Function
I2CS_SDA	I/O	1	I^{2}C slave data input/output pin This pin inputs serial data from the I^{2}C bus and outputs serial data to the I^{2}C bus.
I2CS_SCL	I/O	1	I2CS clock input/output pin Inputs SCL line status from the I^{2}C bus. Also outputs a low level to put the I^{2}C bus into clock stretch status.
#I2CS_RST	I	1	I^{2}C slave bus free request input pin A low pulse input to this pin requests the I^{2}C slave to release the I^{2}C bus. When the bus free request input has been enabled with software, a low pulse initializes the communication process of the I^{2}C slave module and sets the I2CS_SDA and I2CS_SCL pins to high impedance state.

The I^{2}C slave input/output pins (I2CS_SDA, I2CS_SCL, and #I2CS_RST) are shared with the I/O ports and they are initialized as general-purpose I/O port pins by default. Before using these pins for the I^{2}C slave, the pin functions must be switched using the Port Function Select Register.

For details on switching pin function, Section I.3.3, “Switching Over the Multiplexed Pin Functions.”

V.3.3 I^{2}C Slave Clock

The I^{2}C slave module operates with the clock output from the external I^{2}C master device by inputting it from the I2CS_SCL pin.

The I^{2}C slave module also uses the system clock (PCLK) for its operations. The PCLK frequency must be set eight-times or higher than the I2CS_SCL 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. See “Asynchronous address detection” in Section V.3.4.3, “Optional Functions,” for details.

V.3.4 Initializing the I²C Slave

V.3.4.1 Reset

The I²C slave 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 I²C slave module can be reset by altering SOFTRESET (D6/I2CS_CTL register).

∗ **SOFTRESET**: Software Reset Bit in the I²C Slave Control (I2CS_CTL) Register (D6/0x4366)

To reset the I²C slave module, write 1 to SOFTRESET to place the I²C slave 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 I²C slave module initializes the I²C slave communication process and put the I2CS_SDA and I2CS_SCL pins into high-impedance state to be ready to detect a start condition. Furthermore, the I²C slave 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 #I2CS_RST pin

The I²C slave module can accept bus free requests using the #I2CS_RST pin input. The bus free request support is disabled by default. To enable this function, set BFREQ_EN (D4/I2CS_CTL register) to 1.

∗ **BFREQ_EN**: Bus Free Request Enable Bit in the I²C Slave Control (I2CS_CTL) Register (D4/0x4366)

When this function is enabled, a low pulse (One system clock (PCLK) cycle or more pulse width is required. Two PCLK clock cycles or more pulse width is recommended.) input to the #I2CS_RST pin sets BFREQ (D4/I2CS_STAT register) to 1. This initializes the I²C slave communication process and puts the I2CS_SDA and I2CS_SCL pins into high-impedance state. The control registers will not be initialized as distinct from the software reset described above.

∗ **BFREQ**: Bus Free Request Bit in the I²C Slave Status (I2CS_STAT) Register (D4/0x4368)

Note: When BFREQ is set to 1 (an interrupt can be used for this check), perform the software reset and set the registers again.

V.3.4.2 Setting the Slave Address

I²C slave devices have a unique slave address to identify each device.

The I²C slave module supports 7-bit address (does not support 10-bit address), and the address of this module must be set to the I2CS_SADRS register (0x4364).

V.3.4.3 Optional Functions

The I²C slave module has a clock stretch, asynchronous address detection, and noise remove 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 I2CS_SCL line down to low. The I²C slave module supports this clock stretch function. The master device enters a standby state until the wait request is canceled (the I2CS_SCL input goes high). The clock stretch function in this module is disabled by default. When using the clock stretch function, set CLKSTR_EN (D3/I2CS_CTL register) to 1 before starting data communication. Note that the data setup time (after the I2CS_SDA pin outputs the MSB of SDATA[7:0] (D[7:0]/I2CS_TRNS register) until the I²C slave module turns the I2CS_SCL pin pull-down resistor off) while the I²C slave module is operating with the clock stretch function enabled varies depending on the I²C slave module operating clock (PCLK) frequency.

∗ **CLKSTR_EN**: Clock Stretch On/Off Bit in the I²C Slave Control (I2CS_CTL) Register (D3/0x4366)

Asynchronous address detection

The I^{2}C slave 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 I^{2}C 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 (D1/I2CS_CTL register) to 1.

∗ **ASDET_EN**: Async. Address Detection On/Off Bit in the I^{2}C Slave Control (I2CS_CTL) Register (D1/0x4366)

If the slave address sent from the master has matched with one that has been set in this I^{2}C slave module when the asynchronous address detection function has been enabled, the I^{2}C slave module generates a bus status interrupt and returns NAK to the I^{2}C 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 I^{2}C 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 I^{2}C 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 I^{2}C slave module contains a function to remove noise from the I2CS_SDA and I2CS_SCL input signals. This function is enabled by setting NF_EN (D2/I2CS_CTL register) to 1.

∗ **NF_EN**: Noise Filter On/Off Bit in the I^{2}C Slave Control (I2CS_CTL) Register (D2/0x4366)

V.3.5 Data Transmit/Receive Control

Before starting data transfer, set up the conditions by the procedure below.

(1) Initialize the I2C slave module. See Section V.3.4.

(2) Set up the interrupt conditions if the I2C slave interrupt is used. See Section V.3.6.

Note: Make sure that the I2C slave module is disabled (I2C_EN/I2CS_CTL register = 0) before setting the conditions above.

* **I2C_EN**: I2C Slave Enable Bit in the I2C Slave Control (I2CS_CTL) Register (D7/0x4366)

Enabling data transmission/reception

First, set the I2C_EN bit (D7/I2CS_CTL register) to 1 to enable I2C slave operation. This makes the I2C slave in ready-to-transmit/receive status in which a START condition can be detected.

Note: Do not set the I2C_EN bit to 0 while the I2C slave module is transmitting/receiving data.

Starting data transmission/reception

To start data transmission/reception, set COM_MODE (D0/I2CS_CTL register) to 1 to enable the data communication.

* **COM_MODE**: I2C Slave Communication Mode Bit in the I2C Slave Control (I2CS_CTL) Register (D0/0x4366)

When the slave address for this module that has been sent from the master is received after a START condition is detected, the I2C slave module returns an ACK (I2CS_SDA = 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 I2C slave module does not send back a response if the master has sent the slave address of this module (it is regarded as that the I2C module has returned a NAK to the master).

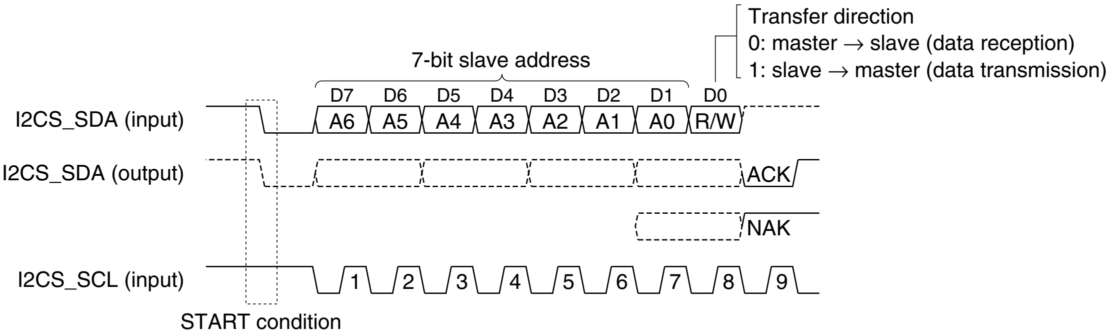


Figure V.3.5.1 Receiving Slave Address and Data Direction Bit

When a START condition is detected, BUSY (D2/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 (D1/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. Furthermore, the value of the transfer direction bit is set to R/W (D0/I2CS_ASTAT register), so use R/W to select the transmit- or receive-handling.

* **BUSY**: I2C Bus Status Bit in the I2C Slave Access Status (I2CS_ASTAT) Register (D2/0x436a)
* **SELECTED**: I2C Slave Select Status Bit in the I2C Slave Access Status (I2CS_ASTAT) Register (D1/0x436a)
* **R/W**: Read/Write Direction Bit in the I2C Slave Access Status (I2CS_ASTAT) Register (D0/0x436a)

If the slave address of this module is detected when the asynchronous address detection function has been enabled, ASDET (D2/I2CS_STAT register) is set to 1. The I2C slave 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.

* **ASDET**: Async. Address Detection Status Bit in the I2C Slave Status (I2CS_STAT) Register (D2/0x4368)

Data transmission

The following describes a data transmission procedure.
The I2C slave module starts data transmit process when both SELECTED and R/W are set to 1. It sets TXEMP (D3/I2CS_ASTAT register) to 1 to issue a request to the application program to write transmit data. Write transmit data to SDATA[7:0] (D[7:0]/I2CS_TRNS register).

* **TXEMP**: Transmit Data Empty Bit in the I2C Slave Access Status (I2CS_ASTAT) Register (D3/0x436a)
* **SDATA[7:0]**: I2C Slave Transmit Data Bits in the I2C Slave Transmit Data (I2CS_TRNS) Register (D[7:0]/0x4360)

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 slave clock (I2CS_SCL) 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 (D8/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.

* **TBUF_CLR**: I2CS_TRNS Register Clear Bit in the I2C Slave Control (I2CS_CTL) Register (D8/0x4366)

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 slave clock (I2CS_SCL) 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 (D5/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.

* **TXUDF**: Transmit Data Underflow Bit in the I2C Slave Status (I2CS_STAT) Register (D5/0x4368)

When the clock stretch function is enabled

When the clock stretch function has been enabled, the I2C slave module pulls down the I2CS_SCL 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 I2CS_SDA pin in sync with the I2CS_SCL 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.

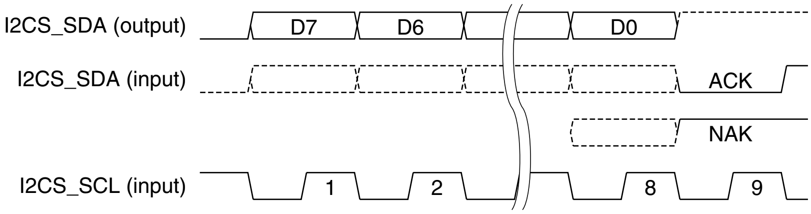


Figure V.3.5.2 ACK and NAK

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 ACK generates a clock stretch status when the clock stretch function has been enabled, so data can be written after an ACK is received.

An 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 (D1/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.

* **DA_NAK**: NAK Receive Status Bit in the I2C Slave Status (I2CS_STAT) Register (D1/0x4368)

The I2CS_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 (D3/I2CS_STAT register). DMS is set to 0 when data is output correctly. If the I2CS_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 I2CS_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.

* **DMS**: Output Data Mismatch Bit in the I2C Slave Status (I2CS_STAT) Register (D3/0x4368)

Note: If the I2C slave 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 I2C slave address again).

1. The transfer rate is set to 320 kbps or higher.
2. The asynchronous address detection function is enabled.
3. The I2C slave 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 I2C slave module starts data receive process when SELECTED is set to 1 and R/W is set to 0. The receive data bits are input from the I2CS_SDA pin in sync with the I2CS_SCL 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] (D[7:0]/I2CS_RECV register).

* **RDATA[7:0]**: I2C Slave Receive Data Bits in the I2C Slave Receive Data (I2CS_RECV) Register (D[7:0]/0x4362)

When the received data is loaded to RDATA[7:0], RXRDY (D4/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 writing 1.

* **RXRDY**: Receive Data Ready Bit in the I2C Slave Access Status (I2CS_ASTAT) Register (D4/0x436a)

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 slave clock (I2CS_SCL) from RXRDY being set to 1.

When the clock stretch function is enabled

When the clock stretch function has been enabled, the I2C slave module pulls down the I2CS_SCL 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 (D5/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.

* **RXOVF**: Receive Data Overflow Bit in the I2C Slave Status (I2CS_STAT) Register (D5/0x4368)

To return NAK during data reception

During data reception (master transmission), the I2C slave module sends back an ACK (I2CS_SDA = low) every time an 8-bit data has been received (by default setting). The response code can be changed to NAK (I2CS_SDA = Hi-Z) by setting NAK_ANS (D5/I2CS_CTL register). ACK will be sent when NAK_ANS is 0 or NAK will be sent when NAK_ANS is set to 1.

∗ **NAK_ANS**: NAK Answer Bit in the I2C Slave Control (I2CS_CTL) Register (D5/0x4366)

NAK_ANS should be set within 7 cycles of the I2C slave clock (I2CS_SCL) after RXRDY has been set to 1 by receiving data just prior to one required for returning NAK.

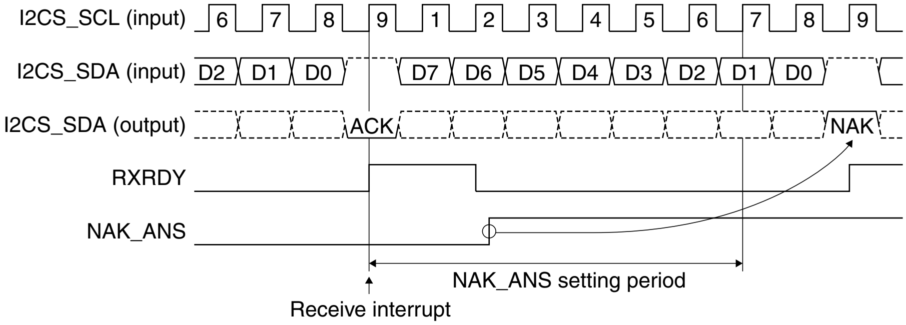


Figure V.3.5.3 Setting NAK_ANS and NAK Response Timing

Terminating data transmission/reception (detecting a STOP condition)

Data transfer will be terminated when the master generates a STOP condition. The STOP condition is a state in which the I2CS_SDA line is pulled up from low to high with the I2CS_SCL line held at high.

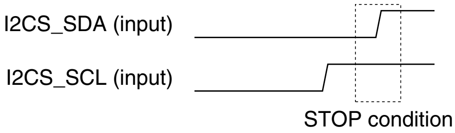


Table V.3.5.4 STOP Condition

If a STOP condition is detected while the I2C slave module is selected as the slave device (SELECTED = 1), the I2C slave module sets DA_STOP (D0/I2CS_STAT register) to 1. At the same time, it puts the I2CS_SDA and I2CS_SCL pins into high-impedance state and initializes the I2C slave communication process to enter standby state that is ready to detect the next START condition. Also SELECTED and BUSY are reset to 0.

∗ **DA_STOP**: Stop Condition Detect Bit in the I2C Slave Status (I2CS_STAT) Register (D0/0x4368)

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 transmission/reception

After data transfer has finished, write 0 to the COM_MODE (D0/I2CS_CTL register) to disable data transmission/reception.
Always make sure that the BUSY and SELECTED flags are 0 before data transmission/reception is disabled.
To deactivate the I2C slave module, set I2C_EN (D7/I2CS_CTL register) to 0.

Timing charts

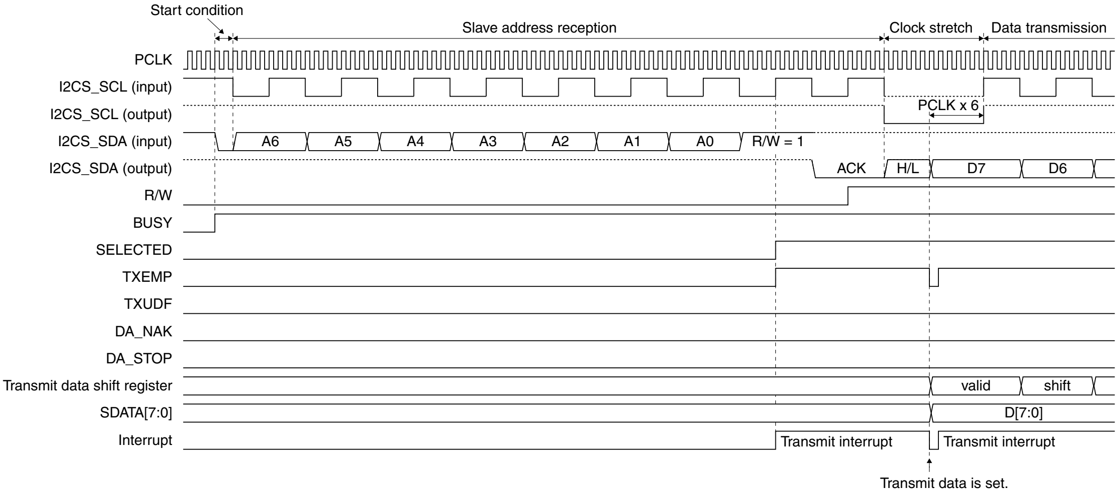


Figure V.3.5.5 I2C Slave Timing Chart 1 (START condition → data transmission)

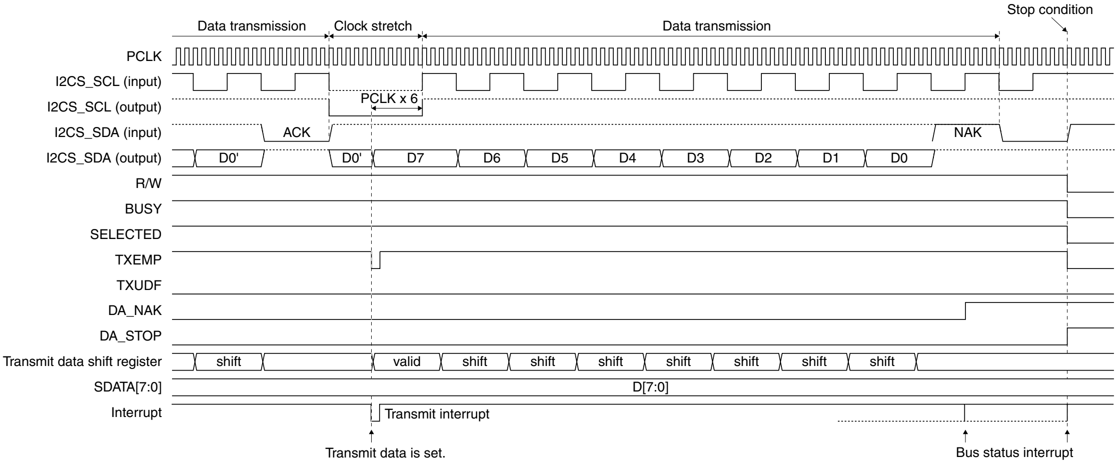


Figure V.3.5.6 I2C Slave Timing Chart 2 (data transmission → STOP condition)

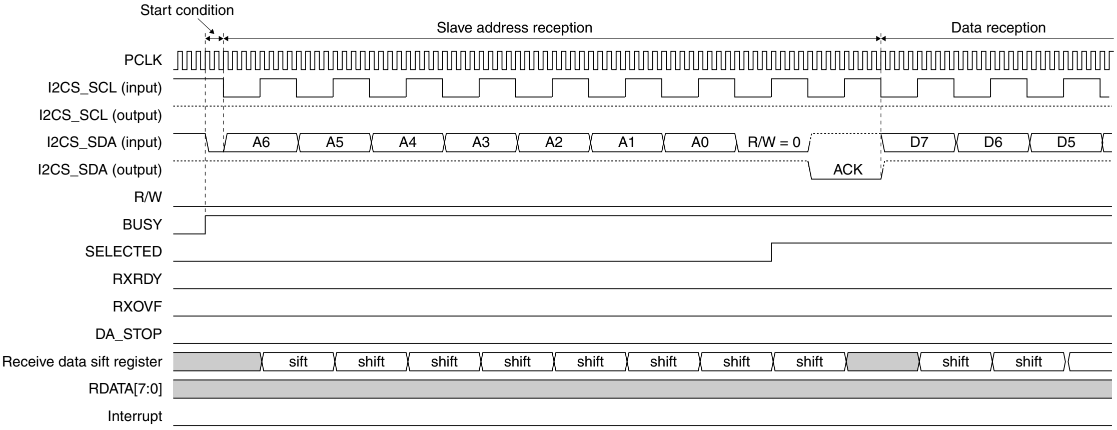


Figure V.3.5.7 I2C Slave Timing Chart 3 (START condition → data reception)

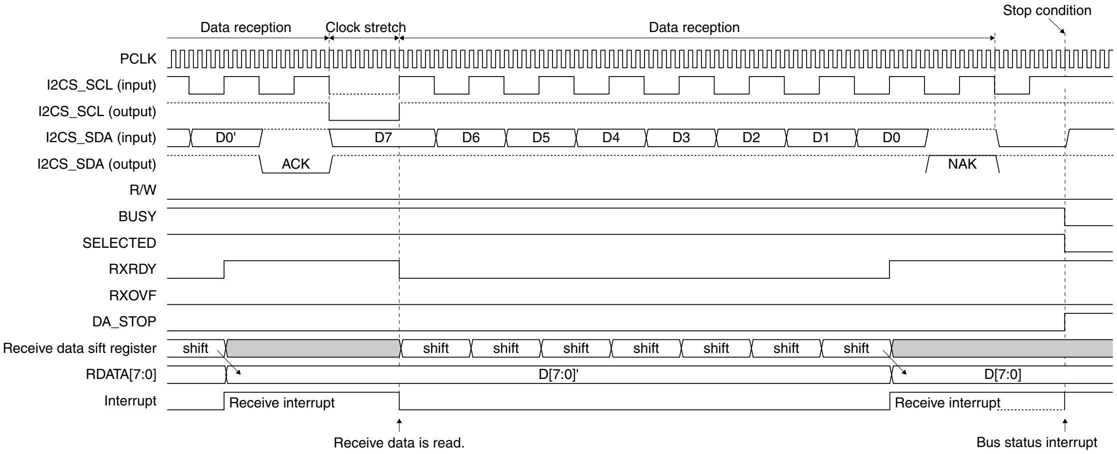


Figure V.3.5.8 I^{2}C Slave Timing Chart 4 (data reception → STOP condition)

V.3.6 I^{2}C Slave Interrupt

The I^{2}C slave module can generate the following three types of interrupts:
- Transmit interrupt
- Receive interrupt
- Bus status interrupt

Transmit interrupt

When the transmit data written to SDATA[7:0] (D[7:0]/I2CS_TRNS register) is sent to the shift register, TXEMP (D3/I2CS_ASTAT register) is set to 1 and an interrupt signal is output to the ITC. This interrupt can be used to write the next transmit data to SDATA[7:0].

∗ **SDATA[7:0]**: I^{2}C Slave Transmit Data Bits in the I^{2}C Slave Transmit Data (I2CS_TRNS) Register (D[7:0]/0x4360)
∗ **TXEMP**: Transmit Data Empty Bit in the I^{2}C Slave Access Status (I2CS_ASTAT) Register (D3/0x436a)

Set TXEMP_IEN (D0/I2CS_ICTL register) to 1 when using this interrupt. If TXEMP_IEN is set to 0 (default), an interrupt request by this cause will not be sent to the ITC.

∗ **TXEMP_IEN**: Transmit Interrupt Enable Bit in the I^{2}C Slave Interrupt Control (I2CS_ICTL) Register (D0/0x436c)

Receive interrupt

When the received data is loaded to RDATA[7:0] (D[7:0]/I2CS_RECV register), RXRDY (D4/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].

∗ **RDATA[7:0]**: I^{2}C Slave Receive Data Bits in the I^{2}C Slave Receive Data (I2CS_RECV) Register (D[7:0]/0x4362)
∗ **RXRDY**: Receive Data Ready Bit in the I^{2}C Slave Access Status (I2CS_ASTAT) Register (D4/0x436a)

Set RXRDY_IEN (D1/I2CS_ICTL register) to 1 when using this interrupt. If RXRDY_IEN is set to 0 (default), an interrupt request by this cause will not be sent to the ITC.

∗ **RXRDY_IEN**: Receive Interrupt Enable Bit in the I^{2}C Slave Interrupt Control (I2CS_ICTL) Register (D1/0x436c)

Bus status interrupt

The I^{2}C slave module provides the status bits listed below to represent the transmit/receive and I^{2}C bus statuses (see Section V.3.5 for details of each function).

1. ASDET: set to 1 when the slave address is detected by the asynchronous address detection function
   ∗ **ASDET**: Async. Address Detection Status Bit in the I^{2}C Slave Status (I2CS_STAT) Register (D2/0x4368)
2. TXUDF: set to 1 when a transmit operation has started before transmit data is written
   (when the clock stretch function is disabled)
   ∗ **TXUDF**: Transmit Data Underflow Bit in the I^{2}C Slave Status (I2CS_STAT) Register (D5/0x4368)
3. DA_NAK: set to 1 when a NAK is returned from the master during transmission
   ∗ **DA_NAK**: NAK Receive Status Bit in the I^{2}C Slave Status (I2CS_STAT) Register (D1/0x4368)
4. DMS: set to 1 when the I2CS_SDA line status is different from transfer data
   ∗ **DMS**: Output Data Mismatch Bit in the I^{2}C Slave Status (I2CS_STAT) Register (D3/0x4368)

   DMA will also be set to 1 when another slave device issues ACK to this I^{2}C slave address (when ASDET_EN (D1/I2CS_CTL register) = 0).

   Note: When the master device of the I^{2}C bus, which has multiple slave devices connected including this IC, starts communication with another slave device, the I^{2}C slave 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 I2CS_SDA line status. When SELECTED (D1/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 I^{2}C slave is placed into asynchronous address detection mode, a DMS does not occur as in the condition above.

5. RXOVF: 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)
   * **RXOVF**: Receive Data Overflow Bit in the I2C Slave Status (I2CS_STAT) Register (D5/0x4368)
6. BFREQ: set to 1 when a bus free request is accepted
   * **BFREQ**: Bus Free Request Bit in the I2C Slave Status (I2CS_STAT) Register (D4/0x4368)
7. DA_STOP: set to 1 if a STOP condition or a repeated start condition is detected while this module is selected as the slave device
   * **DA_STOP**: Stop Condition Detect Bit in the I2C Slave Status (I2CS_STAT) Register (D0/0x4368)

When one of the bits shown above is set to 1, BSTAT (D7/I2CS_STAT register) is set to 1 and an interrupt signal is output to the ITC. This interrupt can be used to perform an error or terminate handling.

* **BSTAT**: Bus Status Transition Bit in the I2C Slave Status (I2CS_STAT) Register (D7/0x4368)

Set BSTAT_IEN (D2/I2CS_ICTL register) to 1 when using this interrupt. If BSTAT_IEN is set to 0 (default), an interrupt request by this cause will not be sent to the ITC.

* **BSTAT_IEN**: Bus Status Interrupt Enable Bit in the I2C Slave Interrupt Control (I2CS_ICTL) Register (D2/0x436c)

ITC registers for I2C slave interrupts

When a cause of interrupt that has been enabled occurs, the I2C slave module asserts the interrupt signal sent to the ITC. To generate an I2C slave interrupt, set the interrupt level and enable the interrupt using the ITC registers. Table V.3.6.1 shows the control bits for the I2C slave interrupt in the ITC.

Table V.3.6.1 ITC Registers

Cause of interrupt	Interrupt flag	Interrupt enable bit	Interrupt level setup bits
Transmit/receive	AIFT12 (D12/ITC_AIFLG)	AIEN12 (D12/ITC_AEN)	AILV12[2:0] (D[2:0]/ITC_AILV6)
Bus status	AIFT13 (D13/ITC_AIFLG)	AIEN13 (D13/ITC_AEN)	AILV13[2:0] (D[10:8]/ITC_AILV6)

ITC_AIFLG register (0x42e0)
ITC_AEN register (0x42e2)
ITC_AILV6 register (0x42f2)

When the I2C slave module outputs an interrupt signal, the corresponding interrupt flag is set to 1.
If the interrupt enable bit corresponding to that interrupt flag has been set to 1, the ITC sends an interrupt request to the S1C17 Core. To disable the timer interrupt, set the interrupt enable bit to 0.
The interrupt flag is always set to 1 by the I2C slave interrupt signal, regardless of how the interrupt enable bit is set (even when set to 0).
The interrupt level setup bits set the interrupt level (0 to 7) of the timer interrupt. If the same interrupt level is set, the transmit/receive interrupt has highest priority and the bus status interrupt has lowest priority.
An interrupt request to the S1C17 Core is accepted only when all the conditions described below are met.

- The interrupt enable bit is set to 1.
- The IE (Interrupt Enable) bit of the PSR (Processor Status Register) in the S1C17 Core is set to 1.
- The I2C slave interrupt has a higher interrupt level than the value that is set in the IL field of the PSR.
- No other cause of interrupt having higher priority, such as NMI, has occurred.

For details on these interrupt control registers, as well as the device operation when an interrupt has occurred, see Section III.1, "Interrupt Controller (ITC)."

Interrupt vector

The following shows the vector number and vector address for the I2C slave interrupt:

Table V.3.6.2 I2C Slave Interrupt Vectors

Cause of interrupt	Vector number	Vector address
Transmit/receive	28 (0x1c)	TTBR + 0x70
Bus status	29 (0x1d)	TTBR + 0x74

V.3.7 Details of Control Registers

Table V.3.7.1 List of I^{2}C Slave Registers

Address	Register name		Function
0x4360	I2CS_TRNS	I^{2}C Slave Transmit Data Write Register	I^{2}C slave transmit data
0x4362	I2CS_RECV	I^{2}C Slave Receive Data Read Register	I^{2}C slave receive data
0x4364	I2CS_SADRS	I^{2}C Slave Address Setup Register	Sets the I^{2}C slave address.
0x4366	I2CS_CTL	I^{2}C Slave Control Register	Controls the I^{2}C slave module.
0x4368	I2CS_STAT	I^{2}C Slave Status Register	Indicates the I^{2}C slave bus status.
0x436a	I2CS_ASTAT	I^{2}C Slave Access Status Register	Indicates the I^{2}C slave access status.
0x436c	I2CS_ICTL	I^{2}C Slave Interrupt Control Register	Controls the I^{2}C slave interrupt.

The following describes each I^{2}C slave register. These are all 16-bit registers.

Note: When setting the registers, be sure to write a 0, and not a 1, for all “reserved bits.”

0x4360: 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

Set a transmit data in this register. (Default: 0x0)

The serial-converted data is output from the I2CS_SDA 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 slave clock (I2CS_SCL) 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 slave clock (I2CS_SCL) 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 (D8/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.

0x4362: I^{2}C Slave Receive Data Register (I2CS_RECV)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
I^{2}C Slave Receive Data Register (I2CS_RECV)	**0x4362** (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	**RDATA[7:0]**	I^{2}C slave receive data	0–0xff	0x0	R	

D[15:8] Reserved

D[7:0] RDATA[7:0]: I^{2}C Slave Receive Data Bits

The received data can be read from this register. (Default: 0x0)

The serial data input from the I2CS_SDA pin is converted into parallel data beginning with the MSB, 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 (D4/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 I^{2}C slave clock (I2CS_SCL) 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.

0x4364: 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]: I2C Slave Address Bits

Set the slave address of the I2C slave module to this register. (Default: 0x0)

0x4366: I^{2}C Slave Control Register (I2CS_CTL)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
I^{2}C 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	**I2C_EN**	I^{2}C 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**	I^{2}C 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 (0x4360).

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 (D3/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 I2C_EN: I^{2}C Slave Enable Bit

Enables/disables operation of the I^{2}C slave module.

1 (R/W): Enable

0 (R/W): Disable (default)

When I2C_EN is set to 1, the I^{2}C slave module is activated and data transfer is enabled.

When I2C_EN is set to 0, the I^{2}C slave module goes off.

D6 SOFTRESET: Software Reset Bit

Resets the I^{2}C slave module.

1 (R/W): Reset

0 (R/W): Cancel reset state (default)

To reset the I^{2}C slave module, write 1 to SOFTRESET to place the I^{2}C slave 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 I^{2}C slave module initializes the I^{2}C slave communication process and put the I2CS_SDA and I2CS_SCL pins into high-impedance state to be ready to detect a start condition. Furthermore, the I^{2}C slave 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 I^{2}C slave module sends back an ACK (I2CS_SDA = low) or a NAK (I2CS_SDA = Hi-Z). Either ACK or NAK should be specified using NAK_ANS within 7 cycles of the I^{2}C slave clock (I2CS_SCL) after RXRDY has been set to 1 by receiving the previous data.

D4 BFREQ_EN: Bus Free Request Enable Bit

Enables/disables I^{2}C bus free requests by inputting a low pulse to the #I2CS_RST pin.
1 (R/W): Enable
0 (R/W): Disable (default)

To accept I^{2}C bus free requests, set BFREQ_EN to 1. When a bus free request is accepted, BFREQ (D4/I2CS_STAT register) is set to 1. This initializes the I^{2}C slave communication process and puts the I2CS_SDA and I2CS_SCL pins into high-impedance state. The control registers will not be initialized in this process.
When BFREQ_EN is set to 0, low pulse inputs to the #I2CS_RST 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 I2CS_SCL line down to low. The I^{2}C slave module supports this clock stretch function. The master device enters a standby state until the wait request is canceled (the I2CS_SCL input 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 I^{2}C slave module contains a function to remove noise from the I2CS_SDA and I2CS_SCL 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 I^{2}C slave 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 I^{2}C 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 I^{2}C slave module when the asynchronous address detection function has been enabled, the I^{2}C slave module generates a bus status interrupt and returns NAK to the I^{2}C 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 I^{2}C 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 I^{2}C 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: I^{2}C Slave Communication Mode Bit

Enables/disables data communication.

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

Set COM_MODE to 1 to enable data communication after setting the I2C_EN bit (D7) to 1 to enable I^{2}C slave operation. When COM_MODE is 0 (default), the I^{2}C slave module does not send back a response if the master has sent the slave address of this module (it is regarded as that the I^{2}C module has returned a NAK to the master).

0x4368: I^{2}C Slave Status Register (I2CS_STAT)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
I^{2}C 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 (D5), BFREQ (D4), DMS (D3), ASDET (D2), DA_NAK (D1), and DA_STOP (D0) 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 (D2/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 (D5), BFREQ (D4), DMS (D3), ASDET (D2), DA_NAK (D1), and DA_STOP (D0) 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 (D2/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

Indicate the I^{2}C bus free request input status.

1 (R/W): Request has been issued

0 (R/W): Request has not been issued (default)

If BFREQ_EN (D4/I2CS_CTL register) has been set to 1 (bus free request enabled), a low pulse longer than five system clock (PCLK) cycles input to the #I2CS_RST pin sets BFREQ to 1 and the bus free request is accepted. When a bus free request is accepted, the I^{2}C slave module initializes the I^{2}C communication process and puts the I2CS_SDA and I2CS_SCL pins into high-impedance state. 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 (D2/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 #I2CS_RST 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 I2CS_SDA line status.
1 (R/W): Error has been occurred
0 (R/W): Error has not been occurred (default)

The I2CS_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. DMS is set to 0 when data is output correctly. If the I2CS_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 I2CS_SDA line. At the same time, an interrupt signal is output to the ITC if the interrupt is enabled with BSTAT_IEN (D2/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 I^{2}C bus, which has multiple slave devices connected including this IC, starts communication with another slave device, the I^{2}C slave 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 I2CS_SDA line status. When SELECTED (D1/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 I^{2}C slave is placed into asynchronous address detection mode, 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 I^{2}C slave 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 I^{2}C slave address sent from the master in this status. ASDET is set to 1 if the slave address of the I^{2}C slave module is detected when the asynchronous address detection function has been enabled by setting ASDET_EN (D1/I2CS_CTL register). The I^{2}C slave module returns a NAK to the I^{2}C 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 (D2/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 (D2/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 I^{2}C slave module is selected as the slave device (SELECTED (D1/I2CS_ASTAT register) = 1), the I^{2}C slave module sets DA_STOP to 1. At the same time, it initializes the I^{2}C 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 (D2/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.

0x436a: I^{2}C Slave Access Status Register (I2CS_ASTAT)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
I^{2}C 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**	I^{2}C bus status	1	Busy	0	Free	0	R	
		D1	**SELECTED**	I^{2}C 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 (D1/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)

When the transmit data written to the I2CS_TRNS register is sent, TXEMP is set to 1. At the same time, an interrupt signal is output to the ITC if the interrupt is enabled with TXEMP_IEN (D0/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: I^{2}C Bus Status Bit

Indicates the I^{2}C bus status.

1 (R): Bus busy status
0 (R): Bus free status (default)

When the I^{2}C slave module detects a START condition or detects that the I2CS_SCL or I2CS_SDA signal goes low, BUSY is set to 1 to indicate that the I^{2}C 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: I^{2}C Slave Select Status Bit

Indicates that this module is selected as the I^{2}C 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 I^{2}C 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.

0x436c: I^{2}C Slave Interrupt Control Register (I2CS_ICTL)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
I^{2}C 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/disables the bus status interrupt.

1 (R/W): Enable

0 (R/W): Disable (default)

When BSTAT_IEN is set to 1, I^{2}C slave bus status interrupt requests to the ITC are enabled. A bus status interrupt request occurs when BSTAT (D7/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/disables the I^{2}C slave receive interrupt.

1 (R/W): Enable

0 (R/W): Disable (default)

When RXRDY_IEN is set to 1, I^{2}C slave 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/disables the I^{2}C slave transmit interrupt.

1 (R/W): Enable

0 (R/W): Disable (default)

When TXEMP_IEN is set to 1, I^{2}C slave 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.

V.3.8 Precautions

- The I^{2}C slave module operating clock (PCLK) frequency must be set to eight-times or higher than the transfer rate during data transfer.
- When the asynchronous address detection function is enabled, the I^{2}C 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.
- When the master device of the I^{2}C bus, which has multiple slave devices connected including this IC, starts communication with another slave device, the I^{2}C slave 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 I2CS_SDA line status. When SELECTED (D1/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 I^{2}C slave is placed into asynchronous address detection mode, a DMS does not occur as in the condition above.
- 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 I^{2}C slave clock (I2CS_SCL) 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 (D8/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.
- When the clock stretch function has been disabled, transmit data/receive data must be written/read within the time shown below.

  During data transmission:

  Within 7 cycles of the I^{2}C slave clock (I2CS_SCL) after TXEMP is set (a transmit interrupt occurs)
  (See the precaution above for the first transmit data after slave selection.)

  During data reception:

  Within 7 cycles of the I^{2}C slave clock (I2CS_SCL) after RXRDY is set (a receive interrupt occurs)
  To return NAK, NAK_ANS should be set within this period.
- If the I^{2}C slave 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 I^{2}C slave address again).
  1. The transfer rate is set to 320 kbps or higher.
  2. The asynchronous address detection function is enabled.
  3. The I^{2}C slave module is placed into transfer standby state and OSC1 is used as the operating clock (PCLK).

V.4 SPI (SPI CH.0)

V.4.1 Configuration of the SPI CH.0

The S1C17002 equipped with a synchronous serial interface module (hereafter SPI CH.0). The SPI CH.0 supports both master and slave modes and performs 8-bit serial data transfer. Data transfer timing (clock phase and polarity variations) is selectable from among 4 types.

The SPI CH.0 includes a transmit data buffer and a receive data buffer separately from the shift registers, and can generate two types of interrupts (transmit data buffer empty and receive data buffer full), this makes it possible to process continuous serial data transfers simply in an interrupt handler.

Figure V.4.1.1 shows the structure of the SPI CH.0.

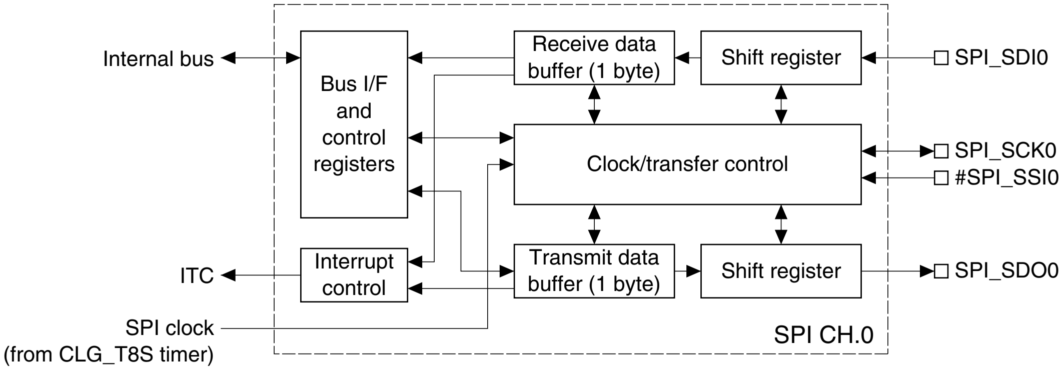


Figure V.4.1.1 Structure of SPI CH.0

V.4.2 SPI CH.0 I/O Pins

Table V.4.2.1 lists the SPI CH.0 pins.

Table V.4.2.1 List of SPI CH.0 Pins

Pin name	I/O	Size	Function
SPI_SDI0	I	1	SPI CH.0 data input pin This pin inputs serial data from the SPI bus.
SPI_SDO0	O	1	SPI CH.0 data output pin This pin outputs serial data to the SPI bus.
SPI_SCK0	I/O	1	SPI CH.0 external clock input/output pin This pin outputs the SPI clock when the SPI CH.0 is in master mode. This pin inputs an external clock when the SPI CH.0 is in slave mode.
#SPI_SSI0	I	1	SPI CH.0 slave select signal (active low) input pin A low level input to this pin selects this SPI CH.0 device in slave mode.

The SPI CH.0 input/output pins (SPI_SDI0, SPI_SDO0, SPI_SCK0, #SPI_SSI0) are shared with the I/O ports and they are initialized as general-purpose I/O port pins by default. Before using these pins for the SPI CH.0, the pin functions must be switched using the Port Function Select registers.

For details on switching pin function, Section I.3.3, “Switching Over the Multiplexed Pin Functions.”

V.4.3 SPI Clock

In master mode, the SPI CH.0 uses the internal clock output from the CLG_T8S timer as the SPI clock. This clock drives the shift register and is output from the SPI_SCK0 pin to the slave device.

Program the CLG_T8S timer so that it will output a clock according to the transfer rate. See Section II.4 "Clock Generator (CLG)," for controlling the timer.

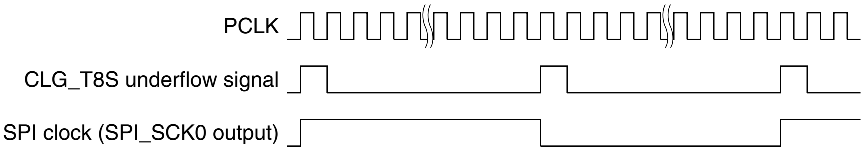


Figure V.4.3.1 SPI Clock in Master Mode

In slave mode, the SPI CH.0 inputs the SPI clock from the SPI_SCK0 pin.

Note: The duty ratio of the clock input from the SPI_SCK0 pin must be 50%.

V.4.4 Setting the Data Transfer Conditions

The SPI CH.0 can be set in master or slave mode and the SPI clock polarity and phase can be set using the SPI_CTL0 register.

The data length is fixed at eight bits.

Note: Make sure that the SPI CH.0 is disabled (SPEN/SPI_CTL0 register = 0) before selecting master/slave mode and setting the clock conditions.

∗ **SPEN**: SPI CH.0 Enable Bit in the SPI CH.0 Control (SPI_CTL0) Register (D0/0x4326)

Selecting master/slave mode

Use MSSL (D1/SPI_CTL0 register) to select whether the SPI CH.0 is set in master mode or slave mode. Setting MSSL to 1 selects master mode, and setting to 0 (default) selects slave mode. In master mode, the SPI CH.0 performs data transfer using the internal clock. In slave mode, the SPI CH.0 performs data transfer using a clock input from the master device.

∗ **MSSL**: Master/Slave Mode Select Bit in the SPI CH.0 Control (SPI_CTL0) Register (D1/0x4326)

Setting the SPI clock polarity and phase

Use CPOL (D2/SPI_CTL0 register) to select the SPI clock polarity. The SPI clock is configured as active low when CPOL is set to 1 or active high when CPOL is set to 0 (default).

∗ **CPOL**: Clock Polarity Select Bit in the SPI CH.0 Control (SPI_CTL0) Register (D2/0x4326)

The SPI clock phase is selected with CPHA (D3/SPI_CTL0 register).

∗ **CPHA**: Clock Phase Select Bit in the SPI CH.0 Control (SPI_CTL0) Register (D3/0x4326)

Setting these control bits determines the transfer timing as in the figure shown below.

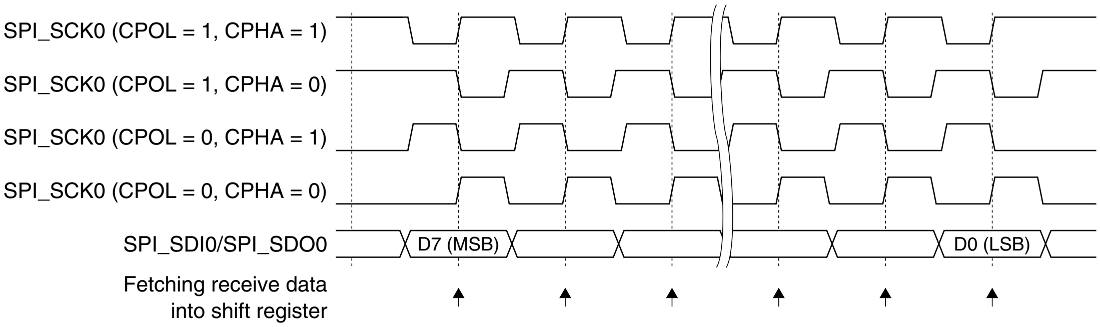

SPI_SCK0 (CPOL = 1, CPHA = 1)
SPI_SCK0 (CPOL = 1, CPHA = 0)
SPI_SCK0 (CPOL = 0, CPHA = 1)
SPI_SCK0 (CPOL = 0, CPHA = 0)
SPI_SDI0/SPI_SDO0
D7 (MSB)
D0 (LSB)
Fetching receive data into shift register

Figure V.4.4.1 Clock and Data Transfer Timing

V.4.5 Data Transmit/Receive Control

Before starting data transfer, set up the conditions as shown below.

