EPSON
EXCEED YOUR VISION

CMOS 16-BIT SINGLE CHIP MICROCONTROLLER

S1C17003
Technical Manual

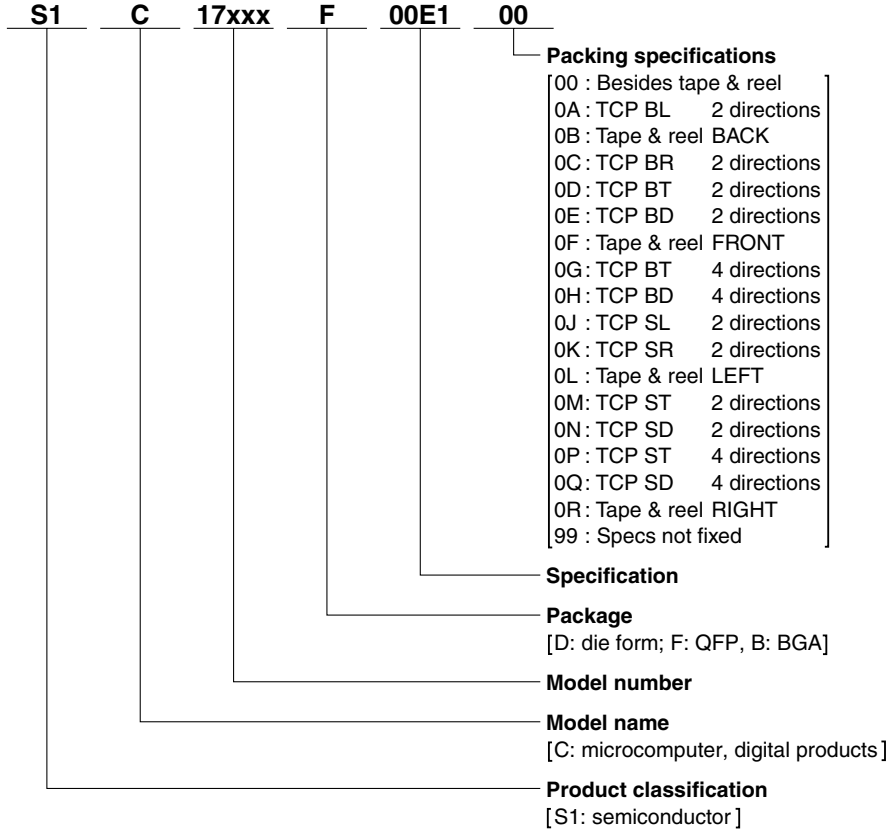
SEIKO EPSON CORPORATION

Rev. 1.1

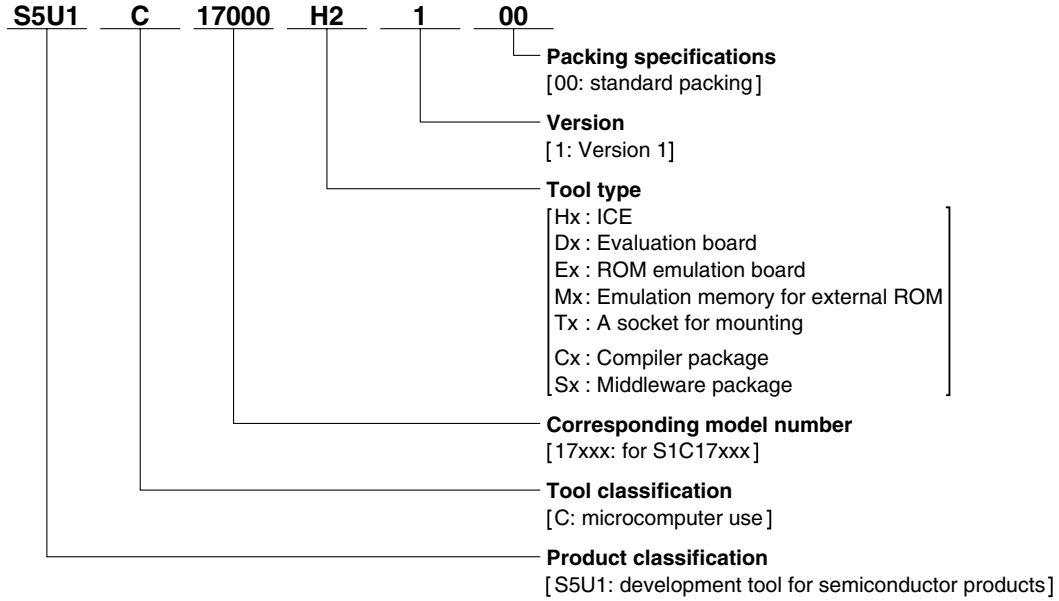
NOTICE

Configuration of product number

Devices



Development tools



- CONTENTS -

1 Overview

The S1C17003 is a 16-bit MCU featuring high-speed low-power operations, compact dimensions, wide address space and on-chip ICE. A/D converter is built in and sensor of various analog I/F can be connected. It is suitable for the application of health care product, sports watch and meter module etc. with sensor that is required a small size and micro display in the battery driven.

1.1 Features

The main features of the S1C17003 are listed below.

●CPU	• Epson original 16-bit RISC CPU core S1C17	
	• 16 bit x 16 bit + 32 bit product-sum operation, 16 bit ÷ 16bit division arithmetic unit	
●OSC3 oscillator circuit	• Crystal oscillator circuit or ceramic oscillator circuit, 20 MHz (max.)	
●OSC1 oscillator circuit	• Crystal oscillator circuit 32.786 kHz (typ.)	
●Internal Mask ROM	• 64 Kbytes (for both instructions and data)	
●Internal RAM	• 4 Kbytes	
●A/D Converter	• 10 bit resolution	4ch.
●Input/output port	• Max. 30-bit general purpose input/output port, 4-bit input only port	
●Serial interface	• SPI (master/slave)	1ch.
	• I^2C (master)	1ch.
	• I^2C (slave)	1ch.
	• UART (460,800bps, IrDA1.0 compatible)	2ch.
	• Remote controller (REMC)	1ch.
●Timer	• 8-bit timer (T8F)	2ch.
	• 16-bit timer (T16)	3ch.
	• PWM timer (T16E)	1ch.
	• Clock timer (CT)	1ch.
	• Stopwatch timer (SWT)	1ch.
	• Watchdog timer (WDT)	1ch.
	• 8-bit OSC1 PWM timer (T8OSC1)	1ch.
●Interrupt	• NMI, P Port Input interrupt	3ch.
	• Serial Interface interrupt	5ch.
	• Timer interrupt	9ch.
●Power supply voltage	• HV_{DD}(I/O) : 1.65 to 3.6V	
	• LV_{DD}(Core) : 1.65 to 1.95V	
	• AV_{DD}(I/O) : 2.7V to 3.6V	
●Operating temperatures	• -40°C to 85°C	
●Current consumption	• SLEEP mode: 1 µA (typ.) off/1.8V	
	• HALT mode: 3.3 µA (typ.) 32kHz/1.8V	
	• When operating: 4.0 mA (typ.) 20MHz/1.8V	
●Configuration as shipped	• TQFP12-64pin (7 mm x 7 mm x 1.2 mm, 0.4 mm pin pitch)	
	• WCSP-48pin (3.124 mm x 3.124 mm x 0.78 mm, 0.4 mm ball pitch)	
	• Chip (3.124 mm x 3.124 mm x 0.40 mm)	