(1) Set up the CLG_T8S timer to output the SPI clock. See Section II.4.
(2) Select either master or slave mode. See Section V.4.4.
(3) Set up the clock conditions. See Section V.4.4.
(4) Set up the interrupt conditions if the SPI CH.0 interrupt is used. See Section V.4.6.

Note: Make sure that the SPI CH.0 is disabled (SPEN/SPI_CTL0 register = 0) before setting the conditions above.

∗ **SPEN**: SPI CH.0 Enable Bit in the SPI CH.0 Control (SPI_CTL0) Register (D0/0x4326)

Enabling data transmission/reception

First, set the SPEN bit (D0/SPI_CTL0 register) to 1 to enable SPI CH.0 operation. This puts the SPI CH.0 in ready-to-transmit/receive status and enables clock input/output.

Note: Do not set the SPEN bit to 0 while the SPI CH.0 is transmitting/receiving data.

Data transmit control

To start transmission, write transmit data to the SPI_TXD0 register (0x4322).

∗ **SPI_TXD0**: SPI CH.0 Transmit Data Register (0x4322)

Data is written to the transmit data buffer and the SPI CH.0 starts data transmission.
The buffered data is sent to the shift register for transmission. In master mode, the SPI CH.0 starts outputting the clock from the SPI_SCK0 pin. In slave mode, the SPI CH.0 waits for clock input from the SPI_SCK0 pin. The data bits in the shift register are shifted one by one at the rising or falling edge of the clock configured with CPHA (D3/SPI_CTL0 register) and CPOL (D2/SPI_CTL0 register) (see Figure V.4.4.1), and are output from the SPI_SDO0 pin. The MSB of data is transmitted first.

∗ **CPHA**: Clock Phase Select Bit in the SPI CH.0 Control (SPI_CTL0) Register (D3/0x4326)
∗ **CPOL**: Clock Polarity Select Bit in the SPI CH.0 Control (SPI_CTL0) Register (D2/0x4326)

The SPI CH.0 provides two status flags for data transmit control, SPTBE (D0/SPI_ST0 register) and SPBSY (D2/SPI_ST0 register).

∗ **SPTBE**: Transmit Data Buffer Empty Flag Bit in the SPI CH.0 Status (SPI_ST0) Register (D0/0x4320)
∗ **SPBSY**: Transfer Busy Flag (Master)/ss Signal Low Flag (Slave) Bit in the SPI CH.0 Status (SPI_ST0) Register (D2/0x4320)

The SPTBE flag indicates the transmit data buffer status; it goes 0 when the application program writes data to the SPI_TXD0 register (transmit data buffer) and returns to 1 when the data in the transmit data buffer is sent to the shift register for transmitting. An interrupt can be generated when this flag goes 1 (see Section V.4.6). Use this interrupt or read the SPTBE flag to check that the transmit data buffer becomes empty when transmitting the next data. Although the transmit data buffer size is one byte, transmit data can be written while the previous data is being transmitted as the shift register is separately provided. However, make sure that the transmit data buffer is empty before writing transmit data. If data is written when the SPTBE flag is 0, the previous transmit data in the transmit data buffer is overwritten with the new data.

In master mode, the SPBSY flag indicates the shift register status; it goes 1 when transmit data is loaded from the transmit data buffer and returns to 0 upon completion of data transmission. Read this flag to check whether the SPI CH.0 is busy or idle.
In slave mode, the SPBSY flag indicates the SPI CH.0 slave select signal (#SPI_SSI0 pin) status; it goes 1 when this SPI CH.0 is selected as a slave or goes 0 when this SPI CH.0 is deselected.

Note: When the SPI CH.0 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.

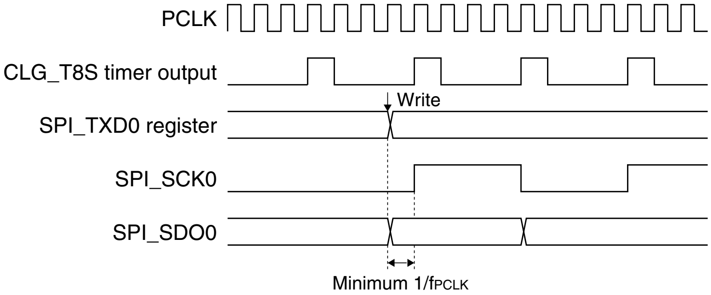


Figure V.4.5.1 SPI_SDO0 and SPI_SCK0 Change Timings when CPHA = 0

The half SPI_SCK0 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 receive control

In master mode, write dummy data to the SPI_TXD0 register (0x4322). Writing to the SPI_TXD0 register is used as the trigger for data receiving as well as starting data transmission. Also actual data to be transmitted can be written as the SPI CH.0 performs data transmission and reception simultaneously.
The SPI CH.0 starts output of the SPI clock from the SPI_SCK0 pin.

In slave mode, the SPI CH.0 waits for clock input from the SPI_SCK0 pin. When receiving data in slave mode without any data transmission, it is not necessary to write data to the SPI_TXD0 register. The receive process activates by the clock input from the master device. When performing data transmission and reception simultaneously, the transmit data should be written to the SPI_TXD0 register before a clock is input.

The data bits are fetched in the shift register one by one at the rising or falling edge of the clock configured with CPHA (D3/SPI_CTL0 register) and CPOL (D2/SPI_CTL0 register) (see Figure V.4.4.1). The MSB of data is received first.
When eight data bits are received in the shift register, the received data is loaded into the receive data buffer.

The received data in the buffer can be read from the SPI_RXD0 register (0x4324).

∗ **SPI_RXD0**: SPI CH.0 Receive Data Register (0x4324)

The SPI CH.0 provides the SPRBF flag (D1/SPI_ST0 register) for data receive control.

∗ **SPRBF**: Receive Data Buffer Full Flag Bit in the SPI CH.0 Status (SPI_ST0) Register (D1/0x4320)

The SPRBF flag indicates the receive data buffer status; it goes 1 when the data received in the shift register is loaded to the receive data buffer to indicate that the receive data can be read and returns to 0 when the data in the receive data buffer is read out from the SPI_RXD0 register. An interrupt can be generated when this flag goes 1 (see Section V.4.6). Use this interrupt or read the SPRBF flag to check that the receive data buffer contains valid data when reading received data. Although the receive data buffer size is one byte, the previous received data can be maintained while the next data is being received as the shift register is separately provided. However, be sure to read the receive data before the next data has been received. If the next data is received before the previous received data in the receive data buffer has been read, the previous received data is overwritten with the new data.

In master mode, the SPBSY flag that indicates the shift register status can be used as in data transmission.

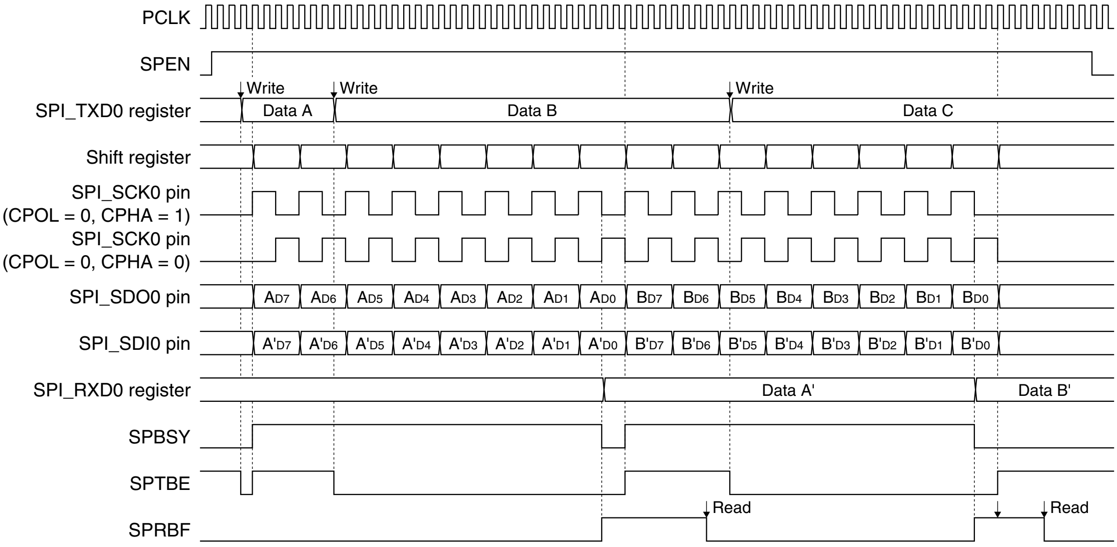

PCLK
SPEN
Write
Write
Write
SPI_TXD0 register
Data A
Data B
Data C
Shift register
SPI_SCK0 pin (CPOL = 0, CPHA = 1)
SPI_SCK0 pin (CPOL = 0, CPHA = 0)
SPI_SDO0 pin
AD7 AD6 AD5 AD4 AD3 AD2 AD1 AD0 BD7 BD6 BD5 BD4 BD3 BD2 BD1 BD0
SPI_SDI0 pin
A'D7 A'D6 A'D5 A'D4 A'D3 A'D2 A'D1 A'D0 B'D7 B'D6 B'D5 B'D4 B'D3 B'D2 B'D1 B'D0
SPI_RXD0 register
Data A'
Data B'
SPBSY
SPTBE
Read
Read
SPRBF

Figure V.4.5.2 Data Transmit/Receive Timing Chart

Disabling data transmission/reception

After data transfer (both transmission and reception) has finished, write 0 to the SPEN bit to disable data transmission/reception.

Always make sure that the SPTBE flag is 1 and SPBSY flag is 0 before data transmission/reception is disabled. The data being transferred cannot be guaranteed if SPEN is set to 0 during transmitting/receiving.

V.4.6 SPI CH.0 Interrupt

The SPI CH.0 can generate the following two types of interrupts:
- Transmit buffer empty interrupt
- Receive buffer full interrupt

The SPI CH.0 has one interrupt signal to be output to the interrupt controller (ITC) and it is shared with the two causes of interrupt. To determine the cause of interrupt that has occurred, read the status flags.

Transmit buffer empty interrupt

Set the SPTIE bit (D4/SPI_CTL0 register) to 1 when using this interrupt. If SPTIE is set to 0 (default), an interrupt request by this cause will not be sent to the ITC.

∗ **SPTIE**: Transmit Data Buffer Empty Interrupt Enable Bit in the SPI CH.0 Control (SPI_CTL0) Register (D4/0x4326)

When the transmit data set in the transmit data buffer is transferred to the shift register, the SPI CH.0 sets the SPTBE bit (D0/SPI_ST0 register) to 1 to indicate that the transmit data buffer is empty. At the same time, the SPI CH.0 outputs an interrupt request pulse to the ITC if the transmit buffer empty interrupt has been enabled (SPTIE = 1).

∗ **SPTBE**: Transmit Data Buffer Empty Flag Bit in the SPI CH.0 Status (SPI_ST0) Register (D0/0x4320)

If other interrupt conditions are satisfied, an interrupt is generated.
The SPI CH.0 interrupt handler routine should read the SPTBE flag to check if the interrupt has occurred due to a transmit buffer empty or another cause. When SPTBE = 1, the SPI CH.0 interrupt handler routine can write the next transmit data to the transmit data buffer.

Receive buffer full interrupt

Set the SPRIE bit (D5/SPI_CTL0 register) to 1 when using this interrupt. If SPRIE is set to 0 (default), an interrupt request by this cause will not be sent to the ITC.

∗ **SPRIE**: Receive Data Buffer Full Interrupt Enable Bit in the SPI CH.0 Control (SPI_CTL0) Register (D5/0x4326)

When data received in the shift register is loaded to the receive data buffer, the SPI CH.0 sets the SPRBF bit (D1/SPI_ST0 register) to 1 to indicate that the received data buffer is full. At the same time, the SPI CH.0 outputs an interrupt request pulse to the ITC if the receive buffer full interrupt has been enabled (SPRIE = 1).

∗ **SPRBF**: Receive Data Buffer Full Flag Bit in the SPI CH.0 Status (SPI_ST0) Register (D1/0x4320)

If other interrupt conditions are satisfied, an interrupt is generated.
The SPI CH.0 interrupt handler routine should read the SPRBF flag to check if the interrupt has occurred due to a receive buffer full or another cause. When SPRBF = 1, the SPI CH.0 interrupt handler routine can read the received data from the receive data buffer.

ITC registers for SPI CH.0 interrupts

The following shows the control bits of the ITC provided for the SPI CH.0:

Interrupt flag
∗ **IIFT6**: SPI CH.0 Interrupt Flag Bit in the Interrupt Flag (ITC_IFLG) Register (D14/0x4300)

Interrupt enable bit
∗ **IIEN6**: SPI CH.0 Interrupt Enable Bit in the Interrupt Enable (ITC_EN) Register (D14/0x4302)

Interrupt level setup bits
∗ **IILV6[2:0]**: SPI CH.0 Interrupt Level Bits in the Internal Interrupt Level Setup (ITC_ILV3) Register 3 (D[2:0]/0x4314)

When the SPI CH.0 outputs an interrupt request pulse, the interrupt flag IIFT6 is set to 1.
If the interrupt enable bit IIEN6 has been set to 1, the ITC sends an interrupt request to the S1C17 Core. To disable the SPI CH.0 interrupt, set the IIEN6 bit to 0.
The IIFT6 flag is always set to 1 by the SPI CH.0 interrupt request pulse, regardless of how the IIEN6 bit is set (even when set to 0).
The interrupt level setup bits IILV6[2:0] set the interrupt level (0 to 7) of the SPI CH.0 interrupt.

An interrupt request to the S1C17 Core is accepted only when all the conditions described below are met.

- The interrupt enable bit is set to 1.
- The IE (Interrupt Enable) bit of the PSR (Processor Status Register) in the S1C17 Core is set to 1.
- The SPI CH.0 interrupt has a higher interrupt level than the value that is set in the IL field of the PSR.
- No other cause of interrupt having higher priority, such as NMI, has occurred.

For details on these interrupt control registers, as well as the device operation when an interrupt has occurred, see Section III.1, “Interrupt Controller (ITC).”

Interrupt vector

The following shows the vector number and vector address for the SPI CH.0 interrupt:

Vector number: 18 (0x12)
Vector address: TTBR + 0x48

V.4.7 Details of Control Registers

Table V.4.7.1 List of SPI CH.0 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 CH.0 mode and enables data transfer.

The following describes each SPI CH.0 register. These are all 16-bit registers.

Notes:
- When setting the registers, be sure to write a 0, and not a 1, for all “reserved bits.”
- Be sure to use 16-bit access instructions for reading/writing from/to the SPI CH.0 registers. The SPI CH.0 registers do not allow reading/writing using 32-bit and 8-bit access instructions.

0x4320: SPI CH.0 Status Register (SPI_ST0)

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	

D[15:3] Reserved

D2 SPBSY: Transfer Busy Flag (Master)/ss Signal Low Flag (Slave) Bit

Master mode

Indicates the SPI CH.0 transmit/receive operation status.

1 (R): Busy

0 (R): Idle (default)

SPBSY is set to 1 when the SPI CH.0 starts data transmission/reception in master mode and stays 1 while data transmission/reception is in progress. SPBSY is reset to 0 upon completion of the transmit/receive operation.

Slave mode

Indicates the slave select (#SPI_SSI0) signal status.

1 (R): Low level (SPI CH.0 is selected)

0 (R): High level (SPI CH.0 is deselected) (default)

SPBSY is set to 1 when the master device activates the #SPI_SSI0 signal to select this SPI CH.0 (slave device), and is reset to 0 when the master device negates the #SPI_SSI0 signal to deselect this SPI CH.0.

D1 SPRBF: Receive Data Buffer Full Flag Bit

Indicates the receive data buffer status.

1 (R): Full

0 (R): Not full (default)

SPRBF is set to 1 when the data received in the shift register is loaded to the receive data buffer (receive operation completed), indicating that the received data can be read out. This bit is reset to 0 when the data is read out from the SPI_RXD0 register (0x4324).

D0 SPTBE: Transmit Data Buffer Empty Flag Bit

Indicates the transmit data buffer status.

1 (R): Empty (default)

0 (R): Not empty

SPTBE is reset to 0 when transmit data is written to the SPI_TXD0 register (transmit data buffer, 0x4322) and is set to 1 when the written data is transferred to the shift register (transmit operation started).

Transmit data should be written to the SPI_TXD0 register when this bit = 1.

0x4322: SPI CH.0 Transmit Data Register (SPI_TXD0)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SPI CH.0 Transmit Data Register (SPI_TXD0)	**0x4322** (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	**SPTDB[7:0]**	SPI CH.0 transmit data buffer SPTDB7 = MSB SPTDB0 = LSB	0x0 to 0xff	0x0	R/W	

D[15:8] Reserved

D[7:0] SPTDB[7:0]: SPI CH.0 Transmit Data Buffer Bits

Set transmit data to be written to the transmit data buffer. (Default: 0x0)

In master mode, data transmission begins by writing data to this register. In slave mode, the register contents are transferred to the shift register to start data transmission when a clock is input from the master device.

SPTBE (D0/SPI_ST0 register) is set to 1 (empty) when the data is transferred to the shift register. At the same time, a cause of transmit data buffer empty interrupt occurs. The next transmit data can be written to the register at any time thereafter, even when the SPI CH.0 is sending data.

The serial-converted data is output from the SPI_SDO0 pin beginning with the MSB, in which the bits set to 1 are output as high-level signals and those set to 0 output as low-level signals.

Note: Make sure that SPEN is set to 1 before writing data to the SPI_TXD0 register to start data transmission/reception.

0x4324: SPI CH.0 Receive Data Register (SPI_RXD0)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SPI CH.0 Receive Data Register (SPI_RXD0)	**0x4324** (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	**SPRDB[7:0]**	SPI CH.0 receive data buffer SPRDB7 = MSB SPRDB0 = LSB	0x0 to 0xff	0x0	R	

D[15:8] Reserved

D[7:0] SPRDB[7:0]: SPI CH.0 Receive Data Buffer Bits

Stores received data. (Default: 0x0)

When a receive operation is completed and the data received in the shift register is loaded to the receive data buffer, SPRBF (D1/SPI_ST0 register) is set to 1 (buffer full). At the same time, a cause of receive data buffer full interrupt occurs. Thereafter, the data can be read out at any time before a receive operation for the next data is completed.

If the next data receive operation is completed before this register is read out, the data in it is overwritten with the newly received data.

The serial data input from the SPI_SDI0 pin is converted into parallel data beginning with the MSB, with the high-level signals changed to 1s and the low-level signals changed to 0s. The resulting data is stored in this register.

SPI_RXD0 is a read-only register, so no data can be written to it.

0x4326: SPI CH.0 Control Register (SPI_CTL0)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
SPI CH.0 Control Register (SPI_CTL0)	**0x4326** (16 bits)	D15–6	–	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 CH.0 enable	1	Enable	0	Disable	0	R/W	

D[15:6] Reserved

D5 SPRIE: Receive Data Buffer Full Interrupt Enable Bit

Enables/disables SPI CH.0 interrupt caused by receive data buffer full.

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

When SPRIE is set to 1, SPI CH.0 (receive data buffer full) interrupt requests to the ITC are enabled. A receive data buffer full interrupt request occurs when the data received in the shift register is loaded to the receive data buffer (receive operation completed).
When SPRIE is set to 0, SPI CH.0 interrupts caused by receive data full are not generated.

D4 SPTIE: Transmit Data Buffer Empty Interrupt Enable Bit

Enables/disables SPI CH.0 interrupt caused by transmit data buffer empty.

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

When SPTIE is set to 1, SPI CH.0 (transmit data buffer empty) interrupt requests to the ITC are enabled. A transmit data buffer empty interrupt request occurs when the data written to the transmit data buffer is transferred to the shift register (transmit operation started).
When SPTIE is set to 0, SPI CH.0 interrupts caused by transmit data buffer empty are not generated.

D3 CPHA: Clock Phase Select Bit

Selects the phase of the SPI clock. (Default: 0)
This bit controls the data transfer timing in conjunction with the CPOL (D2) bit (see Figure V.4.7.1).

D2 CPOL: Clock Polarity Select Bit

Selects the polarity of the SPI clock.

1 (R/W): Active low
0 (R/W): Active high (default)

This bit controls the data transfer timing in conjunction with the CPHA (D3) bit (see Figure V.4.7.1).

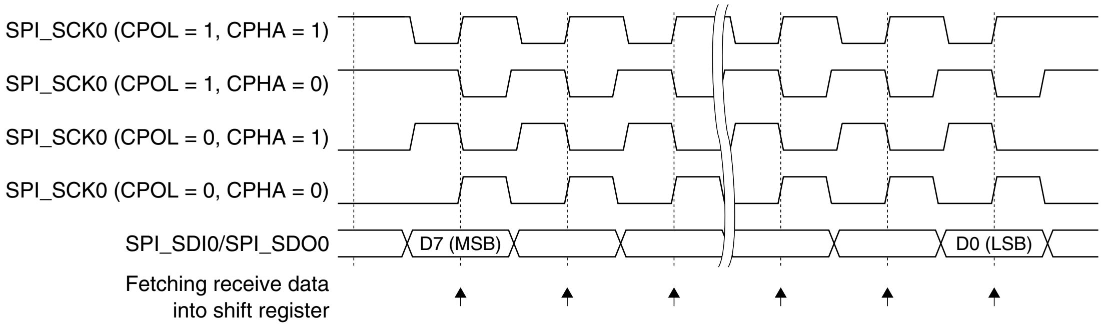

SPI_SCK0 (CPOL = 1, CPHA = 1)
SPI_SCK0 (CPOL = 1, CPHA = 0)
SPI_SCK0 (CPOL = 0, CPHA = 1)
SPI_SCK0 (CPOL = 0, CPHA = 0)
SPI_SDI0/SPI_SDO0
D7 (MSB)
D0 (LSB)
Fetching receive data into shift register

Figure V.4.7.1 Clock and Data Transfer Timing

D1 MSSL: Master/Slave Mode Select Bit

Sets the SPI CH.0 in master or slave mode.
1 (R/W): Master mode
0 (R/W): Slave mode (default)

Setting MSSL to 1 selects master mode, and setting to 0 selects slave mode. In master mode, the SPI CH.0 performs data transfer using the clock generated by the CLG_T8S timer. In slave mode, the SPI CH.0 performs data transfer using a clock input from the master device.

D0 SPEN: SPI CH.0 Enable Bit

Enables/disables operation of the SPI CH.0.
1 (R/W): Enable
0 (R/W): Disable (default)

When SPEN is set to 1, the SPI CH.0 is activated and data transfer is enabled.
When SPEN is set to 0, the SPI CH.0 goes off.

Note: Make sure that the SPEN bit is 0 before setting the CPHA, CPOL, and MSSL bits.

V.4.8 Precautions

- Be sure to use 16-bit access instructions for reading/writing from/to the SPI CH.0 registers (0x4320 to 0x4326). The SPI CH.0 registers do not allow reading/writing using 32-bit and 8-bit access instructions.
- Do not access the SPI_CTL0 register (0x4326), while the SPBSY flag (D2/SPI_ST0 register) is set to 1 (during data transfer).

  * **SPBSY**: Transfer Busy Flag (Master)/ss Signal Low Flag (Slave) Bit in the SPI CH.0 Status (SPI_ST0) Register (D2/0x4320)
- When the SPI CH.0 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.

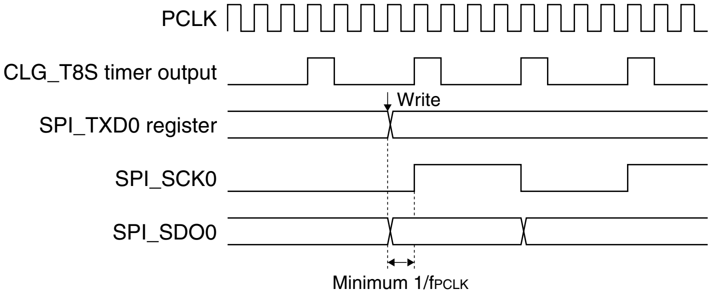


Figure V.4.8.1 SPI_SDO0 and SPI_SCK0 Change Timings when CPHA = 0

The half SPI_SCK0 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.

- Make sure that SPEN is set to 1 before writing data to the SPI_TXD0 register to start data transmission/reception.

V.5 Extended SPI (SPI CH.1)

V.5.1 Configuration of the SPI CH.1

The S1C17002 equipped with a synchronous serial interface module (hereafter SPI CH.1). The SPI CH.1 supports both master and slave modes and performs 8-bit serial data transfer. Data transfer timing (clock phase and polarity variations) is selectable from among 4 types.

The SPI CH.1 includes a transmit data buffer and a receive data buffer separately from the shift registers, and can generate two types of interrupts (transmit data buffer empty and receive data buffer full), this makes it possible to process continuous serial data transfers simply in an interrupt handler.

Figure V.5.1.1 shows the structure of the SPI CH.1.

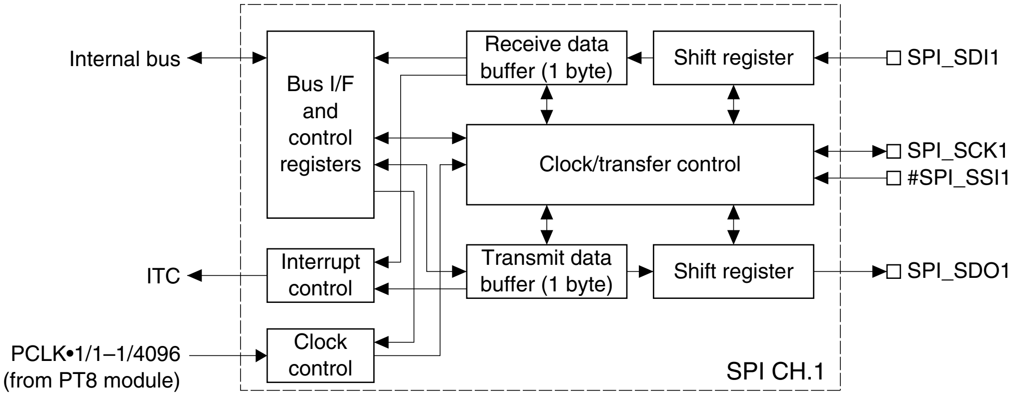


Figure V.5.1.1 Structure of SPI CH.1

V.5.2 SPI CH.1 I/O Pins

Table V.5.2.1 lists the SPI CH.1 pins.

Table V.5.2.1 List of SPI CH.1 Pins

Pin name	I/O	Size	Function
SPI_SDI1	I	1	SPI CH.1 data input pin This pin inputs serial data from the SPI bus.
SPI_SDO1	O	1	SPI CH.1 data output pin This pin outputs serial data to the SPI bus.
SPI_SCK1	I/O	1	SPI CH.1 external clock input/output pin This pin outputs the SPI clock when the SPI CH.1 is in master mode. This pin inputs an external clock when the SPI CH.1 is in slave mode.
#SPI_SSI1	I	1	SPI CH.1 slave select signal (active low) input pin A low level input to this pin selects this SPI CH.1 device in slave mode.

The SPI CH.1 input/output pins (SPI_SDI1, SPI_SDO1, SPI_SCK1, #SPI_SSI1) are shared with the I/O ports and they are initialized as general-purpose I/O port pins by default. Before using these pins for the SPI CH.1, the pin functions must be switched using the Port Function Select registers.

For details on switching pin function, Section I.3.3, “Switching Over the Multiplexed Pin Functions.”

V.5.3 SPI Clock

In master mode, the SPI CH.1 uses the internal clock output from the prescaler in the 8-bit timer (PT8) module.
To operate the PT8 prescaler for SPI CH.1, the operating clocks for both the PT8 and SPI CH.1 modules must be supplied from the CMU by setting the control bits shown below to 1.

∗ **PT8_CLK_EN**: 8-bit Programmable Timer Clock Control Bit in the Gated Clock Control 1 (CMU_GATEDCLK1) Register (D1/0x4907)
∗ **SPI_CLK_EN**: SPI CH.1 Module Clock Control Bit in the Gated Clock Control 2 (CMU_GATEDCLK2) Register (D5/0x4908)

The PT8 prescaler divides PCLK (with the same frequency as the system clock) by 1 to 4096 to generate 13 clocks. Select one of the prescaler output clocks using SPI_CLK[3:0] (D[3:0]/SPI_CLK1 register).

∗ **SPI_CLK[3:0]**: SPI CH.1 Clock Division Ratio Selection Bits in the SPI CH.1 Clock Control (SPI_CLK1) Register (D[3:0]/0x5708)

Table V.5.3.1 Selecting the SPI Clock

SPI_CLK[3:0]	Prescaler output clock
0xf	Reserved
0xe	Reserved
0xd	Reserved
0xc	PCLK•1/4096
0xb	PCLK•1/2048
0xa	PCLK•1/1024
0x9	PCLK•1/512
0x8	PCLK•1/256
0x7	PCLK•1/128
0x6	PCLK•1/64
0x5	PCLK•1/32
0x4	PCLK•1/16
0x3	PCLK•1/8
0x2	PCLK•1/4
0x1	PCLK•1/2
0x0	PCLK•1/1

(Default: 0x0)

The selected clock is input to the SPI CH.1 by setting SPI_CKE (D4/SPI_CLK1 register) to 1.

∗ **SPI_CKE**: SPI CH.1 Clock Enable Bit in the SPI CH.1 Clock Control (SPI_CLK1) Register (D4/0x5708)

In slave mode, the SPI CH.1 inputs the SPI clock from the SPI_SCK1 pin.

Note: The duty ratio of the clock input from the SPI_SCK1 pin must be 50%.

V.5.4 Setting the Data Transfer Conditions

The SPI CH.1 can be set in master or slave mode and the SPI clock polarity and phase can be set using the SPI_CTL1 register.

The data length is fixed at eight bits.

Note: Make sure that the SPI CH.1 is disabled (SPEN/SPI_CTL1 register = 0) before selecting master/slave mode and setting the clock conditions.

∗ **SPEN**: SPI CH.1 Enable Bit in the SPI CH.1 Control (SPI_CTL1) Register (D0/0x5706)

Selecting master/slave mode

Use MSSL (D1/SPI_CTL1 register) to select whether the SPI CH.1 is set in master mode or slave mode. Setting MSSL to 1 selects master mode, and setting to 0 (default) selects slave mode. In master mode, the SPI CH.1 performs data transfer using the internal clock. In slave mode, the SPI CH.1 performs data transfer using a clock input from the master device.

∗ **MSSL**: Master/Slave Mode Select Bit in the SPI CH.1 Control (SPI_CTL1) Register (D1/0x5706)

Setting the SPI clock polarity and phase

Use CPOL (D2/SPI_CTL1 register) to select the SPI clock polarity. The SPI clock is configured as active low when CPOL is set to 1 or active high when CPOL is set to 0 (default).

∗ **CPOL**: Clock Polarity Select Bit in the SPI CH.1 Control (SPI_CTL1) Register (D2/0x5706)

The SPI clock phase is selected with CPHA (D3/SPI_CTL1 register).

∗ **CPHA**: Clock Phase Select Bit in the SPI CH.1 Control (SPI_CTL1) Register (D3/0x5706)

Setting these control bits determines the transfer timing as in the figure shown below.

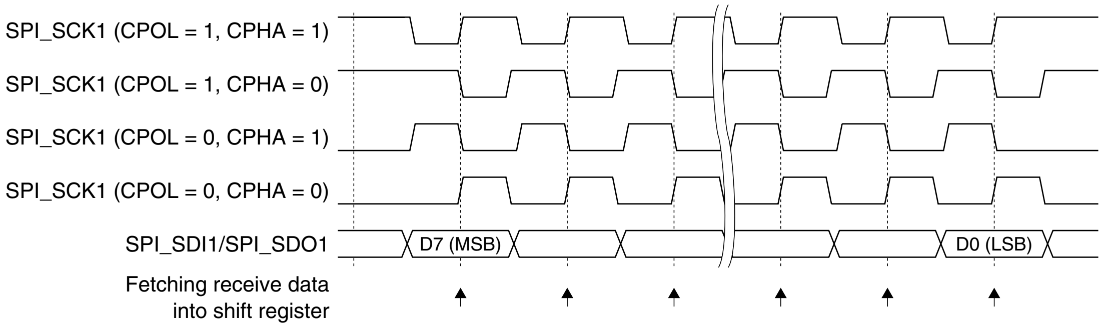

SPI_SCK1 (CPOL = 1, CPHA = 1)
SPI_SCK1 (CPOL = 1, CPHA = 0)
SPI_SCK1 (CPOL = 0, CPHA = 1)
SPI_SCK1 (CPOL = 0, CPHA = 0)
SPI_SDI1/SPI_SDO1
D7 (MSB)
D0 (LSB)
Fetching receive data into shift register

Figure V.5.4.1 Clock and Data Transfer Timing

V.5.5 Data Transmit/Receive Control

Before starting data transfer, set up the conditions as shown below.

(1) Set up the PT8 prescaler to output the SPI clock. See Section V.5.3.
(2) Select either master or slave mode. See Section V.5.4.
(3) Set up the clock conditions. See Section V.5.4.
(4) Set up the interrupt conditions if the SPI CH.1 interrupt is used. See Section V.5.6.

Note: Make sure that the SPI CH.1 is disabled (SPEN/SPI_CTL1 register = 0) before setting the conditions above.

∗ **SPEN**: SPI CH.1 Enable Bit in the SPI CH.1 Control (SPI_CTL1) Register (D0/0x5706)

Enabling data transmission/reception

First, set the SPEN bit (D0/SPI_CTL1 register) to 1 to enable SPI CH.1 operation. This puts the SPI CH.1 in ready-to-transmit/receive status and enables clock input/output.

Note: Do not set the SPEN bit to 0 while the SPI CH.1 is transmitting/receiving data.

Data transmit control

To start transmission, write transmit data to the SPI_TXD1 register (0x5702).

∗ **SPI_TXD1**: SPI CH.1 Transmit Data Register (0x5702)

Data is written to the transmit data buffer and the SPI CH.1 starts data transmission.
The buffered data is sent to the shift register for transmission. In master mode, the SPI CH.1 starts outputting the clock from the SPI_SCK1 pin. In slave mode, the SPI CH.1 waits for clock input from the SPI_SCK1 pin. The data bits in the shift register are shifted one by one at the rising or falling edge of the clock configured with CPHA (D3/SPI_CTL1 register) and CPOL (D2/SPI_CTL1 register) (see Figure V.5.4.1), and are output from the SPI_SDO1 pin. The MSB of data is transmitted first.

∗ **CPHA**: Clock Phase Select Bit in the SPI CH.1 Control (SPI_CTL1) Register (D3/0x5706)
∗ **CPOL**: Clock Polarity Select Bit in the SPI CH.1 Control (SPI_CTL1) Register (D2/0x5706)

The SPI CH.1 provides two status flags for data transmit control, SPTBE (D0/SPI_ST1 register) and SPBSY (D2/SPI_ST1 register).

∗ **SPTBE**: Transmit Data Buffer Empty Flag Bit in the SPI CH.1 Status (SPI_ST1) Register (D0/0x5700)
∗ **SPBSY**: Transfer Busy Flag (Master)/ss Signal Low Flag (Slave) Bit in the SPI CH.1 Status (SPI_ST1) Register (D2/0x5700)

The SPTBE flag indicates the transmit data buffer status; it goes 0 when the application program writes data to the SPI_TXD1 register (transmit data buffer) and returns to 1 when the data in the transmit data buffer is sent to the shift register for transmitting. An interrupt can be generated when this flag goes 1 (see Section V.5.6). Use this interrupt or read the SPTBE flag to check that the transmit data buffer becomes empty when transmitting the next data. Although the transmit data buffer size is one byte, transmit data can be written while the previous data is being transmitted as the shift register is separately provided. However, make sure that the transmit data buffer is empty before writing transmit data. If data is written when the SPTBE flag is 0, the previous transmit data in the transmit data buffer is overwritten with the new data.

In master mode, the SPBSY flag indicates the shift register status; it goes 1 when transmit data is loaded from the transmit data buffer and returns to 0 upon completion of data transmission. Read this flag to check whether the SPI CH.1 is busy or idle.
In slave mode, the SPBSY flag indicates the SPI CH.1 slave select signal (#SPI_SSI1 pin) status; it goes 1 when this SPI CH.1 is selected as a slave or goes 0 when this SPI CH.1 is deselected.

Note: When the SPI CH.1 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.

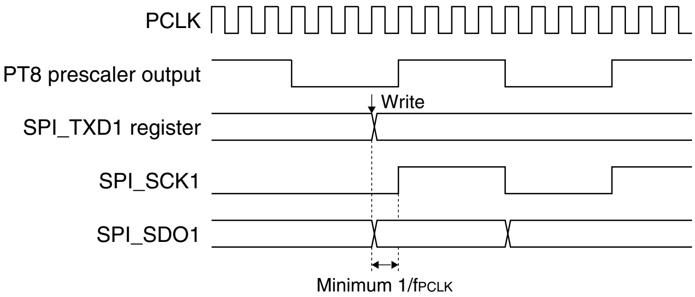


Figure V.5.5.1 SPI_SDO1 and SPI_SCK1 Change Timings when CPHA = 0

The half SPI_SCK1 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 receive control

In master mode, write dummy data to the SPI_TXD1 register (0x5702). Writing to the SPI_TXD1 register is used as the trigger for data receiving as well as starting data transmission. Also actual data to be transmitted can be written as the SPI CH.1 performs data transmission and reception simultaneously.
The SPI CH.1 starts output of the SPI clock from the SPI_SCK1 pin.

In slave mode, the SPI CH.1 waits for clock input from the SPI_SCK1 pin. When receiving data in slave mode without any data transmission, it is not necessary to write data to the SPI_TXD1 register. The receive process activates by the clock input from the master device. When performing data transmission and reception simultaneously, the transmit data should be written to the SPI_TXD1 register before a clock is input.

The data bits are fetched in the shift register one by one at the rising or falling edge of the clock configured with CPHA (D3/SPI_CTL1 register) and CPOL (D2/SPI_CTL1 register) (see Figure V.5.4.1). The MSB of data is received first.
When eight data bits are received in the shift register, the received data is loaded into the receive data buffer.

The received data in the buffer can be read from the SPI_RXD1 register (0x5704).

∗ **SPI_RXD1**: SPI CH.1 Receive Data Register (0x5704)

The SPI CH.1 provides the SPRBF flag (D1/SPI_ST1 register) for data receive control.

∗ **SPRBF**: Receive Data Buffer Full Flag Bit in the SPI CH.1 Status (SPI_ST1) Register (D1/0x5700)

The SPRBF flag indicates the receive data buffer status; it goes 1 when the data received in the shift register is loaded to the receive data buffer to indicate that the receive data can be read and returns to 0 when the data in the receive data buffer is read out from the SPI_RXD1 register. An interrupt can be generated when this flag goes 1 (see Section V.5.6). Use this interrupt or read the SPRBF flag to check that the receive data buffer contains valid data when reading received data. Although the receive data buffer size is one byte, the previous received data can be maintained while the next data is being received as the shift register is separately provided. However, be sure to read the receive data before the next data has been received. If the next data is received before the previous received data in the receive data buffer has been read, the previous received data is overwritten with the new data.

In master mode, the SPBSY flag that indicates the shift register status can be used as in data transmission.

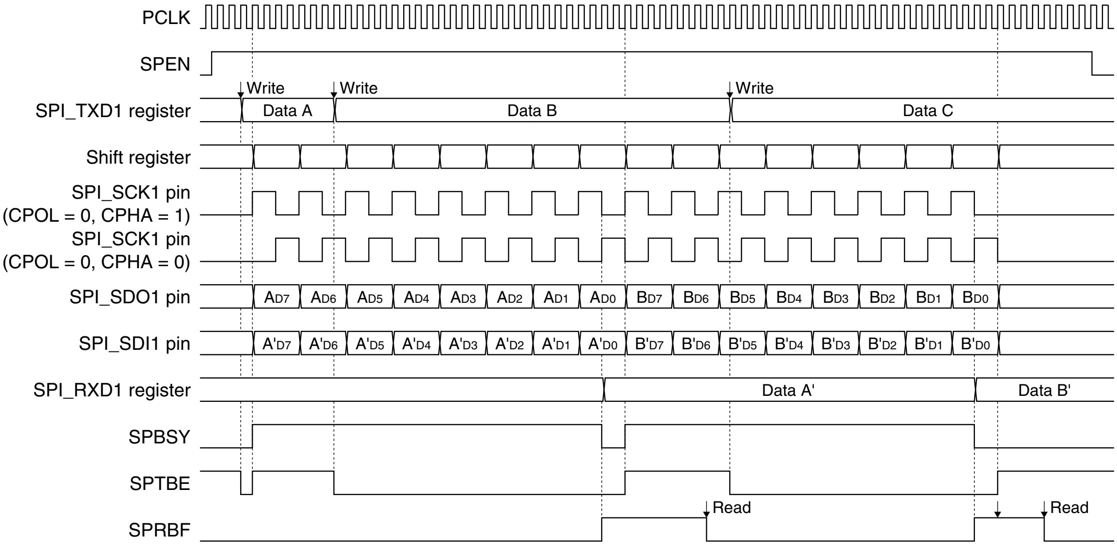

PCLK
SPEN
Write
Write
Write
SPI_TXD1 register
Data A
Data B
Data C
Shift register
SPI_SCK1 pin (CPOL = 0, CPHA = 1)
SPI_SCK1 pin (CPOL = 0, CPHA = 0)
SPI_SDO1 pin
AD7 AD6 AD5 AD4 AD3 AD2 AD1 AD0 BD7 BD6 BD5 BD4 BD3 BD2 BD1 BD0
SPI_SDI1 pin
A'D7 A'D6 A'D5 A'D4 A'D3 A'D2 A'D1 A'D0 B'D7 B'D6 B'D5 B'D4 B'D3 B'D2 B'D1 B'D0
SPI_RXD1 register
Data A'
Data B'
SPBSY
SPTBE
Read
Read
SPRBF

Figure V.5.5.2 Data Transmit/Receive Timing Chart

Disabling data transmission/reception

After data transfer (both transmission and reception) has finished, write 0 to the SPEN bit to disable data transmission/reception.

Always make sure that the SPTBE flag is 1 and SPBSY flag is 0 before data transmission/reception is disabled. The data being transferred cannot be guaranteed if SPEN is set to 0 during transmitting/receiving.

V.5.6 SPI CH.1 Interrupt

The SPI CH.1 can generate the following two types of interrupts:
- Transmit buffer empty interrupt
- Receive buffer full interrupt

The SPI CH.1 has one interrupt signal to be output to the interrupt controller (ITC) and it is shared with the two causes of interrupt. To determine the cause of interrupt that has occurred, read the status flags.

Transmit buffer empty interrupt

Set the SPTIE bit (D4/SPI_CTL1 register) to 1 when using this interrupt. If SPTIE is set to 0 (default), an interrupt request by this cause will not be sent to the ITC.

∗ **SPTIE**: Transmit Data Buffer Empty Interrupt Enable Bit in the SPI CH.1 Control (SPI_CTL1) Register (D4/0x5706)

When the transmit data set in the transmit data buffer is transferred to the shift register, the SPI CH.1 sets the SPTBE bit (D0/SPI_ST1 register) to 1 to indicate that the transmit data buffer is empty. At the same time, the SPI CH.1 outputs an interrupt request pulse to the ITC if the transmit buffer empty interrupt has been enabled (SPTIE = 1).

∗ **SPTBE**: Transmit Data Buffer Empty Flag Bit in the SPI CH.1 Status (SPI_ST1) Register (D0/0x5700)

If other interrupt conditions are satisfied, an interrupt is generated.
The SPI CH.1 interrupt handler routine should read the SPTBE flag to check if the interrupt has occurred due to a transmit buffer empty or another cause. When SPTBE = 1, the SPI CH.1 interrupt handler routine can write the next transmit data to the transmit data buffer.

Receive buffer full interrupt

Set the SPRIE bit (D5/SPI_CTL1 register) to 1 when using this interrupt. If SPRIE is set to 0 (default), an interrupt request by this cause will not be sent to the ITC.

∗ **SPRIE**: Receive Data Buffer Full Interrupt Enable Bit in the SPI CH.1 Control (SPI_CTL1) Register (D5/0x5706)

When data received in the shift register is loaded to the receive data buffer, the SPI CH.1 sets the SPRBF bit (D1/SPI_ST1 register) to 1 to indicate that the received data buffer is full. At the same time, the SPI CH.1 outputs an interrupt request pulse to the ITC if the receive buffer full interrupt has been enabled (SPRIE = 1).

∗ **SPRBF**: Receive Data Buffer Full Flag Bit in the SPI CH.1 Status (SPI_ST1) Register (D1/0x5700)

If other interrupt conditions are satisfied, an interrupt is generated.
The SPI CH.1 interrupt handler routine should read the SPRBF flag to check if the interrupt has occurred due to a receive buffer full or another cause. When SPRBF = 1, the SPI CH.1 interrupt handler routine can read the received data from the receive data buffer.

ITC registers for SPI CH.1 interrupts

The following shows the control bits of the ITC provided for the SPI CH.1:

Interrupt flag
∗ **AIFT10**: SPI CH.1 Interrupt Flag Bit in the Additional Interrupt Flag (ITC_AIFLG) Register (D10/0x42e0)

Interrupt enable bit
∗ **AIEN10**: SPI CH.1 Interrupt Enable Bit in the Additional Interrupt Enable (ITC_AEN) Register (D10/0x42e2)

Interrupt level setup bits
∗ **AILV10[2:0]**: SPI CH.1 Interrupt Level Bits in the Additional Interrupt Level Setup (ITC_AILV5) Register 5 (D[2:0]/0x42f0)

When the SPI CH.1 outputs an interrupt request pulse, the interrupt flag AIFT10 is set to 1.
If the interrupt enable bit AIEN10 has been set to 1, the ITC sends an interrupt request to the S1C17 Core. To disable the SPI CH.1 interrupt, set the AIEN10 bit to 0.
The AIFT10 flag is always set to 1 by the SPI CH.1 interrupt request pulse, regardless of how the AIEN10 bit is set (even when set to 0).
The interrupt level setup bits AILV10[2:0] set the interrupt level (0 to 7) of the SPI CH.1 interrupt.