1.2 Block Diagram

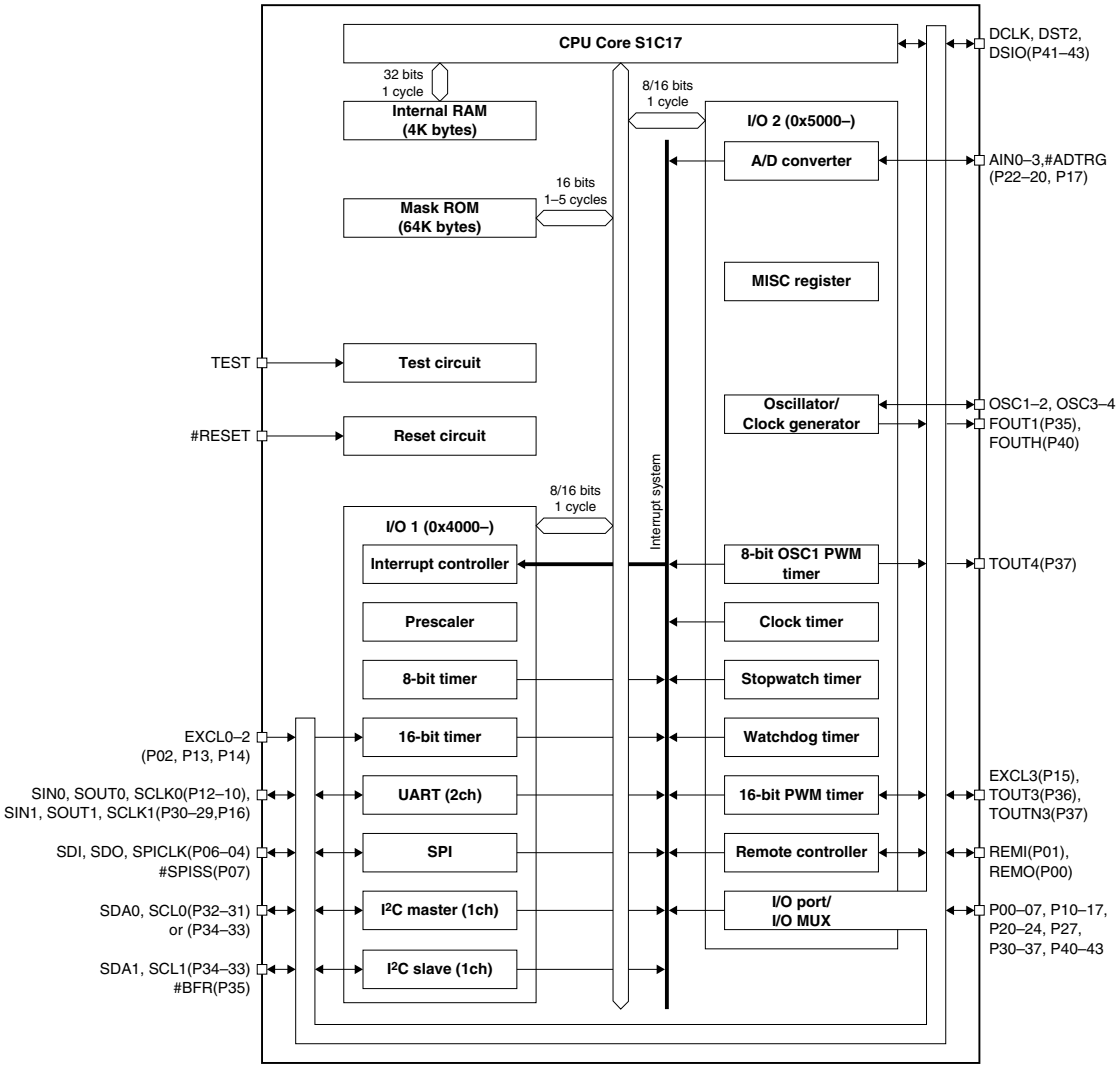


Figure 1.2.1: Block diagram

1.3 Pins

1.3.1 Pinout Diagram

The S1C17003 comes in a TQFP12-64 pin or a WCSP-48 package.