An interrupt request to the S1C17 Core is accepted only when all the conditions described below are met.

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

For details on these interrupt control registers, as well as the device operation when an interrupt has occurred, see Section III.1, “Interrupt Controller (ITC).”

Interrupt vector

The following shows the vector number and vector address for the SPI CH.1 interrupt:

Vector number: 26 (0x1a)
Vector address: TTBR + 0x68

V.5.7 Details of Control Registers

Table V.5.7.1 List of SPI CH.1 Registers

Address	Register name		Function
0x5700	SPI_ST1	SPI CH.1 Status Register	Indicates transfer and buffer statuses.
0x5702	SPI_TXD1	SPI CH.1 Transmit Data Register	Transmit data
0x5704	SPI_RXD1	SPI CH.1 Receive Data Register	Receive data
0x5706	SPI_CTL1	SPI CH.1 Control Register	Sets the SPI CH.1 mode and enables data transfer.
0x5708	SPI_CLK1	SPI CH.1 Clock Control Register	Sets up the SPI clock.

The following describes each SPI CH.1 register. These are all 16-bit registers.

Notes:
- When setting the registers, be sure to write a 0, and not a 1, for all “reserved bits.”
- Be sure to use 16-bit access instructions for reading/writing from/to the SPI CH.1 registers. The SPI CH.1 registers do not allow reading/writing using 32-bit and 8-bit access instructions.

0x5700: SPI CH.1 Status Register (SPI_ST1)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
SPI CH.1 Status Register (SPI_ST1)	**0x5700** (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 (Master)/ss Signal Low Flag (Slave) Bit

Master mode

Indicates the SPI CH.1 transmit/receive operation status.

1 (R): Busy
0 (R): Idle (default)

SPBSY is set to 1 when the SPI CH.1 starts data transmission/reception in master mode and stays 1 while data transmission/reception is in progress. SPBSY is reset to 0 upon completion of the transmit/receive operation.

Slave mode

Indicates the slave select (#SPI_SSI1) signal status.

1 (R): Low level (SPI CH.1 is selected)
0 (R): High level (SPI CH.1 is deselected) (default)

SPBSY is set to 1 when the master device activates the #SPI_SSI1 signal to select this SPI CH.1 (slave device), and is reset to 0 when the master device negates the #SPI_SSI1 signal to deselect this SPI CH.1.

D1 SPRBF: Receive Data Buffer Full Flag Bit

Indicates the receive data buffer status.

1 (R): Full
0 (R): Not full (default)

SPRBF is set to 1 when the data received in the shift register is loaded to the receive data buffer (receive operation completed), indicating that the received data can be read out. This bit is reset to 0 when the data is read out from the SPI_RXD1 register (0x5704).

D0 SPTBE: Transmit Data Buffer Empty Flag Bit

Indicates the transmit data buffer status.

1 (R): Empty (default)
0 (R): Not empty

SPTBE is reset to 0 when transmit data is written to the SPI_TXD1 register (transmit data buffer, 0x5702) and is set to 1 when the written data is transferred to the shift register (transmit operation started).

Transmit data should be written to the SPI_TXD1 register when this bit = 1.

0x5702: SPI CH.1 Transmit Data Register (SPI_TXD1)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SPI CH.1 Transmit Data Register (SPI_TXD1)	**0x5702** (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	**SPTDB[7:0]**	SPI CH.1 transmit data buffer SPTDB7 = MSB SPTDB0 = LSB	0x0 to 0xff	0x0	R/W	

D[15:8] Reserved

D[7:0] SPTDB[7:0]: SPI CH.1 Transmit Data Buffer Bits

Set transmit data to be written to the transmit data buffer. (Default: 0x0)

In master mode, data transmission begins by writing data to this register. In slave mode, the register contents are transferred to the shift register to start data transmission when a clock is input from the master device.

SPTBE (D0/SPI_ST1 register) is set to 1 (empty) when the data is transferred to the shift register. At the same time, a cause of transmit data buffer empty interrupt occurs. The next transmit data can be written to the register at any time thereafter, even when the SPI CH.1 is sending data.

The serial-converted data is output from the SPI_SDO1 pin beginning with the MSB, in which the bits set to 1 are output as high-level signals and those set to 0 output as low-level signals.

Note: Make sure that SPEN is set to 1 before writing data to the SPI_TXD1 register to start data transmission/reception.

0x5704: SPI CH.1 Receive Data Register (SPI_RXD1)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SPI CH.1 Receive Data Register (SPI_RXD1)	**0x5704** (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	**SPRDB[7:0]**	SPI CH.1 receive data buffer SPRDB7 = MSB SPRDB0 = LSB	0x0 to 0xff	0x0	R	

D[15:8] Reserved

D[7:0] SPRDB[7:0]: SPI CH.1 Receive Data Buffer Bits

Stores received data. (Default: 0x0)

When a receive operation is completed and the data received in the shift register is loaded to the receive data buffer, SPRBF (D1/SPI_ST1 register) is set to 1 (buffer full). At the same time, a cause of receive data buffer full interrupt occurs. Thereafter, the data can be read out at any time before a receive operation for the next data is completed.

If the next data receive operation is completed before this register is read out, the data in it is overwritten with the newly received data.

The serial data input from the SPI_SDI1 pin is converted into parallel data beginning with the MSB, with the high-level signals changed to 1s and the low-level signals changed to 0s. The resulting data is stored in this register.

SPI_RXD1 is a read-only register, so no data can be written to it.

0x5706: SPI CH.1 Control Register (SPI_CTL1)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
SPI CH.1 Control Register (SPI_CTL1)	**0x5706** (16 bits)	D15–6	–	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 CH.1 enable	1	Enable	0	Disable	0	R/W	

D[15:6] Reserved

D5 SPRIE: Receive Data Buffer Full Interrupt Enable Bit

Enables/disables SPI CH.1 interrupt caused by receive data buffer full.

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

When SPRIE is set to 1, SPI CH.1 (receive data buffer full) interrupt requests to the ITC are enabled. A receive data buffer full interrupt request occurs when the data received in the shift register is loaded to the receive data buffer (receive operation completed).
When SPRIE is set to 0, SPI CH.1 interrupts caused by receive data full are not generated.

D4 SPTIE: Transmit Data Buffer Empty Interrupt Enable Bit

Enables/disables SPI CH.1 interrupt caused by transmit data buffer empty.

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

When SPTIE is set to 1, SPI CH.1 (transmit data buffer empty) interrupt requests to the ITC are enabled. A transmit data buffer empty interrupt request occurs when the data written to the transmit data buffer is transferred to the shift register (transmit operation started).
When SPTIE is set to 0, SPI CH.1 interrupts caused by transmit data buffer empty are not generated.

D3 CPHA: Clock Phase Select Bit

Selects the phase of the SPI clock. (Default: 0)
This bit controls the data transfer timing in conjunction with the CPOL (D2) bit (see Figure V.5.7.1).

D2 CPOL: Clock Polarity Select Bit

Selects the polarity of the SPI clock.

1 (R/W): Active low
0 (R/W): Active high (default)

This bit controls the data transfer timing in conjunction with the CPHA (D3) bit (see Figure V.5.7.1).

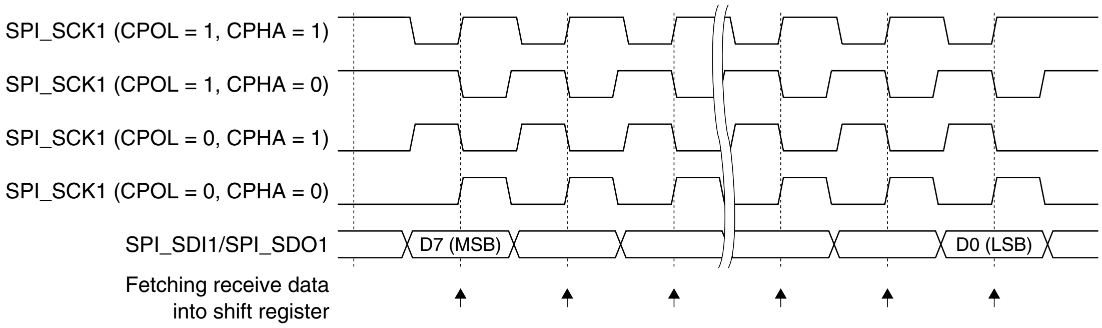

SPI_SCK1 (CPOL = 1, CPHA = 1)
SPI_SCK1 (CPOL = 1, CPHA = 0)
SPI_SCK1 (CPOL = 0, CPHA = 1)
SPI_SCK1 (CPOL = 0, CPHA = 0)
SPI_SDI1/SPI_SDO1
D7 (MSB)
D0 (LSB)
Fetching receive data into shift register

Figure V.5.7.1 Clock and Data Transfer Timing

D1 MSSL: Master/Slave Mode Select Bit

Sets the SPI CH.1 in master or slave mode.
1 (R/W): Master mode
0 (R/W): Slave mode (default)

Setting MSSL to 1 selects master mode, and setting to 0 selects slave mode. In master mode, the SPI CH.1 performs data transfer using the clock generated by the PT8 prescaler. In slave mode, the SPI CH.1 performs data transfer using a clock input from the master device.

D0 SPEN: SPI CH.1 Enable Bit

Enables/disables operation of the SPI CH.1.
1 (R/W): Enable
0 (R/W): Disable (default)

When SPEN is set to 1, the SPI CH.1 is activated and data transfer is enabled.
When SPEN is set to 0, the SPI CH.1 goes off.

Note: Make sure that the SPEN bit is 0 before setting the CPHA, CPOL, and MSSL bits.

0x5708: SPI CH.1 Clock Control Register (SPI_CLK1)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
SPI CH.1 Clock Control Register (SPI_CLK1)	**0x5708** (16 bits)	D15–5	–	reserved	–		–	–	0 when being read.
		D4	**SPI_CKE**	SPI CH.1 clock enable	1 Enable	0 Disable	0	R/W	
		D3–0	**SPI_CLK [3:0]**	SPI CH.1 clock division ratio selection (Prescaler output clock)	SPI_CLK[3:0]	Clock	0x0	R/W	
					0xf–0xd	reserved			
					0xc	PCLK•1/4096			
					0xb	PCLK•1/2048			
					0xa	PCLK•1/1024			
					0x9	PCLK•1/512			
					0x8	PCLK•1/256			
					0x7	PCLK•1/128			
					0x6	PCLK•1/64			
					0x5	PCLK•1/32			
					0x4	PCLK•1/16			
					0x3	PCLK•1/8			
					0x2	PCLK•1/4			
					0x1	PCLK•1/2			
					0x0	PCLK•1/1			

D[15:5] Reserved

D4 SPI_CKE: SPI CH.1 Clock Enable Bit

Enables the internally generated SPI clock input to the SPI CH.1.

1 (R/W): Enable

0 (R/W): Disable (default)

Write 1 to this bit before the SPI CH.1 can operate in master mode.

D[3:0] SPI_CLK[3:0]: SPI CH.1 Clock Division Ratio Selection Bits

These bits select the SPI clock for the SPI CH.1 from 13 PT8 prescaler output clocks.

Table V.5.7.2 Selecting the SPI Clock

SPI_CLK[3:0]	Prescaler output clock	SPI_CLK[3:0]	Prescaler output clock
0xf	Reserved	0x7	PCLK•1/128
0xe	Reserved	0x6	PCLK•1/64
0xd	Reserved	0x5	PCLK•1/32
0xc	PCLK•1/4096	0x4	PCLK•1/16
0xb	PCLK•1/2048	0x3	PCLK•1/8
0xa	PCLK•1/1024	0x2	PCLK•1/4
0x9	PCLK•1/512	0x1	PCLK•1/2
0x8	PCLK•1/256	0x0	PCLK•1/1

(Default: 0x0)

V.5.8 Precautions

- When using SPI CH.1 in master mode, the operating clocks for both the PT8 and SPI CH.1 modules must be supplied from the CMU, since the PT8 module includes the prescaler for generating the SPI clock for the SPI CH.1.
- Be sure to use 16-bit access instructions for reading/writing from/to the SPI CH.1 registers (0x5700 to 0x5708). The SPI CH.1 registers do not allow reading/writing using 32-bit and 8-bit access instructions.
- Do not access the SPI_CTL1 register (0x5706), while the SPBSY flag (D2/SPI_ST1 register) is set to 1 (during data transfer).

  * **SPBSY**: Transfer Busy Flag (Master)/ss Signal Low Flag (Slave) Bit in the SPI CH.1 Status (SPI_ST1) Register (D2/0x5700)

- When the SPI CH.1 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.

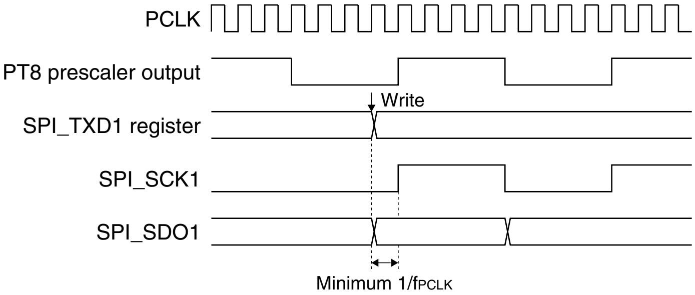


Figure V.5.8.1 SPI_SDO1 and SPI_SCK1 Change Timings when CPHA = 0

The half SPI_SCK1 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.

- Make sure that SPEN is set to 1 before writing data to the SPI_TXD1 register to start data transmission/reception.

THIS PAGE IS BLANK.

V.6 Remote Controller (REMC)

V.6.1 Outline of the REMC

The S1C17002 is equipped with a remote controller (REMC) module for generating/receiving infrared remote control signals. The REMC module consists of a carrier generator for generating a carrier signal, a 16-bit envelope counter for counting the transmit/receive data length, a modulator for generating transmit data with a designated carrier length, and an edge detector for detecting rising and falling edges from the input signal.

Also the REMC module supports three types of interrupts: (1) counter underflow interrupt that will occur when the envelope counter has reaches 0 indicating that a specified length of data has been sent during transmission or the received data length is too long during reception, (2) input rising edge interrupt that will occur when a rising edge of the input signal (received data) has been detected during reception, and (3) falling edge interrupt that will occur when a falling edge of the input signal has been detected.

Figure V.6.1.1 shows the structure of the REMC module.

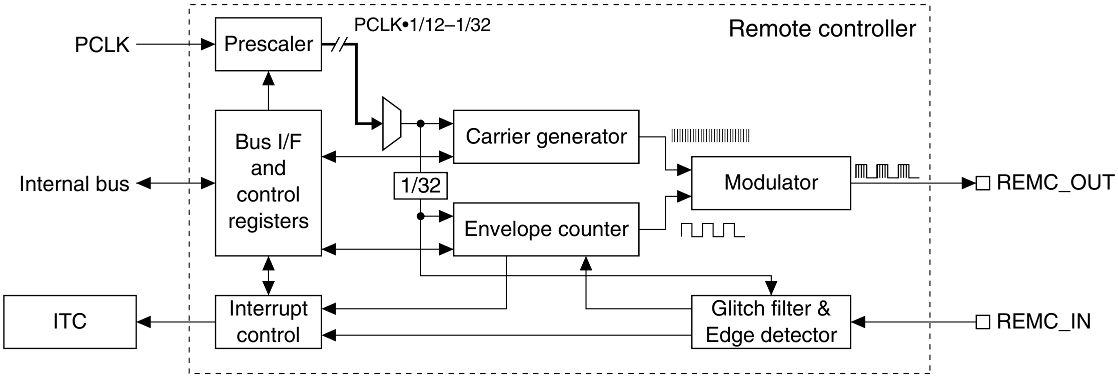


Figure V.6.1.1 Structure of REMC Module

V.6.2 REMC I/O Pins

Table V.6.2.1 lists the REMC input/output pins.

Table V.6.2.1 List of REMC Pins

Pin name	I/O	Size	Function
REMC_IN	I	1	Remote controller receive data input pin This pin inputs receive data.
REMC_OUT	O	1	Remote controller transmit data output pin This pin outputs the modulated Remote control transmit data.

The REMC input/output pins (REMC_IN, REMC_OUT) are shared with the I/O ports and they are initialized as general-purpose I/O port pins by default. Before using these pins for the REMC, the pin functions must be switched using the Port Function Select Register.

For details on switching pin function, Section I.3.3, "Switching Over the Multiplexed Pin Functions."

V.6.3 Prescaler and Carrier Generator

The REMC module contains a carrier generator with a prescaler that generates a transmit carrier signal according to the H carrier length, and L carrier length set with software.

Setting the prescaler

The REMC module also contains a prescaler that divides the PCLK clock input from the CMU to generate the count clock for the carrier generator. It outputs four clocks, PCLK divided by 12 to PCLK divided by 32. Use REMPSDIV[1:0] (D[1:0]/REMC_PSC register) to select one them.

∗ **REMPSDIV[1:0]**: REMC Prescaler Division Ratio Select Bits in the REMC Prescaler Control (REMC_PSC) Register (D[1:0]/0x5400)

Table V.6.3.1 Selecting a Clock for Carrier Generator

REMPSDIV[1:0]	Output clock
0x3	PCLK•1/32
0x2	PCLK•1/24
0x1	PCLK•1/16
0x0	PCLK•1/12

(Default: 0x0)

Most infrared remote carrier frequencies fall within the range from 30 kHz to 56 kHz and their duty ratio is 1/2, 1/3, or 1/4. To insure the accuracy of the carrier frequency, the prescaler output clock frequency should be set to 1 MHz (min.) to 8 MHz (max.).
Examples:

1. When the PCLK frequency is 48 MHz, PCLK•1/24 (2 MHz) is recommended.
2. When the PCLK frequency is 60 MHz, PCLK•1/32 (1.875 MHz) is recommended.

The prescaler is disabled (turned off) by default. Set REMPSON (D2/REMC_PSC register) to 1 to turn the prescaler on before the REMC module can be used.

∗ **REMPSON**: REMC Prescaler Control Bit in the REMC Prescaler Control (REMC_PSC) Register (D2/0x5400)

Also the prescaler clock is supplied to the envelope counter and the edge detector after it is divided by 32.

If the REMC module is not used, the prescaler should be turned off (REMPSON = 0) to reduce current consumption.

Setting the carrier signal

The high period and low period widths of the carrier signal can be set using CLDH[7:0] (D[15:8]/REMC_CARL register) and CLDL[7:0] (D[7:0]/REMC_CARL register). The carrier high and low widths should be set as the number of clock (selected as above) cycles +1.
Carrier high width [s] = (CLDH[7:0] + 1) / f_{PSOUT} [Hz]
Carrier low width [s] = (CLDL[7:0] + 1) / f_{PSOUT} [Hz]

∗ **CLDH[7:0]**: REMC Carrier High Width Setup Bits in the REMC Carrier Load (REMC_CARL) Register (D[15:8]/0x540c)
∗ **CLDL[7:0]**: REMC Carrier Low Width Setup Bits in the REMC Carrier Load (REMC_CARL) Register (D[7:0]/0x540c)

The table below shows examples of carrier settings when the prescaler clock frequency is 2 MHz.

Table V.6.3.2 Carrier Setting Examples

Prescaler clock frequency	2 MHz		
Carrier frequency	30 kHz	38 kHz	56 kHz
Carrier duty	1/2	1/3	1/4
Ideal period	33.333 µs	26.316 µs	17.857 µs
CLDH[7:0] setting	32	17	8
CLDL[7:0] setting	32	34	26
Period error margin	1%	0.7%	0.8%

The carrier signal is generated as shown in Figure V.6.3.1.

Example: REMPSDIV[1:0] = 0x2 (PCLK•1/24, 2 MHz), CLDH[7:0] = 17, CLDL[7:0] = 34
(carrier 38 kHz, 1/3 duty)

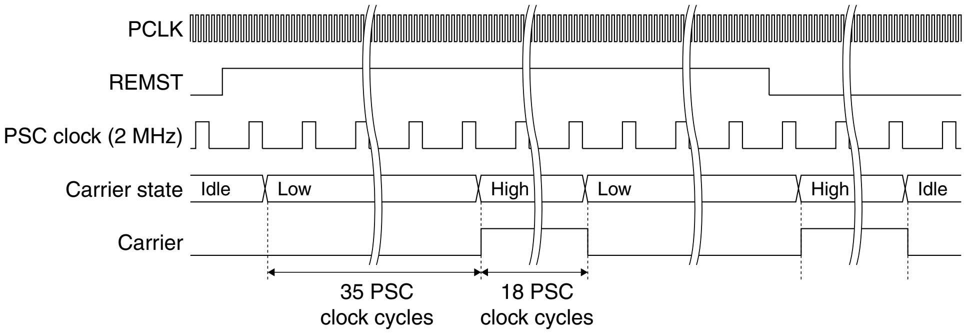


Figure V.6.3.1 Carrier Signal Generation

V.6.4 Controlling Data Transmission/Reception

Before starting data transfer, set up the conditions by the procedure below.

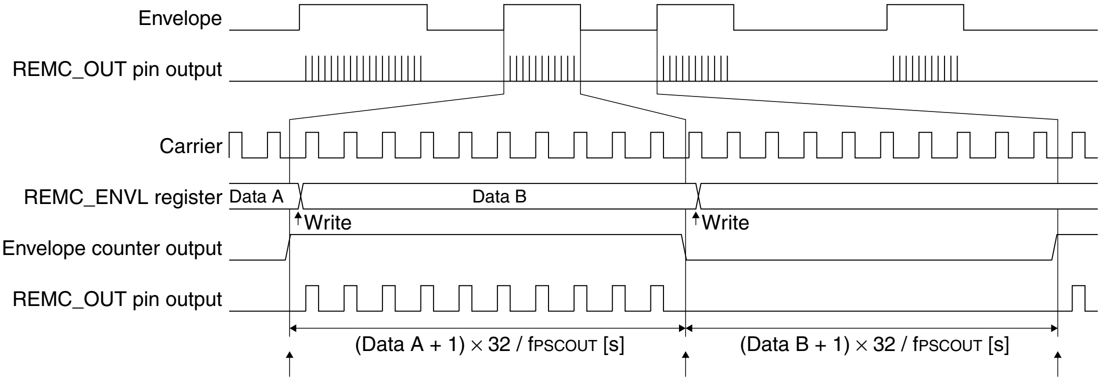
(1) Configure the carrier signal. See Section V.6.3.

(2) Set up the interrupt conditions. See Section V.6.5.

Note: Make sure that the REMC module is idle (REMST/REMC_CTL register = 0) before setting the conditions above.

∗ **REMST**: REMC Start/Stop Control Bit in the REMC Control (REMC_CTL) Register (D0/0x5408)

Data transmit control



An underflow occurs in the envelope counter. The envelope counter outputs an underflow signal for interrupt and inverts the envelope output signal. At the same time, it loads the REMC_ENVL register value to the counter and continues counting from the loaded new data.

Figure V.6.4.1 Data Transmission

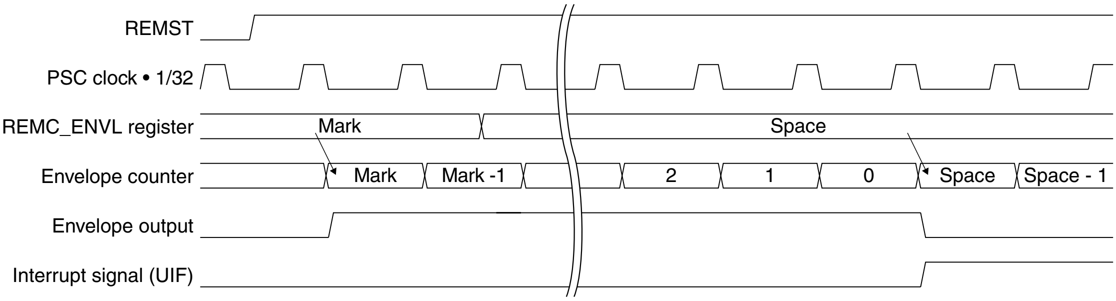


Figure V.6.4.2 Underflow Interrupt Generation Timing

(1) Setting data transmit mode

Write 0 to MODE (D7/REMC_CFG register) to set the REMC in data transmit mode.

∗ **MODE**: REMC Mode Select Bit in the REMC Configuration (REMC_CFG) Register (D7/0x5404)

(2) Setting the REMC_ENVL register

Write the value equivalent to the first Mark width (high period) of the transmit data to REMC_ENVL register (0x540e).

Use the following equation to determine the value to be set to the envelope counter.

$$\text{REMC\_ENVL} = \frac{\text{Mark/Space width [second]} \times \text{Prescaler output clock frequency [Hz]}}{32} - 1$$

(3) Starting transmission

Set REMST (D0/REMC_CTL register) to 1 to start data transmit operation. The carrier generator starts outputting the carrier signal. The envelope counter turns the envelope output signal to high at the next count clock (PSC clock • 1/32) and loads the value set in the REMC_ENVL register into the counter. The envelope counter starts counting down from the loaded value.

(4) Setting the next envelope period

After starting transmission, set the Space width (low period) that follows the Mark period being currently output to REMC_ENVL register. Then, wait occurrence of an underflow interrupt.

(5) Underflow interrupt

When the envelope counter underflows, it inverts the envelope output signal and loads the REMC_ENVL register value to the counter. The counting operation continues.

At the same time, the REMC module sends an interrupt request to the interrupt controller (ITC) if the interrupt is enabled.

Use this interrupt to set the next period count data to the REMC_ENVL register.

(6) Terminating data transmission

After the last data transfer has finished (after an underflow interrupt occurs), write 0 to REMST to terminate data transmission. Note that the envelope counter stops immediately after REMST is set to 0.

Data receive control

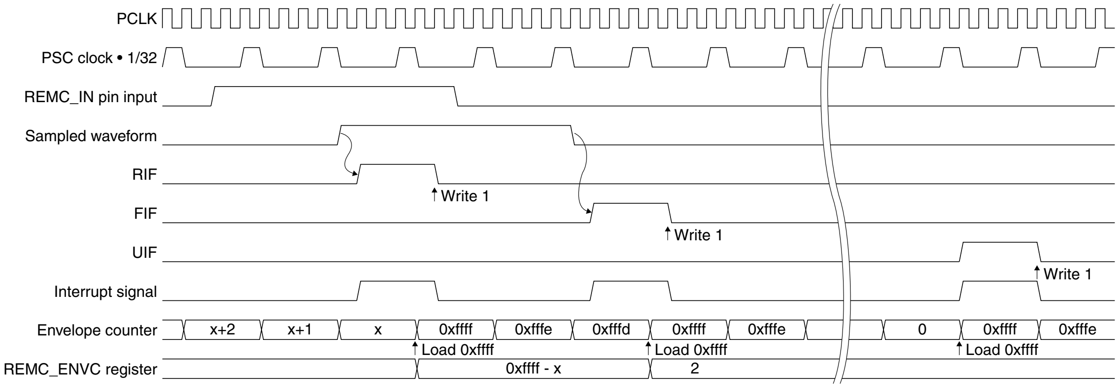


Figure V.6.4.3 Data Reception

(1) Setting data receive mode

Write 1 to MODE (D7/REMC_CFG register) to set the REMC in data receive mode.

(2) Setting the REMC_ENVL register

Write 0xffff to REMC_ENVL register (0x540e). Note that the correct envelope pulse width cannot be obtained if a value other than 0xffff is set to the REMC_ENVL register.

(3) Start data reception

Set REMST (D0/REMC_CTL register) to 1 to start the data receive operation (input signal edge detection).

(4) Rising edge and falling edge detection and interrupt

The edge detector samples the signal input to the REMC_IN pin with the envelope count clock (PSC clock • 1/32) to detect input transition (rising edge or falling edge of the signal). When a signal edge is detected, a cause of rising edge or falling edge interrupt occurs and the REMC module sends an interrupt request to the ITC if the interrupt is enabled. The rising edge and falling edge interrupts can be enabled individually. A rising edge interrupt notifies the application program that a Space period ends and a Mark period starts. A falling edge interrupt notifies the application program that a Mark period ends and a Space period starts.

Note that a signal transition is regarded as noise by the glitch filter if the signal level after the input changes is not sampled for two or more sampling clock cycles. In this case no rising edge or falling edge interrupt occurs.

When an input signal edge is detected, the envelope counter data bits are inverted and loaded to the REMC_ENVC register (0x5410). This value represents the width of the envelope pulse that has been received. The envelope counter loads 0xffff and starts counting for the next envelope pulse.

If the envelope counter reaches 0 without an interrupt generated after the counter is set to 0xffff, either no receive data remains or a receive error has occurred. An underflow interrupt occurs even in data reception, use it for a terminate/error processing.

Note: After REMST is set to 1, the envelope counter loads 0xffff written in the REMC_ENVL register when the first input signal edge is detected. At this time, a rising edge interrupt occurs if the interrupt is enabled. This interrupt should be ignored.

(5) Obtain envelope pulse width

By using the interrupt above, read the counted value from the REMC_ENVC register. It represents the width of the envelope pulse received previously.

The pulse width can be determined by the following equation:

$$\text{Envelope pulse width} = \frac{(\text{REMC\_ENVC} + 1) \times 32}{f_{\text{PSOUT}}\ [\text{Hz}]}\ [\text{s}]$$

where REMC_ENVC is the REMC_ENVC register value and f_{PSOUT} is the prescaler output clock frequency.

The REMC_ENVC register value must be read out until the next rising or falling edge interrupt occurs. Otherwise, the REMC_ENVC register will be overwritten with the next count data.

(6) Terminating data reception

After the last data transfer has finished, write 0 to REMST to terminate data reception.

V.6.5 REMC Interrupt

The REMC module can generate the following three types of interrupts:

- Underflow interrupt
- Rising edge interrupt
- Falling edge interrupt

The REMC module has one interrupt signal to be output to the interrupt controller (ITC) and it is shared with the three causes of interrupt. To determine the cause of interrupt that has occurred, read the interrupt flags in the REMC module.

Underflow interrupt

This interrupt request occurs when the envelope counter underflows during count-down, and it sets the interrupt flag UIF (D4/REMC_CFG register) in the REMC module to 1.
During data transmission, this interrupt notifies the application program that an envelope pulse with the period length specified has completed. During data reception, this interrupt notifies the application program that a data reception has completed or a receive error has occurred.

∗ **UIF**: Underflow Interrupt Flag Bit in the REMC Configuration (REMC_CFG) Register (D4/0x5404)

Set UIE (D0/REMC_CFG register) to 1 when using this interrupt. If UIE is set to 0 (default), UIF will not be set to 1 and an interrupt request by this cause will not be sent to the ITC.

∗ **UIE**: Underflow Interrupt Enable Bit in the REMC Configuration (REMC_CFG) Register (D0/0x5404)

If UIF is set to 1, the REMC module outputs the interrupt request signal to the ITC. The interrupt request signal sets the REMC interrupt flag in the ITC to 1 and an interrupt occurs if other interrupt conditions meet the ITC and S1C17 Core settings.
The REMC interrupt handler routine should read the UIF flag to check if the interrupt has occurred due to an envelope counter underflow or another cause.
Furthermore, the interrupt handler routine must reset (write 1 to) UIF in the REMC module as well as the REMC interrupt flag in the ITC, to clear the cause of interrupt.

Rising edge interrupt

This interrupt request occurs when the signal input to the REMC_IN pin goes high from low status, and it sets the interrupt flag RIF (D6/REMC_CFG register) in the REMC module to 1.
The envelope counter value that has been counted from the time the previous falling edge detected is loaded to the REMC_ENVC register (0x5410). The application program can read the REMC_ENVC register to obtain the received Space pulse width after this interrupt has occurred.
At the same time this interrupt occurs, the envelope counter loads 0xffff and starts counting to measure the length between this interrupt and the next falling edge interrupt (Mark pulse width).

∗ **RIF**: Rising Edge Interrupt Flag Bit in the REMC Configuration (REMC_CFG) Register (D6/0x5404)

Set RIE (D2/REMC_CFG register) to 1 when using this interrupt. If RIE is set to 0 (default), RIF will not be set to 1 and an interrupt request by this cause will not be sent to the ITC.

∗ **RIE**: Rising Edge Interrupt Enable Bit in the REMC Configuration (REMC_CFG) Register (D2/0x5404)

If RIF is set to 1, the REMC module outputs the interrupt request signal to the ITC. The interrupt request signal sets the REMC interrupt flag in the ITC to 1 and an interrupt occurs if other interrupt conditions meet the ITC and S1C17 Core settings.
The REMC interrupt handler routine should read the RIF flag to check if the interrupt has occurred due to detection of a rising edge of the input signal or another cause.
Furthermore, the interrupt handler routine must reset (write 1 to) RIF in the REMC module as well as the REMC interrupt flag in the ITC, to clear the cause of interrupt.

Falling edge interrupt

This interrupt request occurs when the signal input to the REMC_IN pin goes low from high status, and it sets the interrupt flag FIF (D5/REMC_CFG register) in the REMC module to 1.

The envelope counter value that has been counted from the time the previous rising edge detected is loaded to the REMC_ENVC register (0x5410). The application program can read the REMC_ENVC register to obtain the received Mark pulse width after this interrupt has occurred.

At the same time this interrupt occurs, the envelope counter loads 0xffff and starts counting to measure the length between this interrupt and the next rising edge interrupt (Space pulse width).

* **FIF**: Falling Edge Interrupt Flag Bit in the REMC Configuration (REMC_CFG) Register (D5/0x5404)

Set the FIE bit (D1/REMC_CFG register) to 1 when using this interrupt. If FIE is set to 0 (default), FIF will not be set to 1 and an interrupt request by this cause will not be sent to the ITC.

* **FIE**: Falling Edge Interrupt Enable Bit in the REMC Configuration (REMC_CFG) Register (D1/0x5404)

If FIF is set to 1, the REMC module outputs the interrupt request signal to the ITC. The interrupt request signal sets the REMC interrupt flag in the ITC to 1 and an interrupt occurs if other interrupt conditions meet the ITC and S1C17 Core settings.

The REMC interrupt handler routine should read the FIF flag to check if the interrupt has occurred due to detection of a falling edge of the input signal or another cause.

Furthermore, the interrupt handler routine must reset (write 1 to) FIF in the REMC module as well as the REMC interrupt flag in the ITC, to clear the cause of interrupt.

ITC registers for REMC interrupt

When a cause of interrupt that has been enabled occurs according to the interrupt condition settings shown above, the REMC module asserts the interrupt signal sent to the ITC. To generate a REMC interrupt, set the interrupt level and enable the interrupt using the ITC registers.

The following shows the control bits for the REMC interrupt in the ITC.

Interrupt flag in the ITC

* **AIFT14**: REMC Interrupt Flag Bit in the Additional Interrupt Flag (ITC_AIFLG) Register (D14/0x42e0)

Interrupt enable bit in the ITC

* **AIEN14**: REMC Interrupt Enable Bit in the Additional Interrupt Enable (ITC_AEN) Register (D14/0x42e2)

Interrupt level setup bits in the ITC

* **AILV14[2:0]**: REMC Interrupt Level Bits in the Additional Interrupt Level Setup (ITC_AILV7) Register 7 (D[2:0]/0x42f4)

The interrupt signal sent from the REMC module sets AIFT14 to 1. If AIEN14 has been set to 1, the ITC sends an interrupt request to the S1C17 Core. To disable the REMC interrupt, set AIEN14 to 0. AIFT14 is always set to 1 by the interrupt signal sent from the REMC module, regardless of how AIEN14 is set (even when set to 0).

AILV14[2:0] sets the interrupt level (0 to 7) of the REMC interrupt.

An interrupt request to the S1C17 Core is accepted only when all the conditions described below are met.

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

For details on these interrupt control registers, as well as the device operation when an interrupt has occurred, see Section III.1, “Interrupt Controller (ITC).”

Interrupt vector

The following shows the vector number and vector address for the REMC interrupt:

Vector number: 30 (0x1e)
Vector address: TTBR + 0x78

V.6.6 Details of Control Registers

Table V.6.6.1 List of REMC Registers

Address	Register name		Function
0x5400	REMC_PSC	REMC Prescaler Control Register	Sets up the REMC prescaler.
0x5404	REMC_CFG	REMC Configuration Register	Sets the REMC modes and controls the REMC interrupt.
0x5408	REMC_CTL	REMC Control Register	Starts/stops transmission.
0x540c	REMC_CARL	REMC Carrier Load Register	Configures the carrier signal.
0x540e	REMC_ENVL	REMC Envelope Load Register	Configures the envelope pulse width.
0x5410	REMC_ENVC	REMC Envelope Capture Register	Input envelope pulse width

The following describes each REMC register. These are all 16-bit registers.

Note: When setting the registers, be sure to write a 0, and not a 1, for all “reserved bits.”

0x5400: REMC Prescaler Control Register (REMC_PSC)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
REMC Prescaler Control Register (REMC_PSC)	**0x5400** (16 bits)	D15–3	–	reserved	–		–	–	0 when being read.
		D2	**REMPSON**	REMC prescaler control	1 On	0 Off	0	R/W	
		D1–0	**REMPSDIV [1:0]**	REMC prescaler division ratio select (Prescaler output clock)	REMPSDIV[1:0] 0x3 0x2 0x1 0x0	Clock PCLK•1/32 PCLK•1/24 PCLK•1/16 PCLK•1/12	0x0	R/W	

D[15:3] Reserved

D2 REMPSON: REMC Prescaler Control Bit

Turns the prescaler in the REMC module on and off.

1 (R/W): On

0 (R/W): Off (default)

Set REMPSON to 1 to turn the prescaler on before the REMC module can be used.

D[1:0] REMPSDIV[1:0]: REMC Prescaler Division Ratio Select Bits

These bits select the count clock for the carrier generator.

Table V.6.6.2 Selecting a Clock for Carrier Generator

REMPSDIV[1:0]	Output clock
0x3	PCLK•1/32
0x2	PCLK•1/24
0x1	PCLK•1/16
0x0	PCLK•1/12

(Default: 0x0)

Most infrared remote carrier frequencies fall within the range from 30 kHz to 56 kHz and their duty ratio is 1/2, 1/3, or 1/4. To insure the accuracy of the carrier frequency, the prescaler output clock frequency should be set to 1 MHz (min.) to 8 MHz (max.).

Examples:

1. When the PCLK frequency is 48 MHz, PCLK•1/24 (2 MHz) is recommended.
2. When the PCLK frequency is 60 MHz, PCLK•1/32 (1.875 MHz) is recommended.

Also the prescaler clock is supplied to the envelope counter and the edge detector after it is divided by 32.

0x5404: REMC Configuration Register (REMC_CFG)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
REMC Configuration Register (REMC_CFG)	**0x5404** (16 bits)	D15–8	**–**	reserved	–				–	–	0 when being read.
		D7	**MODE**	REMC mode select	1	Receive	0	Transmit	0	R/W	
		D6	**RIF**	Rising edge interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D5	**FIF**	Falling edge interrupt flag					0	R/W	
		D4	**UIF**	Underflow interrupt flag					0	R/W	
		D3	**–**	reserved	–				–	–	0 when being read.
		D2	**RIE**	Rising edge interrupt enable	1	Enable	0	Disable	0	R/W	
		D1	**FIE**	Falling edge interrupt enable	1	Enable	0	Disable	0	R/W	
		D0	**UIE**	Underflow interrupt enable	1	Enable	0	Disable	0	R/W	

D[15:8] Reserved

D7 MODE: REMC Mode Select Bit

Selects the data transmit/receive direction.

1 (R/W): Receive mode

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

Note: Make sure REMST (D0/REMC_CTL register) is 0 (REMC is idle) before changing the mode.

D6 RIF: Rising Edge Interrupt Flag Bit

This is the interrupt flag to indicate the rising edge interrupt cause occurrence status.

1 (R): Cause of interrupt has occurred

0 (R): No cause of interrupt has occurred (default)

1 (W): Flag is reset

0 (W): Has no effect

RIF is set to 1 when a rising edge is detected in the input signal. This flag is effective only when the REMC is in receive mode and RIE (D2) is set to 1. When RIE is 0, RIF will not be set to 1 even if a rising edge is detected.

When this flag is set to 1, the REMC interrupt request signal is output to the ITC. The interrupt request signal sets the REMC interrupt flag in the ITC to 1 and an interrupt occurs if other interrupt conditions meet the ITC and S1C17 Core settings.

RIF is reset by writing 1.

D5 FIF: Falling Edge Interrupt Flag Bit

This is the interrupt flag to indicate the falling edge interrupt cause occurrence status.

1 (R): Cause of interrupt has occurred

0 (R): No cause of interrupt has occurred (default)

1 (W): Flag is reset

0 (W): Has no effect

FIF is set to 1 when a falling edge is detected in the input signal. This flag is effective only when the REMC is in receive mode and FIE (D1) is set to 1. When FIE is 0, FIF will not be set to 1 even if a falling edge is detected.

When this flag is set to 1, the REMC interrupt request signal is output to the ITC. The interrupt request signal sets the REMC interrupt flag in the ITC to 1 and an interrupt occurs if other interrupt conditions meet the ITC and S1C17 Core settings.

FIF is reset by writing 1.

D4 UIF: Underflow Interrupt Flag Bit

This is the interrupt flag to indicate the underflow interrupt cause occurrence status.

1 (R): Cause of interrupt has occurred

0 (R): No cause of interrupt has occurred (default)

1 (W): Flag is reset

0 (W): Has no effect

UIF is set to 1 when the envelope counter underflows during data transmission or reception. This flag is effective only when the UIE (D0) is set to 1. When UIE is 0, UIF will not be set to 1 even if an envelope counter underflow occurs.

When this flag is set to 1, the REMC interrupt request signal is output to the ITC. The interrupt request signal sets the REMC interrupt flag in the ITC to 1 and an interrupt occurs if other interrupt conditions meet the ITC and S1C17 Core settings.
UIF is reset by writing 1.

D3 Reserved

D2 RIE: Rising Edge Interrupt Enable Bit

Enables or disables the interrupt by detecting a rising edge of the input signal.
1 (R/W): Enable interrupt
0 (R/W): Disable interrupt (default)

Setting RIE to 1 enables the rising edge interrupt; setting to 0 disables the interrupt.
In addition, it is necessary to set the REMC interrupt enable bits in the ITC to interrupt enabled to actually generate an interrupt.

D1 FIE: Falling Edge Interrupt Enable Bit

Enables or disables the interrupt by detecting a falling edge of the input signal.
1 (R/W): Enable interrupt
0 (R/W): Disable interrupt (default)

Setting FIE to 1 enables the falling edge interrupt; setting to 0 disables the interrupt.
In addition, it is necessary to set the REMC interrupt enable bits in the ITC to interrupt enabled to actually generate an interrupt.

D0 UIE: Underflow Interrupt Enable Bit

Enables or disables the interrupt by an envelope counter underflow.
1 (R/W): Enable interrupt
0 (R/W): Disable interrupt (default)

Setting UIE to 1 enables the underflow interrupt; setting to 0 disables the interrupt.
In addition, it is necessary to set the REMC interrupt enable bits in the ITC to interrupt enabled to actually generate an interrupt.

Note: To avoid occurrence of unnecessary interrupts, be sure to reset the interrupt flag before the REMC interrupt is enabled using the interrupt enable bit.

0x5408: REMC Control Register (REMC_CTL)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
REMC Control Register (REMC_CTL)	**0x5408** (16 bits)	D15–1	–	reserved	–				–	–	0 when being read.
		D0	**REMST**	REMC start/stop control	1	Start	0	Stop	0	R/W	

D[15:1] Reserved

D0 REMST: REMC Start/Stop Control Bit

Starts/stops the REMC module.

1 (R): REMC is transmitting/receiving data
0 (R): REMC is idle (default)
1 (W): Start REMC operation
0 (W): Stop REMC operation

When REMST is set to 1, the REMC module starts data transmission or data reception according to the MODE (D7/REMC_CFG register) setting. When REMST is set to 0, the REMC module stops operating.

0x540c: REMC Carrier Load Register (REMC_CARL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
REMC Carrier Load Register (REMC_CARL)	**0x540c** (16 bits)	D15–8	**CLDH[7:0]**	REMC carrier high width setup	0x0 to 0xff	0x0	R/W	
		D7–0	**CLDL[7:0]**	REMC carrier low width setup	0x0 to 0xff	0x0	R/W	

Note: This register is effective only in transmit mode and is not used in receive mode.

D[15:8] CLDH[7:0]: REMC Carrier High Width Setup Bits

Sets the high period width of the carrier signal. (Default: 0x0)

The carrier high width should be set as the number of the prescaler clock cycles +1.

Carrier high width [s] = (CLDH[7:0] + 1) / fPSOUT [Hz] (fPSOUT: Prescaler clock frequency)

D[7:0] CLDL[7:0]: REMC Carrier Low Width Setup Bits

Sets the low period width of the carrier signal. (Default: 0x0)

The carrier low widths should be set as the number of prescaler clock cycles +1.

Carrier low width [s] = (CLDL[7:0] + 1) / fPSOUT [Hz] (fPSOUT: Prescaler clock frequency)

The table below shows examples of carrier settings when the prescaler clock frequency is 2 MHz.

Table V.6.6.3 Carrier Setting Examples

Prescaler clock frequency	2 MHz		
Carrier frequency	30 kHz	38 kHz	56 kHz
Carrier duty	1/2	1/3	1/4
Ideal period	33.333 µs	26.316 µs	17.857 µs
CLDH[7:0] setting	32	17	8
CLDL[7:0] setting	32	34	26
Period error margin	1%	0.7%	0.8%

The carrier signal is generated as shown in Figure V.6.6.1.