TQFP12-64pin

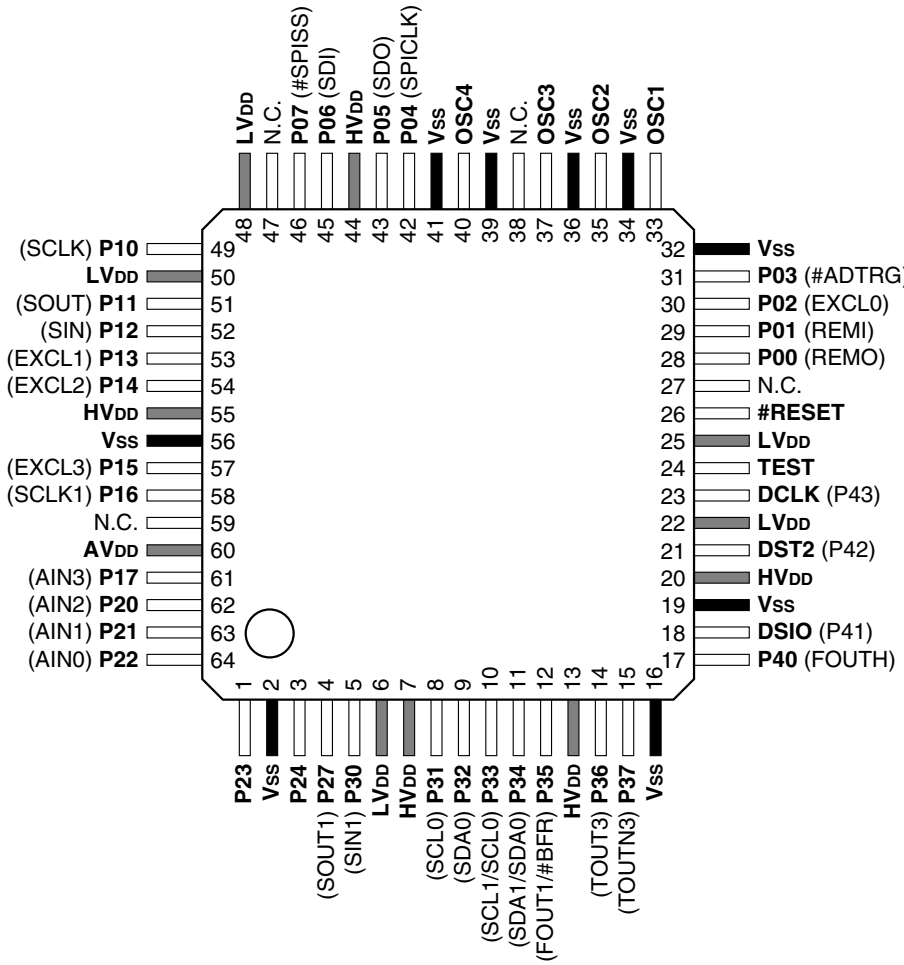
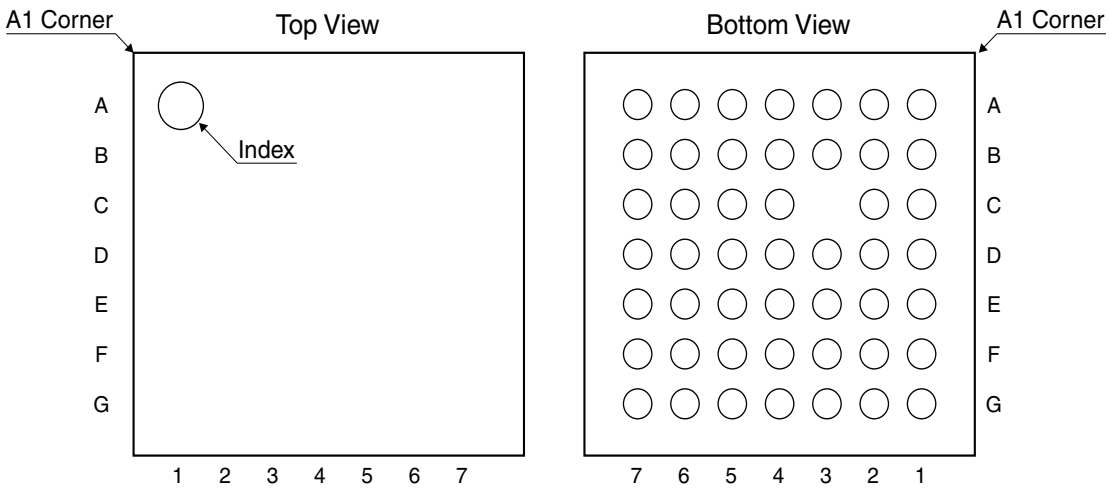


Figure 1.3.1.1: Pinout diagram (TQFP12-64pin)

WCSP-48 package

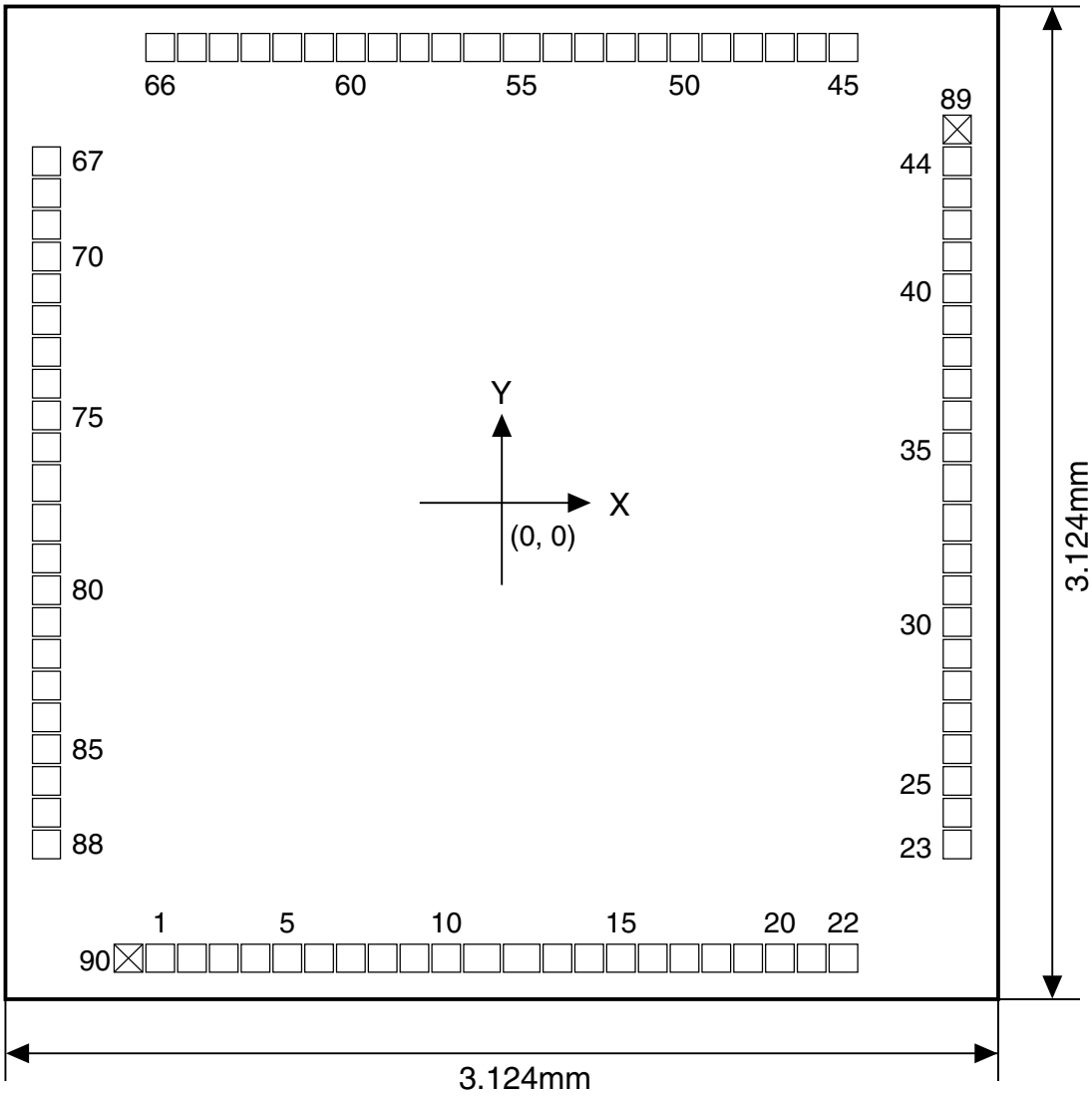


Top View

	1	2	3	4	5	6	7
A	**P23**	**P24**	**P30** SIN1	**P31** SCL0	**P34** SDA1 SDA0	**P36** TOUT3	**P40** FOUTH
B	**P21** AIN1	**P22** AIN0	**P27** SOUT1	**P32** SDA0	**P35** FOUT1 #BFR	**P37** TOUTN3	**DSIO** P41
C	**P17** AIN3	**P20** AIN2		**HVDD**	**P33** SCL1 SCL0	**VSS**	**DST2** P42
D	**P15** EXCL3	**P16** SCLK1	**AVDD**	**VSS**	**LVDD**	**DCLK** P43	**TEST**
E	**P14** EXCL2	**P13** EXCL1	**P12** SIN	**HVDD**	**P01** REMI	**P00** REMO	**#RESET**
F	**P11** SOUT	**LVDD**	**P06** SDI	**P04** SPICLK	**VSS**	**P03** #ADTRG	**P02** EXCL0
G	**P10** SCLK	**P07** #SPISS	**P05** SDO	**OSC4**	**OSC3**	**OSC2**	**OSC1**

Figure 1.3.1.2: Pinout diagram (WCSP-48)

CHIP-88pad



Opening of Pad
Pad No. 1～10, 13 ～ 22, 45 ～ 54, 57 ～ 66 : 90 μm×88 μm
Pad No. 11, 12, 55, 56 : 115 μm×88 μm
Pad No. 23 ～ 32, 35 ～ 44, 67 ～ 76, 79 ～ 88 : 88 μm×90 μm
Pad No. 33, 34, 77, 78 : 88 μm×115 μm
Chip thickness
400 μm