Example: REMPSDIV[1:0] = 0x2 (PCLK•1/24, 2 MHz), CLDH[7:0] = 17, CLDL[7:0] = 34
(carrier 38 kHz, 1/3 duty)

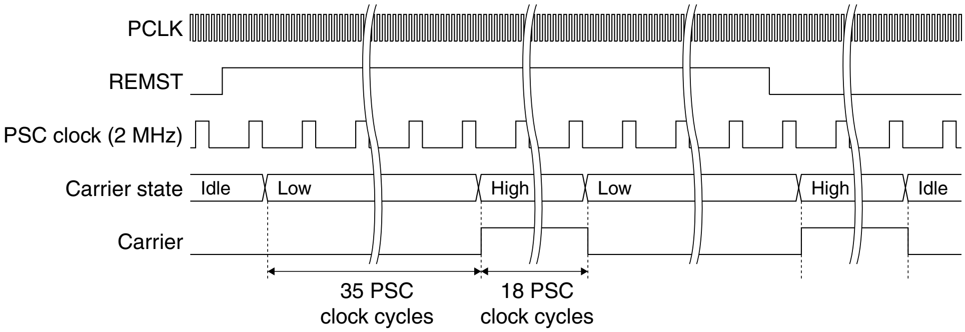


Figure V.6.6.1 Carrier Signal Generation

0x540e: REMC Envelope Load Register (REMC_ENVL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
REMC Envelope Load Register (REMC_ENVL)	**0x540e** (16 bits)	D15–0	**ELD[15:0]**	Envelope counter preset data	0x0 to 0xffff	0x0	R/W	

D[15:0] ELD[15:0]: Envelope Counter Preset Data Bits

Sets the initial value to be set to the envelope counter. (Default: 0x0)

The counter stops when it reaches 0 and generates a cause of underflow interrupt.

In data transmit mode

During data transmission, set the envelope pulse width to be transmitted.

Use the following equation to determine the value to be set to this register.

$$\text{REMC\_ENVL} = \frac{\text{Mark/Space width [second]} \times \text{Prescaler output clock frequency [Hz]}}{32} - 1$$

When data transmission is started by writing 1 to REMST (D0/REMC_CTL register), the envelope counter starts counting down from the value set in this register and generates a cause of underflow interrupt when the counter underflows. The next envelope pulse width can be set using this interrupt.

The envelope counter loads the value set in this register when it underflows and continues counting down. It stops when REMST is set to 0.

In data receive mode

During data reception, set 0xffff to this register. It is used as the initial value for counting input envelope pulse width using the envelope counter. When a rising or falling edge of the input signal is detected, an interrupt occurs and the envelope counter loads 0xffff set in this register and starts counting down from the set value. Before loading 0xffff, the count value in the envelope counter is inverted and sent to the REMC_ENVC register (0x5410). This value represents the width of the envelope pulse previously received.

0x5410: REMC Envelope Capture Register (REMC_ENVC)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
REMC Envelope Capture Register (REMC_ENVC)	**0x5410** (16 bits)	D15–0	**ECP[15:0]**	Receive envelope pulse width	0x0 to 0xffff	0x0	R	

D[15:0] ECP[15:0]: Receive Envelope Pulse Width Bits

The envelope pulse width measured by the envelope counter during reception is loaded in this register. (Default: 0x0)

When an input signal edge is detected, the envelope counter data bits are inverted and loaded to this register. This value represents the width of the envelope pulse that has been received.

After a rising or falling edge interrupt has occurred, read this register. The pulse width can be determined by the following equation:

$$\text{Envelope pulse width} = \frac{(\text{REMC\_ENVC} + 1) \times 32}{f_{\text{PSOUT}}\ [\text{Hz}]}\ [\text{s}]$$

where REMC_ENVC is the REMC_ENVC register value and f_{PSOUT} is the prescaler output clock frequency.

The REMC_ENVC register value must be read out until the next rising or falling edge interrupt occurs. Otherwise, the REMC_ENVC register will be overwritten with the next count data.

V.6.7 Precautions

- Before the REMC module can start transmission/reception, the prescaler must be run.
- Make sure that the REMC module is idle (REMST/REMC_CTL register = 0) before setting the prescaler and carrier generator conditions.

  ∗ **REMST**: REMC Start/Stop Control Bit in the REMC Control (REMC_CTL) Register (D0/0x5408)

S1C17002 Technical Manual

VI S1C17002 I/O PORTS

VI.1 General-Purpose I/O Ports (GPIO)

VI.1.1 Structure of I/O Port

The S1C17002 contains 4 input ports (P0[3:0]) and 30 I/O ports (P1[6:0], P2[7:0], P3[2:0], P3[7:5], P4[4:0], and P5[3:0]) that can be directed for input or output through the use of a program. Although each pin is used for input/output from/to the internal peripheral circuits, some pins can be used as general-purpose input/output ports unless they are used for the peripheral circuits.

Also the I/O ports support 8 systems of port input interrupts.

Figure VI.1.1.1 shows the structure of a typical I/O port.

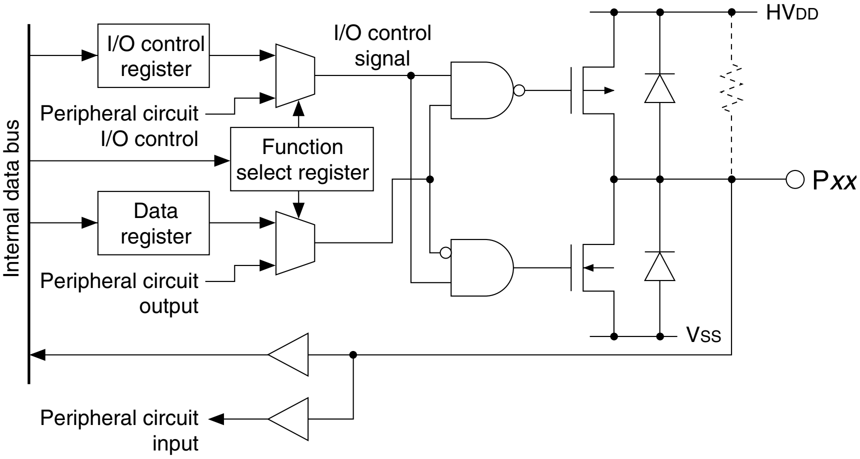


Figure VI.1.1.1 Structure of I/O Port

Notes:
- The P36 port provides a 100 kΩ internal pull-up resistor. Other ports have no pull-up/down resistor.
- The P0[3:0] are input only ports and the interface voltage level is AVDD, not HVDD.
- The AVDD voltage range can be changed to 1.65 to 3.60 V only when the ADC is not used and the P0[3:0] pins are used as digital signal input pins, not analog input pins. However, the high and low level input voltages of the digital signals must be AVDD and GND, respectively.

VI.1.2 Selecting the I/O Pin Functions

The I/O ports concurrently serve as the input/output pins for peripheral circuits or bus signals. Whether they are used as I/O ports or for peripheral circuits/bus signals can be selected bit-for-bit using the Port Function Select Registers. All pins not used for peripheral circuits/bus signals can be used as general-purpose I/O ports.

Each I/O port pin (P*xx*) is initialized for a default function at initial reset.

For details of pin functions and how to switch over, see Section I.3.3, "Switching Over the Multiplexed Pin Functions."

The subsequent sections explain the port functions assuming that the pin has been set as a general-purpose I/O port.

Note: The P3[7:5] pins are configured as the debug interface pins by default. Other P*xx* pins are all configured as general-purpose I/O ports.

VI.1.3 I/O Control Register and I/O Modes

The I/O ports are directed for input or output modes by writing data to IOC*x* corresponding to each port bit.

∗ **IOC1[6:0]**: P1[6:0] I/O Control Bits in the P1 I/O Control (P1_IOC) Register (D[6:0]/0x4403)
∗ **IOC2[7:0]**: P2[7:0] I/O Control Bits in the P2 I/O Control (P2_IOC) Register (D[7:0]/0x4405)
∗ **IOC3[2:0]**: P3[2:0] I/O Control Bits in the P3 I/O Control (P3_IOC) Register (D[2:0]/0x4407)
∗ **IOC3[7:5]**: P3[7:5] I/O Control Bits in the P3 I/O Control (P3_IOC) Register (D[7:5]/0x4407)
∗ **IOC4[4:0]**: P4[4:0] I/O Control Bits in the P4 I/O Control (P4_IOC) Register (D[4:0]/0x4409)
∗ **IOC5[3:0]**: P5[3:0] I/O Control Bits in the P5 I/O Control (P5_IOC) Register (D[3:0]/0x440b)

To set an I/O port for input, write 0 to the I/O control bit. I/O ports set for input mode are placed in the high-impedance state, and thus function as input ports. In the input mode, the state of the input pin is read directly, so the data is 1 when the pin state is high (HVDD level) or 0 when the pin state is low (VSS level).

To set an I/O port for output, write 1 to the I/O control bit. I/O port set for output function as output ports. When the port output data is 1, the port outputs a high level (HVDD level); when the data is 0, the port outputs a low level (VSS level).

VI.1.4 I/O Data Register

The registers listed below are used to read data from the I/O-port pins or to set output data.

∗ **P0D[3:0]**: P0[3:0] Port Input Data Bits in the P0 Port Input Data (P0_DAT) Register (D[3:0]/0x4400)
∗ **P1D[6:0]**: P1[6:0] Port Input/Output Data Bits in the P1 Port Input/Output Data (P1_DAT) Register (D[6:0]/0x4402)
∗ **P2D[7:0]**: P2[7:0] Port Input/Output Data Bits in the P2 Port Input/Output Data (P2_DAT) Register (D[7:0]/0x4404)
∗ **P3D[2:0]**: P3[2:0] Port Input/Output Data Bits in the P3 Port Input/Output Data (P3_DAT) Register (D[2:0]/0x4406)
∗ **P3D[7:5]**: P3[7:5] Port Input/Output Data Bits in the P3 Port Input/Output Data (P3_DAT) Register (D[7:5]/0x4406)
∗ **P4D[4:0]**: P4[4:0] Port Input/Output Data Bits in the P4 Port Input/Output Data (P4_DAT) Register (D[4:0]/0x4408)
∗ **P5D[3:0]**: P5[3:0] Port Input/Output Data Bits in the P5 Port Input/Output Data (P5_DAT) Register (D[3:0]/0x440a)

When an I/O port is set for output, the data written to the register is directly output to the I/O port pin. If the data written to the register is 1, the port pin is set high (HVDD level); if the data is 0, the port pin is set low (VSS level). Even in the input mode, data can be written to the data register without affecting the pin state.

When the register is read, the voltage level on the port pin is read out regardless of whether an I/O port is set for input or output mode. If the pin voltage is high (HVDD level), 1 is read out as input data; if the pin voltage is low (VSS level), 0 is read out as input data.

VI.1.5 Port Input Interrupt

The GPIO module has 8 interrupt systems (port input interrupts 0 to 7) and a port can be selected for generating each cause of interrupt.

The interrupt condition can also be selected from between input signal edge (rising edge or falling edge) and input signal level (high level or low level) in the interrupt controller (ITC).

Figure VI.1.5.1 shows the configuration of the port input interrupt circuit.

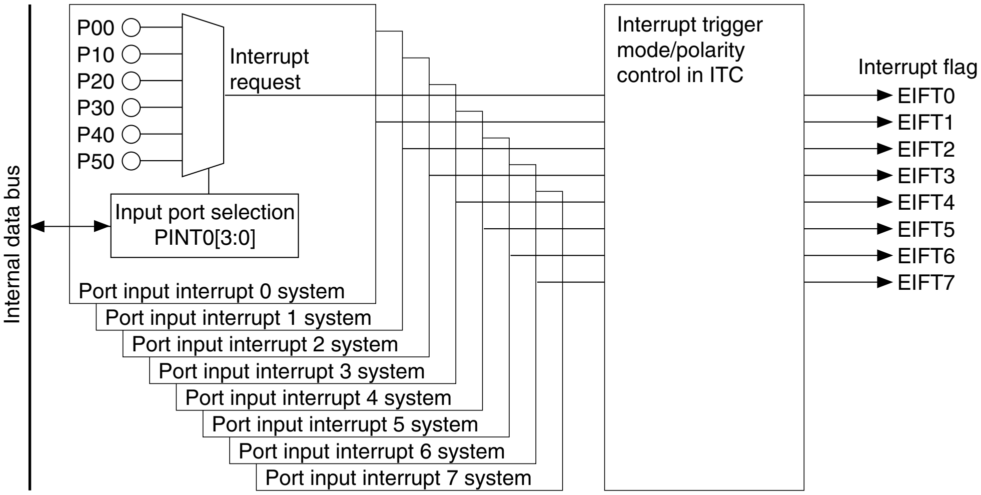

Internal data bus
P00
P10
P20
P30
P40
P50
Interrupt request
Input port selection PINT0[3:0]
Port input interrupt 0 system
Port input interrupt 1 system
Port input interrupt 2 system
Port input interrupt 3 system
Port input interrupt 4 system
Port input interrupt 5 system
Port input interrupt 6 system
Port input interrupt 7 system
Interrupt trigger mode/polarity control in ITC
Interrupt flag
EIFT0
EIFT1
EIFT2
EIFT3
EIFT4
EIFT5
EIFT6
EIFT7

Figure VI.1.5.1 Configuration of Port Input Interrupt Circuit

VI.1.5.1 Selecting Input Pins

Each port input interrupt system allows selection of an I/O port pin from the eight predefined pins.

The following lists the control bits and registers to select a port for each interrupt system.

- **PINT0[3:0]**: Interrupt Port Select Bits in the Port Input Interrupt 0 Select (PINTSEL0) Register (D[7:4]/0x4440)
- **PINT1[3:0]**: Interrupt Port Select Bits in the Port Input Interrupt 1 Select (PINTSEL1) Register (D[7:4]/0x4441)
- **PINT2[3:0]**: Interrupt Port Select Bits in the Port Input Interrupt 2 Select (PINTSEL2) Register (D[7:4]/0x4442)
- **PINT3[3:0]**: Interrupt Port Select Bits in the Port Input Interrupt 3 Select (PINTSEL3) Register (D[7:4]/0x4443)
- **PINT4[3:0]**: Interrupt Port Select Bits in the Port Input Interrupt 4 Select (PINTSEL4) Register (D[7:4]/0x4444)
- **PINT5[3:0]**: Interrupt Port Select Bits in the Port Input Interrupt 5 Select (PINTSEL5) Register (D[7:4]/0x4445)
- **PINT6[3:0]**: Interrupt Port Select Bits in the Port Input Interrupt 6 Select (PINTSEL6) Register (D[7:4]/0x4446)
- **PINT7[3:0]**: Interrupt Port Select Bits in the Port Input Interrupt 7 Select (PINTSEL7) Register (D[7:4]/0x4447)

Table VI.1.5.1.1 shows the selectable pins for each interrupt system.

Table VI.1.5.1.1 Selecting Pins for Port Input Interrupts

Port input interrupt system	PINT*x*[3:0] settings					
	0x0	0x1	0x2	0x3	0x4	0x5
0	P00	P10	P20	P30	P40	P50
1	P01	P11	P21	P31	P41	P51
2	P02	P12	P22	P32	P42	P52
3	P03	P13	P23	–	P43	P53
4	–	P14	P24	–	P44	–
5	–	P15	P25	P35	–	–
6	–	P16	P26	P36	–	–
7	–	–	P27	P37	–	–

VI.1.5.2 Control Registers of the Interrupt Controller

Selecting the trigger mode and polarity

The interrupt controller (ITC) provides two trigger modes for the port interrupts, the pulse trigger mode and the level trigger mode, to accept either a pulse signal or a level signal as interrupt requests.
The trigger mode can be selected using the EITG*x* bit in the ITC_ELV*x* registers (0x4306 to 0x430c). When EITG*x* is set to 1, level trigger mode is selected; when EITG*x* is set to 0 (default), pulse trigger mode is selected.

The ITC allows these interrupt sources to select the polarity of the interrupt request signal to be sent to the ITC. The signal polarity can be selected using the EITP*x* bit in the ITC_ELV*x* registers (0x4306 to 0x430c). When EITP*x* is set to 1, positive pulse/rising edge (in pulse trigger mode) or active high (in level mode) is selected; when EIPT*x* is set to 0 (default), negative pulse/falling edge (in pulse trigger mode) or active low (in level mode) is selected.

Table VI.1.5.2.1 Trigger Mode/Polarity Select Bits

Interrupt system	Trigger mode select bit	Trigger polarity select bit	Register address
Port input interrupt 0	EITG0 (D4/ITC_ELV0 register)	EITP0 (D5/ITC_ELV0 register)	0x4306
Port input interrupt 1	EITG1 (D12/ITC_ELV0 register)	EITP1 (D13/ITC_ELV0 register)	0x4306
Port input interrupt 2	EITG2 (D4/ITC_ELV1 register)	EITP2 (D5/ITC_ELV1 register)	0x4308
Port input interrupt 3	EITG3 (D12/ITC_ELV1 register)	EITP3 (D13/ITC_ELV1 register)	0x4308
Port input interrupt 4	EITG4 (D4/ITC_ELV2 register)	EITP4 (D5/ITC_ELV2 register)	0x430a
Port input interrupt 5	EITG5 (D12/ITC_ELV2 register)	EITP5 (D13/ITC_ELV2 register)	0x430a
Port input interrupt 6	EITG6 (D4/ITC_ELV3 register)	EITP6 (D5/ITC_ELV3 register)	0x430c
Port input interrupt 7	EITG7 (D12/ITC_ELV3 register)	EITP7 (D13/ITC_ELV3 register)	0x430c

With these registers, the port input interrupt condition is determined as shown in Table VI.1.5.2.2.

Table VI.1.5.2.2 Port Input Interrupt Condition

EITG*x*	EITP*x*	Port input interrupt condition
1	1	High level
1	0	Low level
0	1	Rising edge
0	0	Falling edge

ITC registers to control interrupt generation

Table VI.1.5.2.3 shows the interrupt control registers of the ITC that are provided for each port input interrupt system.

Table VI.1.5.2.3 Control Registers of Interrupt Controller

Interrupt system	Interrupt flag	Interrupt enable bit	Interrupt level setup bits
Port input interrupt 0	EIFT0 (D0/ITC_IFLG)	EIEN0 (D0/ITC_EN)	EILV0[2:0] (D[2:0]/ITC_ELV0)
Port input interrupt 1	EIFT1 (D1/ITC_IFLG)	EIEN1 (D1/ITC_EN)	EILV1[2:0] (D[10:8]/ITC_ELV0)
Port input interrupt 2	EIFT2 (D2/ITC_IFLG)	EIEN2 (D2/ITC_EN)	EILV2[2:0] (D[2:0]/ITC_ELV1)
Port input interrupt 3	EIFT3 (D3/ITC_IFLG)	EIEN3 (D3/ITC_EN)	EILV3[2:0] (D[10:8]/ITC_ELV1)
Port input interrupt 4	EIFT4 (D4/ITC_IFLG)	EIEN4 (D4/ITC_EN)	EILV4[2:0] (D[2:0]/ITC_ELV2)
Port input interrupt 5	EIFT5 (D5/ITC_IFLG)	EIEN5 (D5/ITC_EN)	EILV5[2:0] (D[10:8]/ITC_ELV2)
Port input interrupt 6	EIFT6 (D6/ITC_IFLG)	EIEN6 (D6/ITC_EN)	EILV6[2:0] (D[2:0]/ITC_ELV3)
Port input interrupt 7	EIFT7 (D7/ITC_IFLG)	EIEN7 (D7/ITC_EN)	EILV7[2:0] (D[10:8]/ITC_ELV3)

ITC_IFLG register (0x4300)
ITC_EN register (0x4302)
ITC_ELV0, ITC_ELV1, ITC_ELV2, ITC_ELV3 registers (0x4306, 0x4308, 0x430a, 0x430c)

When the interrupt generation condition described above is met, the corresponding interrupt flag is set to 1.
If the interrupt enable bit corresponding to that interrupt flag has been set to 1, the ITC sends an interrupt request to the S1C17 Core. To disable the port input interrupt, set the interrupt enable bit to 0.
The interrupt flag is always set to 1 by the designated port input, regardless of how the interrupt enable bit is set (even when set to 0).
The interrupt level setup bits set the interrupt level (0 to 7) of the port input interrupt. If the same interrupt level is set, port input interrupt 0 has highest priority and port input interrupt 7 has lowest priority.

An interrupt request to the S1C17 Core is accepted only when all the conditions described below are met.

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

For details on these interrupt control registers, as well as the device operation when an interrupt has occurred, see Section III.1, "Interrupt Controller (ITC)."

Interrupt vectors

The following shows the vector numbers and vector addresses for the port input interrupts:

Table VI.1.5.2.4 Port Input Interrupt Vectors

Interrupt system	Vector number	Vector address
Port input interrupt 0	4 (0x4)	TTBR + 0x10
Port input interrupt 1	5 (0x5)	TTBR + 0x14
Port input interrupt 2	6 (0x6)	TTBR + 0x18
Port input interrupt 3	7 (0x7)	TTBR + 0x1c
Port input interrupt 4	12 (0xc) or 16 (0x10)	TTBR + 0x30 or TTBR + 0x40
Port input interrupt 5	13 (0xd) or 17 (0x11)	TTBR + 0x34 or TTBR + 0x44
Port input interrupt 6	14 (0xe) or 18 (0x12)	TTBR + 0x38 or TTBR + 0x48
Port input interrupt 7	15 (0xf) or 19 (0x13)	TTBR + 0x3c or TTBR + 0x4c

Note: The interrupt vector numbers 12–16 and 18–19 for port input interrupts 4–7 are shared with another cause of interrupt. To use the vector number for the port input interrupt, set the interrupt enable bit for the port input interrupt to 1 and clear one for another to 0.

Table VI.1.5.2.5 Interrupt Vectors 12 and 16 (UART CH.0, CLG_T16U0, and Port 4 Interrupts)

Interrupt enable bit			Interrupt vector 12	Interrupt vector 16
IIEN4 (UART CH.0)	IIEN0 (CLG_T16U0)	EIEN4 (Port 4)		
1	1	1	CLG_T16U0 interrupt	UART CH.0 interrupt
1	1	0	CLG_T16U0 interrupt	UART CH.0 interrupt
1	0	1	Port interrupt 4	UART CH.0 interrupt
1	0	0	–	UART CH.0 interrupt
0	1	1	CLG_T16U0 interrupt	Port interrupt 4
0	1	0	CLG_T16U0 interrupt	–
0	0	1	Port interrupt 4	Port interrupt 4
0	0	0	–	–

(Interrupt enable bit: 1 = enable, 0 = disable)

Table VI.1.5.2.6 Interrupt Vector 13 (CLG_T8FU0 and Port 5 Interrupts)

Interrupt enable bit		Interrupt vector 13
IIEN1 (CLG_T8FU0)	EIEN5 (Port 5)	
1	1	CLG_T8FU0 interrupt
1	0	CLG_T8FU0 interrupt
0	1	Port interrupt 5
0	0	–

Table VI.1.5.2.7 Interrupt Vectors 14 and 18 (SPI CH.0, CLG_T8S, and Port 6 Interrupts)

Interrupt enable bit			Interrupt vector 14	Interrupt vector 18
IIEN6 (SPI CH.0)	IIEN2 (CLG_T8S)	EIEN6 (Port 6)		
1	1	1	CLG_T8S interrupt	SPI CH.0 interrupt
1	1	0	CLG_T8S interrupt	SPI CH.0 interrupt
1	0	1	Port interrupt 6	SPI CH.0 interrupt
1	0	0	–	SPI CH.0 interrupt
0	1	1	CLG_T8S interrupt	Port interrupt 6
0	1	0	CLG_T8S interrupt	–
0	0	1	Port interrupt 6	Port interrupt 6
0	0	0	–	–

Table VI.1.5.2.8 Interrupt Vectors 15 and 19 (I^2C master, CLG_T8I, and Port 7 Interrupts)

Interrupt enable bit			Interrupt vector 15	Interrupt vector 19
IIEN7 (I^2C master)	IIEN3 (CLG_T8I)	EIEN7 (Port 7)		
1	1	1	CLG_T8I interrupt	I^2C master interrupt
1	1	0	CLG_T8I interrupt	I^2C master interrupt
1	0	1	Port interrupt 7	I^2C master interrupt
1	0	0	–	I^2C master interrupt
0	1	1	CLG_T8I interrupt	Port interrupt 7
0	1	0	CLG_T8I interrupt	–
0	0	1	Port interrupt 7	Port interrupt 7
0	0	0	–	–

VI.1.6 Details of Control Registers

Table VI.1.6.1 List of I/O Port Registers

Address	Register name		Function
0x4400	P0_DAT	P0 Port Input Data Register	P0 port input data
0x4402	P1_DAT	P1 Port Input/Output Data Register	P1 port input/output data
0x4403	P1_IOC	P1 Port I/O Control Register	Selects the P1 port I/O direction.
0x4404	P2_DAT	P2 Port Input/Output Data Register	P2 port input/output data
0x4405	P2_IOC	P2 Port I/O Control Register	Selects the P2 port I/O direction.
0x4406	P3_DAT	P3 Port Input/Output Data Register	P3 port input/output data
0x4407	P3_IOC	P3 Port I/O Control Register	Selects the P3 port I/O direction.
0x4408	P4_DAT	P4 Port Input/Output Data Register	P4 port input/output data
0x4409	P4_IOC	P4 Port I/O Control Register	Selects the P4 port I/O direction.
0x440a	P5_DAT	P5 Port Input/Output Data Register	P5 port input/output data
0x440b	P5_IOC	P5 Port I/O Control Register	Selects the P5 port I/O direction.
0x4420	P0_03_CFP	P00–P03 Port Function Select Register	Selects the P00–P03 port functions.
0x4422	P1_03_CFP	P10–P13 Port Function Select Register	Selects the P10–P13 port functions.
0x4423	P1_46_CFP	P14–P16 Port Function Select Register	Selects the P14–P16 port functions.
0x4425	P2_57_CFP	P25–P27 Port Function Select Register	Selects the P25–P27 port functions.
0x4426	P3_02_CFP	P30–P32 Port Function Select Register	Selects the P30–P32 port functions.
0x4427	P3_57_CFP	P35–P37 Port Function Select Register	Selects the P35–P37 port functions.
0x4428	P4_03_CFP	P40–P43 Port Function Select Register	Selects the P40–P43 port functions.
0x4429	P4_4_CFP	P44 Port Function Select Register	Selects the P44 port function.
0x442a	P5_03_CFP	P50–P53 Port Function Select Register	Selects the P50–P53 port functions.
0x4440	PINTSEL0	Port Input Interrupt 0 Select Register	Selects a P*x*0 port for input interrupt.
0x4441	PINTSEL1	Port Input Interrupt 1 Select Register	Selects a P*x*1 port for input interrupt.
0x4442	PINTSEL2	Port Input Interrupt 2 Select Register	Selects a P*x*2 port for input interrupt.
0x4443	PINTSEL3	Port Input Interrupt 3 Select Register	Selects a P*x*3 port for input interrupt.
0x4444	PINTSEL4	Port Input Interrupt 4 Select Register	Selects a P*x*4 port for input interrupt.
0x4445	PINTSEL5	Port Input Interrupt 5 Select Register	Selects a P*x*5 port for input interrupt.
0x4446	PINTSEL6	Port Input Interrupt 6 Select Register	Selects a P*x*6 port for input interrupt.
0x4447	PINTSEL7	Port Input Interrupt 7 Select Register	Selects a P*x*7 port for input interrupt.

The following describes each I/O port register. These are all 8-bit registers.

Note: When setting the registers, be sure to write a 0, and not a 1, for all “reserved bits.”

0x4400–0x440a: P*x* Port Input/Output Data Registers (P*x*_DAT)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
P*x* Port Input /Output Data Register (P*x*_DAT)	**0x4400** **|** **0x440a** (8 bits)	D7–0	**P*x*D[7:0]**	P*x*[7:0] port input/output data	1	1 (High)	0	0 (Low)	Ext.	R/W	Ext.: The initial value depends on the external pin status.

Note: The letter '*x*' in bit names, etc., denotes a port number from 0 to 5.

0x4400	P0 Port Input Data Register (P0_DAT) ∗ Input only
0x4402	P1 Port Input/Output Data Register (P1_DAT)
0x4404	P2 Port Input/Output Data Register (P2_DAT)
0x4406	P3 Port Input/Output Data Register (P3_DAT)
0x4408	P4 Port Input/Output Data Register (P4_DAT)
0x440a	P5 Port Input/Output Data Register (P5_DAT)

These registers are used to read data from I/O-port pins or to set output data. (Default: external pin status)

1 (R/W): High level
0 (R/W): Low level

When an I/O port is set for output, the data written to the register is directly output to the I/O port pin. If the data written to the port is 1, the port pin is set high (HVDD level); if the data is 0, the port pin is set low (VSS level). Even in input mode, data can be written to the port data register.

When the register is read, the voltage level on the port pin is read out regardless of whether an I/O port is set for input or output mode. If the pin voltage is high (HVDD level), 1 is read out as input data; if the pin voltage is low (VSS level), 0 is read out as input data.

Note: If noise may affect the input data, read it twice and verify that the read two data are the same.

0x4403–0x440b: P*x* I/O Control Registers (P*x*_IOC)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
P*x* I/O Control Register (P*x*_IOC)	**0x4403** | **0x440b** (8 bits)	D7–0	**IOC*x*[7:0]**	P*x*[7:0] I/O control	1	Output	0	Input	0	R/W	

Note: The letter '*x*' in bit names, etc., denotes a port number from 1 to 5.

0x4403	P1 I/O Control Register (P1_IOC)
0x4405	P2 I/O Control Register (P2_IOC)
0x4407	P3 I/O Control Register (P3_IOC)
0x4409	P4 I/O Control Register (P4_IOC)
0x440b	P5 I/O Control Register (P5_IOC)

Directs the I/O port for input or output and indicates the I/O control signal value of the port.

1 (R/W): Output mode
0 (R/W): Input mode (default)

Each I/O control register bit corresponds to each I/O port. When IOC*x* is set to 1, the corresponding I/O port is directed for output; if it is set to 0, the I/O port is directed for input.
When the pin is used for a peripheral function, the input/output direction depends on the peripheral function.

When the register is read, the I/O control signal value for the port pin is read out. When I/O port function is selected using the Port Function Select Register, the value written to the I/O Control Register is read out as is. When peripheral function is selected, the read value depends on the peripheral circuit status and may not indicate the value written to IOC*x*.

0x4420–0x442a: Port Function Select Registers (P*x*_*xx*_CFP)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
P*x*0–P*x*3 Port Function Select Register (P*x*_03_CFP) or P*x*4–P*x*7 Port Function Select Register (P*x*_47_CFP)	**0x4420 | 0x4439** (8 bits)	D7–6	**CFP*x*3[1:0] or CFP*x*7[1:0]**	P*x*3/P*x*7 port function select	CFP*x*3/7[1:0]	Function	0x0	R/W	
					0x3	Pin function 3			
					0x2	Pin function 2			
					0x1	Pin function 1			
					0x0	Pin function 0			
		D5–4	**CFP*x*2[1:0] or CFP*x*6[1:0]**	P*x*2/P*x*6 port function select	CFP*x*2/6[1:0]	Function	0x0	R/W	
					0x3	Pin function 3			
					0x2	Pin function 2			
					0x1	Pin function 1			
					0x0	Pin function 0			
		D3–2	**CFP*x*1[1:0] or CFP*x*5[1:0]**	P*x*1/P*x*5 port function select	CFP*x*1/5[1:0]	Function	0x0	R/W	
					0x3	Pin function 3			
					0x2	Pin function 2			
					0x1	Pin function 1			
					0x0	Pin function 0			
		D1–0	**CFP*x*0[1:0] or CFP*x*4[1:0]**	P*x*0/P*x*4 port function select	CFP*x*0/4[1:0]	Function	0x0	R/W	
					0x3	Pin function 3			
					0x2	Pin function 2			
					0x1	Pin function 1			
					0x0	Pin function 0			

Note: The letter '*x*' in bit names, etc., denotes a port number from 0 to 5.

0x4420	P00–P03 Port Function Select Register (P0_03_CFP)
0x4422	P10–P13 Port Function Select Register (P1_03_CFP)
0x4423	P14–P16 Port Function Select Register (P1_46_CFP)
0x4425	P25–P27 Port Function Select Register (P2_57_CFP)
0x4426	P30–P32 Port Function Select Register (P3_02_CFP)
0x4427	P35–P37 Port Function Select Register (P3_57_CFP)
0x4428	P40–P43 Port Function Select Register (P4_03_CFP)
0x4429	P44 Port Function Select Register (P4_4_CFP)
0x442a	P50–P53 Port Function Select Register (P5_03_CFP)

These bits select the function of each I/O port pin. (Default: 0x0 = Pin function 0)

The I/O ports concurrently serve as the input/output pins for peripheral circuits or bus signals. Whether they are used as I/O ports or for peripheral circuits/bus signals can be selected bit-for-bit using these registers. All pins that are not used for peripheral circuits/bus signals can be used as general-purpose I/O ports.

For details of pin functions, see Section I.3.3, "Switching Over the Multiplexed Pin Functions."

0x4440–0x4447: Port Input Interrupt *x* Select Registers (PINTSEL*x*)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
Port Input Interrupt *x* Select Register (PINTSEL*x*)	**0x4440** **|** **0x4447** (8 bits)	D7–4	**PINT*x*[3:0]**	Interrupt port select	PINT*x*[3:0]	Function	0xf	R/W	
					0xf–0x6	reserved			
					0x5	P5*x*			
					0x4	P4*x*			
					0x3	P3*x*			
					0x2	P2*x*			
					0x1	P1*x*			
					0x0	P0*x*			
		D3–0	–	reserved	–		X	R	

Note: The letter '*x*' in bit names, etc., denotes a port input interrupt number from 0 to 7.

0x4440	Port Input Interrupt 0 Select Register (PINTSEL0)
0x4441	Port Input Interrupt 1 Select Register (PINTSEL1)
0x4442	Port Input Interrupt 2 Select Register (PINTSEL2)
0x4443	Port Input Interrupt 3 Select Register (PINTSEL3)
0x4444	Port Input Interrupt 4 Select Register (PINTSEL4)
0x4445	Port Input Interrupt 5 Select Register (PINTSEL5)
0x4446	Port Input Interrupt 6 Select Register (PINTSEL6)
0x4447	Port Input Interrupt 7 Select Register (PINTSEL7)

PINT*x*[3:0]: Interrupt Port Select Bits

Selects an I/O port used to generate the port input interrupt.

Table VI.1.6.2 Selecting Pins for Port Input Interrupts

Port input interrupt system	PINT*x*[3:0] settings					
	0x0	0x1	0x2	0x3	0x4	0x5
0	P00	P10	P20	P30	P40	P50
1	P01	P11	P21	P31	P41	P51
2	P02	P12	P22	P32	P42	P52
3	P03	P13	P23	–	P43	P53
4	–	P14	P24	–	P44	–
5	–	P15	P25	P35	–	–
6	–	P16	P26	P36	–	–
7	–	–	P27	P37	–	–

(Default: 0xf)

VI
GPIO

VI.1.7 Precautions

- The interrupt vector numbers 12–16 and 18–19 for port input interrupts 4–7 are shared with another cause of interrupt. To use the vector number for the port input interrupt, set the interrupt enable bit for the port input interrupt to 1 and clear one for another to 0.
- When using a port input interrupt as the trigger to restart from SLEEP mode, level interrupt mode must be specified as an interrupt condition. Edge interrupt signals from ports cannot cancel SLEEP mode.
- When a port input interrupt (edge interrupt) is used, the input pulse width must be longer than 1 cycle of the system clock to be certain an interrupt will be generated.
- If noise may affect the input data, read it twice and verify that the read two data are the same.
- When using the port input interrupt (level interrupt), the input signal must be set to the active level for more than the time shown below.

  (1) When the clock is stopped during SLEEP mode
  OSC3 oscillation start time + OSC3 oscillation stabilization wait time (set by the user) + 10 system clock cycle time

  (2) When the clock is not stopped during SLEEP mode, or in HALT mode
  10 system clock cycle time

  (3) During normal operation
  10 system clock cycle time

S1C17002 Technical Manual

VII S1C17002 ANALOG MODULE

VII.1 A/D Converter (ADC)

VII.1.1 Features and Structure of A/D Converter

The S1C17002 contains an A/D converter with the following features:

- Conversion method: Successive comparison
- Resolution: 10 bits
- Input channels: 4 channels
- A/D converter input clock: Maximum of 2 MHz, minimum of 16 kHz
- Conversion time: 9 clocks (sampling time) + 11 clocks (conversion time) = 20 clocks
  Minimum of 10 µs (when a 2-MHz input clock is selected)
  Maximum of 1250 µs (when a 16-kHz input clock is selected)
- Conversion range: Between V_{SS} and AV_{DD}
- Two conversion modes can be selected:
  Normal mode: Conversion is completed in one operation.
  Continuous mode: Conversion is continuous and terminated through software control.
  Continuous conversion of multiple channels can be performed in each mode.
- Four types of A/D-conversion start triggers can be selected:
  Triggered by the external pin (#ADTRG)
  Triggered by the period-match of the 16-bit multi-function timer (MFT)
  Triggered by the underflow of the 8-bit programmable timer (PT8) CH.0
  Triggered by the software
- A/D conversion results can be read out from the 10-bit data register or the conversion result buffer* for each channel.
- An interrupt is generated upon completion of A/D conversion or when the conversion result is out of the specified range (upper and lower-limit values can be specified)*.
- The upper-limit and lower-limit out-of-range signals of CH.0 are sent to the MFT for controlling IGBT.

* These functions can be used in the advanced mode. The A/D converter of the S1C17002 has two operating modes, standard mode of which functions are compatible with the legacy analog block for the existing models and an advanced mode allowing use of the extended functions.

Figure VII.1.1.1 shows the structure of the A/D converter.

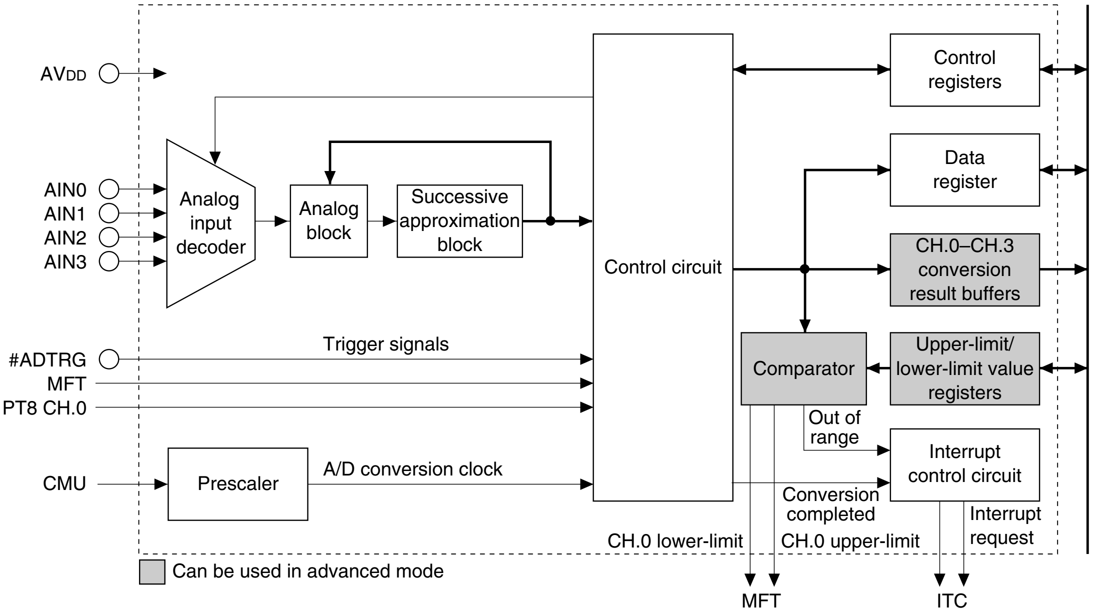


Figure VII.1.1.1 Structure of A/D Converter

VII.1.2 Input Pins of A/D Converter

Table VII.1.2.1 shows the pins used by the A/D converter.

Table VII.1.2.1 Input Pins of A/D Converter

Pin name	I/O	Function
AV_{DD}	–	Analog power-supply pin AV_{DD} is the power-supply pin for the analog circuit.
AIN0–AIN3	I	Analog signal input pins AIN0 (CH.0)–AIN3 (CH.3) The analog input voltage AV_{IN} can be input in the range of $V_{SS} \le AV_{IN} \le AV_{DD}$.
#ADTRG	I	External trigger input pin This pin is used to input a trigger signal to start A/D conversion from an external source.

The A/D converter input pins (AIN[3:0], #ADTRG) are shared with the input ports and they are initialized as general-purpose input port pins by default. Before using these pins for the A/D converter, the pin functions must be switched using the Port Function Select registers.

For details on switching pin function, Section I.3.3, "Switching Over the Multiplexed Pin Functions."

Notes:
- When the A/D converter is enabled, a current flows between AV_{DD} and V_{SS}, and power is consumed, even when A/D operations are not performed. Therefore, when the A/D converter is not used, it must be disabled (default 0 setting of ADE (D2/AD_CTL register)).

  * **ADE**: A/D Enable Bit in the A/D Control/Status (AD_CTL) Register (D2/0x5544)

- Take measures against noise when designing the #ADTRG signal path on the printed circuit board.
- Be aware that the interface voltage level is AV_{DD} even if the AIN*x* pin is used as an input port (P0*x*) pin.

VII.1.3 A/D Converter Operating Clock

The A/D converter use the ADC_CLK clock (= PCLK) generated by the CMU as the operating clock. The conversion clock is generated in the A/D converter module.

Controlling the supply of the operating clock

ADC_CLK is supplied to the A/D converter with default settings. It can be turned off using ADC_CLK_EN (D3/CMU_GATEDCLK2 register) to reduce the amount of power consumed on the chip if the A/D converter is not used.

∗ **ADC_CLK_EN**: ADC Module Clock Control Bit in the Gated Clock Control 2 (CMU_GATEDCLK2) Register (D3/0x4908)

Setting ADC_CLK_EN to 0 (1 by default) turns off the clock supply to the A/D converter. When the clock supply is turned off, the A/D converter control registers cannot be accessed.
For details on how to set and control the clock, refer to Section II.2, "Clock Management Unit (CMU)."

Note: The Gated Clock Control 2 Register (0x4908) is write-protected. Write protection of this and other CMU registers at addresses 0x4900 to 0x4908 to be rewritten must be removed by writing 0x96 to the CMU Write Protect Register (0x4920). Since unnecessary rewrites to addresses 0x4900 to 0x4908 could cause the system to operate erratically, make sure the data set in the CMU Write Protect Register (0x4920) is other than 0x96, unless rewriting said registers.

Clock state in standby mode

The clock supply to the A/D converter stops depending on type of standby mode.
HALT mode: The operating clock is supplied the same way as in normal mode.
SLEEP mode: The operating clock supply stops.

Therefore, the A/D converter also stops operating in SLEEP mode.

VII.1.4 Setting A/D Converter

When the A/D converter is used, the following settings must be made before an A/D conversion can be performed:

1. Setting analog input pins ... See Sections VII.1.2 and I.3.3.
2. Setting the operating mode (standard mode/advanced mode)
3. Setting the input clock
4. Selecting the analog-conversion start and end channels
5. Setting the A/D conversion mode
6. Selecting a trigger
7. Setting the sampling time
8. Setting the upper-limit and lower-limit values (advanced mode)
9. Setting the interrupt mode (advanced mode)
10. Setting interrupt ... See Section VII.1.6.

Note: Before making these settings, make sure the A/D converter is disabled (ADE (D2/AD_CTL register) = 0). Changing the settings while the A/D converter is enabled could cause a malfunction.

∗ **ADE**: A/D Enable Bit in the A/D Control/Status (AD_CTL) Register (D2/0x5544)

Setting the operating mode (standard mode / advanced mode)

The A/D converter of the S1C17002 has two operating modes, standard mode of which functions are compatible with the legacy analog block for the existing models and an advanced mode allowing use of the extended functions. Table VII.1.4.1 shows differences between the standard mode and the advanced mode.

Table VII.1.4.1 Differences Between Standard Mode and Advanced Mode

Function	Standard mode	Advanced mode
Reading conversion results	The conversion results are read from the A/D conversion result register common to all channels. When converting for multiple channels, the A/D conversion result register must be read before conversion for the next channel has completed.	The conversion results can be read from the conversion result buffer provided for each channel. Thus the conversion result for the current channel will not be lost even when the conversion for the next channel is completed during a multiple channel conversion.
Conversion-complete flag, overwrite error flag	One bit is assigned for the flag and is commonly used in all channels.	Different flags are provided for each channel.
Comparison with upper/lower-limit values	Not supported.	An upper-limit value and a lower-limit value can be set and conversion results of the specified channel can be checked whether they are within the specified range or not.
Interrupts	Conversion-complete interrupt only can be generated. The interrupts cannot be masked in channel units.	Conversion-complete interrupts and out-of-range interrupts can be generated. Conversion complete interrupts for the specified channels can be masked.

To configure the A/D converter in the advanced mode, set ADCADV (D8/AD_ADVMODE register) to 1. The control bits for the extended functions can be accessed after this setting. At initial reset, ADCADV is set to 0 and the A/D converter enters the standard mode.

∗ **ADCADV**: Standard/Advanced Mode Selection Bit in the A/D Converter Mode Select/Internal Status (AD_ADVMODE) Register (D8/0x555e)

The following descriptions unless otherwise specified are common contents for both modes.
The extended functions in the advanced mode are explained assuming that ADCADV has been set to 1.

Setting the input clock

The A/D converter contains a prescaler and the A/D conversion clock can be selected from among the eight types shown in Table VII.1.4.2 below. Use PSAD[2:0] (D[2:0]/AD_CLKCTL register) for this selection.

∗ **PSAD[2:0]**: A/D Converter Clock Division Ratio Selection Bits in the A/D Clock Control (AD_CLKCTL) Register (D[2:0]/0x5520)

Table VII.1.4.2 Input Clock Selection

PSAD[2:0]	A/D clock
0x7	PCLK•1/256
0x6	PCLK•1/128
0x5	PCLK•1/64
0x4	PCLK•1/32
0x3	PCLK•1/16
0x2	PCLK•1/8
0x1	PCLK•1/4
0x0	PCLK•1/2

(Default: 0x0)

The selected clock is output from the prescaler by writing 1 to PSONAD (D3/AD_CLKCTL register).