Pad Coordinates

PAD No.	X (mm)	Y (mm)	Assignment	PAD No.	X (mm)	Y (mm)	Assignment
1	-1.075	-1.433	P23	45	1.433	1.075	NC
2	-0.975	-1.433	V_{SS}	46	1.075	1.433	OSC1
3	-0.875	-1.433	V_{SS}	47	0.975	1.433	V_{SS}
4	-0.775	-1.433	P24	48	0.875	1.433	V_{SS}
5	-0.675	-1.433	NC	49	0.775	1.433	OSC2
6	-0.575	-1.433	P27	50	0.675	1.433	NC
7	-0.475	-1.433	P30	51	0.475	1.433	V_{SS}
8	-0.375	-1.433	LV_{DD}	52	0.375	1.433	OSC3
9	-0.275	-1.433	LV_{DD}	53	0.275	1.433	NC
10	-0.175	-1.433	HV_{DD}	54	0.175	1.433	V_{SS}
11	-0.063	-1.433	HV_{DD}	55	0.063	1.433	OSC4
12	0.063	-1.433	P31	56	-0.063	1.433	V_{SS}
13	0.175	-1.433	P32	57	-0.175	1.433	NC
14	0.275	-1.433	P33	58	-0.275	1.433	P04
15	0.375	-1.433	P34	59	-0.375	1.433	P05
16	0.475	-1.433	P35	60	-0.475	1.433	HV_{DD}
17	0.575	-1.433	HV_{DD}	61	-0.575	1.433	P06
18	0.675	-1.433	P36	62	-0.675	1.433	P07
19	0.775	-1.433	P37	63	-0.775	1.433	NC
20	0.875	-1.433	V_{SS}	64	-0.875	1.433	LV_{DD}
21	0.975	-1.433	V_{SS}	65	-0.975	1.433	NC
22	1.075	-1.433	NC	66	-1.075	1.433	NC
23	1.433	-1.075	NC	67	-1.433	1.075	P10
24	1.433	-0.975	P40	68	-1.433	0.975	NC
25	1.433	-0.875	NC	69	-1.433	0.875	LV_{DD}
26	1.433	-0.775	DSIO	70	-1.433	0.775	LV_{DD}
27	1.433	-0.675	V_{SS}	71	-1.433	0.675	P11
28	1.433	-0.575	V_{SS}	72	-1.433	0.575	P12
29	1.433	-0.475	HV_{DD}	73	-1.433	0.475	P13
30	1.433	-0.375	DST2	74	-1.433	0.375	P14
31	1.433	-0.275	LV_{DD}	75	-1.433	0.275	HV_{DD}
32	1.433	-0.175	DCLK	76	-1.433	0.175	V_{SS}
33	1.433	-0.063	TEST	77	-1.433	0.063	P15
34	1.433	0.063	LV_{DD}	78	-1.433	-0.063	V_{SS}
35	1.433	0.175	LV_{DD}	79	-1.433	-0.175	P16
36	1.433	0.275	#RESET	80	-1.433	-0.275	AV_{DD}
37	1.433	0.375	NC	81	-1.433	-0.375	NC