∗ **PSONAD**: A/D Converter Clock Control Bit in the A/D Clock Control (AD_CLKCTL) Register (D3/0x5520)

Notes:
- The recommended input clock frequency is a maximum of 2 MHz and a minimum of 16 kHz.
- Do not start an A/D conversion when the clock output from the prescaler is turned off, and do not turn off the prescaler's clock output when an A/D conversion is underway. This could cause the A/D converter to operate erratically.

Selecting analog-conversion start and end channels

Select the channel in which the A/D conversion is to be performed from among the pins (channels) that have been set for analog input. To enable A/D conversions in multiple channels to be performed successively through one convert operation, specify the conversion start and conversion end channels using CS[2:0] (D[10:8]/AD_TRIG_CH register) and CE[2:0] (D[13:11]/AD_TRIG_CH register) respectively.

∗ **CS[2:0]**: A/D Converter Start Channel Selection Bits in the A/D Trigger/Channel Select (AD_TRIG_CH) Register (D[10:8]/0x5542)
∗ **CE[2:0]**: A/D Converter End Channel Selection Bits in the A/D Trigger/Channel Select (AD_TRIG_CH) Register (D[13:11]/0x5542)

Table VII.1.4.3 Relationship between CS/CE and Input Channel

CS[2:0]/CE[2:0]	Channel selected
0x7–0x4	Reserved
0x3	AIN3
0x2	AIN2
0x1	AIN1
0x0	AIN0

(Default: 0x0)

Example: Operation of one A/D conversion

CS[2:0] = 0, CE[2:0] = 0: Converted only in AIN0
CS[2:0] = 0, CE[2:0] = 3: Converted in the following order: AIN0→AIN1→AIN2→AIN3
CS[2:0] = 2, CE[2:0] = 1: Converted in the following order: AIN2→AIN3→(AIN4)→(AIN5)→(AIN6)→(AIN7)→AIN0→AIN1

Note: The control circuits in the A/D converter supports up to eight channels for expansion in the future, and it performs A/D conversion if a channel (AIN4–AIN7) without an analog input is specified. In this case, the results that will be stored to ADD[9:0] (A/D Conversion Result Register) is 0x0.
To avoid A/D conversion for the channels without an input, set the CS[2:0] to equal or smaller than CE[2:0] within the available analog inputs.

Setting the A/D conversion mode

The A/D converter can operate in one of the following two modes. This operation mode is selected using MS (D5/AD_TRIG_CH register).

∗ **MS**: A/D Conversion Mode Selection Bit in the A/D Trigger/Channel Select (AD_TRIG_CH) Register (D5/0x5542)

1. Normal mode (MS = 0)

All inputs in the range of channels set using CS[2:0] (D[10:8]/AD_TRIG_CH register) and CE[2:0] (D[13:11]/AD_TRIG_CH register) are A/D converted once and then stopped.

2. Continuous mode (MS = 1)

A/D conversions in the range of channels set using CS[2:0] and CE[2:0] are executed successively until stopped by the software.

At initial reset, the normal mode is selected.

Selecting a trigger

Use TS[1:0] (D[4:3]/AD_TRIG_CH register) to select a trigger to start A/D conversion from among the four types shown in Table VII.1.4.4.

∗ **TS[1:0]**: A/D Conversion Trigger Selection Bits in the A/D Trigger/Channel Select (AD_TRIG_CH) Register (D[4:3]/0x5542)

Table VII.1.4.4 Trigger Selection

TS[1:0]	Trigger source
0x3	External trigger (#ADTRG pin)
0x2	8-bit programmable timer (PT8) CH.0
0x1	16-bit multi-function timer (MFT)
0x0	Software trigger

(Default: 0x0)

1. External (#ADTRG) trigger

The signal input to the #ADTRG pin is used as a trigger. When this trigger is used, the #ADTRG pin must be set in advance using the port function select register. A/D conversion is started when a low level of the #ADTRG signal is detected.

2. 8-bit programmable timer (PT8) CH.0

The underflow signal of the PT8 CH.0 is used as a trigger. Since the cycle can be programmed using the timer, this trigger is effective when cyclic A/D conversions are required.
For details on how to set the timer, refer to the explanation of the PT8 in this manual.

3. 16-bit multi-function timer (MFT)

The period-match signal of the 16-bit multi-function timer (MFT) is used as a trigger. Since the cycle can be programmed using the timer, this trigger is effective when cyclic A/D conversions are required.
For details on how to set the timer, refer to the explanation of the MFT in this manual.

4. Software trigger

Writing 1 to ADST (D1/AD_CTL register) in the software serves as a trigger to start A/D conversion.

∗ **ADST**: A/D Conversion Control/Status Bit in the A/D Control/Status (AD_CTL) Register (D1/0x5544)

Setting the sampling time

The A/D converter contains ST[1:0] (D[9:8]/AD_CTL register) that allows the analog-signal input sampling time to be set in four steps (3, 5, 7, or 9 times the conversion clock period).
However, this register should be used as set by default (ST[1:0] = 11; 9 clock periods).

∗ **ST[1:0]**: Input Signal Sampling Time Setup Bits in the A/D Control/Status (AD_CTL) Register (D[9:8]/0x5544)

Setting the upper-limit and lower-limit values (advanced mode)

The advanced mode allows a range check of the conversion results by setting the upper-limit and lower-limit values. When the conversion result is out of the range, the A/D converter can output the interrupt signal to the interrupt controller (ITC). Furthermore, the A/D converter CH.0 outputs the upper-limit out-of-range or lower-limit out-of-range signal to the MFT for controlling IGBT (see Section IV.2 for more information).

1. Selecting the channel

Select the channel to compare the A/D conversion results and the upper-limit and lower-limit value using ADCMP[2:0] (D[14:12]/AD_CTL register).

∗ **ADCMP[2:0]**: Upper/Lower-Limit Comparison Channel Selection Bits in the A/D Control/Status (AD_CTL) Register (D[14:12]/0x5544)

Table VII.1.4.5 Selecting the Channel for Checking Conversion Results

ADCMP[2:0]	Channel selected
0x7–0x4	Reserved
0x3	AIN3
0x2	AIN2
0x1	AIN1
0x0	AIN0

(Default: 0x0)

ADCMP[2:0] selects the channel to generate the out-of-range interrupt. The A/D converter CH.0 can output the out-of-range signal to the MFT even if another channel is selected by ADCMP[2:0].

2. Setting upper-limit and lower-limit values

Set the upper-limit value to ADUPR[9:0] (D[9:0]/AD_UPPER register) and the lower-limit value to ADLWR[9:0] (D[9:0]/AD_LOWER register).

∗ **ADUPR[9:0]**: A/D Conversion Upper Limit Value Setup Bits in the A/D Upper Limit Value (AD_UPPER) Register (D[9:0]/0x5558)

∗ **ADLWR[9:0]**: A/D Conversion Lower Limit Value Setup Bits in the A/D Lower Limit Value (AD_LOWER) Register (D[9:0]/0x555a)

When the conversion result exceeds the upper-limit value set or is lower than the lower-limit value, it is determined as out of range. If the conversion result is the same value as the upper-limit or lower-limit value, it is determined as within the range.

3. Enabling comparison with the upper-limit and lower-limit values

Set ADCMPE (D15/AD_CTL register) to 1 to enable the range check function.

∗ **ADCMPE**: Upper/Lower-Limit Comparison Enable Bit in the A/D Control/Status (AD_CTL) Register (D15/0x5544)

The A/D converter CH.0 can output the out-of-range signal to the MFT when the ADCMPE is set.

Setting the interrupt mode (advanced mode)

The interrupt functions are extended in the advanced mode, so the following configuration is necessary.

1. Enabling/disabling the conversion-complete interrupt

The conversion-complete interrupt can be enabled/disabled using CNVINTEN (D4/AD_CTL register). Set CNVINTEN to 1 when using the conversion-complete interrupt, or to 0 when it is not used. At initial reset, CNVINTEN is set to 1, so the conversion-complete interrupt function is enabled.

∗ **CNVINTEN**: Conversion-Complete Interrupt Enable Bit in the A/D Control/Status (AD_CTL) Register (D4/0x5544)

2. Enabling/disabling the out-of-range interrupt

The out-of-range interrupt can be enabled/disabled using CMPINTEN (D5/AD_CTL register). Set CMPINTEN to 1 when using the out-of range interrupt, or to 0 when it is not used. At initial reset, CMPINTEN is set to 0, so the out-of-range interrupt function is disabled.

∗ **CMPINTEN**: Out-of-Range Interrupt Enable Bit in the A/D Control/Status (AD_CTL) Register (D5/0x5544)

3. Setting the interrupt signal mode

The S1C17002 A/D converter has two interrupt request outputs for the interrupt sources above and each interrupt can be handled individually. In the initial setting, the out-of-range interrupt signal is ORed with the conversion-complete interrupt signal to send to the ITC. So, the conversion-complete interrupt flag in the ITC is set when an A/D conversion has completed or when the conversion results are out of range. This signal mode can be canceled using INTMODE (D6/AD_CTL register). To handle each interrupt individually, set INTMODE to 1. In this setting, the out-of-range interrupt signal is not ORed with the conversion-complete interrupt signal.

∗ **INTMODE**: Interrupt Signal Mode Bit in the A/D Control/Status (AD_CTL) Register (D6/0x5544)

4. Masking conversion-complete interrupt for the specified channels

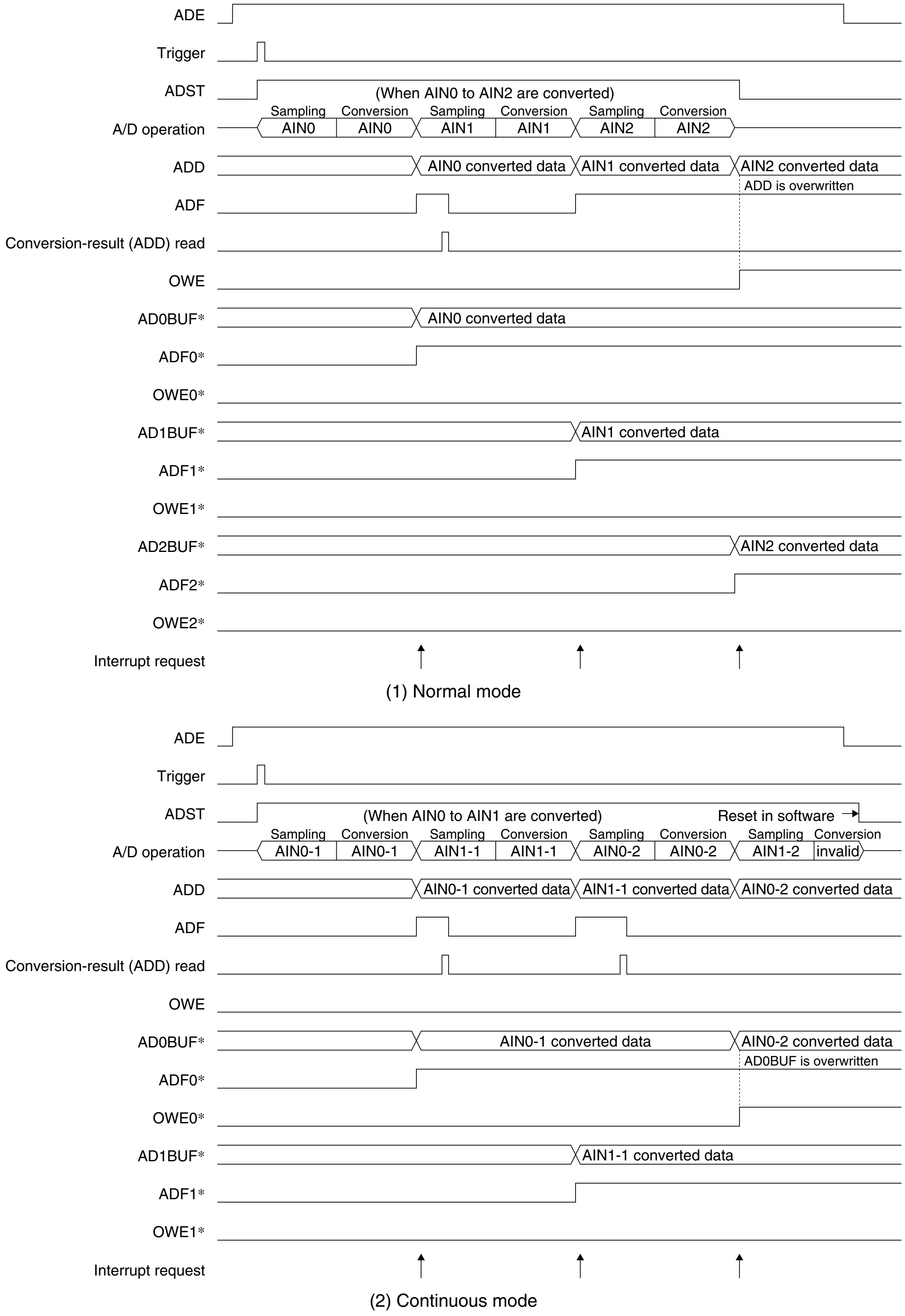
The A/D conversion-complete interrupt mask register is used to mask the conversion-complete interrupts of the specified channels. When INTMASK*x* (D*x*/AD_INTMASK register) for channel '*x*' in the register is set to 0, channel '*x*' does not generate conversion-complete interrupts. For instance, by masking the conversion-complete interrupt of the channel used for range checking, it is possible to generate out-of range interrupts only.

∗ **INTMASK*x***: CH.*x* Conversion-Complete Interrupt Mask Bit in the A/D Conversion Complete Interrupt Mask (AD_INTMASK) Register (D*x*/0x555c)

At initial reset, INTMASK*x* are all set to 1 to enable conversion-complete interrupts.

VII.1.5 Control and Operation of A/D Conversion

Figure VII.1.5.1 shows the operation of the A/D converter.



(1) Normal mode

(2) Continuous mode

∗ Extended functions that can be used when ADCADV = 1

Figure VII.1.5.1 Operation of A/D Converter

Starting up the A/D converter circuit

After the settings specified in the preceding section have been made, write 1 to ADE (D2/AD_CTL register) to enable the A/D converter. The A/D converter is thereby ready to accept a trigger to start A/D conversion. To set the A/D converter again, or if it is not used, set ADE to 0.

∗ **ADE**: A/D Enable Bit in the A/D Control/Status (AD_CTL) Register (D2/0x5544)

Starting A/D conversion

When a trigger is input while ADE = 1, A/D conversion is started. If a software trigger has been selected, A/D conversion is started by writing 1 to ADST (D1/AD_CTL register).

∗ **ADST**: A/D Conversion Control/Status Bit in the A/D Control/Status (AD_CTL) Register (D1/0x5544)

Only the trigger selected using TS[1:0] (D[4:3]/AD_TRIG_CH register) are valid; no other trigger is accepted.

∗ **TS[1:0]**: A/D Conversion Trigger Selection Bits in the A/D Trigger/Channel Select (AD_TRIG_CH) Register (D[4:3]/0x5542)

When a trigger is input, the A/D converter samples and A/D-converts the analog input signal, beginning with the conversion start channel selected by CS[2:0] (D[10:8]/AD_TRIG_CH register).

∗ **CS[2:0]**: A/D Converter Start Channel Selection Bits in the A/D Trigger/Channel Select (AD_TRIG_CH) Register (D[10:8]/0x5542)

ADST used for the software trigger is set to 1 during A/D conversion, even when it is started by some other trigger, so it can be used as an A/D-conversion status bit.
The channel in which conversion is underway can be identified by reading CH[2:0] (D[2:0]/AD_TRIG_CH register).

∗ **CH[2:0]**: A/D Conversion Channel Status Bits in the A/D Trigger/Channel Select (AD_TRIG_CH) Register (D[2:0]/0x5542)

Reading out A/D conversion results

• Standard mode

Upon completion of the A/D conversion in the start channel, the A/D converter stores the conversion result into the 10-bit data registers ADD[9:0] (D[9:0]/AD_DAT register) and sets the conversion-complete flag ADF (D3/AD_CTL register). If multiple channels are specified using CS[2:0] (D[10:8]/AD_TRIG_CH register) and CE[2:0] (D[13:11]/AD_TRIG_CH register), A/D conversions in the subsequent channels are performed in succession.

∗ **ADD[9:0]**: A/D Converted Data Bits in the A/D Conversion Result (AD_DAT) Register (D[9:0]/0x5540)
∗ **ADF**: Conversion-Complete Flag Bit in the A/D Control/Status (AD_CTL) Register (D3/0x5544)
∗ **CE[2:0]**: A/D Converter End Channel Selection Bits in the A/D Trigger/Channel Select (AD_TRIG_CH) Register (D[13:11]/0x5542)

The results of A/D conversion are stored in ADD[9:0] each time conversion in one channel is completed. Since an interrupt can be generated simultaneously, this interrupt is normally used to read out the converted data. In addition, be sure to reset the interrupt flag (by writing 0) to prepare the A/D converter for the next operation.

If multiple A/D conversion channels are specified, the conversion results in one channel must be read out prior to completion of conversion in the next channel. If the A/D conversion currently under way is completed before the previous conversion results are read out, ADD[9:0] is overwritten with the new conversion results.
If ADD[9:0] is updated when the conversion-complete flag ADF (D3/AD_CTL register) = 1 (before the converted data is read out), the overwrite-error flag OWE (D0/AD_CTL register) is set to 1. The conversion-complete flag ADF is reset to 0 when the converted data is read out. If ADD[9:0] is updated when ADF = 0, OWE remains at 0, indicating that the operation has been completed normally. When reading out data, also read OWE to make sure the data is valid. Once OWE is set, it remains set until it is reset to 0 in the software. Note also that if OWE is set, ADF also is set. In this case, read out the converted data and reset ADF.

∗ **OWE**: Overwrite Error Flag Bit in the A/D Control/Status (AD_CTL) Register (D0/0x5544)

- **Advanced mode**

Upon completion of the A/D conversion in the start channel (CH.*x*), the A/D converter stores the conversion result to the 10-bit CH.*x* conversion result buffer AD*x*BUF[9:0] (D[9:0]/AD_CH*x*_BUF register) and sets the CH.*x* conversion-complete flag ADF*x* (D*x*/AD_CH_STAT register). If multiple channels are specified using CS[2:0] (D[10:8]/AD_TRIG_CH register) and CE[2:0] (D[13:11]/AD_TRIG_CH register), A/D conversions in the subsequent channels are performed in succession.

∗ **AD*x*BUF[9:0]**: A/D CH.*x* Converted Data Bits in the A/D CH.*x* Conversion Result Buffer (AD_CH*x*_BUF) Register (D[9:0]/0x5548 + 2•*x*)
∗ **ADF*x***: CH.*x* Conversion-Complete Flag Bit in the A/D Channel Status Flag (AD_CH_STAT) Register (D*x*/0x5546)

The results of A/D conversion are stored in the A/D conversion result buffer for each channel each time conversion in one channel is completed. Since an interrupt can be generated simultaneously, this interrupt is normally used to read out the converted data. In addition, be sure to reset the interrupt flag (by writing 0) to prepare the A/D converter for the next operation.

In the advanced mode, each channel has a conversion result buffer, so it is not necessary to read the conversion results prior to completion of conversion in the next channel. However, if the next A/D conversion in the same channel is completed before the previous conversion results are read out, the conversion result buffer is overwritten with the new conversion results. If AD*x*BUF[9:0] is updated when the conversion-complete flag ADF*x* = 1 (before the converted data is read out), the overwrite-error flag OWE*x* (D*x* + 8/AD_CH_STAT register) is set to 1. ADF*x* is reset to 0 when the converted data is read out. If AD*x*BUF[9:0] is updated when ADF*x* = 0, OWE*x* remains at 0, indicating that the operation has been completed normally. When reading out data, also read OWE*x* to make sure the data is valid. Once OWE*x* is set, it remains set until it is reset to 0 by writing 0 in the software. Note also that if OWE*x* is set, ADF*x* is also set. In this case, read out the converted data and reset ADF*x*.

∗ **OWE*x***: CH.*x* Overwrite Error Flag Bit in the A/D Channel Status Flag (AD_CH_STAT) Register (D*x* + 8/0x5546)

ADD[9:0] (D[9:0]/AD_DAT register), ADF (D3/AD_CTL register) and OWE (D0/AD_CTL register) used in the standard mode are also effective in the advanced mode as well. The functions and actions of the register/bits are the same as those of the standard mode. OWE is set during conversion in multiple-channels, but it is not necessary to reset it.

Range check (comparison with upper-limit/lower-limit values in advanced mode)

When the range check function is enabled (ADCMPE (D15/AD_CTL register) = 1) and an A/D conversion in the channel specified using ADCMP[2:0] (D[14:12]/AD_CTL register) has completed, the conversion results are compared with the contents of ADUPR[9:0] (D[9:0]/AD_UPPER register) and ADLWR[9:0] (D[9:0]/AD_LOWER register).

∗ **ADCMPE**: Upper/Lower-Limit Comparison Enable Bit in the A/D Control/Status (AD_CTL) Register (D15/0x5544)
∗ **ADCMP[2:0]**: Upper/Lower-Limit Comparison Channel Selection Bits in the A/D Control/Status (AD_CTL) Register (D[14:12]/0x5544)
∗ **ADUPR[9:0]**: A/D Conversion Upper Limit Value Bits in the A/D Upper Limit Value (AD_UPPER) Register (D[9:0]/0x5558)
∗ **ADLWR[9:0]**: A/D Conversion Lower Limit Value Bits in the A/D Lower Limit Value (AD_LOWER) Register (D[9:0]/0x555a)

If the conversion results exceed the upper-limit value, the upper-limit comparison status bit ADUPRST (D11/AD_CTL register) is set to 1. If the results are less than the lower-limit value, the lower-limit comparison status bit ADLWRST (D10/AD_CTL register) is set to 1. When the out-of range interrupt is enabled, an interrupt occurs if one of the status bits has been set.

∗ **ADUPRST**: Upper-Limit Comparison Status Bit in the A/D Control/Status (AD_CTL) Register (D11/0x5544)
∗ **ADLWRST**: Lower-Limit Comparison Status Bit in the A/D Control/Status (AD_CTL) Register (D10/0x5544)

When the conversion results are the same as the upper-limit or lower-limit values, it is assumed within the range and an interrupt is not generated.

When the A/D converter CH.0 has finished an A/D conversion, the range check is always performed regardless of how the ADCMP[2:0] is set. If the conversion results exceed the upper-limit value, the upper-limit out-of-range signal is output to the MFT. If the results are less than the lower-limit value, the lower-limit out-of-range signal is output to the MFT.

Terminating A/D conversion

- **For normal mode (MS = 1)**

In the normal mode, A/D conversion is performed successively from the conversion start channel specified using CS[2:0] (D[10:8]/AD_TRIG_CH register) to the conversion end channel specified using CE[2:0] (D[13:11]/AD_TRIG_CH register), and is completed after these conversions are executed in one operation. ADST (D1/AD_CTL register) is reset to 0 upon completion of the conversion.

* **MS**: A/D Conversion Mode Selection Bit in the A/D Trigger/Channel Select (AD_TRIG_CH) Register (D5/0x5542)

- **For continuous mode (MS = 0)**

In the continuous mode, A/D conversion from the conversion-start to the conversion-end channels is executed repeatedly, without being stopped in the hardware. To terminate conversion, therefore, ADST (D1/AD_CTL register) must be reset to 0 in the software. However, the A/D conversion being executed will be completed normally or forcibly stopped depending on the timing of writing 0 to ADST. When the A/D conversion has completed normally, ADF (D3/AD_CTL register) is set to 1 and the conversion results can be obtained. If it is forcibly stopped, ADF maintains its previous status, therefore, conversion results cannot be obtained.

- **Forced termination**

A/D conversion is immediately terminated by writing 0 to ADST. The results of the conversion then under-way cannot be obtained.

VII.1.6 A/D Converter Interrupt

The A/D converter can generate the following two types of interrupts:

- Conversion-complete interrupt
- Out-of-range interrupt

Conversion-complete interrupt

When A/D conversion in one channel has completed, the A/D converter outputs the conversion-complete interrupt signal to the interrupt controller (ITC) to request an interrupt.
When using this interrupt in advanced mode, set CNVINTEN (D4/AD_CTL register) to 1. If CNVINTEN is set to 0, an interrupt request by this cause will not be sent to the ITC. In standard mode, it is not necessary to set CNVINTEN.

∗ **CNVINTEN**: Conversion-Complete Interrupt Enable Bit in the A/D Control/Status (AD_CTL) Register (D4/0x5544)

If other interrupt conditions are satisfied, an interrupt is generated.

In advanced mode, the specified channels can be configured to disable interrupt generation using INTMASK*x* (D*x*/AD_INTMASK register). When INTMASK*x* is set to 0, the A/D converter CH.*x* does not generate conversion-complete interrupts.

∗ **INTMASK*x***: CH.*x* Conversion-Complete Interrupt Mask Bit in the A/D Conversion Complete Interrupt Mask (AD_INTMASK) Register (D*x*/0x555c)

Out-of-range interrupt (advanced mode)

When the range check function has been enabled and the conversion results are out of the range from the lower- to upper limits that have been set with software, the A/D converter outputs the out-of-range interrupt signal to the ITC to request an interrupt.
When using this interrupt, set the A/D converter to advanced mode and set CMPINTEN (D5/AD_CTL register) to 1. If CMPINTEN is set to 0, an interrupt request by this cause will not be sent to the ITC.

∗ **CMPINTEN**: Out-of-Range Interrupt Enable Bit in the A/D Control/Status (AD_CTL) Register (D5/0x5544)

By default, the A/D converter uses the conversion-complete interrupt signal line to send an out-of-range interrupt request to the ITC.
So, the conversion-complete interrupt flag in the ITC is set when an A/D conversion has completed or when the conversion results are out of range. To generate the out-of-range interrupt independently of the conversion complete interrupt, set INTMODE (D6/AD_CTL register) to 1.

∗ **INTMODE**: Interrupt Signal Mode Bit in the A/D Control/Status (AD_CTL) Register (D6/0x5544)

In standard mode, this interrupt cannot be used.

ITC registers for A/D converter interrupts

Table VII.1.6.1 shows the control registers of the ITC provided for each cause of A/D converter interrupt.

Table VII.1.6.1 ITC Registers

Cause of interrupt	Interrupt flag	Interrupt enable bit	Interrupt level setup bits
End of conversion	AIFT3 (D3/ITC_AIFLG)	AIEN3 (D3/ITC_AEN)	AILV3[2:0] (D[10:8]/ITC_AILV1)
Out of range	AIFT2 (D2/ITC_AIFLG)	AIEN2 (D2/ITC_AEN)	AILV2[2:0] (D[2:0]/ITC_AILV1)

ITC_AIFLG register (0x42e0)
ITC_AEN register (0x42e2)
ITC_AILV1 register (0x42e8)

When the A/D converter outputs an interrupt signal, the corresponding interrupt flag is set to 1.

If the interrupt enable bit corresponding to that interrupt flag has been set to 1, the ITC sends an interrupt request to the S1C17 Core. To disable the A/D converter interrupt, set the interrupt enable bit to 0.

The interrupt flag is always set to 1 by the interrupt signal, regardless of how the interrupt enable bit is set (even when set to 0).

The interrupt level setup bits set the interrupt level (0 to 7) of the A/D converter interrupt. If the same interrupt level is set, the out-of-range interrupt has higher priority than the conversion-complete interrupt.

An interrupt request to the S1C17 Core is accepted only when all the conditions described below are met.

- The interrupt enable bit is set to 1.
- The IE (Interrupt Enable) bit of the PSR (Processor Status Register) in the S1C17 Core is set to 1.
- The A/D converter interrupt has a higher interrupt level than the value that is set in the IL field of the PSR.
- No other cause of interrupt having higher priority, such as NMI, has occurred.

For details on these interrupt control registers, as well as the device operation when an interrupt has occurred, see Section III.1, "Interrupt Controller (ITC)."

Interrupt vectors

The following shows the vector numbers and vector addresses for the A/D converter interrupts:

Table VII.1.6.2 A/D Converter Interrupt Vectors

Cause of interrupt	Vector number	Vector address
Out of range	10 (0xa)	TTBR + 0x28
End of conversion	11 (0xb)	TTBR + 0x2c

VII.1.7 Details of Control Registers

Table VII.1.7.1 List of A/D Converter Registers

Address	Register name		Function
0x5520	AD_CLKCTL	A/D Clock Control Register	Controls A/D converter clock.
0x5540	AD_DAT	A/D Conversion Result Register	A/D converted data
0x5542	AD_TRIG_CH	A/D Trigger/Channel Select Register	Sets start/end channels and conversion mode.
0x5544	AD_CTL	A/D Control/Status Register	Controls A/D converter and indicates conversion status.
0x5546	AD_CH_STAT	A/D Channel Status Flag Register	Indicates overwrite error and conversion complete status.
0x5548	AD_CH0_BUF	A/D CH.0 Conversion Result Buffer Register	A/D CH.0 converted data
0x554a	AD_CH1_BUF	A/D CH.1 Conversion Result Buffer Register	A/D CH.1 converted data
0x554c	AD_CH2_BUF	A/D CH.2 Conversion Result Buffer Register	A/D CH.2 converted data
0x554e	AD_CH3_BUF	A/D CH.3 Conversion Result Buffer Register	A/D CH.3 converted data
0x5558	AD_UPPER	A/D Upper Limit Value Register	Specifies A/D conversion upper limit value.
0x555a	AD_LOWER	A/D Lower Limit Value Register	Specifies A/D conversion lower limit value.
0x555c	AD_INTMASK	A/D Conversion Complete Interrupt Mask Register	Masks A/D conversion complete interrupt.
0x555e	AD_ADVMODE	A/D Converter Mode Select/Internal Status Register	Selects A/D operating mode and indicates internal status and internal counter value.

The following describes each A/D converter register. These are all 16-bit registers.

Note: When setting the registers, be sure to write a 0, and not a 1, for all “reserved bits.”

0x5520: A/D Clock Control Register (AD_CLKCTL)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
A/D Clock Control Register (AD_CLKCTL)	**0x5520** (16 bits)	D15–4	–	reserved	–		–	–	0 when being read.
		D3	**PSONAD**	A/D converter clock control	1 On	0 Off	0	R/W	
		D2–0	**PSAD[2:0]**	A/D converter clock division ratio selection	PSAD[2:0] 0x7 0x6 0x5 0x4 0x3 0x2 0x1 0x0	A/D clock PCLK•1/256 PCLK•1/128 PCLK•1/64 PCLK•1/32 PCLK•1/16 PCLK•1/8 PCLK•1/4 PCLK•1/2	0x0	R/W	

D[15:4] Reserved

D3 PSONAD: A/D Converter Clock Control Bit

Controls the A/D conversion clock supply to the A/D converter.

1 (R/W): On

0 (R/W): Off (default)

D[2:0] PSAD[2:0]: A/D Converter Clock Division Ratio Selection Bits

Selects a division ratio to generate the A/D converter clock.

Table VII.1.7.2 Selecting Division Ratio

PSAD[2:0]	A/D clock
0x7	PCLK•1/256
0x6	PCLK•1/128
0x5	PCLK•1/64
0x4	PCLK•1/32
0x3	PCLK•1/16
0x2	PCLK•1/8
0x1	PCLK•1/4
0x0	PCLK•1/2

(Default: 0x0)

Notes:
- The recommended A/D clock frequency is a maximum of 2 MHz and a minimum of 16 kHz.
- Do not start an A/D conversion when the clock output from the prescaler is turned off, and do not turn off the prescaler's clock output when an A/D conversion is underway. This could cause the A/D converter to operate erratically.

0x5540: A/D Conversion Result Register (AD_DAT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
A/D Conversion Result Register (AD_DAT)	**0x5540** (16 bits)	D15–10	–	reserved	–	–	–	0 when being read.
		D9–0	**ADD[9:0]**	A/D converted data ADD9 = MSB ADD0 = LSB	0x0 to 0x3ff	0x0	R	

D[15:10] Reserved

D[9:0] ADD[9:0]: A/D Converted Data Bits

Stores the results of A/D conversion. (Default: 0x0)

The LSB is stored in ADD0, and the MSB is stored in ADD9.

This is a read-only register, so writing to this register is ignored.

0x5542: A/D Trigger/Channel Select Register (AD_TRIG_CH)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
A/D Trigger/ Channel Select Register (AD_TRIG_CH)	**0x5542** (16 bits)	D15–14	–	reserved	–		–	–	0 when being read.
		D13–11	**CE[2:0]**	A/D converter end channel selection	0 to 3		0x0	R/W	
		D10–8	**CS[2:0]**	A/D converter start channel selection	0 to 3		0x0	R/W	
		D7–6	–	reserved	–		–	–	0 when being read.
		D5	**MS**	A/D conversion mode selection	1 Continuous	0 Normal	0	R/W	
		D4–3	**TS[1:0]**	A/D conversion trigger selection	TS[1:0] 0x3 0x2 0x1 0x0	Trigger #ADTRG pin PT8 CH.0 MFT Software	0x0	R/W	
		D2–0	**CH[2:0]**	A/D conversion channel status	0 to 3		0x0	R	

D[15:14] Reserved

D[13:11] CE[2:0]: A/D Converter End Channel Selection Bits

Sets the conversion end channel by selecting a channel number from 0 to 3. (Default: 0x0 = AIN0)

Analog inputs can be A/D-converted successively from the channel set using CS[2:0] (D[10:8]) to the channel set using these bits in one operation. If only one channel is to be A/D converted, set the same channel number in both CS[2:0] and CE[2:0].

D[10:8] CS[2:0]: A/D Converter Start Channel Selection Bits

Sets the conversion start channel by selecting a channel number from 0 to 3. (Default: 0x0 = AIN0)

Analog inputs can be A/D-converted successively from the channel set using these bits to the channel set using CE[2:0] (D[13:11]) in one operation. If only one channel is to be A/D converted, set the same channel number in both CS[2:0] and CE[2:0].

D[7:6] Reserved

D5 MS: A/D Conversion Mode Selection Bit

Selects an A/D conversion mode.

1 (R/W): Continuous mode

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

The A/D converter is set for the continuous mode by writing 1 to MS. In this mode, A/D conversions in the range of the channels selected using CS[2:0] (D[10:8]) and CE[2:0] (D[13:11]) are executed continuously until stopped in the software.

When MS = 0, the A/D converter operates in the normal mode. In this mode, A/D conversion is completed after all inputs in the range of the channels selected by CS[2:0] and CE[2:0] are converted in one operation.

D[4:3] TS[1:0]: A/D Conversion Trigger Selection Bits

Selects a trigger to start A/D conversion.

Table VII.1.7.3 Trigger Selection

TS[1:0]	Trigger source
0x3	External trigger (#ADTRG pin)
0x2	8-bit programmable timer (PT8) CH.0
0x1	16-bit multi-function timer (MFT)
0x0	Software trigger

(Default: 0x0)

When an external trigger is used, the #ADTRG pin must be set in advance using the Port Function Select Register. A/D conversion is started when a low level of the #ADTRG signal is detected.

When the 8-bit programmable timer CH.0 is used, since its underflow signal serves as a trigger, set the cycle and other parameters for the timer.

When the 16-bit multi-function timer is used, since its period-match signal serves as a trigger, set the cycle and other parameters for the timer.

D[2:0] CH[2:0]: A/D Conversion Channel Status Bits

Indicates the channel number (0 to 3) currently being A/D-converted. (Default: 0x0 = AIN0)

When A/D conversion is performed in multiple channels, read this bit to identify the channel in which conversion is underway.

Note that CH[2:0] may indicate an unavailable (reserved) channel depending on the CS[2:0] and CE[2:0] settings as the A/D converter control circuits supports eight channels.

0x5544: A/D Control/Status Register (AD_CTL)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
A/D Control/ Status Register (AD_CTL)	**0x5544** (16 bits)	D15	**ADCMPE**	Upper/lower-limit comparison enable	1	Enable	0	Disable	0	R/W	Can be used when ADCADV = 1.
		D14–12	**ADCMP[2:0]**	Upper/lower-limit comparison channel selection	0 to 3				0x0	R/W	
		D11	**ADUPRST**	Upper-limit comparison status	1	Out of range	0	Within range	0	R	
		D10	**ADLWRST**	Lower-limit comparison status	1	Out of range	0	Within range	0	R	
		D9–8	**ST[1:0]**	Input signal sampling time setup	ST[1:0]		Sampling time		0x3	R/W	Use with 9 clocks.
					0x3 0x2 0x1 0x0		9 clocks 7 clocks 5 clocks 3 clocks				
		D7	–	reserved	–				–	–	0 when being read.
		D6	**INTMODE**	Interrupt signal mode	1	Complete only	0	OR	0	R/W	Can be used when ADCADV = 1.
		D5	**CMPINTEN**	Out-of-range int. enable	1	Enable	0	Disable	0	R/W	
		D4	**CNVINTEN**	Conversion-complete int. enable	1	Enable	0	Disable	1	R/W	Can be changed when ADCADV = 1.
		D3	**ADF**	Conversion-complete flag	1	Completed	0	Run/ Standby	0	R	Reset when ADD is read.
		D2	**ADE**	A/D enable	1	Enable	0	Disable	0	R/W	
		D1	**ADST**	A/D conversion control/status	1	Start/Run	0	Stop	0	R/W	
		D0	**OWE**	Overwrite error flag	1	Error	0	Normal	0	R/W	Reset by writing 0.

D15 ADCMPE: Upper/Lower-Limit Comparison Enable Bit (for advanced mode)

Enables/disables comparison between converted data and upper-/lower-limit values.

1 (R/W): Enabled

0 (R/W): Disabled (default)

ADCMPE selects whether the converted data is compared with the upper-/lower-limit values after A/D conversion of the channel specified using ADCMP[2:0] (D[14:12]). Set ADCMPE to 1 when using the comparison function or set to 0 when not used.

D[14:12] ADCMP[2:0]: Upper/Lower-Limit Comparison Channel Selection Bits (for advanced mode)

Set the channel number (0–3) to compare its converted data with the upper-/ lower-limit values. (Default: 0x0 = AIN0)

These bits do not affect the CH.0 out-of-range signal output for the MFT.

D11 ADUPRST: Upper-Limit Comparison Status Bit (for advanced mode)

Indicates the results of comparison between the A/D converted data and the upper-limit value.

1 (R): Exceeded the upper limit

0 (R): Within the range (default)

When the upper-/lower-limit comparison function is enabled (ADCMPE (D15) = 1), the converted data is compared with the upper-/lower-limit values after A/D conversion of the channel specified using ADCMP[2:0] (D[14:12]) has completed. If the converted data exceeds the upper-limit value set in ADUPR[9:0] (D[9:0]/AD_UPPER register), ADUPRST is set to 1. If the converted data is equal to or less than the upper-limit value, ADUPRST is set to 0. An interrupt occurs when ADUPRST is set to 1 if the out-of-range interrupt is enabled.

D10 ADLWRST: Lower-Limit Comparison Status Bit (for advanced mode)

Indicates the results of comparison between the A/D converted data and the lower-limit value.

1 (R): Under the lower limit

0 (R): Within the range (default)

When the upper-/lower-limit comparison function is enabled (ADCMPE (D15) = 1), the converted data is compared with the upper-/lower-limit values after A/D conversion of the channel specified using ADCMP[2:0] (D[14:12]) has completed. If the converted data is less than the lower-limit value set in ADLWR[9:0] (D[9:0]/AD_LOWER register), ADLWRST is set to 1. If the converted data is equal to or more than the lower-limit value, ADLWRST is set to 0. An interrupt occurs when ADLWRST is set to 1 if the out-of-range interrupt is enabled.

D[9:8] ST[1:0]: Input Signal Sampling Time Setup Bits

Sets the analog input sampling time.

Table VII.1.7.4 Sampling Time

ST[1:0]	Sampling time
0x3	9-clock period
0x2	7-clock period
0x1	5-clock period
0x0	3-clock period

(Default: 0x3)

The A/D converter conversion clock is used for counting.
To maintain the conversion accuracy, use ST as set by default (9-clock period).
The conversion time is fixed at 11-clock period.

D7 Reserved

D6 INTMODE: Interrupt Signal Mode Bit (for advanced mode)

Configures the conversion-complete interrupt signal delivered to the ITC.
1 (R/W): Conversion-complete signal only
0 (R/W): OR between conversion-complete and out-of-range signals (default)

INTMODE selects whether the conversion-complete interrupt signal line connected to the ITC is used to send the conversion-complete signal only or used to send the signal of which the conversion-complete and out-of-range signal are ORed.
Set INTMODE to 1 when handling the out-of-range interrupt as another interrupt. When using the out-of-range interrupt, set CMPINTEN (D5) to 1.

D5 CMPINTEN: Out-of-Range Interrupt Enable Bit (for advanced mode)

Enables/disables the out-of-range interrupt.
1 (R/W): Enabled
0 (R/W): Disabled (default)

When CMPINTEN is set to 1, upper and lower-limit comparison results become a cause of interrupt. When it is set to 0, an out-of-range interrupt is not generated.

D4 CNVINTEN: Conversion-Complete Interrupt Enable Bit

Enables/disables the conversion-complete interrupt.
1 (R/W): Enabled (default)
0 (R/W): Disabled

When CNVINTEN is set to 1, completion of an A/D conversion becomes a cause of interrupt. When it is set to 0, a conversion-complete interrupt is not generated.

Note: CNVINTEN is effective in standard mode. Set CNVINTEN to 1 to enable the A/D converter interrupt in standard and advanced modes. However, CNVINTEN can only be changed in advanced mode (ADCADV = 1).

D3 ADF: Conversion-Complete Flag Bit

Indicates that A/D conversion has been completed.
1 (R): Conversion completed
0 (R): Being converted or standing by (default)

This flag is set to 1 when A/D conversion is completed, and the converted data is stored in the data register and is reset to 0 when the converted data is read out. When A/D conversion is performed in multiple channels, if the next A/D conversion is completed while ADF = 1 (before the converted data is read out), the data register is overwritten with the new conversion results, causing an overrun error to occur. Therefore, ADF must be reset by reading out the converted data before the next A/D conversion is completed.

D2 ADE: A/D Enable Bit

Enables the A/D converter (ready for conversion).

1 (R/W): Enabled

0 (R/W): Disabled (default)

When ADE is set to 1, the A/D converter is enabled, meaning it is ready to start A/D conversion (i.e., ready to accept a trigger). When ADE = 0, the A/D converter is disabled, meaning it is unable to accept a trigger.

Before setting the conversion mode, start/end channels, etc. for the A/D converter, be sure to reset ADE to 0. This helps to prevent the A/D converter from operating erratically.

D1 ADST: A/D Conversion Control/Status Bit

Controls A/D conversion.

1 (R/W): Software trigger

0 (R/W): A/D conversion is stopped (default)

If A/D conversion is to be started by a software trigger, set ADST to 1. If any other trigger is used, ADST is automatically set to 1 by the hardware.

ADST remains set while A/D conversion is underway.

In normal mode, upon completion of A/D conversion in selected channels, ADST is reset to 0 and the A/D conversion circuit is turned off. To stop A/D conversion during operation in continuous mode, reset ADST by writing 0.

When ADE (D2) = 0 (A/D conversion disabled), ADST is fixed to 0, with no trigger accepted.

D0 OWE: Overwrite Error Flag Bit

Indicates that the converted data has been overwritten.

1 (R):	Overwritten
0 (R):	Normal (default)
1 (W):	Has no effect
0 (W):	Flag is reset

During A/D conversion in multiple channels, if the conversion results for the next channel are written to the converted-data register (overwritten) before the converted data is read out to reset the conversion-complete flag ADF (D3) that has been set through conversion of the preceding channel, OWE is set to 1. When ADF (D3) is reset, because this means that the converted data has been read out, OWE is not set. Once OWE is set to 1, it remains set until it is reset by writing 0 in the software.

0x5546: A/D Channel Status Flag Register (AD_CH_STAT)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
A/D Channel Status Flag Register (AD_CH_STAT)	**0x5546** (16 bits)	D15–12	–	reserved	–				–	–	0 when being read.
		D11	**OWE3**	CH.3 overwrite error flag	1	Error	0	Normal	0	R/W	Can be used when ADCADV = 1. Reset by writing 0.
		D10	**OWE2**	CH.2 overwrite error flag					0	R/W	
		D9	**OWE1**	CH.1 overwrite error flag					0	R/W	
		D8	**OWE0**	CH.0 overwrite error flag					0	R/W	
		D7–4	–	reserved	–				–	–	0 when being read.
		D3	**ADF3**	CH.3 conversion-complete flag	1	Completed	0	Run/ Standby	0	R	Can be used when ADCADV = 1. Reset when AD*x*BUF [9:0] is read.
		D2	**ADF2**	CH.2 conversion-complete flag					0	R	
		D1	**ADF1**	CH.1 conversion-complete flag					0	R	
		D0	**ADF0**	CH.0 conversion-complete flag					0	R	

Note: The letter '*x*' in bit names, etc., denotes a channel number from 0 to 3.

D[15:12] Reserved

D[11:8] OWE[3:0]: CH.*x* Overwrite Error Flag Bits (for advanced mode)

These bits indicate that the conversion result buffer for each channel has been overwritten.

1 (R): Overwritten
0 (R): Normal (default)
1 (W): Has no effect
0 (W): Flag is reset

During A/D conversion in continuous mode, if the new conversion results in the same channel are written to the conversion result buffer (overwritten) before the converted data is read out to reset the ADF*x* conversion-complete flag that has been set through the previous conversion, OWE*x* is set to 1. When ADF*x* is reset, because this means that the converted data has been read out, OWE*x* is not set.
Once OWE*x* is set to 1, it remains set until it is reset by writing 0 in the software.

D[7:4] Reserved

D[3:0] ADF[3:0]: CH.*x* Conversion-Complete Flag Bits (for advanced mode)

These bits indicate that A/D conversion in each channel has been completed.

1 (R): Conversion completed
0 (R): Being converted or standing by (default)

This flag is set to 1 when A/D conversion of the corresponding channel is completed, and the converted data is stored in the conversion result buffer and is reset to 0 when the conversion result buffer is read out. When A/D conversion is performed in continuous mode, if the next A/D conversion of the same channel is completed while ADF*x* = 1 (before the conversion result buffer is read out), the buffer is overwritten with the new conversion results, causing an overrun error to occur. Therefore, ADF*x* must be reset by reading out the converted data before the next A/D conversion is completed.

0x5548–0x554e: A/D CH.*x* Conversion Result Buffer Registers (AD_CH*x*_BUF)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
A/D CH.*x* Conversion Result Buffer Register (AD_CH*x*_BUF)	**0x5548** **|** **0x554e** (16 bits)	D15–10	**–**	reserved	–	–	–	0 when being read.
		D9–0	**AD*x*BUF [9:0]**	A/D CH.*x* converted data AD*x*BUF9 = MSB AD*x*BUF0 = LSB	0x0 to 0x3ff	0x0	R	Can be used when ADCADV = 1.

Note: The letter '*x*' in bit names, etc., denotes a channel number from 0 to 3.

0x5548	A/D CH.0 Conversion Result Buffer Register (AD_CH0_BUF)
0x554a	A/D CH.1 Conversion Result Buffer Register (AD_CH1_BUF)
0x554c	A/D CH.2 Conversion Result Buffer Register (AD_CH2_BUF)
0x554e	A/D CH.3 Conversion Result Buffer Register (AD_CH3_BUF)

D[15:10] Reserved

D[9:0] AD*x*BUF[9:0]: A/D CH.*x* Converted Data Bits (for advanced mode)

The conversion results in each channel are stored. (Default: 0x0)

This is a read-only register, so writing to this register is ignored.

0x5558: A/D Upper Limit Value Register (AD_UPPER)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
A/D Upper Limit Value Register (AD_UPPER)	**0x5558** (16 bits)	D15–10	–	reserved	–	–	–	0 when being read.
		D9–0	**ADUPR[9:0]**	A/D conversion upper limit value ADUPR9 = MSB ADUPR0 = LSB	0x0 to 0x3ff	0x0	R/W	Can be used when ADCADV = 1.

D[15:10] Reserved

D[9:0] ADUPR[9:0]: A/D Conversion Upper Limit Value Bits (for advanced mode)

Set the upper-limit value to be compared with the A/D conversion results. (Default: 0x0)

The value set in this register is used for the range check of the A/D conversion results in the channel specified with ADCMP[2:0] (D[14:12]/AD_CTL register). If the converted data exceeds the set value, an interrupt can be generated.

0x555a: A/D Lower Limit Value Register (AD_LOWER)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
A/D Lower Limit Value Register (AD_LOWER)	**0x555a** (16 bits)	D15–10	–	reserved	–	–	–	0 when being read.
		D9–0	**ADLWR [9:0]**	A/D conversion lower limit value ADLWR9 = MSB ADLWR0 = LSB	0x0 to 0x3ff	0x0	R/W	Can be used when ADCADV = 1.

D[15:10] Reserved

D[9:0] ADLWR[9:0]: A/D Conversion Lower Limit Value Bits (for advanced mode)

Set the lower-limit value to be compared with the A/D conversion results. (Default: 0x0)

The value set in this register is used for the range check of the A/D conversion results in the channel specified with ADCMP[2:0] (D[14:12]/AD_CTL register). If the converted data is less than the set value, an interrupt can be generated.

0x555c: A/D Conversion Complete Interrupt Mask Register (AD_INTMASK)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
A/D Conversion Complete Interrupt Mask Register (AD_INTMASK)	**0x555c** (16 bits)	D15–4	–	reserved	–				–	–	0 when being read.
		D3	**INTMASK3**	CH.3 conversion-complete int. mask	1	Interrupt enabled	0	Interrupt mask	1	R/W	Can be used when ADCADV = 1.
		D2	**INTMASK2**	CH.2 conversion-complete int. mask					1	R/W	
		D1	**INTMASK1**	CH.1 conversion-complete int. mask					1	R/W	
		D0	**INTMASK0**	CH.0 conversion-complete int. mask					1	R/W	

Note: The letter '*x*' in bit names, etc., denotes a channel number from 0 to 3.

D[15:4] Reserved

D[3:0] INTMASK[3:0]: CH.*x* Conversion-Complete Interrupt Mask Bits (for advanced mode)

These bits mask the A/D conversion-complete interrupt for each channel individually.

1 (R/W): Interrupt is enabled (default)

0 (R/W): Interrupt is masked

When INTMASK*x* is set to 0, the conversion-completed interrupt request of the CH.*x* is masked and the interrupt flag in the ITC will not be set to 1 even if A/D conversion is completed. When INTMASK*x* is 1, the A/D converter can generate an interrupt upon completion of A/D conversion in CH.*x*.

Note: INTMASK[3:0] must be set to 1 to generate A/D converter interrupts in standard and advanced modes.

0x555e: A/D Converter Mode Select/Internal Status Register (AD_ADVMODE)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
A/D Converter Mode Select/ Internal Status Register (AD_ADVMODE)	**0x555e** (16 bits)	D15–9	–	reserved	–				–	–	0 when being read.
		D8	**ADCADV**	Standard/advanced mode selection	1	Advanced	0	Standard	0	R/W	
		D7–6	–	reserved	–				–	–	0 when being read.
		D5–4	**ISTATE[1:0]**	Internal status	ISTATE[1:0] 0x3 0x2 0x1 0x0		Status Converting reserved Sampling Idle		0x0	R	
		D3–0	**ICOUNTER [3:0]**	Internal counter value	0 to 15				0x0	R	

D[15:9] Reserved

D8 ADCADV: Standard/Advanced Mode Selection Bit

Selects the A/D converter operating mode.

1 (R/W): Advanced mode

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

When ADCADV is set to 1, the A/D converter is set in the advanced mode, and the registers/bits for the extended function can be used.

When ADCADV is set to 0, only the standard A/D converter functions can be used. In this mode, the extended registers/bits for advanced mode become read only and writing operation is disabled.

D[7:6] Reserved

D[5:4] ISTATE[1:0]: Internal Status Bits

Indicates the A/D converter internal status.

Table VII.1.7.5 Internal Status

ISTATE[1:0]	Status
0x3	Converting
0x2	Reserved
0x1	Sampling
0x0	Idle

(Default: 0x0)

D[3:0] ICOUNTER[3:0]: Internal Counter Value Bits

Indicates the internal counter value. (Default: 0x0)

VII.1.8 Precautions

- Before setting the conversion mode, start/end channels, etc. for the A/D converter, be sure to disable ADE (D2/AD_CTL register). A change in settings while the A/D converter is enabled could cause it to operate erratically.

  ∗ **ADE**: A/D Enable Bit in the A/D Control/Status (AD_CTL) Register (D2/0x5544)

- In consideration of the conversion accuracy, we recommend that the A/D conversion clock be min. 16 kHz to max. 2 MHz.
- Do not start an A/D conversion when the clock supplied from the prescaler to the A/D converter is turned off, and do not turn off the prescaler's clock output when an A/D conversion is underway, as doing so could cause the A/D converter to operate erratically.
- When the A/D converter is set to enabled state, a current flows between AV_{DD} and V_{SS}, and power is consumed, even when A/D operations are not performed. Therefore, when the A/D converter is not used, it must be set to the disabled state (default 0 setting of ADE (D2/AD_CTL register)).
- When the MFT period-match signal is used as a trigger factor, the division ratio of the prescaler in the MFT module must not be set to PCLK/1.
- When using an external trigger to start A/D conversion, the low period of the trigger signal to be applied to the #ADTRG pin must be two or more CPU operating clock cycles. Furthermore, return the #ADTRG input level to high within 20 cycles of the A/D input clock set. Otherwise, it will be detected as the trigger for the next A/D conversion.
- When the A/D converter is set to enabled state, a current flows between AV_{DD} and V_{SS}, and power is consumed, even when A/D operations are not performed. Therefore, when the A/D converter is not used, it must be set to the disabled state (default 0 setting of ADE (D2/AD_CTL register)).
  The A/D converter must always be enabled if a timer output or an external input is used as the trigger but it increases current consumption. To reduce current consumption, an A/D conversion control procedure as shown below is recommended so that the A/D converter will be enabled as short as possible using the software trigger.

  (1) Place the S1C17002 into HALT mode with the OSC1 clock set as the CPU clock if A/D conversion control is not necessary.
  (2) Generate an interrupt according to the sampling frequency to cancel HALT mode.
  (3) Run the CPU with the OSC3 clock.
  (4) Enable the A/D converter.
  (5) Start an A/D conversion using the software trigger.
  (6) Read the conversion results.
  (7) Disable the A/D converter.
  (8) Return to Step (1) after the necessary processing has been finished.

THIS PAGE IS BLANK.

S1C17002 Technical Manual

APPENDIX

Appendix A List of I/O Registers

Peripheral	Address	Register name		Function
Prescaler (8-bit device)	0x4020	PSC_CTL	Prescaler Control Register	Starts/stops the prescaler.
	0x4021–0x403f	–	–	Reserved
UART (with IrDA) (8-bit device)	0x4100	UART_ST	UART Status Register	Indicates transfer, buffer and error statuses.
	0x4101	UART_TXD	UART Transmit Data Register	Transmit data
	0x4102	UART_RXD	UART Receive Data Register	Receive data
	0x4103	UART_MOD	UART Mode Register	Sets transfer data format.
	0x4104	UART_CTL	UART Control Register	Controls data transfer.
	0x4105	UART_EXP	UART Expansion Register	Sets IrDA mode.
	0x4106–0x411f	–	–	Reserved
CLG_T16U0 timer (16-bit device)	0x4200	CLG_T16U0_CLK	CLG_T16U0 Input Clock Select Register	Selects a prescaler output clock.
	0x4202	CLG_T16U0_TR	CLG_T16U0 Reload Data Register	Sets reload data.
	0x4204	CLG_T16U0_TC	CLG_T16U0 Counter Data Register	Counter data
	0x4206	CLG_T16U0_CTL	CLG_T16U0 Control Register	Sets the timer mode and starts/stops the timer.
	0x4208–0x421f	–	–	Reserved
CLG_T8FU0 timer (16-bit device)	0x4220	CLG_T8FU0_CLK	CLG_T8S Input Clock Select Register	Selects a prescaler output clock.
	0x4222	CLG_T8FU0_TR	CLG_T8S Reload Data Register	Sets reload data.
	0x4224	CLG_T8FU0_TC	CLG_T8S Counter Data Register	Counter data
	0x4226	CLG_T8FU0_CTL	CLG_T8S Control Register	Sets the timer mode and starts/stops the timer.
	0x4228–0x423f	–	–	Reserved
CLG_T8S timer (16-bit device)	0x4240	CLG_T8S_CLK	CLG_T8S Input Clock Select Register	Selects a prescaler output clock.
	0x4242	CLG_T8S_TR	CLG_T8S Reload Data Register	Sets reload data.
	0x4244	CLG_T8S_TC	CLG_T8S Counter Data Register	Counter data
	0x4246	CLG_T8S_CTL	CLG_T8S Control Register	Sets the timer mode and starts/stops the timer.
	0x4248–0x425f	–	–	Reserved
CLG_T8I timer (16-bit device)	0x4260	CLG_T8I_CLK	CLG_T8I Input Clock Select Register	Selects a prescaler output clock.
	0x4262	CLG_T8I_TR	CLG_T8I Reload Data Register	Sets reload data.
	0x4264	CLG_T8I_TC	CLG_T8I Counter Data Register	Counter data
	0x4266	CLG_T8I_CTL	CLG_T8I Control Register	Sets the timer mode and starts/stops the timer.
	0x4268–0x427f	–	–	Reserved
Interrupt controller (16-bit device)	0x42e0	ITC_AIFLG	Additional Interrupt Flag Register	Indicates/resets interrupt occurrence status.
	0x42e2	ITC_AEN	Additional Interrupt Enable Register	Enables/disables each maskable interrupt.
	0x42e4	–	–	Reserved
	0x42e6	ITC_AILV0	Additional Interrupt Level Setup Register 0	Sets the MFT interrupt level.
	0x42e8	ITC_AILV1	Additional Interrupt Level Setup Register 1	Sets the ADC interrupt level.
	0x42ea	ITC_AILV2	Additional Interrupt Level Setup Register 2	Sets the RTC and PT8 CH.0/T8OSC1 CH.0 interrupt levels.
	0x42ec	ITC_AILV3	Additional Interrupt Level Setup Register 3	Sets the PT8 CH.1/T8OSC1 CH.1 and PT8 CH.2 interrupt levels.
	0x42ee	ITC_AILV4	Additional Interrupt Level Setup Register 4	Sets the PT8 CH.3 interrupt level.
	0x42f0	ITC_AILV5	Additional Interrupt Level Setup Register 5	Sets the SPI CH.1 interrupt level.
	0x42f2	ITC_AILV6	Additional Interrupt Level Setup Register 6	Sets the I^{2}C slave interrupt levels.
	0x42f4	ITC_AILV7	Additional Interrupt Level Setup Register 7	Sets the REMC interrupt level.
	0x42f6–0x42ff	–	–	Reserved
	0x4300	ITC_IFLG	Interrupt Flag Register	Indicates/resets interrupt occurrence status.
	0x4302	ITC_EN	Interrupt Enable Register	Enables/disables each maskable interrupt.
	0x4304	ITC_CTL	ITC Control Register	Enables/disables the ITC.
	0x4306	ITC_ELV0	External Interrupt Level Setup Register 0	Sets the port 0 and port 1 interrupt levels and trigger modes.
	0x4308	ITC_ELV1	External Interrupt Level Setup Register 1	Sets the port 2 and port 3 interrupt levels and trigger modes.
	0x430a	ITC_ELV2	External Interrupt Level Setup Register 2	Sets the port 4 and port 5 interrupt levels and trigger modes.
	0x430c	ITC_ELV3	External Interrupt Level Setup Register 3	Sets the port 6 and port 7 interrupt levels and trigger modes.
	0x430e	ITC_ILV0	Internal Interrupt Level Setup Register 0	Sets the CLG_T16U0 and CLG_T8FU0 timer interrupt levels.
	0x4310	ITC_ILV1	Internal Interrupt Level Setup Register 1	Sets the CLG_T8S and CLG_T8I timer interrupt levels.
	0x4312	ITC_ILV2	Internal Interrupt Level Setup Register 2	Sets the UART interrupt level.
	0x4314	ITC_ILV3	Internal Interrupt Level Setup Register 3	Sets the SPI CH.0 and I^{2}C master interrupt levels.
	0x4316–0x431f	–	–	Reserved

Peripheral	Address	Register name		Function
SPI (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 CH.0 mode and enables data transfer.
	0x4328–0x433f	–	–	Reserved
I^{2}C master (16-bit device)	0x4340	I2CM_EN	I^{2}C Master Enable Register	Enables the I^{2}C master module.
	0x4342	I2CM_CTL	I^{2}C Master Control Register	Controls the I^{2}C master operation and indicates transfer status.
	0x4344	I2CM_DAT	I^{2}C Master Data Register	I^{2}C master transmit/receive data
	0x4346	I2CM_ICTL	I^{2}C Master Interrupt Control Register	Controls the I^{2}C master interrupt.
	0x4348–0x435f	–	–	Reserved
I^{2}C slave (16-bit device)	0x4360	I2CS_TRNS	I^{2}C Slave Transmit Data Write Register	I^{2}C slave transmit data
	0x4362	I2CS_RECV	I^{2}C Slave Receive Data Read Register	I^{2}C slave receive data
	0x4364	I2CS_SADRS	I^{2}C Slave Address Setup Register	Sets the I^{2}C slave address.
	0x4366	I2CS_CTL	I^{2}C Slave Control Register	Controls the I^{2}C slave module.
	0x4368	I2CS_STAT	I^{2}C Slave Status Register	Indicates the I^{2}C slave bus status.
	0x436a	I2CS_ASTAT	I^{2}C Slave Access Status Register	Indicates the I^{2}C slave access status
	0x436c	I2CS_ICTL	I^{2}C Slave Interrupt Control Register	Controls the I^{2}C slave interrupt.
	0x436e–0x437f	–	–	Reserved
GPIO (8-bit device)	0x4400	P0_DAT	P0 Port Input Data Register	P0 port input data
	0x4401	–	–	Reserved
	0x4402	P1_DAT	P1 Port Input/Output Data Register	P1 port input/output data
	0x4403	P1_IOC	P1 Port I/O Control Register	Selects the P1 port I/O direction.
	0x4404	P2_DAT	P2 Port Input/Output Data Register	P2 port input/output data
	0x4405	P2_IOC	P2 Port I/O Control Register	Selects the P2 port I/O direction.
	0x4406	P3_DAT	P3 Port Input/Output Data Register	P3 port input/output data
	0x4407	P3_IOC	P3 Port I/O Control Register	Selects the P3 port I/O direction.
	0x4408	P4_DAT	P4 Port Input/Output Data Register	P4 port input/output data
	0x4409	P4_IOC	P4 Port I/O Control Register	Selects the P4 port I/O direction.
	0x440a	P5_DAT	P5 Port Input/Output Data Register	P5 port input/output data
	0x440b	P5_IOC	P5 Port I/O Control Register	Selects the P5 port I/O direction.
	0x440c–0x441f	–	–	Reserved
	0x4420	P0_03_CFP	P00–P03 Port Function Select Register	Selects the P00–P03 port functions.
	0x4421	–	–	Reserved
	0x4422	P1_03_CFP	P10–P13 Port Function Select Register	Selects the P10–P13 port functions.
	0x4423	P1_46_CFP	P14–P16 Port Function Select Register	Selects the P14–P16 port functions.
	0x4424	–	–	Reserved
	0x4425	P2_57_CFP	P25–P27 Port Function Select Register	Selects the P25–P27 port functions.
	0x4426	P3_02_CFP	P30–P32 Port Function Select Register	Selects the P30–P32 port functions.
	0x4427	P3_57_CFP	P35–P37 Port Function Select Register	Selects the P35–P37 port functions.
	0x4428	P4_03_CFP	P40–P43 Port Function Select Register	Selects the P40–P43 port functions.
	0x4429	P4_4_CFP	P44 Port Function Select Register	Selects the P44 port functions.
	0x442a	P5_03_CFP	P50–P53 Port Function Select Register	Selects the P50–P53 port functions.
	0x442b–0x443f	–	–	Reserved
	0x4440	PINTSEL0	Port Input Interrupt 0 Select Register	Selects a P*x*0 port for input interrupt.
	0x4441	PINTSEL1	Port Input Interrupt 1 Select Register	Selects a P*x*1 port for input interrupt.
	0x4442	PINTSEL2	Port Input Interrupt 2 Select Register	Selects a P*x*2 port for input interrupt.
	0x4443	PINTSEL3	Port Input Interrupt 3 Select Register	Selects a P*x*3 port for input interrupt.
	0x4444	PINTSEL4	Port Input Interrupt 4 Select Register	Selects a P*x*4 port for input interrupt.
	0x4445	PINTSEL5	Port Input Interrupt 5 Select Register	Selects a P*x*5 port for input interrupt.
	0x4446	PINTSEL6	Port Input Interrupt 6 Select Register	Selects a P*x*6 port for input interrupt.
	0x4447	PINTSEL7	Port Input Interrupt 7 Select Register	Selects a P*x*7 port for input interrupt.
	0x4448–0x44ff	–	–	Reserved
Real-time clock (8-bit device)	0x4600	RTC_INTSTAT	RTC Interrupt Status Register	Indicates RTC interrupt status.
	0x4601	RTC_INTMODE	RTC Interrupt Mode Register	Sets up RTC interrupt modes.
	0x4602	RTC_CNTL0	RTC Control 0 Register	Controls the RTC.
	0x4603	RTC_CNTL1	RTC Control 1 Register	
	0x4604–0x4613	–	–	Reserved
	0x4614	RTC_SEC	RTC Second Register	Second counter data
	0x4615	RTC_MIN	RTC Minute Register	Minute counter data
	0x4616	RTC_HOUR	RTC Hour Register	Hour counter data
	0x4617	RTC_DAY	RTC Day Register	Day counter data
	0x4618–0x4627	–	–	Reserved
	0x4628	RTC_MONTH	RTC Month Register	Month counter data
	0x4629	RTC_YEAR	RTC Year Register	Year counter data
	0x462a	RTC_WEEK	RTC Days of Week Register	Days of week counter data
	0x462b–0x46ff	–	–	Reserved

Peripheral	Address	Register name		Function
8-bit programmable timer CH.0 (8-bit device)	0x4800	PT8_CLK0	PT8 CH.0 Input Clock Select Register	Selects the count clock.
	0x4801	PT8_RLD0	PT8 CH.0 Reload Data Register	Sets reload data.
	0x4802	PT8_PTD0	PT8 CH.0 Counter Data Register	Counter data
	0x4803	PT8_CTL0	PT8 CH.0 Control Register	Sets the timer mode and starts/stops the timer.
8-bit programmable timer CH.1 (8-bit device)	0x4804	PT8_CLK1	PT8 CH.1 Input Clock Select Register	Selects the count clock.
	0x4805	PT8_RLD1	PT8 CH.1 Reload Data Register	Sets reload data.
	0x4806	PT8_PTD1	PT8 CH.1 Counter Data Register	Counter data
	0x4807	PT8_CTL1	PT8 CH.1 Control Register	Sets the timer mode and starts/stops the timer.
8-bit programmable timer CH.2 (8-bit device)	0x4808	PT8_CLK2	PT8 CH.2 Input Clock Select Register	Selects the count clock.
	0x4809	PT8_RLD2	PT8 CH.2 Reload Data Register	Sets reload data.
	0x480a	PT8_PTD2	PT8 CH.2 Counter Data Register	Counter data
	0x480b	PT8_CTL2	PT8 CH.2 Control Register	Sets the timer mode and starts/stops the timer.
8-bit programmable timer CH.3 (8-bit device)	0x480c	PT8_CLK3	PT8 CH.3 Input Clock Select Register	Selects the count clock.
	0x480d	PT8_RLD3	PT8 CH.3 Reload Data Register	Sets reload data.
	0x480e	PT8_PTD3	PT8 CH.3 Counter Data Register	Counter data
	0x480f	PT8_CTL3	PT8 CH.3 Control Register	Sets the timer mode and starts/stops the timer.
	0x4810–0x48ff	–	–	Reserved
Clock management unit (8-bit device)	0x4900	CMU_SYSCLKCTL	System Clock Control Register	Controls the system clock.
	0x4901	CMU_OSC3_WCNT	OSC3 Wait Timer Register	Sets the OSC3 wait timer for system wake-up.
	0x4902	CMU_NF	Noise Filter Control Register	Enables noise filters.
	0x4903	CMU_OSC3DIV	OSC3 Clock Divider Register	Selects an OSC3 system clock frequency.
	0x4904	–	–	Reserved
	0x4905	CMU_CMUCLK	CMU_CLK Select Register	Selects the output CMU_CLK frequency.
	0x4906	CMU_GATEDCLK0	Gated Clock Control 0 Register	Controls clock supply to peripheral modules.
	0x4907	CMU_GATEDCLK1	Gated Clock Control 1 Register	
	0x4908	CMU_GATEDCLK2	Gated Clock Control 2 Register	
	0x4909–0x491f	–	–	Reserved
	0x4920	CMU_PROTECT	CMU Write Protect Register	Enables writing to the CMU registers (0x4900–0x4908).
	0x4921–0x49ff	–	–	Reserved
8-bit OSC1 timer CH.0 (8-bit device)	0x4a00	T8OSC1_CTL0	T8OSC1 CH.0 Control Register	Sets the timer mode and starts/stops the timer.
	0x4a01	T8OSC1_CNT0	T8OSC1 CH.0 Timer Counter Data Register	Counter data
	0x4a02	T8OSC1_CMP0	T8OSC1 CH.0 Timer Compare Data Register	Sets compare data.
	0x4a03	T8OSC1_IMSK0	T8OSC1 CH.0 Timer Interrupt Mask Register	Enables/disables interrupt.
	0x4a04	T8OSC1_IFLG0	T8OSC1 CH.0 Timer Interrupt Flag Register	Indicates/resets interrupt occurrence status.
	0x4a05–0x4aff	–	–	Reserved
8-bit OSC1 timer CH.1 (8-bit device)	0x4b00	T8OSC1_CTL1	T8OSC1 CH.1 Timer Control Register	Sets the timer mode and starts/stops the timer.
	0x4b01	T8OSC1_CNT1	T8OSC1 CH.1 Timer Counter Data Register	Counter data
	0x4b02	T8OSC1_CMP1	T8OSC1 CH.1 Timer Compare Data Register	Sets compare data.
	0x4b03	T8OSC1_IMSK1	T8OSC1 CH.1 Timer Interrupt Mask Register	Enables/disables interrupt.
	0x4b04	T8OSC1_IFLG1	T8OSC1 CH.1 Timer Interrupt Flag Register	Indicates/resets interrupt occurrence status.
	0x4b05–0x4bff	–	–	Reserved
SRAM controller (16-bit device)	0x5000–0x5017	–	–	Reserved
	0x5018	RTC_WAIT	RTC Wait Control Register	Sets up RTC access cycle.
	0x501a–0x50ff	–	–	Reserved
Multi-function timer (16-bit device)	0x5200	MFT_TC	MFT Counter Data Register	Counter data
	0x5202	MFT_PRD	MFT Period Data Register	Sets period data.
	0x5204	MFT_CMP	MFT Compare Data Register	Sets compare data.
	0x5206	MFT_CTL	MFT Control Register	Sets the timer mode and starts/stops the timer.
	0x5208–0x521d	–	–	Reserved
	0x521e	MFT_IOCTL	MFT Input/Output Control Register	Controls the clock input/output.
	0x5230	MFT_IE	MFT Interrupt Enable Register	Enables the MFT interrupt.
	0x5238	MFT_IF	MFT Interrupt Flag Register	Indicates the MFT interrupt status.
	0x523a–0x527d	–	–	Reserved
	0x527e	MFT_TST	MFT Test Register	Controls the MFT test.
	0x5280–0x52ff	–	–	Reserved
Remote controller (16-bit device)	0x5400	REMC_PSC	REMC Prescaler Control Register	Sets up the REMC prescaler.
	0x5404	REMC_CFG	REMC Configuration Register	Sets the REMC modes and controls the REMC interrupt.
	0x5406	–	–	Reserved
	0x5408	REMC_CTL	REMC Control Register	Starts/stops transmission.
	0x540a	–	–	Reserved
	0x540c	REMC_CARL	REMC Carrier Load Register	Configures the carrier signal.
	0x540e	REMC_ENVL	REMC Envelope Load Register	Configures the envelope pulse width.
	0x5410	REMC_ENVC	REMC Envelope Capture Register	Input envelope pulse width
	0x5412–0x54ff	–	–	Reserved

AP
IOmap

Peripheral	Address	Register name		Function
A/D converter (16-bit device)	0x5500–0x551f	–	–	Reserved
	0x5520	AD_CLKCTL	A/D Clock Control Register	Controls A/D converter clock.
	0x5522–0x553f	–	–	Reserved
	0x5540	AD_DAT	A/D Conversion Result Register	A/D converted data
	0x5542	AD_TRIG_CH	A/D Trigger/Channel Select Register	Sets start/end channels and conversion mode.
	0x5544	AD_CTL	A/D Control/Status Register	Controls A/D converter and indicates conversion status.
	0x5546	AD_CH_STAT	A/D Channel Status Flag Register	Indicates overwrite error and conversion complete status.
	0x5548	AD_CH0_BUF	A/D CH.0 Conversion Result Buffer Register	A/D CH.0 converted data
	0x554a	AD_CH1_BUF	A/D CH.1 Conversion Result Buffer Register	A/D CH.1 converted data
	0x554c	AD_CH2_BUF	A/D CH.2 Conversion Result Buffer Register	A/D CH.2 converted data
	0x554e	AD_CH3_BUF	A/D CH.3 Conversion Result Buffer Register	A/D CH.3 converted data
	0x5550–0x5557	–	–	Reserved
	0x5558	AD_UPPER	A/D Upper Limit Value Register	Specifies A/D conversion upper limit value.
	0x555a	AD_LOWER	A/D Lower Limit Value Register	Specifies A/D conversion lower limit value.
	0x555c	AD_INTMASK	A/D Conversion Complete Interrupt Mask Register	Masks A/D conversion complete interrupt.
	0x555e	AD_ADVMODE	A/D Converter Mode Select/Internal Status Register	Selects A/D operating mode and indicates internal status and internal counter value.
Watchdog timer (16-bit device)	0x5600–0x565f	–	–	Reserved
	0x5660	WD_WP	WDT Write Protect Register	Enables WDT control registers for writing.
	0x5662	WD_EN	WDT Enable and Setup Register	Configures and starts watchdog timer.
	0x5664	WD_CMP_L	WDT Comparison Data L Register	Comparison data
	0x5666	WD_CMP_H	WDT Comparison Data H Register	
	0x5668	WD_CNT_L	WDT Count Data L Register	Watchdog timer counter data
	0x566a	WD_CNT_H	WDT Count Data H Register	
	0x566c	WD_CTL	WDT Control Register	Resets watchdog timer.
	0x566e–0x56ff	–	–	Reserved
Extended SPI (16-bit device)	0x5700	SPI_ST1	SPI CH.1 Status Register	Indicates transfer and buffer statuses.
	0x5702	SPI_TXD1	SPI CH.1 Transmit Data Register	Transmit data
	0x5704	SPI_RXD1	SPI CH.1 Receive Data Register	Receive data
	0x5706	SPI_CTL1	SPI CH.1 Control Register	Sets the SPI CH.1 mode and enables data transfer.
	0x5708	SPI_CLK1	SPI CH.1 Clock Control Register	Sets up the SPI clock.
	0x570a–0x57ff	–	–	Reserved
ROM controller (16-bit device)	0x5800–0x5803	–	–	Reserved
	0x5804	ROMC_WAIT	ROMC Wait Register	Sets the wait cycle for ROM read.
	0x5806–0x580f	–	–	Reserved
	0x5810	ROMC_PRT	ROMC Protect Register	Enables ROMC registers for writing.
	0x5812–0x5813	–	–	Reserved
	0x5814	TTBR_LOW	Trap Table Base Register 0	Sets the vector table address.
	0x5816	TTBR_HIGH	Trap Table Base Register 1	
	0x5818–0x58ff	–	–	Reserved
S1C17 Core I/O	0xffff90	DBRAM	Debug RAM Base Register	Indicates the debug RAM base address.

Note: Do not access the “Reserved” address in the table above and unused areas in the peripheral area that are not described in the table from the application program.

0x4020 Prescaler

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Prescaler Control Register (PSC_CTL)	**0x4020** (8 bits)	D7–2	**–**	reserved	–				–	–	0 when being read.
		D1	**PRUND**	Prescaler run/stop in debug mode	1	Run	0	Stop	0	R/W	
		D0	**PRUN**	Prescaler run/stop control	1	Run	0	Stop	0	R/W	

0x4100–0x4105 UART (with IrDA)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
UART Status Register (UART_ST)	**0x4100** (8 bits)	D7	**–**	reserved	–				–	–	0 when being read.
		D6	**FER**	Framing error flag	1	Error	0	Normal	0	R/W	Reset by writing 1.
		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 Transmit Data Register (UART_TXD)	**0x4101** (8 bits)	D7–0	**TXD[7:0]**	Transmit data TXD7(6) = MSB TXD0 = LSB	0x0 to 0xff (0x7f)				0x0	R/W	
UART Receive Data Register (UART_RXD)	**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 Mode Register (UART_MOD)	**0x4103** (8 bits)	D7–6	**–**	reserved	–				–	–	0 when being read.
		D5	**TMSEL**	Timer select	1	CLG_T8FU0	0	CLG_T16U0	0	R/W	
		D4	**CHLN**	Character length	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	**SSCK**	Input clock select	1	External	0	Internal	0	R/W	
UART Control Register (UART_CTL)	**0x4104** (8 bits)	D7	**–**	reserved	–				–	–	0 when being read.
		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	1	2 bytes	0	1 byte	0	R/W	
		D0	**RXEN**	UART enable	1	Enable	0	Disable	0	R/W	
UART Expansion Register (UART_EXP)	**0x4105** (8 bits)	D7	**–**	reserved	–				–	–	0 when being read.
		D6–4	**IRCLK[2:0]**	IrDA receive detection clock select	IRCLK[2:0]		Clock		0x0	R/W	
					0x7		PCLK•1/128				
					0x6		PCLK•1/64				
					0x5		PCLK•1/32				
					0x4		PCLK•1/16				
					0x3		PCLK•1/8				
					0x2		PCLK•1/4				
					0x1		PCLK•1/2				
					0x0		PCLK•1/1				
		D3–1	**–**	reserved	–				–	–	0 when being read.
		D0	**IRMD**	IrDA mode select	1	On	0	Off	0	R/W	

0x4200–0x4206 CLG_T16U0 Timer

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
CLG_T16U0 Input Clock Select Register (CLG_T16U0 _CLK)	**0x4200** (16 bits)	D15–4	**–**	reserved	–	–	–	0 when being read.
		D3–0	**DF[3:0]**	Timer input clock select (Prescaler output clock)	DF[3:0] : Clock 0xf : reserved 0xe : PCLK•1/16384 0xd : PCLK•1/8192 0xc : PCLK•1/4096 0xb : PCLK•1/2048 0xa : PCLK•1/1024 0x9 : PCLK•1/512 0x8 : PCLK•1/256 0x7 : PCLK•1/128 0x6 : PCLK•1/64 0x5 : PCLK•1/32 0x4 : PCLK•1/16 0x3 : PCLK•1/8 0x2 : PCLK•1/4 0x1 : PCLK•1/2 0x0 : PCLK•1/1	0x0	R/W	
CLG_T16U0 Reload Data Register (CLG_T16U0 _TR)	**0x4202** (16 bits)	D15–0	**TR[15:0]**	16-bit timer reload data TR15 = MSB TR0 = LSB	0x0 to 0xffff	0x0	R/W	
CLG_T16U0 Counter Data Register (CLG_T16U0 _TC)	**0x4204** (16 bits)	D15–0	**TC[15:0]**	16-bit timer counter data TC15 = MSB TC0 = LSB	0x0 to 0xffff	0xffff	R	
CLG_T16U0 Control Register (CLG_T16U0 _CTL)	**0x4206** (16 bits)	D15–5	**–**	reserved	–	–	–	0 when being read.
		D4	**TRMD**	Count mode select	1 One shot / 0 Repeat	0	R/W	
		D3–2	**–**	reserved	–	–	–	0 when being read.
		D1	**PRESER**	Timer reset	1 Reset / 0 Ignored	0	W	
		D0	**PRUN**	Timer run/stop control	1 Run / 0 Stop	0	R/W	

0x4220–0x4226 CLG_T8FU0 Timer

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
CLG_T8FU0 Input Clock Select Register (CLG_T8FU0_CLK)	**0x4220** (16 bits)	D15–4	–	reserved	–		–	–	0 when being read.
		D3–0	**DF[3:0]**	Timer input clock select (Prescaler output clock)	DF[3:0]	Clock	0x0	R/W	
					0xf	reserved			
					0xe	PCLK•1/16384			
					0xd	PCLK•1/8192			
					0xc	PCLK•1/4096			
					0xb	PCLK•1/2048			
					0xa	PCLK•1/1024			
					0x9	PCLK•1/512			
					0x8	PCLK•1/256			
					0x7	PCLK•1/128			
					0x6	PCLK•1/64			
					0x5	PCLK•1/32			
					0x4	PCLK•1/16			
					0x3	PCLK•1/8			
					0x2	PCLK•1/4			
					0x1	PCLK•1/2			
					0x0	PCLK•1/1			
CLG_T8FU0 Reload Data Register (CLG_T8FU0_TR)	**0x4222** (16 bits)	D15–8	–	reserved	–		–	–	0 when being read.
		D7–0	**TR[7:0]**	CLG_T8FU0 reload data TR7 = MSB TR0 = LSB	0x0 to 0xff		0x0	R/W	
CLG_T8FU0 Counter Data Register (CLG_T8FU0_TC)	**0x4224** (16 bits)	D15–8	–	reserved	–		–	–	0 when being read.
		D7–0	**TC[7:0]**	CLG_T8FU0 counter data TC7 = MSB TC0 = LSB	0x0 to 0xff		0xff	R	
CLG_T8FU0 Control Register (CLG_T8FU0_CTL)	**0x4226** (16 bits)	D15–12	–	reserved	–		–	–	0 when being read.
		D11–8	**TFMD[3:0]**	Fine mode setup	0x0 to 0xf		0x0	R/W	Set a number of times to insert delay into a 16-underflow period.
		D7–5	–	reserved	–		–	–	0 when being read.
		D4	**TRMD**	Count mode select	1 One shot	0 Repeat	0	R/W	
		D3–2	–	reserved	–		–	–	0 when being read.
		D1	**PRESER**	Timer reset	1 Reset	0 Ignored	0	W	
		D0	**PRUN**	Timer run/stop control	1 Run	0 Stop	0	R/W	

0x4240–0x4246 CLG_T8S Timer

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
CLG_T8S Input Clock Select Register (CLG_T8S_CLK)	**0x4240** (16 bits)	D15–4	**–**	reserved	–		–	–	0 when being read.
		D3–0	**DF[3:0]**	Timer input clock select (Prescaler output clock)	DF[3:0]	Clock	0x0	R/W	
					0xf	reserved			
					0xe	PCLK•1/16384			
					0xd	PCLK•1/8192			
					0xc	PCLK•1/4096			
					0xb	PCLK•1/2048			
					0xa	PCLK•1/1024			
					0x9	PCLK•1/512			
					0x8	PCLK•1/256			
					0x7	PCLK•1/128			
					0x6	PCLK•1/64			
					0x5	PCLK•1/32			
					0x4	PCLK•1/16			
					0x3	PCLK•1/8			
					0x2	PCLK•1/4			
					0x1	PCLK•1/2			
					0x0	PCLK•1/1			
CLG_T8S Reload Data Register (CLG_T8S_TR)	**0x4242** (16 bits)	D15–8	**–**	reserved	–		–	–	0 when being read.
		D7–0	**TR[7:0]**	8-bit timer reload data TR7 = MSB TR0 = LSB	0x0 to 0xff		0x0	R/W	
CLG_T8S Counter Data Register (CLG_T8S_TC)	**0x4244** (16 bits)	D15–8	**–**	reserved	–		–	–	0 when being read.
		D7–0	**TC[7:0]**	8-bit timer counter data TC7 = MSB TC0 = LSB	0x0 to 0xff		0xff	R	
CLG_T8S Control Register (CLG_T8S_CTL)	**0x4246** (16 bits)	D15–5	**–**	reserved	–		–	–	0 when being read.
		D4	**TRMD**	Count mode select	1 One shot	0 Repeat	0	R/W	
		D3–2	**–**	reserved	–		–	–	0 when being read.
		D1	**PRESER**	Timer reset	1 Reset	0 Ignored	0	W	
		D0	**PRUN**	Timer run/stop control	1 Run	0 Stop	0	R/W	

0x4260–0x4266 CLG_T8I Timer

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
CLG_T8I Input Clock Select Register (CLG_T8I_CLK)	**0x4260** (16 bits)	D15–4	**–**	reserved	–		–	–	0 when being read.
		D3–0	**DF[3:0]**	Timer input clock select (Prescaler output clock)	DF[3:0]	Clock	0x0	R/W	
					0xf	reserved			
					0xe	PCLK•1/16384			
					0xd	PCLK•1/8192			
					0xc	PCLK•1/4096			
					0xb	PCLK•1/2048			
					0xa	PCLK•1/1024			
					0x9	PCLK•1/512			
					0x8	PCLK•1/256			
					0x7	PCLK•1/128			
					0x6	PCLK•1/64			
					0x5	PCLK•1/32			
					0x4	PCLK•1/16			
					0x3	PCLK•1/8			
					0x2	PCLK•1/4			
					0x1	PCLK•1/2			
					0x0	PCLK•1/1			
CLG_T8I Reload Data Register (CLG_T8I_TR)	**0x4262** (16 bits)	D15–8	**–**	reserved	–		–	–	0 when being read.
		D7–0	**TR[7:0]**	8-bit timer reload data TR7 = MSB TR0 = LSB	0x0 to 0xff		0x0	R/W	
CLG_T8I Counter Data Register (CLG_T8I_TC)	**0x4264** (16 bits)	D15–8	**–**	reserved	–		–	–	0 when being read.
		D7–0	**TC[7:0]**	8-bit timer counter data TC7 = MSB TC0 = LSB	0x0 to 0xff		0xff	R	
CLG_T8I Control Register (CLG_T8I_CTL)	**0x4266** (16 bits)	D15–5	**–**	reserved	–		–	–	0 when being read.
		D4	**TRMD**	Count mode select	1 One shot	0 Repeat	0	R/W	
		D3–2	**–**	reserved	–		–	–	0 when being read.
		D1	**PRESER**	Timer reset	1 Reset	0 Ignored	0	W	
		D0	**PRUN**	Timer run/stop control	1 Run	0 Stop	0	R/W	

0x42e0–0x42f2 Interrupt Controller

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Additional Interrupt Flag Register (ITC_AIFLG)	**0x42e0** (16 bits)	D15	–	reserved	–				–	–	0 when being read.
		D14	**AIFT14**	REMC interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D13	**AIFT13**	I^{2}C slave bus status interrupt flag					0	R/W	
		D12	**AIFT12**	I^{2}C slave transmit/receive interrupt flag					0	R/W	
		D11	–	reserved	–				–	–	0 when being read.
		D10	**AIFT10**	SPI CH.1 interrupt flag	1	Occurred	0	Not occurred	0	R/W	Reset by writing 1.
		D9	–	reserved	–				–	–	0 when being read.
		D8	**AIFT8**	PT8 CH.3 interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D7	**AIFT7**	PT8 CH.2 interrupt flag					0	R/W	
		D6	**AIFT6**	PT8 CH.1/T8OSC1 CH.1 interrupt flag					0	R/W	
		D5	**AIFT5**	PT8 CH.0/T8OSC1 CH.0 interrupt flag					0	R/W	
		D4	**AIFT4**	RTC interrupt flag					0	R/W	
		D3	**AIFT3**	ADC end-of-conversion interrupt flag					0	R/W	
		D2	**AIFT2**	ADC out-of-range interrupt flag					0	R/W	
		D1	–	reserved	–				–	–	0 when being read.
		D0	**AIFT0**	MFT interrupt flag	1	Occurred	0	Not occurred	0	R/W	Reset by writing 1.
Additional Interrupt Enable Register (ITC_AEN)	**0x42e2** (16 bits)	D15	–	reserved	–				–	–	0 when being read.
		D14	**AIEN14**	REMC interrupt enable	1	Enable	0	Disable	0	R/W	
		D13	**AIEN13**	I^{2}C slave bus status interrupt enable					0	R/W	
		D12	**AIEN12**	I^{2}C slave transmit/receive interrupt enable					0	R/W	
		D11	–	reserved	–				–	–	0 when being read.
		D10	**AIEN10**	SPI CH.1 interrupt enable	1	Enable	0	Disable	0	R/W	
		D9	–	reserved	–				–	–	0 when being read.
		D8	**AIEN8**	PT8 CH.3 interrupt enable	1	Enable	0	Disable	0	R/W	
		D7	**AIEN7**	PT8 CH.2 interrupt enable					0	R/W	
		D6	**AIEN6**	PT8 CH.1/T8OSC1 CH.1 interrupt enable					0	R/W	
		D5	**AIEN5**	PT8 CH.0/T8OSC1 CH.0 interrupt enable					0	R/W	
		D4	**AIEN4**	RTC interrupt enable					0	R/W	
		D3	**AIEN3**	ADC end-of-conversion interrupt enable					0	R/W	
		D2	**AIEN2**	ADC out-of-range interrupt enable					0	R/W	
		D1	–	reserved	–				–	–	0 when being read.
		D0	**AIEN0**	MFT interrupt enable	1	Enable	0	Disable	0	R/W	
Additional Interrupt Level Setup Register 0 (ITC_AILV0)	**0x42e6** (16 bits)	D15–3	–	reserved	–				–	–	0 when being read.
		D2–0	**AILV0[2:0]**	MFT interrupt level	0 to 7				0x0	R/W	
Additional Interrupt Level Setup Register 1 (ITC_AILV1)	**0x42e8** (16 bits)	D15–11	–	reserved	–				–	–	0 when being read.
		D10–8	**AILV3[2:0]**	ADC end-of-conversion interrupt level	0 to 7				0x0	R/W	
		D7–3	–	reserved	–				–	–	0 when being read.
		D2–0	**AILV2[2:0]**	ADC out-of-range interrupt level	0 to 7				0x0	R/W	
Additional Interrupt Level Setup Register 2 (ITC_AILV2)	**0x42ea** (16 bits)	D15–11	–	reserved	–				–	–	0 when being read.
		D10–8	**AILV5[2:0]**	PT8 CH.0/T8OSC1 CH.0 interrupt level	0 to 7				0x0	R/W	
		D7–3	–	reserved	–				–	–	0 when being read.
		D2–0	**AILV4[2:0]**	RTC interrupt level	0 to 7				0x0	R/W	
Additional Interrupt Level Setup Register 3 (ITC_AILV3)	**0x42ec** (16 bits)	D15–11	–	reserved	–				–	–	0 when being read.
		D10–8	**AILV7[2:0]**	PT8 CH.2 interrupt level	0 to 7				0x0	R/W	
		D7–3	–	reserved	–				–	–	0 when being read.
		D2–0	**AILV6[2:0]**	PT8 CH.1/T8OSC1 CH.1 interrupt level	0 to 7				0x0	R/W	
Additional Interrupt Level Setup Register 4 (ITC_AILV4)	**0x42ee** (16 bits)	D15–3	–	reserved	–				–	–	0 when being read.
		D2–0	**AILV8[2:0]**	PT8 CH.3 interrupt level	0 to 7				0x0	R/W	
Additional Interrupt Level Setup Register 5 (ITC_AILV5)	**0x42f0** (16 bits)	D15–3	–	reserved	–				–	–	0 when being read.
		D2–0	**AILV10[2:0]**	SPI CH.1 interrupt level	0 to 7				0x0	R/W	
Additional Interrupt Level Setup Register 6 (ITC_AILV6)	**0x42f2** (16 bits)	D15–11	–	reserved	–				–	–	0 when being read.
		D10–8	**AILV13[2:0]**	I^{2}C slave bus status interrupt level	0 to 7				0x0	R/W	
		D7–3	–	reserved	–				–	–	0 when being read.
		D2–0	**AILV12[2:0]**	I^{2}C slave transmit/receive interrupt level	0 to 7				0x0	R/W	

0x42f4–0x430a Interrupt Controller

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Additional Interrupt Level Setup Register 7 (ITC_AILV7)	**0x42f4** (16 bits)	D15–3	–	reserved	–				–	–	0 when being read.
		D2–0	**AILV14[2:0]**	REMC interrupt level	0 to 7				0x0	R/W	
Interrupt Flag Register (ITC_IFLG)	**0x4300** (16 bits)	D15	**IIFT7**	I^{2}C master interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D14	**IIFT6**	SPI CH.0 interrupt flag					0	R/W	
		D13	–	reserved	–				–	–	0 when being read.
		D12	**IIFT4**	UART interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D11	**IIFT3**	CLG_T8I timer interrupt flag					0	R/W	
		D10	**IIFT2**	CLG_T8S timer interrupt flag					0	R/W	
		D9	**IIFT1**	CLG_T8FU0 timer interrupt flag					0	R/W	
		D8	**IIFT0**	CLG_T16U0 timer interrupt flag					0	R/W	
		D7	**EIFT7**	Port interrupt 7 flag					0	R/W	Reset by writing 1 in pulse trigger mode. Cannot be reset by software in level trigger mode.
		D6	**EIFT6**	Port interrupt 6 flag					0	R/W	
		D5	**EIFT5**	Port interrupt 5 flag					0	R/W	
		D4	**EIFT4**	Port interrupt 4 flag					0	R/W	
		D3	**EIFT3**	Port interrupt 3 flag					0	R/W	
		D2	**EIFT2**	Port interrupt 2 flag					0	R/W	
		D1	**EIFT1**	Port interrupt 1 flag					0	R/W	
		D0	**EIFT0**	Port interrupt 0 flag					0	R/W	
Interrupt Enable Register (ITC_EN)	**0x4302** (16 bits)	D15	**IIEN7**	I^{2}C master interrupt enable	1	Enable	0	Disable	0	R/W	
		D14	**IIEN6**	SPI CH.0 interrupt enable					0	R/W	
		D13	–	reserved	–				–	–	0 when being read.
		D12	**IIEN4**	UART interrupt enable	1	Enable	0	Disable	0	R/W	
		D11	**IIEN3**	CLG_T8I timer interrupt enable					0	R/W	
		D10	**IIEN2**	CLG_T8S timer interrupt enable					0	R/W	
		D9	**IIEN1**	CLG_T8FU0 timer interrupt enable					0	R/W	
		D8	**IIEN0**	CLG_T16U0 timer interrupt enable					0	R/W	
		D7	**EIEN7**	Port interrupt 7 enable					0	R/W	
		D6	**EIEN6**	Port interrupt 6 enable					0	R/W	
		D5	**EIEN5**	Port interrupt 5 enable					0	R/W	
		D4	**EIEN4**	Port interrupt 4 enable					0	R/W	
		D3	**EIEN3**	Port interrupt 3 enable					0	R/W	
		D2	**EIEN2**	Port interrupt 2 enable					0	R/W	
		D1	**EIEN1**	Port interrupt 1 enable					0	R/W	
		D0	**EIEN0**	Port interrupt 0 enable					0	R/W	
ITC Control Register (ITC_CTL)	**0x4304** (16 bits)	D15–1	–	reserved	–				–	–	0 when being read.
		D0	**ITEN**	ITC enable	1	Enable	0	Disable	0	R/W	
External Interrupt Level Setup Register 0 (ITC_ELV0)	**0x4306** (16 bits)	D15–14	–	reserved	–				–	–	0 when being read.
		D13	**EITP1**	Port interrupt 1 trigger polarity	1	Positive	0	Negative	0	R/W	
		D12	**EITG1**	Port interrupt 1 trigger mode	1	Level	0	Pulse	0	R/W	
		D11	–	reserved	–				–	–	0 when being read.
		D10–8	**EILV1[2:0]**	Port interrupt 1 level	0 to 7				0x0	R/W	
		D7–6	–	reserved	–				–	–	0 when being read.
		D5	**EITP0**	Port interrupt 0 trigger polarity	1	Positive	0	Negative	0	R/W	
		D4	**EITG0**	Port interrupt 0 trigger mode	1	Level	0	Pulse	0	R/W	
		D3	–	reserved	–				–	–	0 when being read.
		D2–0	**EILV0[2:0]**	Port interrupt 0 level	0 to 7				0x0	R/W	
External Interrupt Level Setup Register 1 (ITC_ELV1)	**0x4308** (16 bits)	D15–14	–	reserved	–				–	–	0 when being read.
		D13	**EITP3**	Port interrupt 3 trigger polarity	1	Positive	0	Negative	0	R/W	
		D12	**EITG3**	Port interrupt 3 trigger mode	1	Level	0	Pulse	0	R/W	
		D11	–	reserved	–				–	–	0 when being read.
		D10–8	**EILV3[2:0]**	Port interrupt 3 level	0 to 7				0x0	R/W	
		D7–6	–	reserved	–				–	–	0 when being read.
		D5	**EITP2**	Port interrupt 2 trigger polarity	1	Positive	0	Negative	0	R/W	
		D4	**EITG2**	Port interrupt 2 trigger mode	1	Level	0	Pulse	0	R/W	
		D3	–	reserved	–				–	–	0 when being read.
		D2–0	**EILV2[2:0]**	Port interrupt 2 level	0 to 7				0x0	R/W	
External Interrupt Level Setup Register 2 (ITC_ELV2)	**0x430a** (16 bits)	D15–14	–	reserved	–				–	–	0 when being read.
		D13	**EITP5**	Port interrupt 5 trigger polarity	1	Positive	0	Negative	0	R/W	
		D12	**EITG5**	Port interrupt 5 trigger mode	1	Level	0	Pulse	0	R/W	
		D11	–	reserved	–				–	–	0 when being read.
		D10–8	**EILV5[2:0]**	Port interrupt 5 level	0 to 7				0x0	R/W	
		D7–6	–	reserved	–				–	–	0 when being read.
		D5	**EITP4**	Port interrupt 4 trigger polarity	1	Positive	0	Negative	0	R/W	
		D4	**EITG4**	Port interrupt 4 trigger mode	1	Level	0	Pulse	0	R/W	
		D3	–	reserved	–				–	–	0 when being read.
		D2–0	**EILV4[2:0]**	Port interrupt 4 level	0 to 7				0x0	R/W	

0x430c–0x4314 Interrupt Controller

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
External Interrupt Level Setup Register 3 (ITC_ELV3)	**0x430c** (16 bits)	D15–14	–	reserved	–				–	–	0 when being read.
		D13	**EITP7**	Port interrupt 7 trigger polarity	1	Positive	0	Negative	0	R/W	
		D12	**EITG7**	Port interrupt 7 trigger mode	1	Level	0	Pulse	0	R/W	
		D11	–	reserved	–				–	–	0 when being read.
		D10–8	**EILV7[2:0]**	Port interrupt 7 level	0 to 7				0x0	R/W	
		D7–6	–	reserved	–				–	–	0 when being read.
		D5	**EITP6**	Port interrupt 6 trigger polarity	1	Positive	0	Negative	0	R/W	
		D4	**EITG6**	Port interrupt 6 trigger mode	1	Level	0	Pulse	0	R/W	
		D3	–	reserved	–				–	–	0 when being read.
		D2–0	**EILV6[2:0]**	Port interrupt 6 level	0 to 7				0x0	R/W	
Internal Interrupt Level Setup Register 0 (ITC_ILV0)	**0x430e** (16 bits)	D15–11	–	reserved	–				–	–	0 when being read.
		D10–8	**IILV1[2:0]**	CLG_T8FU0 timer interrupt level	0 to 7				0x0	R/W	
		D7–3	–	reserved	–				–	–	0 when being read.
		D2–0	**IILV0[2:0]**	CLG_T16U0 timer interrupt level	0 to 7				0x0	R/W	
Internal Interrupt Level Setup Register 1 (ITC_ILV1)	**0x4310** (16 bits)	D15–11	–	reserved	–				–	–	0 when being read.
		D10–8	**IILV3[2:0]**	CLG_T8I timer interrupt level	0 to 7				0x0	R/W	
		D7–3	–	reserved	–				–	–	0 when being read.
		D2–0	**IILV2[2:0]**	CLG_T8S timer interrupt level	0 to 7				0x0	R/W	
Internal Interrupt Level Setup Register 2 (ITC_ILV2)	**0x4312** (16 bits)	D15–3	–	reserved	–				–	–	0 when being read.
		D2–0	**IILV4[2:0]**	UART interrupt level	0 to 7				0x0	R/W	
Internal Interrupt Level Setup Register 3 (ITC_ILV3)	**0x4314** (16 bits)	D15–11	–	reserved	–				–	–	0 when being read.
		D10–8	**IILV7[2:0]**	I^{2}C master interrupt level	0 to 7				0x0	R/W	
		D7–3	–	reserved	–				–	–	0 when being read.
		D2–0	**IILV6[2:0]**	SPI CH.0 interrupt level	0 to 7				0x0	R/W	

0x4320–0x4326 SPI

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 CH.0 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 CH.0 receive data buffer SPRDB7 = MSB SPRDB0 = LSB	0x0 to 0xff				0x0	R	
SPI CH.0 Control Register (SPI_CTL0)	**0x4326** (16 bits)	D15–6	**–**	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 CH.0 enable	1	Enable	0	Disable	0	R/W	

0x4340–0x4346 I^2C Master

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
I^2C Master Enable Register (I2CM_EN)	**0x4340** (16 bits)	D15–1	**–**	reserved	–				–	–	0 when being read.
		D0	**I2CMEN**	I^2C master enable	1	Enable	0	Disable	0	R/W	
I^2C 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	
I^2C Master Data Register (I2CM_DAT)	**0x4344** (16 bits)	D15–12	**–**	reserved	–				–	–	0 when being read.
		D11	**RBRDY**	Receive buffer ready	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	
I^2C 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 I^{2}C Slave

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
I^{2}C Slave Transmit Data Register (I2CS_TRNS)	**0x4360** (16 bits)	D15–8	–	reserved	–				–	–	0 when being read.
		D7–0	**SDATA[7:0]**	I^{2}C slave transmit data	0–0xff				0x0	R/W	
I^{2}C Slave Receive Data Register (I2CS_RECV)	**0x4362** (16 bits)	D15–8	–	reserved	–				–	–	0 when being read.
		D7–0	**RDATA[7:0]**	I^{2}C slave receive data	0–0xff				0x0	R	
I^{2}C Slave Address Setup Register (I2CS_SADRS)	**0x4364** (16 bits)	D15–7	–	reserved	–				–	–	0 when being read.
		D6–0	**SADRS[6:0]**	I^{2}C slave address	0–0x7f				0x0	R/W	
I^{2}C 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	**I2C_EN**	I^{2}C 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**	I^{2}C slave communication mode	1	Active	0	Standby	0	R/W	
I^{2}C 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	
I^{2}C Slave Access Status Register (I2CS_ASTAT)	**0x436a** (16 bits)	D15–3	–	reserved	–				–	–	0 when being read.
		D2	**BUSY**	I^{2}C bus status	1	Busy	0	Free	0	R	
		D1	**SELECTED**	I^{2}C slave select status	1	Selected	0	Not selected	0	R	
		D0	**R/W**	Read/write direction	1	Output	0	Input	0	R	
I^{2}C Slave Interrupt Control Register (I2CS_ICTL)	**0x436c** (16 bits)	D15–5	–	reserved	–				–	–	0 when being read.
		D4	**RXRDY_DRQ_EN**	RXRDY DMA request enable	1	Enable	0	Disable	0	R/W	Fix at 0.
		D3	**TXEMP_DRQ_EN**	TXEMP DMA request enable	1	Enable	0	Disable	0	R/W	
		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	

0x4400–0x4422 GPIO & Port MUX

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
P0 Port Input Data Register (P0_DAT)	**0x4400** (8 bits)	D7–4	–	reserved	–				–	–	0 when being read.
		D3–0	**P0D[3:0]**	P0[3:0] port input data	1	1 (High)	0	0 (Low)	Ext.	R/W	Ext.: The initial value depends on the external pin status.
P1 Port Input / Output Data Register (P1_DAT)	**0x4402** (8 bits)	D7	–	reserved	–				–	–	0 when being read.
		D6–0	**P1D[6:0]**	P1[6:0] port input/output data	1	1 (High)	0	0 (Low)	Ext.	R/W	Ext.: The initial value depends on the external pin status.
P1 I/O Control Register (P1_IOC)	**0x4403** (8 bits)	D7	–	reserved	–				–	–	0 when being read.
		D6–0	**IOC1[6:0]**	P1[6:0] I/O control	1	Output	0	Input	0	R/W	
P2 Port Input / Output Data Register (P2_DAT)	**0x4404** (8 bits)	D7–0	**P2D[7:0]**	P2[7:0] port input/output data	1	1 (High)	0	0 (Low)	Ext.	R/W	Ext.: The initial value depends on the external pin status.
P2 I/O Control Register (P2_IOC)	**0x4405** (8 bits)	D7–0	**IOC2[7:0]**	P2[7:0] I/O control	1	Output	0	Input	0	R/W	
P3 Port Input / Output Data Register (P3_DAT)	**0x4406** (8 bits)	D7–5	**P3D[7:5]**	P3[7:5] port input/output data	1	1 (High)	0	0 (Low)	Ext.	R/W	Ext.: The initial value depends on the external pin status.
		D4–3	–	reserved	–				–	–	0 when being read.
		D2–0	**P3D[2:0]**	P3[2:0] port input/output data	1	1 (High)	0	0 (Low)	Ext.	R/W	
P3 I/O Control Register (P3_IOC)	**0x4407** (8 bits)	D7–5	**IOC3[7:5]**	P3[7:5] I/O control	1	Output	0	Input	0	R/W	
		D4–3	–	reserved	–				–	–	0 when being read.
		D2–0	**IOC3[2:0]**	P3[2:0] I/O control	1	Output	0	Input	0	R/W	
P4 Port Input / Output Data Register (P4_DAT)	**0x4408** (8 bits)	D7–5	–	reserved	–				–	–	0 when being read.
		D4–0	**P4D[4:0]**	P4[4:0] port input/output data	1	1 (High)	0	0 (Low)	Ext.	R/W	Ext.: The initial value depends on the external pin status.
P4 I/O Control Register (P4_IOC)	**0x4409** (8 bits)	D7–5	–	reserved	–				–	–	0 when being read.
		D4–0	**IOC4[4:0]**	P4[4:0] I/O control	1	Output	0	Input	0	R/W	
P5 Port Input / Output Data Register (P5_DAT)	**0x440a** (8 bits)	D7–4	–	reserved	–				–	–	0 when being read.
		D3–0	**P5D[3:0]**	P5[3:0] port input/output data	1	1 (High)	0	0 (Low)	Ext.	R/W	Ext.: The initial value depends on the external pin status.
P5 I/O Control Register (P5_IOC)	**0x440b** (8 bits)	D7–4	–	reserved	–				–	–	0 when being read.
		D3–0	**IOC5[3:0]**	P5[3:0] I/O control	1	Output	0	Input	0	R/W	
P00–P03 Port Function Select Register (P0_03_CFP)	**0x4420** (8 bits)	D7–6	**CFP03[1:0]**	P03 port function select	CFP03[1:0] 0x3–0x2 0x1 0x0		Function reserved AIN3 P03		0x0	R/W	
		D5–4	**CFP02[1:0]**	P02 port function select	CFP02[1:0] 0x3–0x2 0x1 0x0		Function reserved AIN2 P02		0x0	R/W	
		D3–2	**CFP01[1:0]**	P01 port function select	CFP01[1:0] 0x3–0x2 0x1 0x0		Function reserved AIN1 P01		0x0	R/W	
		D1–0	**CFP00[1:0]**	P00 port function select	CFP00[1:0] 0x3–0x2 0x1 0x0		Function reserved AIN0 P00		0x0	R/W	
P10–P13 Port Function Select Register (P1_03_CFP)	**0x4422** (8 bits)	D7–6	**CFP13[1:0]**	P13 port function select	CFP13[1:0] 0x3 0x2 0x1 0x0		Function SPI_SDI0 #SPI_SSI1 #SPI_SSI0 P13		0x0	R/W	
		D5–4	**CFP12[1:0]**	P12 port function select	CFP12[1:0] 0x3–0x2 0x1 0x0		Function reserved SPI_SCK0 P12		0x0	R/W	
		D3–2	**CFP11[1:0]**	P11 port function select	CFP11[1:0] 0x3–0x2 0x1 0x0		Function reserved SPI_SDO0 P11		0x0	R/W	
		D1–0	**CFP10[1:0]**	P10 port function select	CFP10[1:0] 0x3–0x2 0x1 0x0		Function reserved SPI_SDI0 P10		0x0	R/W	

0x4423–0x4428 GPIO & Port MUX

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
P14–P16 Port Function Select Register (P1_46_CFP)	**0x4423** (8 bits)	D7–6	–	reserved	–		–	–	0 when being read.
		D5–4	**CFP16[1:0]**	P16 port function select	CFP16[1:0]	Function	0x0	R/W	
					0x3–0x2	reserved			
					0x1	SPI_SCK1			
					0x0	P16			
		D3–2	**CFP15[1:0]**	P15 port function select	CFP15[1:0]	Function	0x0	R/W	
					0x3–0x2	reserved			
					0x1	SPI_SDO1			
					0x0	P15			
		D1–0	**CFP14[1:0]**	P14 port function select	CFP14[1:0]	Function	0x0	R/W	
					0x3–0x2	reserved			
					0x1	SPI_SDI1			
					0x0	P14			
P25–P27 Port Function Select Register (P2_57_CFP)	**0x4425** (8 bits)	D7–6	**CFP27[1:0]**	P27 port function select	CFP27[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	I2CS_SCL			
					0x1	reserved			
					0x0	P27			
		D5–4	**CFP26[1:0]**	P26 port function select	CFP26[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	I2CS_SDA			
					0x1	reserved			
					0x0	P26			
		D3–2	**CFP25[1:0]**	P25 port function select	CFP25[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	#I2CS_RST			
					0x1	reserved			
					0x0	P25			
		D1–0	–	reserved	–		–	–	0 when being read.
P30–P32 Port Function Select Register (P3_02_CFP)	**0x4426** (8 bits)	D7–6	–	reserved	–		–	–	0 when being read.
		D5–4	**CFP32[1:0]**	P32 port function select	CFP32[1:0]	Function	0x0	R/W	
					0x3	CMU_CLK			
					0x2	#WDT_NMI			
					0x1	WDT_CLK			
					0x0	P32			
		D3–2	**CFP31[1:0]**	P31 port function select	CFP31[1:0]	Function	0x0	R/W	
					0x3	#ADTRG			
					0x2	#TM0			
					0x1	#I2CS_RST			
					0x0	P31			
		D1–0	**CFP30[1:0]**	P30 port function select	CFP30[1:0]	Function	0x0	R/W	
					0x3–0x2	reserved			
					0x1	TM0			
					0x0	P30			
P35–P37 Port Function Select Register (P3_57_CFP)	**0x4427** (8 bits)	D7–6	**CFP37[1:0]**	P37 port function select	CFP37[1:0]	Function	0x0	R/W	
					0x3–0x2	reserved			
					0x1	P37			
					0x0	DST2			
		D5–4	**CFP36[1:0]**	P36 port function select	CFP36[1:0]	Function	0x0	R/W	
					0x3–0x2	reserved			
					0x1	P36			
					0x0	DSIO			
		D3–2	**CFP35[1:0]**	P35 port function select	CFP35[1:0]	Function	0x0	R/W	
					0x3–0x2	reserved			
					0x1	P35			
					0x0	DCLK			
		D1–0	–	reserved	–		–	–	0 when being read.
P40–P43 Port Function Select Register (P4_03_CFP)	**0x4428** (8 bits)	D7–6	**CFP43[1:0]**	P43 port function select	CFP43[1:0]	Function	0x0	R/W	
					0x3	PWMPRT0			
					0x2	REMC_IN			
					0x1	I2CS_SDA			
					0x0	P43			
		D5–4	**CFP42[1:0]**	P42 port function select	CFP42[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	EXCL0			
					0x1	#SCLK0			
					0x0	P42			
		D3–2	**CFP41[1:0]**	P41 port function select	CFP41[1:0]	Function	0x0	R/W	
					0x3–0x2	reserved			
					0x1	SOUT0			
					0x0	P41			
		D1–0	**CFP40[1:0]**	P40 port function select	CFP40[1:0]	Function	0x0	R/W	
					0x3–0x2	reserved			
					0x1	SIN0			
					0x0	P40			

0x4429–0x4444 GPIO & Port MUX

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
P44 Port Function Select Register (P4_4_CFP)	**0x4429** (8 bits)	D7–2	–	reserved	–		–	–	0 when being read.
		D1–0	**CFP44[1:0]**	P44 port function select	CFP44[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	REMC_OUT			
					0x1	I2CS_SCL			
					0x0	P44			
P50–P53 Port Function Select Register (P5_03_CFP)	**0x442a** (8 bits)	D7–6	**CFP53[1:0]**	P53 port function select	CFP53[1:0]	Function	0x0	R/W	
					0x3–0x2	reserved			
					0x1	REMC_OUT			
					0x0	P53			
		D5–4	**CFP52[1:0]**	P52 port function select	CFP52[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	#TM0			
					0x1	REMC_IN			
					0x0	P52			
		D3–2	**CFP51[1:0]**	P51 port function select	CFP51[1:0]	Function	0x0	R/W	
					0x3–0x2	reserved			
					0x1	I2C_SCL			
					0x0	P51			
		D1–0	**CFP50[1:0]**	P50 port function select	CFP50[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	EXCL0			
					0x1	I2C_SDA			
					0x0	P50			
Port Input Interrupt 0 Select Register (PINTSEL0)	**0x4440** (8 bits)	D7–4	**PINT0[3:0]**	Interrupt port select	PINT0[3:0]	Function	0xf	R/W	
					0xc–0x6	reserved			
					0x5	P50			
					0x4	P40			
					0x3	P30			
					0x2	P20			
					0x1	P10			
					0x0	P00			
		D3–0	–	reserved	–		X	R	
Port Input Interrupt 1 Select Register (PINTSEL1)	**0x4441** (8 bits)	D7–4	**PINT1[3:0]**	Interrupt port select	PINT1[3:0]	Function	0xf	R/W	
					0xc–0x6	reserved			
					0x5	P51			
					0x4	P41			
					0x3	P31			
					0x2	P21			
					0x1	P11			
					0x0	P01			
		D3–0	–	reserved	–		X	R	
Port Input Interrupt 2 Select Register (PINTSEL2)	**0x4442** (8 bits)	D7–4	**PINT2[3:0]**	Interrupt port select	PINT2[3:0]	Function	0xf	R/W	
					0xc–0x6	reserved			
					0x5	P52			
					0x4	P42			
					0x3	P32			
					0x2	P22			
					0x1	P12			
					0x0	P02			
		D3–0	–	reserved	–		X	R	
Port Input Interrupt 3 Select Register (PINTSEL3)	**0x4443** (8 bits)	D7–4	**PINT3[3:0]**	Interrupt port select	PINT3[3:0]	Function	0xf	R/W	
					0xc–0x6	reserved			
					0x5	P53			
					0x4	P43			
					0x3	–			
					0x2	P23			
					0x1	P13			
					0x0	P03			
		D3–0	–	reserved	–		X	R	
Port Input Interrupt 4 Select Register (PINTSEL4)	**0x4444** (8 bits)	D7–4	**PINT4[3:0]**	Interrupt port select	PINT4[3:0]	Function	0xf	R/W	
					0xc–0x6	reserved			
					0x5	–			
					0x4	P44			
					0x3	–			
					0x2	P24			
					0x1	P14			
					0x0	–			
		D3–0	–	reserved	–		X	R	

0x4445–0x4447 GPIO & Port MUX

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
Port Input Interrupt 5 Select Register (PINTSEL5)	**0x4445** (8 bits)	D7–4	**PINT5[3:0]**	Interrupt port select	PINT5[3:0]	Function	0xf	R/W	
					0xc–0x6	reserved			
					0x5	–			
					0x4	–			
					0x3	P35			
					0x2	P25			
					0x1	P15			
					0x0	–			
		D3–0	–	reserved	–		X	R	
Port Input Interrupt 6 Select Register (PINTSEL6)	**0x4446** (8 bits)	D7–4	**PINT6[3:0]**	Interrupt port select	PINT6[3:0]	Function	0xf	R/W	
					0xc–0x6	reserved			
					0x5	–			
					0x4	–			
					0x3	P36			
					0x2	P26			
					0x1	P16			
					0x0	–			
		D3–0	–	reserved	–		X	R	
Port Input Interrupt 7 Select Register (PINTSEL7)	**0x4447** (8 bits)	D7–4	**PINT7[3:0]**	Interrupt port select	PINT7[3:0]	Function	0xf	R/W	
					0xc–0x6	reserved			
					0x5	–			
					0x4	–			
					0x3	P37			
					0x2	P27			
					0x1	–			
					0x0	–			
		D3–0	–	reserved	–		X	R	

0x4600–0x462a Real-time Clock

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
RTC Interrupt Status Register (RTC_INTSTAT)	**0x4600** (8 bits)	D7–1	**–**	reserved	–		–	–	0 when being read.
		D0	**RTCIRQ**	Interrupt status	1 Occurred	0 Not occurred	X	R/W	Reset by writing 1.
RTC Interrupt Mode Register (RTC_INTMODE)	**0x4601** (8 bits)	D7–4	**–**	reserved	–		–	–	0 when being read.
		D3–2	**RTCT[1:0]**	RTC interrupt cycle setup	RTCT[1:0]	Cycle	X	R/W	
					0x3	1 hour			
					0x2	1 minute			
					0x1	1 second			
					0x0	1/64 second			
		D1	**RTCIMD**	RTC interrupt mode select	1 Level sense	0 Edge trigger	X	R/W	
		D0	**RTCIEN**	RTC interrupt enable	1 Enable	0 Disable	X	R/W	
RTC Control 0 Register (RTC_CNTL0)	**0x4602** (8 bits)	D7–5	**–**	reserved	–		–	–	0 when being read.
		D4	**RTC24H**	24H/12H mode select	1 24H	0 12H	X	R/W	
		D3	**–**	reserved	–		–	–	0 when being read.
		D2	**RTCADJ**	30-second adjustment	1 Adjust	0 –	X	R/W	
		D1	**RTCSTP**	Counter run/stop control	1 Stop	0 Run	X	R/W	
		D0	**RTCRST**	Software reset	1 Reset	0 –	X	R/W	
RTC Control 1 Register (RTC_CNTL1)	**0x4603** (8 bits)	D7–2	**–**	reserved	–		–	–	0 when being read.
		D1	**RTCBSY**	Counter busy flag	1 Busy	0 R/W possible	X	R	
		D0	**RTCHLD**	Counter hold control	1 Hold	0 Running	X	R/W	
RTC Second Register (RTC_SEC)	**0x4614** (8 bits)	D7	**–**	reserved	–		–	–	0 when being read.
		D6–4	**RTCSH[2:0]**	RTC 10-second counter	0 to 5		X	R/W	
		D3–0	**RTCSL[3:0]**	RTC 1-second counter	0 to 9		X	R/W	
RTC Minute Register (RTC_MIN)	**0x4615** (8 bits)	D7	**–**	reserved	–		–	–	0 when being read.
		D6–4	**RTCMIH[2:0]**	RTC 10-minute counter	0 to 5		X	R/W	
		D3–0	**RTCMIL[3:0]**	RTC 1-minute counter	0 to 9		X	R/W	
RTC Hour Register (RTC_HOUR)	**0x4616** (8 bits)	D7	**–**	reserved	–		–	–	0 when being read.
		D6	**RTCAP**	AM/PM indicator	1 PM	0 AM	X	R/W	
		D5–4	**RTCHH[1:0]**	RTC 10-hour counter	0 to 2 or 0 to 1		X	R/W	
		D3–0	**RTCHL[3:0]**	RTC 1-hour counter	0–9		X	R/W	
RTC Day Register (RTC_DAY)	**0x4617** (8 bits)	D7–6	**–**	reserved	–		–	–	0 when being read.
		D5–4	**RTCDH[1:0]**	RTC 10-day counter	0 to 3		X	R/W	
		D3–0	**RTCDL[3:0]**	RTC 1-day counter	0 to 9		X	R/W	
RTC Month Register (RTC_MONTH)	**0x4628** (8 bits)	D7–5	**–**	reserved	–		–	–	0 when being read.
		D4	**RTCMOH**	RTC 10-month counter	0 to 1		X	R/W	
		D3–0	**RTCMOL[3:0]**	RTC 1-month counter	0 to 9		X	R/W	
RTC Year Register (RTC_YEAR)	**0x4629** (8 bits)	D7–4	**RTCYH[3:0]**	RTC 10-year counter	0 to 9		X	R/W	
		D3–0	**RTCYL[3:0]**	RTC 1-year counter	0 to 9		X	R/W	
RTC Days of Week Register (RTC_WEEK)	**0x462a** (8 bits)	D7–3	**–**	reserved	–		–	–	0 when being read.
		D2–0	**RTCWK[2:0]**	RTC days of week counter	RTCWK[2:0]	Days of week	X	R/W	
					0x7	–			
					0x6	Saturday			
					0x5	Friday			
					0x4	Thursday			
					0x3	Wednesday			
					0x2	Tuesday			
					0x1	Monday			
					0x0	Sunday			

0x4800–0x4803 8-bit Programmable Timer CH.0

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
PT8 CH.0 Input Clock Select Register (PT8_CLK0)	**0x4800** (8 bits)	D7–5	**–**	reserved	–		–	–	0 when being read.
		D4	**PT8_EN**	PT8 Clock Enable	1 Enable	0 Disable	0	R/W	
		D3–0	**PT8_CLK [3:0]**	PT8 clock division ratio selection (Prescaler output clock)	PT8_CLK[3:0]	Clock	0x0	R/W	
					0xf–0xd	reserved			
					0xc	PT8_CLK•1/4096			
					0xb	PT8_CLK•1/2048			
					0xa	PT8_CLK•1/1024			
					0x9	PT8_CLK•1/512			
					0x8	PT8_CLK•1/256			
					0x7	PT8_CLK•1/128			
					0x6	PT8_CLK•1/64			
					0x5	PT8_CLK•1/32			
					0x4	PT8_CLK•1/16			
					0x3	PT8_CLK•1/8			
					0x2	PT8_CLK•1/4			
					0x1	PT8_CLK•1/2			
					0x0	PT8_CLK•1/1			
PT8 CH.0 Reload Data Register (PT8_RLD0)	**0x4801** (8 bits)	D7–0	**PT8_RLD [7:0]**	8-bit programmable timer reload data PT8_RLD7 = MSB PT8_RLD0 = LSB	0 to 255		X	R/W	
PT8 CH.0 Counter Data Register (PT8_PTD0)	**0x4802** (8 bits)	D7–0	**PT8_PTD [7:0]**	8-bit programmable timer counter data PT8_PTD7 = MSB PT8_PTD0 = LSB	0 to 255		X	R	
PT8 CH.0 Control Register (PT8_CTL0)	**0x4803** (8 bits)	D7–2	**–**	reserved	–		–	–	0 when being read.
		D1	**PT8_PSET**	Timer reset	1 Reset	0 Ignored	0	W	
		D0	**PT8_RUN**	Timer run/stop control	1 Run	0 Stop	0	R/W	

0x4804–0x4807 8-bit Programmable Timer CH.1

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
PT8 CH.1 Input Clock Select Register (PT8_CLK1)	**0x4804** (8 bits)	D7–5	**–**	reserved	–		–	–	0 when being read.
		D4	**PT8_EN**	PT8 Clock Enable	1 Enable	0 Disable	0	R/W	
		D3–0	**PT8_CLK [3:0]**	PT8 clock division ratio selection (Prescaler output clock)	PT8_CLK[3:0]	Clock	0x0	R/W	
					0xf–0xd	reserved			
					0xc	PT8_CLK•1/4096			
					0xb	PT8_CLK•1/2048			
					0xa	PT8_CLK•1/1024			
					0x9	PT8_CLK•1/512			
					0x8	PT8_CLK•1/256			
					0x7	PT8_CLK•1/128			
					0x6	PT8_CLK•1/64			
					0x5	PT8_CLK•1/32			
					0x4	PT8_CLK•1/16			
					0x3	PT8_CLK•1/8			
					0x2	PT8_CLK•1/4			
					0x1	PT8_CLK•1/2			
					0x0	PT8_CLK•1/1			
PT8 CH.1 Reload Data Register (PT8_RLD1)	**0x4805** (8 bits)	D7–0	**PT8_RLD [7:0]**	8-bit programmable timer reload data PT8_RLD7 = MSB PT8_RLD0 = LSB	0 to 255		X	R/W	
PT8 CH.1 Counter Data Register (PT8_PTD1)	**0x4806** (8 bits)	D7–0	**PT8_PTD [7:0]**	8-bit programmable timer counter data PT8_PTD7 = MSB PT8_PTD0 = LSB	0 to 255		X	R	
PT8 CH.1 Control Register (PT8_CTL1)	**0x4807** (8 bits)	D7–2	**–**	reserved	–		–	–	0 when being read.
		D1	**PT8_PSET**	Timer reset	1 Reset	0 Ignored	0	W	
		D0	**PT8_RUN**	Timer run/stop control	1 Run	0 Stop	0	R/W	

AP
IOmap

0x4808–0x480b 8-bit Programmable Timer CH.2

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
PT8 CH.2 Input Clock Select Register (PT8_CLK2)	**0x4808** (8 bits)	D7–5	**–**	reserved	–	–	–	0 when being read.
		D4	**PT8_EN**	PT8 Clock Enable	1 Enable 0 Disable	0	R/W	
		D3–0	**PT8_CLK [3:0]**	PT8 clock division ratio selection (Prescaler output clock)	PT8_CLK[3:0] : Clock 0xf–0xd : reserved 0xc : PT8_CLK•1/4096 0xb : PT8_CLK•1/2048 0xa : PT8_CLK•1/1024 0x9 : PT8_CLK•1/512 0x8 : PT8_CLK•1/256 0x7 : PT8_CLK•1/128 0x6 : PT8_CLK•1/64 0x5 : PT8_CLK•1/32 0x4 : PT8_CLK•1/16 0x3 : PT8_CLK•1/8 0x2 : PT8_CLK•1/4 0x1 : PT8_CLK•1/2 0x0 : PT8_CLK•1/1	0x0	R/W	
PT8 CH.2 Reload Data Register (PT8_RLD2)	**0x4809** (8 bits)	D7–0	**PT8_RLD [7:0]**	8-bit programmable timer reload data PT8_RLD7 = MSB PT8_RLD0 = LSB	0 to 255	X	R/W	
PT8 CH.2 Counter Data Register (PT8_PTD2)	**0x480a** (8 bits)	D7–0	**PT8_PTD [7:0]**	8-bit programmable timer counter data PT8_PTD7 = MSB PT8_PTD0 = LSB	0 to 255	X	R	
PT8 CH.2 Control Register (PT8_CTL2)	**0x480b** (8 bits)	D7–2	**–**	reserved	–	–	–	0 when being read.
		D1	**PT8_PSET**	Timer reset	1 Reset 0 Ignored	0	W	
		D0	**PT8_RUN**	Timer run/stop control	1 Run 0 Stop	0	R/W	

0x480c–0x480f 8-bit Programmable Timer CH.3

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
PT8 CH.3 Input Clock Select Register (PT8_CLK3)	**0x480c** (8 bits)	D7–5	**–**	reserved	–	–	–	0 when being read.
		D4	**PT8_EN**	PT8 Clock Enable	1 Enable 0 Disable	0	R/W	
		D3–0	**PT8_CLK [3:0]**	PT8 clock division ratio selection (Prescaler output clock)	PT8_CLK[3:0] Clock 0xf–0xd reserved 0xc PT8_CLK•1/4096 0xb PT8_CLK•1/2048 0xa PT8_CLK•1/1024 0x9 PT8_CLK•1/512 0x8 PT8_CLK•1/256 0x7 PT8_CLK•1/128 0x6 PT8_CLK•1/64 0x5 PT8_CLK•1/32 0x4 PT8_CLK•1/16 0x3 PT8_CLK•1/8 0x2 PT8_CLK•1/4 0x1 PT8_CLK•1/2 0x0 PT8_CLK•1/1	0x0	R/W	
PT8 CH.3 Reload Data Register (PT8_RLD3)	**0x480d** (8 bits)	D7–0	**PT8_RLD [7:0]**	8-bit programmable timer reload data PT8_RLD7 = MSB PT8_RLD0 = LSB	0 to 255	X	R/W	
PT8 CH.3 Counter Data Register (PT8_PTD3)	**0x480e** (8 bits)	D7–0	**PT8_PTD [7:0]**	8-bit programmable timer counter data PT8_PTD7 = MSB PT8_PTD0 = LSB	0 to 255	X	R	
PT8 CH.3 Control Register (PT8_CTL3)	**0x480f** (8 bits)	D7–2	**–**	reserved	–	–	–	0 when being read.
		D1	**PT8_PSET**	Timer reset	1 Reset 0 Ignored	0	W	
		D0	**PT8_RUN**	Timer run/stop control	1 Run 0 Stop	0	R/W	

0x4900–0x4920 Clock Management Unit

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
System Clock Control Register (CMU _SYSCLKCTL)	0x4900 (8 bits)	D7	**OSC3OFF**	OSC3 disable during SLEEP	1	Stop	0	Run	0	R/W	
		D6	**TMHSP**	Wait-timer high-speed mode	1	High speed	0	Normal	0	R/W	
		D5	**–**	reserved	–				–	–	0 when being read.
		D4	**WAKEUPWT**	Wakeup-wait function enable	1	Wait interrupt	0	No wait	0	R/W	
		D3	**–**	reserved	–				–	–	0 when being read.
		D2	**OSCSEL**	OSC clock selection	1	OSC1	0	OSC3	0	R/W	
		D1	**SOSC3**	OSC3 oscillator on/off	1	On	0	Off	1	R/W	
		D0	**–**	reserved	–				–	–	0 when being read.
OSC3 Wait Timer Register (CMU_OSC3_ WCNT)	0x4901 (8 bits)	D7–0	**OSCTM[7:0]**	OSC oscillation stabilization-wait timer	0–255				0x0	R/W	
Noise Filter Control Register (CMU_NF)	0x4902 (8 bits)	D7–5	**–**	reserved	–				–	–	0 when being read.
		D4	**DSINNF**	DSIO input noise filter enable	1	Enable	0	Disable	0	R/W	
		D3–2	**–**	reserved	–				–	–	0 when being read.
		D1	**INPORTNF**	Input port noise filter enable	1	Disable	0	Enable	1	R/W	
		D0	**OSC3NF**	OSC3 input noise filter enable	1	Disable	0	Enable	1	R/W	
OSC3 Clock Divider Register (CMU_OSC3DIV)	0x4903 (8 bits)	D7–3	**–**	reserved	–				–	–	0 when being read.
		D2–0	**OSC3DIV [2:0]**	OSC3 clock divider selection	OSC3DIV[2:0]		Divider		0x0	R/W	
					0x7		OSC3•1/1				
					0x6		OSC3•1/1				
					0x5		OSC3•1/32				
					0x4		OSC3•1/16				
					0x3		OSC3•1/8				
					0x2		OSC3•1/4				
					0x1		OSC3•1/2				
					0x0		OSC3•1/1				
CMU_CLK Select Register (CMU_CMUCLK)	0x4905 (8 bits)	D7–4	**–**	reserved	–				–	–	0 when being read.
		D3–0	**CMU_ CLKSEL[3:0]**	CMU_CLK selection	CMU_CLKSEL[3:0]		Clock source		0x0	R/W	
					0xf–0xa		reserved				
					0x9		OSC3•1/32				
					0x8		OSC3•1/16				
					0x7		OSC3•1/8				
					0x6		OSC3•1/4				
					0x5		OSC3•1/2				
					0x4		OSC3•1/1				
					0x3		reserved				
					0x2		BCLK				
					0x1		OSC1				
					0x0		OSC3				
Gated Clock Control 0 Register (CMU _GATEDCLK0)	0x4906 (8 bits)	D7	**ROMC _CLK_EN**	ROMC clock control (in HALT mode)	1	On	0	Off	1	R/W	
		D6–1	**–**	reserved	–				–	–	0 when being read.
		D0	**PCLK_EN**	Core peripheral clock control	1	On	0	Off	1	R/W	
Gated Clock Control 1 Register (CMU _GATEDCLK1)	0x4907 (8 bits)	D7	**SRAMC _CLK_EN**	SRAMC clock control (in HALT mode)	1	On	0	Off	1	R/W	
		D6–4	**–**	reserved	–				–	–	0 when being read.
		D3	**T8OSC1 _CLK_EN**	8-bit OSC1 timer clock control	1	On	0	Off	1	R/W	
		D2	**–**	reserved	–				–	–	0 when being read.
		D1	**PT8_CLK _EN**	8-bit programmable timer clock control	1	On	0	Off	1	R/W	
		D0	**MFT_CLK _EN**	Multi-function timer clock control					1	R/W	
Gated Clock Control 2 Register (CMU _GATEDCLK2)	0x4908 (8 bits)	D7–6	**–**	reserved	–				–	–	0 when being read.
		D5	**SPI_CLK _EN**	SPI CH.1 module clock control	1	On	0	Off	1	R/W	
		D4	**REMC_CLK _EN**	REMC module clock control					1	R/W	
		D3	**ADC_CLK _EN**	ADC module clock control					1	R/W	
		D2	**WDT_CLK _EN**	WDT module clock control					1	R/W	
		D1	**PORT_CLK _EN**	I/O port module clock control					1	R/W	
		D0	**RTC_SAPB _CLK_EN**	RTC SAPB I/F clock control					1	R/W	
CMU Write Protect Register (CMU _PROTECT)	0x4920 (8 bits)	D7–0	**CLGP[7:0]**	CMU register protect flag	Writing 10010110 (0x96) removes the write protection of the CMU registers (0x4900–0x4908). Writing another value set the write protection.				0x0	R/W	

0x4a00–0x4a04 8-bit OSC1 Timer CH.0

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
8-bit OSC1 Timer CH.0 Control Register (T8OSC1_ CTL0)	**0x4a00** (8 bits)	D7–5	**–**	reserved	–				–	–	0 when being read.
		D4	**T8OSC1RST**	Timer reset	1	Reset	0	Ignored	0	W	
		D3–2	**–**	reserved	–				–	–	
		D1	**T8OSC1RMD**	Count mode select	1	One shot	0	Repeat	0	R/W	
		D0	**T8OSC1RUN**	Timer run/stop control	1	Run	0	Stop	0	R/W	
8-bit OSC1 Timer CH.0 Counter Data Register (T8OSC1_ CNT0)	**0x4a01** (8 bits)	D7–0	**T8OSC1CNT [7:0]**	Timer counter data T8OSC1CNT7 = MSB T8OSC1CNT0 = LSB	0x0 to 0xff				0x0	R	
8-bit OSC1 Timer CH.0 Compare Data Register (T8OSC1_ CMP0)	**0x4a02** (8 bits)	D7–0	**T8OSC1CMP [7:0]**	Compare data T8OSC1CMP7 = MSB T8OSC1CMP0 = LSB	0x0 to 0xff				0x0	R/W	
8-bit OSC1 Timer CH.0 Interrupt Mask Register (T8OSC1_ IMSK0)	**0x4a03** (8 bits)	D7–1	**–**	reserved	–				–	–	0 when being read.
		D0	**T8OSC1IE**	8-bit OSC1 timer interrupt enable	1	Enable	0	Disable	0	R/W	
8-bit OSC1 Timer CH.0 Interrupt Flag Register (T8OSC1_ IFLG0)	**0x4a04** (8 bits)	D7–1	**–**	reserved	–				–	–	0 when being read.
		D0	**T8OSC1IF**	8-bit OSC1 timer interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.

0x4b00–0x4b04 8-bit OSC1 Timer CH.1

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
8-bit OSC1 Timer CH.1 Control Register (T8OSC1_ CTL1)	**0x4b00** (8 bits)	D7–5	–	reserved	–				–	–	0 when being read.
		D4	**T8OSC1RST**	Timer reset	1	Reset	0	Ignored	0	W	
		D3–2	–	reserved	–				–	–	
		D1	**T8OSC1RMD**	Count mode select	1	One shot	0	Repeat	0	R/W	
		D0	**T8OSC1RUN**	Timer run/stop control	1	Run	0	Stop	0	R/W	
8-bit OSC1 Timer CH.1 Counter Data Register (T8OSC1_ CNT1)	**0x4b01** (8 bits)	D7–0	**T8OSC1CNT [7:0]**	Timer counter data T8OSC1CNT7 = MSB T8OSC1CNT0 = LSB	0x0 to 0xff				0x0	R	
8-bit OSC1 Timer CH.1 Compare Data Register (T8OSC1_ CMP1)	**0x4b02** (8 bits)	D7–0	**T8OSC1CMP [7:0]**	Compare data T8OSC1CMP7 = MSB T8OSC1CMP0 = LSB	0x0 to 0xff				0x0	R/W	
8-bit OSC1 Timer CH.1 Interrupt Mask Register (T8OSC1_ IMSK1)	**0x4b03** (8 bits)	D7–1	–	reserved	–				–	–	0 when being read.
		D0	**T8OSC1IE**	8-bit OSC1 timer interrupt enable	1	Enable	0	Disable	0	R/W	
8-bit OSC1 Timer CH.1 Interrupt Flag Register (T8OSC1_ IFLG1)	**0x4b04** (8 bits)	D7–1	–	reserved	–				–	–	0 when being read.
		D0	**T8OSC1IF**	8-bit OSC1 timer interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.

0x5018 SRAM Controller

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
RTC Wait Control Register (RTC_WAIT)	**0x5018** (16 bits)	D15–3	**–**	reserved	–		–	–	0 when being read.
		D2–0	**RTC_WAIT [2:0]**	RTC access wait cycle setup	RTC_WAIT[2:0]	Wait cycle	0x7	R/W	
					0x7 : 0x0	7 cycles : 0 cycles			

0x5200–0x527e Multi-Function Timer

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
MFT Counter Data Register (MFT_TC)	**0x5200** (16 bits)	D15–0	**TC[15:0]**	Counter data TC15 = MSB TC0 = LSB	0x0 to 0xffff				0x0	R/W	
MFT Period Data Register (MFT_PRD)	**0x5202** (16 bits)	D15–0	**PRD[15:0]**	Period data PRD15 = MSB PRD0 = LSB	0x0 to 0xffff				0x0	R/W	
MFT Compare Data Register (MFT_CMP)	**0x5204** (16 bits)	D15–0	**CR[15:0]**	Compare data CR15 = MSB CR0 = LSB	0x0 to 0xffff				0x0	R/W	
MFT Control Register (MFT_CTL)	**0x5206** (16 bits)	D15–12	–	reserved	–				–	–	0 when being read.
		D11	**TPSON**	Clock control	1	On	0	Off	0	R/W	
		D10–8	**TPS[2:0]**	Clock division ratio selection (Prescaler output clock)	TPS[2:0]		Count clock		0x0	R/W	
					0x7		MFT_CLK•1/128				
					0x6		MFT_CLK•1/64				
					0x5		MFT_CLK•1/32				
					0x4		MFT_CLK•1/16				
					0x3		MFT_CLK•1/8				
					0x2		MFT_CLK•1/4				
					0x1		MFT_CLK•1/2				
					0x0		MFT_CLK•1/1				
		D7	**BUFEN**	Comparison/period buffer enable	1	Enable	0	Disable	0	R/W	
		D6	–	reserved	–				–	–	0 when being read.
		D5	**TCKS**	Input clock selection	1	External	0	Internal	0	R/W	
		D4–2	–	reserved	–				–	–	0 when being read.
		D1	**PRST**	Timer reset	1	Reset	0	Ignored	0	W	
		D0	**TMEN**	Timer run/stop control	1	Run	0	Stop	0	R/W	
MFT Input/ Output Control Register (MFT_IOCTL)	**0x521e** (16 bits)	D15–7	–	reserved	–				–	–	0 when being read.
		D6	**APIL**	ADC protection input selection	1	Lower limit	0	Upper limit	0	R/W	
		D5	**PPIL**	Port protection input level selection	1	Low level	0	High level	0	R/W	
		D4	**IQA**	Port protection input noise filter	1	12 clocks	0	6 clocks	0	R/W	
		D3	–	reserved	–				–	–	0 when being read.
		D2	**PTM**	Clock output enable	1	Enable	0	Disable	0	R/W	
		D1	**INITOL**	Initial output level	1	High	0	Low	0	R/W	
		D0	**OINV**	Inverse output	1	Invert	0	Normal	0	R/W	
MFT Interrupt Enable Register (MFT_IE)	**0x5230** (16 bits)	D15–4	–	reserved	–				–	–	0 when being read.
		D3	**CMPIE**	Compare-match interrupt enable	1	Enable	0	Disable	0	R/W	
		D2	**PRDIE**	Period-match interrupt enable	1	Enable	0	Disable	0	R/W	
		D1	**APIE**	ADC protection interrupt enable	1	Enable	0	Disable	0	R/W	
		D0	**PPIE**	Port protection interrupt enable	1	Enable	0	Disable	0	R/W	
MFT Interrupt Flag Register (MFT_IF)	**0x5238** (16 bits)	D15–4	–	reserved	–				–	–	0 when being read.
		D3	**CMPIF**	Compare-match interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D2	**PRDIF**	Period-match interrupt flag					0	R/W	
		D1	**APIF**	ADC protection interrupt flag					0	R/W	
		D0	**PPIF**	Port protection interrupt flag					0	R/W	
MFT Test Register (MFT_TST)	**0x527e** (16 bits)	D15–7	–	reserved	–				–	–	0 when being read.
		D6	**DBGMD**	MFT operation in debug mode	1	Stop	0	Run	0	R/W	
		D5–3	–	reserved	–				–	–	0 when being read.
		D2–0	–	reserved	0x7				0x7	–	Fix at 0x7.

0x5400–0x5410 Remote Controller

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
REMC Prescaler Control Register (REMC_PSC)	**0x5400** (16 bits)	D15–3	**–**	reserved	–		–	–	0 when being read.
		D2	**REMPSON**	REMC prescaler control	1 On	0 Off	0	R/W	
		D1–0	**REMPSDIV [1:0]**	REMC prescaler division ratio select (Prescaler output clock)	REMPSDIV[1:0]	Clock	0x0	R/W	
					0x3	PCLK•1/32			
					0x2	PCLK•1/24			
					0x1	PCLK•1/16			
					0x0	PCLK•1/12			
REMC Configuration Register (REMC_CFG)	**0x5404** (16 bits)	D15–8	**–**	reserved	–		–	–	0 when being read.
		D7	**MODE**	REMC mode select	1 Receive	0 Transmit	0	R/W	
		D6	**RIF**	Rising edge interrupt flag	1 Cause of interrupt occurred	0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D5	**FIF**	Falling edge interrupt flag			0	R/W	
		D4	**UIF**	Underflow interrupt flag			0	R/W	
		D3	**–**	reserved	–		–	–	0 when being read.
		D2	**RIE**	Rising edge interrupt enable	1 Enable	0 Disable	0	R/W	
		D1	**FIE**	Falling edge interrupt enable	1 Enable	0 Disable	0	R/W	
		D0	**UIE**	Underflow interrupt enable	1 Enable	0 Disable	0	R/W	
REMC Control Register (REMC_CTL)	**0x5408** (16 bits)	D15–1	**–**	reserved	–		–	–	0 when being read.
		D0	**REMST**	REMC start/stop control	1 Start	0 Stop	0	R/W	
REMC Carrier Load Register (REMC_CARL)	**0x540c** (16 bits)	D15–8	**CLDH[7:0]**	REMC carrier high width setup	0x0 to 0xff		0x0	R/W	
		D7–0	**CLDL[7:0]**	REMC carrier low width setup	0x0 to 0xff		0x0	R/W	
REMC Envelope Load Register (REMC_ENVL)	**0x540e** (16 bits)	D15–0	**ELD[15:0]**	Envelope counter preset data	0x0 to 0xffff		0x0	R/W	
REMC Envelope Capture Register (REMC_ENVC)	**0x5410** (16 bits)	D15–0	**ECP[15:0]**	Receive envelope pulse width	0x0 to 0xffff		0x0	R	

0x5520–0x555a A/D Converter

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
A/D Clock Control Register (AD_CLKCTL)	**0x5520** (16 bits)	D15–4	**–**	reserved	–				–	–	0 when being read.
		D3	**PSONAD**	A/D converter clock control	1	On	0	Off	0	R/W	
		D2–0	**PSAD[2:0]**	A/D converter clock division ratio selection	PSAD[2:0]		A/D clock		0x0	R/W	
					0x7		PCLK•1/256				
					0x6		PCLK•1/128				
					0x5		PCLK•1/64				
					0x4		PCLK•1/32				
					0x3		PCLK•1/16				
					0x2		PCLK•1/8				
					0x1		PCLK•1/4				
					0x0		PCLK•1/2				
A/D Conversion Result Register (AD_DAT)	**0x5540** (16 bits)	D15–10	**–**	reserved	–				–	–	0 when being read.
		D9–0	**ADD[9:0]**	A/D converted data ADD9 = MSB ADD0 = LSB	0x0 to 0x3ff				0x0	R	
A/D Trigger/ Channel Select Register (AD_TRIG_CH)	**0x5542** (16 bits)	D15–14	**–**	reserved	–				–	–	0 when being read.
		D13–11	**CE[2:0]**	A/D converter end channel selection	0 to 3				0x0	R/W	
		D10–8	**CS[2:0]**	A/D converter start channel selection	0 to 3				0x0	R/W	
		D7–6	**–**	reserved	–				–	–	0 when being read.
		D5	**MS**	A/D conversion mode selection	1	Continuous	0	Normal	0	R/W	
		D4–3	**TS[1:0]**	A/D conversion trigger selection	TS[1:0]		Trigger		0x0	R/W	
					0x3		#ADTRG pin				
					0x2		PT8 CH.0				
					0x1		MFT				
					0x0		Software				
		D2–0	**CH[2:0]**	A/D conversion channel status	0 to 3				0x0	R	
A/D Control/ Status Register (AD_CTL)	**0x5544** (16 bits)	D15	**ADCMPE**	Upper/lower-limit comparison enable	1	Enable	0	Disable	0	R/W	Can be used when ADCADV = 1.
		D14–12	**ADCMP[2:0]**	Upper/lower-limit comparison channel selection	0 to 3				0x0	R/W	
		D11	**ADUPRST**	Upper-limit comparison status	1	Out of range	0	Within range	0	R	
		D10	**ADLWRST**	Lower-limit comparison status	1	Out of range	0	Within range	0	R	
		D9–8	**ST[1:0]**	Input signal sampling time setup	ST[1:0]		Sampling time		0x3	R/W	Use with 9 clocks.
					0x3		9 clocks				
					0x2		7 clocks				
					0x1		5 clocks				
					0x0		3 clocks				
		D7	**–**	reserved	–				–	–	0 when being read.
		D6	**INTMODE**	Interrupt signal mode	1	Complete only	0	OR	0	R/W	Can be used when ADCADV = 1.
		D5	**CMPINTEN**	Out-of-range int. enable	1	Enable	0	Disable	0	R/W	
		D4	**CNVINTEN**	Conversion-complete int. enable	1	Enable	0	Disable	1	R/W	Can be changed when ADCADV = 1.
		D3	**ADF**	Conversion-complete flag	1	Completed	0	Run/Standby	0	R	Reset when ADD is read.
		D2	**ADE**	A/D enable	1	Enable	0	Disable	0	R/W	
		D1	**ADST**	A/D conversion control/status	1	Start/Run	0	Stop	0	R/W	
		D0	**OWE**	Overwrite error flag	1	Error	0	Normal	0	R/W	Reset by writing 0.
A/D Channel Status Flag Register (AD_CH_STAT)	**0x5546** (16 bits)	D15–12	**–**	reserved	–				–	–	0 when being read.
		D11	**OWE3**	CH.3 overwrite error flag	1	Error	0	Normal	0	R/W	Can be used when ADCADV = 1. Reset by writing 0.
		D10	**OWE2**	CH.2 overwrite error flag					0	R/W	
		D9	**OWE1**	CH.1 overwrite error flag					0	R/W	
		D8	**OWE0**	CH.0 overwrite error flag					0	R/W	
		D7–4	**–**	reserved	–				–	–	0 when being read.
		D3	**ADF3**	CH.3 conversion-complete flag	1	Completed	0	Run/Standby	0	R	Can be used when ADCADV = 1. Reset when AD*x*BUF [9:0] is read.
		D2	**ADF2**	CH.2 conversion-complete flag					0	R	
		D1	**ADF1**	CH.1 conversion-complete flag					0	R	
		D0	**ADF0**	CH.0 conversion-complete flag					0	R	
A/D CH.*x* Conversion Result Buffer Register (AD_CH*x*_BUF)	**0x5548** **|** **0x554e** (16 bits)	D15–10	**–**	reserved	–				–	–	0 when being read.
		D9–0	**AD*x*BUF [9:0]**	A/D CH.*x* converted data AD*x*BUF9 = MSB AD*x*BUF0 = LSB	0x0 to 0x3ff				0x0	R	Can be used when ADCADV = 1.
A/D Upper Limit Value Register (AD_UPPER)	**0x5558** (16 bits)	D15–10	**–**	reserved	–				–	–	0 when being read.
		D9–0	**ADUPR[9:0]**	A/D conversion upper limit value ADUPR9 = MSB ADUPR0 = LSB	0x0 to 0x3ff				0x0	R/W	Can be used when ADCADV = 1.
A/D Lower Limit Value Register (AD_LOWER)	**0x555a** (16 bits)	D15–10	**–**	reserved	–				–	–	0 when being read.
		D9–0	**ADLWR [9:0]**	A/D conversion lower limit value ADLWR9 = MSB ADLWR0 = LSB	0x0 to 0x3ff				0x0	R/W	Can be used when ADCADV = 1.

0x555c–0x555e A/D Converter

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
A/D Conversion Complete Interrupt Mask Register (AD_INTMASK)	**0x555c** (16 bits)	D15–4	**–**	reserved	–	–	–	0 when being read.
		D3	**INTMASK3**	CH.3 conversion-complete int. mask	1 Interrupt enabled 0 Interrupt mask	1	R/W	Can be used when ADCADV = 1.
		D2	**INTMASK2**	CH.2 conversion-complete int. mask		1	R/W	
		D1	**INTMASK1**	CH.1 conversion-complete int. mask		1	R/W	
		D0	**INTMASK0**	CH.0 conversion-complete int. mask		1	R/W	
A/D Converter Mode Select/ Internal Status Register (AD_ADVMODE)	**0x555e** (16 bits)	D15–9	**–**	reserved	–	–	–	0 when being read.
		D8	**ADCADV**	Standard/advanced mode selection	1 Advanced 0 Standard	0	R/W	
		D7–6	**–**	reserved	–	–	–	0 when being read.
		D5–4	**ISTATE[1:0]**	Internal status	ISTATE[1:0] Status 0x3 Converting 0x2 reserved 0x1 Sampling 0x0 Idle	0x0	R	
		D3–0	**ICOUNTER [3:0]**	Internal counter value	0 to 15	0x0	R	

0x5660–0x566c Watchdog Timer

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
WDT Write Protect Register (WD_WP)	**0x5660** (16 bits)	D15–0	**WDPTC [15:0]**	WDT register write protect flag	Writing 0x96 removes the write protection of the WD_EN, WD_CMP_L, and WD_CMP_H registers (0x5662–0x5666). Writing another value set the write protection.				X	W	0 when being read.
WDT Enable and Setup Register (WD_EN)	**0x5662** (16 bits)	D15–7	**–**	reserved	–				–	–	0 when being read.
		D6	**CLKSEL**	WDT input clock select	1	External clk	0	Internal clk	0	R/W	
		D5	**CLKEN**	WDT clock output control	1	On	0	Off	0	R/W	
		D4	**RUNSTP**	WDT Run/Stop control	1	Run	0	Stop	0	R/W	
		D3–2	**–**	reserved	–				–	–	0 when being read.
		D1	**NMIEN**	WDT NMI enable	1	Enable	0	Disable	0	R/W	
		D0	**RESEN**	WDT RESET enable	1	Enable	0	Disable	0	R/W	
WDT Comparison Data L Register (WD_CMP_L)	**0x5664** (16 bits)	D15–0	**CMPDT [15:0]**	WDT comparison data CMPDT0 = LSB	0x0 to 0x3fffffff (low-order 16 bits)				0x0	R/W	
WDT Comparison Data H Register (WD_CMP_H)	**0x5666** (16 bits)	D15–14	**–**	reserved	–				–	–	0 when being read.
		D13-0	**CMPDT [29:16]**	WDT comparison data CMPDT29 = MSB	0x0 to 0x3fffffff (high-order 14 bits)				0x0	R/W	
WDT Count Data L Register (WD_CNT_L)	**0x5668** (16 bits)	D15–0	**CTRDT [15:0]**	WDT counter data CTRDT0 = LSB	0x0 to 0x3fffffff (low-order 16 bits)				X	R	
WDT Count Data H Register (WD_CNT_H)	**0x566a** (16 bits)	D15–14	**–**	reserved	–				–	–	0 when being read.
		D13-0	**CTRDT [29:16]**	WDT counter data CTRDT29 = MSB	0x0 to 0x3fffffff (high-order 14 bits)				X	R	
WDT Control Register (WD_CTL)	**0x566c** (16 bits)	D15–1	**–**	reserved	–				–	–	0 when being read.
		D0	**WDRESEN**	WDT reset	1	Reset	0	ignored	0	W	

0x5700–0x5708 Extended SPI

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
SPI CH.1 Status Register (SPI_ST1)	**0x5700** (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.1 Transmit Data Register (SPI_TXD1)	**0x5702** (16 bits)	D15–8	**–**	reserved	–				–	–	0 when being read.
		D7–0	**SPTDB[7:0]**	SPI CH.1 transmit data buffer SPTDB7 = MSB SPTDB0 = LSB	0x0 to 0xff				0x0	R/W	
SPI CH.1 Receive Data Register (SPI_RXD1)	**0x5704** (16 bits)	D15–8	**–**	reserved	–				–	–	0 when being read.
		D7–0	**SPRDB[7:0]**	SPI CH.1 receive data buffer SPRDB7 = MSB SPRDB0 = LSB	0x0 to 0xff				0x0	R	
SPI CH.1 Control Register (SPI_CTL1)	**0x5706** (16 bits)	D15–6	**–**	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 CH.1 enable	1	Enable	0	Disable	0	R/W	
SPI CH.1 Clock Control Register (SPI_CLK1)	**0x5708** (16 bits)	D15–5	**–**	reserved	–				–	–	0 when being read.
		D4	**SPI_CKE**	SPI CH.1 clock enable	1	Enable	0	Disable	0	R/W	
		D3–0	**SPI_CLK [3:0]**	SPI CH.1 clock division ratio selection (Prescaler output clock)	SPI_CLK[3:0]		Clock		0x0	R/W	
					0xf–0xd		reserved				
					0xc		PCLK•1/4096				
					0xb		PCLK•1/2048				
					0xa		PCLK•1/1024				
					0x9		PCLK•1/512				
					0x8		PCLK•1/256				
					0x7		PCLK•1/128				
					0x6		PCLK•1/64				
					0x5		PCLK•1/32				
					0x4		PCLK•1/16				
					0x3		PCLK•1/8				
					0x2		PCLK•1/4				
					0x1		PCLK•1/2				
					0x0		PCLK•1/1				

0x5804–0x5816 ROM Controller

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
ROMC Wait Register (ROMC_WAIT)	**0x5804** (16 bits)	D15–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**ROM_WAIT [2:0]**	ROM read access wait cycle setup	ROM_WAIT[2:0] Wait cycle 0x7 7 cycles : : 0x0 0 cycles	0x7	R/W	
ROMC Protect Register (ROMC_PRT)	**0x5810** (16 bits)	D15–8	**–**	reserved	–	–	–	0 when being read.
		D7–0	**ROMC_PRT [7:0]**	ROMC register protect flag	Writing 10010110 (0x96) removes the write protection of the Trap Table Base Registers (0x5814–0x5816). Writing another value set the write protection.	0x0	R/W	
Vector Table Base Register 0 (TTBR_LOW)	**0x5814** (16 bits)	D15–8	**TTBR[15:8]**	Vector table base address A[15:8]	0x0–0xff	0x0	R/W	
		D7–0	**TTBR[7:0]**	Vector table base address A[7:0] (fixed at 0)	0x0	0x0	R	
Vector Table Base Register 1 (TTBR_HIGH)	**0x5816** (16 bits)	D15–8	**–**	reserved	–	–	–	0 when being read.
		D7–0	**TTBR[23:16]**	Vector table base address A[23:16]	0x0–0xff	0x2	R/W	

0xffff90 S1C17 Core I/O

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	0x0	0x0	R	

Appendix B Multiplier/Divider

B.1 Outline

The S1C17002 has an embedded coprocessor that provides a signed/unsigned 16 × 16-bit multiplication function, a signed/unsigned 16 ÷ 16-bit division function, and a signed 16 × 16-bit + 32-bit MAC (Multiplication and Accumulation) function with overflow detection.
This section explains how to use these functions.

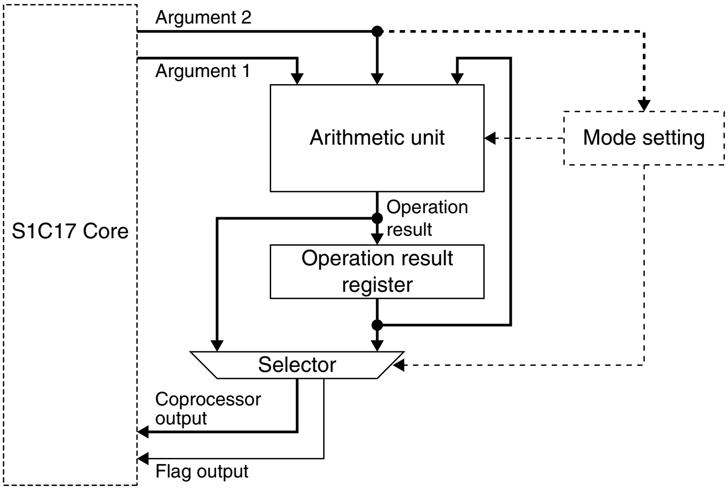


Figure B.1.1 Multiplier/Divider Block Diagram

Table B.1.1 Number of Operation Cycles

Operation	Number of cycles
Multiplication	1 cycle
MAC operation	1 cycle
Division	17 to 20 cycles

B.2 Operation Mode and Output Mode

The Multiplier/divider operates according to the operation mode specified by the application program. As listed in Table B.2.1, the multiplier/divider supports 9 operations.

The multiplication, division and MAC results are 32-bit data, therefore, the S1C17 Core cannot read them in one access. 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)

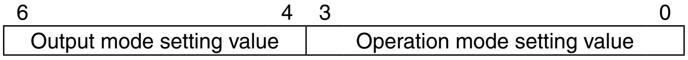


Figure B.2.1 Mode Setting Register

Table B.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

B.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.

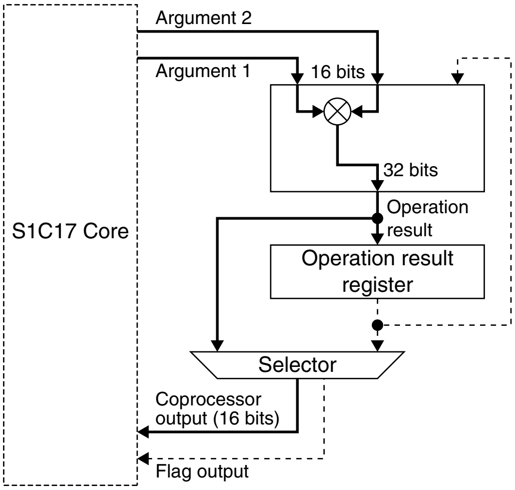


Figure B.3.1 Data Path in Multiplication Mode

Table B.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.
```

B.4 Division

The division function performs “A (16 bits) = B (16 bits) ÷ C (16 bits), D (16 bits) = residue.”

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.

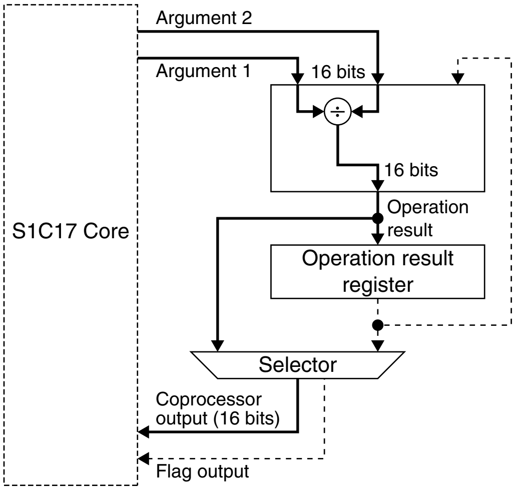


Table B.4.1 Data Path in Division Mode

Table B.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.
```

B.5 MAC

The MAC 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.

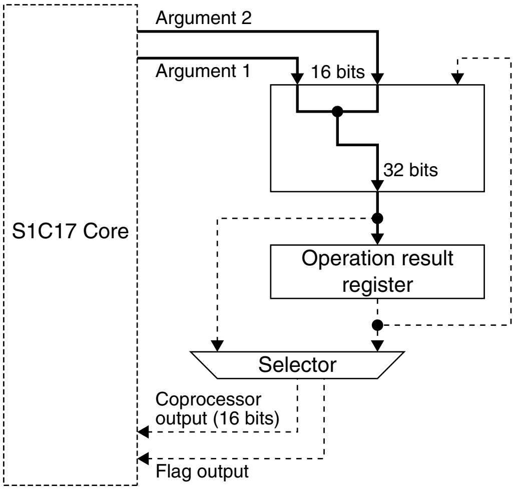


Figure B.5.1 Data Path in Initialize Mode

Table B.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.

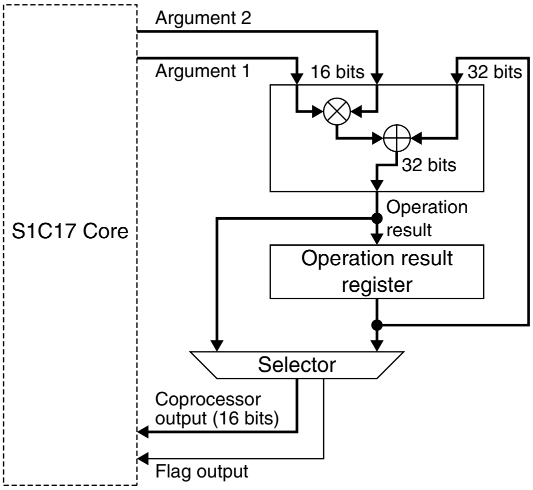


Figure B.5.2 Data Path in MAC Mode

Table B.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 B.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.

B.6 Reading 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.

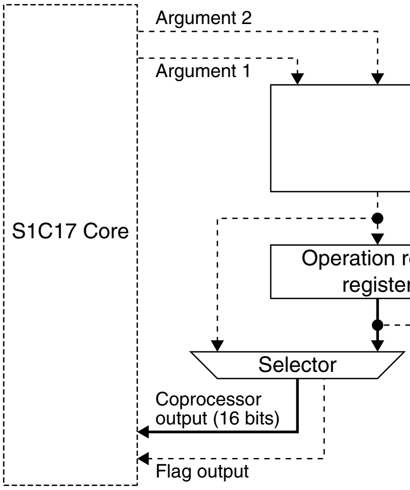


Figure B.6.1 Data Path in Operation Result Read Mode

Table B.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

Appendix C Power Saving

Current consumption depends, to a large degree, on the CPU operating mode, operating clock frequency, and the peripheral circuits to be activated. This chapter summarizes the control to save power.
Figure C.1 shows the S1C17002 clock system.

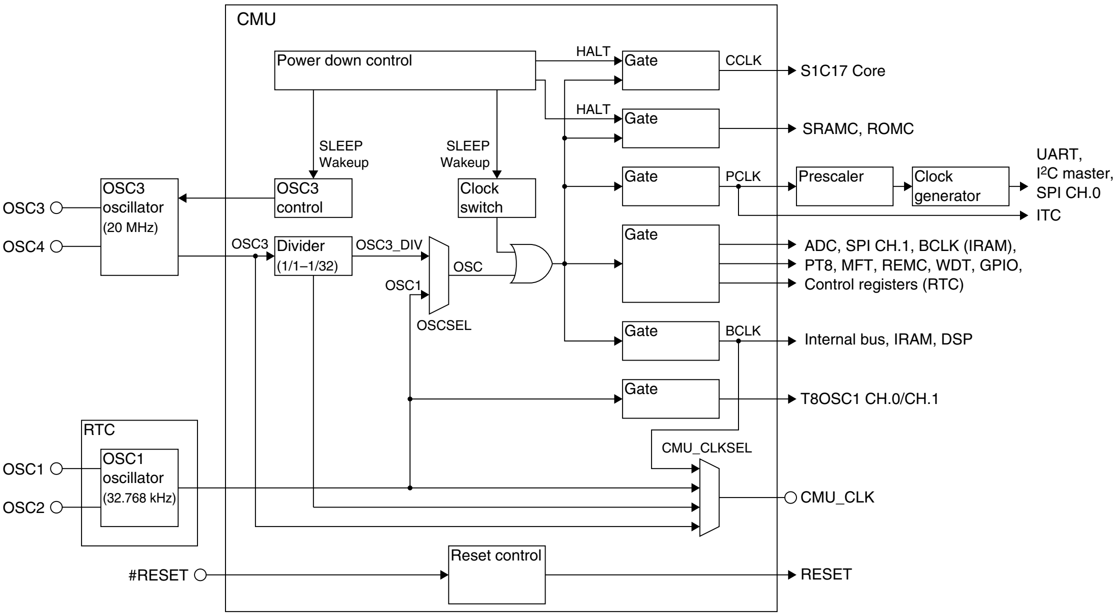


Figure C.1 Clock System

The following shows the clock systems that can be controlled with software and power saving control methods. For details of control registers and control methods, see the chapter for each module.

System sleep (disabling all clocks)

- Executing the `slp` instruction
  Execute the `slp` instruction if all of the system can be stopped. In SLEEP mode, the CPU stops operating and the CMU stops supplying a clock to each functional module (see Section II.2.6 for more information). Therefore, all peripheral circuits (except the OSC1 oscillator circuit and RTC) stop operating.
  The OSC3 oscillator stops oscillating in SLEEP mode if OSC3OFF (D7/CMU_SYSCLKCTL register) is set to 1.
  The CPU is reawaken from SLEEP mode (when WAKEUPWT (D4/CMU_SYSCLKCTL register) = 1) by initial reset, RTC interrupt (level triggered), or other interrupt from an external device (port input interrupt with level triggered).

  * **OSC3OFF**: OSC3 Disable During SLEEP Bit in the System Clock Control (CMU_SYSCLKCTL) Register (D7/0x4900)
  * **WAKEUPWT**: Wakeup-Wait Function Enable Bit in the System Clock Control (CMU_SYSCLKCTL) Register (D4/0x4900)

System clock

- Selecting the clock source (CMU module)
  Either OSC3 or OSC1 can be selected as the system clock source. If the application can process the task with a low-speed clock, select OSC1 as the system clock source to reduce current consumption.

- Disabling the OSC3 oscillator circuit (CMU module)
  Enable the oscillator configured as the system clock source and disable another oscillator if possible. Using OSC1 for the system clock and disabling the OSC3 oscillator circuit achieves more reduction of current consumed.

- Selecting a low clock gear (CMU module)
  The CMU module provides clock gears to set the system clock speed to 1/1 to 1/32 of the OSC3 clock. By running the S1C17002 with the lowest speed required for the application's task, current consumption can be reduced.

CPU clock (CCLK)

- Executing the `halt` instruction
  Execute the `halt` instruction if there is no task to be processed by the CPU such as when the display on the LCD is only required or when the CPU is waiting an interrupt. Although the CPU enters HALT mode and stops operating, the peripheral modules keep the status when the `halt` instruction is executed. So the LCD driver and the peripheral modules used to generate an interrupt can be made to be run. Power saving effect will be enhanced by disabling the unnecessary oscillator and peripheral modules before executing the `halt` instruction. The CPU reactivates from HALT mode by an interrupt from the ports or peripheral modules that are being operated in HALT mode.

Peripheral clocks

- Disabling peripheral clocks (CMU, CLG, and PSC modules)
  The peripheral clock supply can be disabled if the peripheral modules listed below can be placed in standby state.

  - SRAMC clock in HALT mode (CMU)
  - Core peripheral clocks for PSC, CLG, ITC (CMU)
  - PT8 clock (CMU)
  - MFT clock (CMU)
  - T8OSC1 clock (CMU)
  - SPI CH.1 clock (CMU)
  - REMC clock (CMU)
  - ADC clock (CMU)

- WDT clock (CMU)
- I/O port clock (CMU)
- RTC register clock (CMU)
- CLG clocks (PSC)
- UART clocks (CLG)
- SPI CH.0 clock (CLG)
- I^{2}C master/slave clock (CLG)

Table C.1 lists the clock control conditions and how to suspend/resume the CPU operation.

Table C.1 List of Clock Control Conditions

Current consumption	OSC1	OSC3	CPU (CCLK)	Peripherals	CPU suspending method	CPU resuming method
↑ Low	Oscillating	Stop	Stop	Stop	`slp` instruction	1
	Oscillating	Stop	Stop	Stop (only RTC is running)	`slp` instruction	1, 2
	Oscillating (System clock)	Stop	Stop	Stop	`halt` instruction	1, 2
	Oscillating (System clock)	Stop	Stop	Run	`halt` instruction	1, 2, 3
	Oscillating (System clock)	Stop	Run	Run		
	Oscillating	Oscillating (System clock)	Stop	Run	`halt` instruction	1, 2, 3
	Oscillating	Oscillating (System clock)	Run (low gear)	Run		
High ↓	Oscillating	Oscillating (System clock)	Run (OSC3•1/1)	Run		

Clearing HALT and SLEEP modes (CPU resuming methods)

1. Resuming by a port input interrupt (level triggered) or #RESET
   The CPU resumes operating by occurrence of a cause of port input interrupt (level triggered), #RESET, or a debug interrupt (issuing an ICD forced break). See Section II.2.8 for details.

2. Resuming by the RTC
   The CPU resumes operating by occurrence of a cause of RTC interrupt (level triggered). See Sections II.2.8 and II.5.4 for details.

3. Resuming by a peripheral
   The CPU resumes operating by occurrence of a cause of interrupt in a peripheral whose interrupt is enabled by the interrupt controller. If the IE flag in the CPU has been set to 0, the CPU does not accept the interrupt request and starts executing the instructions that follow the `halt` instruction. If the IE flag has been set to 1, the CPU executes the interrupt handler.

Appendix D Developing S1C17002 Mask ROM Code

(1) Use the S1C17511 Flash microcomputer to develop mask ROM code for the S1C17002.

(2) The ROM data file format to submit to SEIKO EPSON should be "*file*.PAn" (output from winmdc). Before submitting the file, perform final verification of the user ROM data using "*file*.psa" (output from sconv32).

(3) Specify the following values as the arguments for the S1C17002 when moto2ff is executed.
- Data start address = 0x20000
- Data block size = 512 × 16 bits

(4) Take the differences listed in the table below into consideration and perform operation check using the S1C17511.
- The IRAM size is different.
- The S1C17511 IRAM2 cannot be used as the S1C17002 IRAM as their addresses are different.

Table D.1 Functional Differences Between S1C17511 and S1C17002

Circuit/function	S1C17511	S1C17002
FLASH	128KB	–
Mask ROM	–	128KB
IRAM	4KB	8KB
IRAM2	2KB	–
Maximum operating frequency	48 MHz (0 to 70°C) 40 MHz (-40 to 85°C)	20 MHz (-40 to 85°C)
OSC3 oscillator	Crystal/CR	←
OSC1 oscillator	Crystal	←
Operating clock input (OSC3)	Supported	←
Operating clock input (OSC1)	–	Supported
I/O port	91 (depending on the package)	30
Input port	8	4
External port interrupt	8	←
SPI (master/slave)	2ch.	←
I^{2}C (master/slave)	2ch.	←
UART (with IrDA1.0)	1ch.	←
CLG 8-bit timer (T8) for SPI	1ch.	←
CLG 8-bit timer (T8) for I^{2}C	1ch.	←
CLG 8-bit timer (T8F) for UART	1ch.	←
8-bit timer (PT8)	4ch.	←
CLG 16-bit timer (T16) for UART	1ch.	←
Multi-function timer (MFT)	1ch.	←
RTC	Available	←
Watchdog timer (WDT)	Available	←
8-bit OSC1 timer (T8OSC1)	2ch.	←
Number of multiplier execution cycles	1 cycle/2 cycles	1 cycle
Number of divider execution cycles	17 to 20 cycles	←
A/D converter	8ch.	4ch.
External bus controller	Available	–
USB function controller	Available	–
Boot address	0x20000	←
TTBR	Available	←
Power supply voltage	V_{DD} = 3.0 to 3.6 V AV_{DD} = 2.7 to 3.6 V $RTCV_{DD}$ = 3.0 to 3.6 V	HV_{DD} = 1.65 to 3.6 V LV_{DD} = 1.65 to 1.95 V AV_{DD} = 2.7 to 3.6 V (1.65 to 3.6 V*) * There are requirements for use. See "I.4.4 Power Supply for Analog Circuits (AV_{DD})."
Maximum external pin drive capability	Different in each pin. See "I.3.4 Input/Output Cells and Input/Output Characteristics."	
Package	TQFP14-100pin TQFP15-128pin PFBGA144	WCSP 48pin (3.124 × 3.124 mm, 0.4 mm pitch) TQFP12-64pin Bare chip

Revision History

Code No.	Page	Contents
411554400	All	New establishment
411554402	I-5-5	CPU: List of S1C17 Core Instructions (Table I.5.3.1) (Old) ipa.d (New) jpa.d
	I-7-6	External clock input characteristics: Input rise/fall times (Old) 5ns (OSC1), 5ns (OSC3) (New) 200ns (OSC1), 10ns (OSC3)
	II-1-1	Clock system diagram Modified Figure II.1.1
	V-1-6	UART: Data reception control (Old) (2) RDRY = 1, RD2B = 0 ... This clears the data inside the buffer and resets the RDRY flag. ... (3) RDRY = 1, RD2B = 1 ... The receive data buffer ... and resetting the RD2B flag. ... Even when the receive data buffer is full, ... and the new data will overwrite the shift register data. (New) (2) RDRY = 1, RD2B = 0 ... This resets the RDRY flag. ... (3) RDRY = 1, RD2B = 1 ... The receive data buffer outputs the oldest data first. This resets the RD2B flag. ... Even when the receive data buffer is full, ... In this case, the last received data cannot be read.
	V-1-8	UART: Overrun error (Old) If there is no space in the buffer ... an overrun error occurs. (New) If there is no space in the buffer ... the third data in the shift register cannot be transferred to the buffer and an overrun error occurs.
	V-2-1	I2CM: Configuration of the I^{2}C master module (Old) ... It supports standard (100 kbps) and fast (400 kbps) modes, and 7-bit slave addressing. ... (New) ... It supports standard (100 kbps) and fast (400 kbps) modes, and 7-bit/10-bit slave addressing. ...
	V-2-5	I2CM: Sending a slave address (Old) After a START condition has been generated, ... 7-bit slave address can be sent at a time. (New) Once the start condition has been generated, ... sent twice or three times under software control.
	V-2-6	I2CM: Transmit data to specify slave address and data direction Modified Figure V.2.5.2
	V-2-7	I2CM: Data receive control (Old) ... In the ninth clock cycle, ... Set RTACK to 0 to send ACK or set to 1 to send NAK. (New) ... In the ninth clock cycle, ... Note: A receive buffer full interrupt occurs ... this may cause communication to fail.
	V-2-8	I2CM: Disabling data transmission/reception (Old) After data transfer ... cannot be guaranteed if I2CMEN is set to 0 during transmitting/receiving. (New) After the stop condition has been generated, ... and transfer data at that point cannot be guaranteed.
	V-2-8, V-2-9, V-2-10	I2CM: Timing charts Modified Figures V.2.5.5, V.2.5.7, V.2.5.8, and V.2.5.9
	V-3-1	I2CS: Structure of the I^{2}C slave module Modified Figure V.3.1.1
	V-3-2	I2CS: List of I^{2}C slave pins - I2CS_SCL Modified Table V.3.2.1
	V-3-4	I2CS: Bus free request with an input from the #I2CS_RST pin (Old) When this function is enabled, a low pulse (five system clock (PCLK) cycles or more pulse width is required) input to the #I2CS_RST pin sets BFREQ (D4/I2CS_STAT register) to 1. (New) When this function is enabled, a low pulse (One system clock (PCLK) cycle or more pulse width is required. Two PCLK clock cycles or more pulse width is recommended.) input to the #I2CS_RST pin sets BFREQ (D4/I2CS_STAT register) to 1.
		I2CS: Clock stretch function (Old) No description (New) Note that the data setup time ... the I^{2}C slave module operating clock (PCLK) frequency.
	V-3-6	I2CS: Starting data transmission/reception (Old) Both BUSY and SELECTED keep 1 until a STOP condition is detected. (New) 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.
	V-3-7, V-3-15, V-3-26	I2CS: Data transmission (Old) No description (New) When the asynchronous address detection function is used, ... after TXEMP has been set to 1.
	V-3-8, V-3-26	I2CS: Note on data transmission (Old) No description (New) Note: If the I^{2}C slave module has sent back a NAK as the response to the address sent ... 1. The transfer rate is set to 320 kbps or higher. ... 3. The I^{2}C slave module is placed into transfer standby state ... used as the operating clock (PCLK).

REVISION HISTORY

Code No.	Page	Contents
411554402	V-3-10, V-3-11	I2CS: Timing charts Modified Figures V.3.5.5 to V.3.5.8
	V-3-13	I2CS: Bus status interrupt (Old) 7. DA_STOP: set to 1 if a STOP condition is detected ... as the slave device. (New) 7. DA_STOP: set to 1 if a STOP condition or a repeated start condition is detected while this module is selected as the slave device
	V-3-23	I2CS: I^{2}C Slave Status Register (I2CS_STAT) - (D0) DA_STOP: Stop Condition Detect Bit (Old) Indicates that a STOP condition is detected. ... communication process to enter standby state that is ready to detect the next START condition. (New) Indicates that a STOP condition or a repeated start condition is detected. ... module sets DA_STOP to 1. At the same time, it initializes the I^{2}C communication process.
	V-3-24	I2CS: I^{2}C Slave Access Status Register (I2CS_ASTAT) - (D1) SELECTED: I^{2}C Slave Select Status Bit (Old) After SELECTED is set to 1, it is reset to 0 when a STOP condition is detected. (New) After SELECTED is set to 1, it is reset to 0 ... or a repeated start condition is detected.
	V-4-3, V-5-3	SPI/ESPI: SPI clock (Old) In slave mode, the SPI CH.*x* inputs ... differentiated and used to sync with the PCLK clock. (New) In slave mode, the SPI clock is input via the SPICLK*x* pin.
		SPI/ESPI: External clock in SPI slave mode Deleted Figures V.4.3.2 and V.5.3.1
	V-4-6, V-5-6	SPI/ESPI: Data transmit timing chart Deleted Figures V.4.5.1 and V.5.5.1
	V-4-6, V-4-16, V-5-6, V-5-17	SPI/ESPI: Data transmit control (Old) No description (New) Note: When the SPI CH.*x* is used in master mode ... (Added Figures V.4.5.1 and V.5.5.1) ... transmit data bits and the second and following bytes during continuous transfer.
	V-4-7, V-5-7	SPI/ESPI: Data transmit/receive timing chart Modified Figures V.4.5.2 and V.5.5.2
	V-4-7, V-5-7	SPI/ESPI: Disabling data transmission/reception (Old) After data transfer ... SPRBF flag is 0 before data transmission/reception is disabled. When the SPEN bit is set to 0, ... guaranteed if SPEN is set to 0 during transmitting/receiving. (New) After data transfer ... SPBSY flag is 0 before data transmission/reception is disabled. The data being transferred cannot be guaranteed if SPEN is set to 0 during transmitting/receiving.
	V-4-12, V-4-16, V-5-12, V-5-17	SPI/ESPI: SPI Ch.*x* Transmit Data Register (SPI_TXD*x*) (Old) No description (New) Note: Make sure that SPEN is set to 1 before writing data ... to start data transmission/reception.

EPSON

International Sales Operations

AMERICA

EPSON ELECTRONICS AMERICA, INC.

2580 Orchard Parkway,
San Jose, CA 95131, USA
Phone: +1-800-228-3964 Fax: +1-408-922-0238

EUROPE

EPSON EUROPE ELECTRONICS GmbH

Riesstrasse 15, 80992 Munich,
GERMANY
Phone: +49-89-14005-0 Fax: +49-89-14005-110

ASIA

EPSON (CHINA) CO., LTD.

7F, Jinbao Bldg., No.89 Jinbao St.,
Dongcheng District,
Beijing 100005, CHINA
Phone: +86-10-8522-1199 Fax: +86-10-8522-1125

SHANGHAI BRANCH

7F, Block B, Hi-Tech Bldg., 900 Yishan Road,
Shanghai 200233, CHINA
Phone: +86-21-5423-5577 Fax: +86-21-5423-4677

SHENZHEN BRANCH

12F, Dawning Mansion, Keji South 12th Road,
Hi-Tech Park, Shenzhen 518057, CHINA
Phone: +86-755-2699-3828 Fax: +86-755-2699-3838

EPSON HONG KONG LTD.

20/F, Harbour Centre, 25 Harbour Road,
Wanchai, Hong Kong
Phone: +852-2585-4600 Fax: +852-2827-4346
Telex: 65542 EPSCO HX

EPSON TAIWAN TECHNOLOGY & TRADING LTD.

14F, No. 7, Song Ren Road,
Taipei 110, TAIWAN
Phone: +886-2-8786-6688 Fax: +886-2-8786-6660

EPSON SINGAPORE PTE., LTD.

1 HarbourFront Place,
#03-02 HarbourFront Tower One, Singapore 098633
Phone: +65-6586-5500 Fax: +65-6271-3182

SEIKO EPSON CORP.
KOREA OFFICE

5F, KLI 63 Bldg., 60 Yoido-dong,
Youngdeungpo-Ku, Seoul 150-763, KOREA
Phone: +82-2-784-6027 Fax: +82-2-767-3677

SEIKO EPSON CORP.
MICRODEVICES OPERATIONS DIVISION

Device Sales & Marketing Dept.

421-8, Hino, Hino-shi, Tokyo 191-8501, JAPAN
Phone: +81-42-587-5814 Fax: +81-42-587-5117

Document Code: 411554403
First Issue September 2008
Revised March 2011 in JAPAN Ⓛ