38	1.433	0.475	P00	82	-1.433	-0.475	AV_{DD}
39	1.433	0.575	P01	83	-1.433	-0.575	P17/AIN3
40	1.433	0.675	P02	84	-1.433	-0.675	P20/AIN2
41	1.433	0.775	NC	85	-1.433	-0.775	AV_{DD}
42	1.433	0.875	P03	86	-1.433	-0.875	P21/AIN1
43	1.433	0.975	V_{SS}	87	-1.433	-0.975	P22/AIN0
44	1.433	1.075	V_{SS}	88	-1.433	-1.075	NC

1.3.2 Pin Descriptions

Table 1.3.2.1: Pin descriptions

PAD/Pin/Ball No.			Name	I/O	Default status	Function (Default/Shared by setting)
CHIP	TQFP	WCSP				
1	1	A1	**P23**	I/O	I(Pull-UP)	I/O common port
*2	2	*2	**VSS**	-	-	Power supply (−)
4	3	A2	**P24**	I/O	I(Pull-UP)	I/O common port
6	4	B3	**P27**/SOUT1	I/O	I(Pull-UP)	I/O common port*1/UART Ch1 data output
7	5	A3	**P30**/SIN1	I/O	I(Pull-UP)	I/O common port*1/UART Ch1 data input
*3	6	*3	**LVDD**	-	-	Core power supply (+)
*4	7	*4	**HVDD**	-	-	I/O Power supply (+)
12	8	A4	**P31**/SCL0	I/O	I(Pull-UP)	I/O common port*1/I2C master clock output
13	9	B4	**P32**/SDA0	I/O	I(Pull-UP)	I/O common port*1/I2C master data I/O
14	10	C5	**P33**/SCL1/SCL0	I/O	I(Pull-UP)	I/O common port*1/I2C slave clock input/I2C master clock output
15	11	A5	**P34**/SDA1/SDA0	I/O	I(Pull-UP)	I/O common port*1/I2C slave data I/O /I2C master data I/O
16	12	B5	**P35**/FOUT1/#BFR	I/O	I(Pull-UP)	I/O common port*1/OSC1 external clock output/I2C slave bus open
*4	13	*4	**HVDD**	-	-	I/O Power supply (+)
18	14	A6	**P36**/TOUT3	I/O	I(Pull-UP)	I/O common port*1/T16E Ch0 PWM signal output (non-inverted)
19	15	B6	**P37**/TOUTN3	I/O	I(Pull-UP)	I/O common port*1/T16E Ch0 PWM signal output (inverted)
*2	16	*2	**VSS**	-	-	Power supply (−)
24	17	A7	**P40**/FOUTH	I/O	I(Pull-UP)	I/O common port*1/HSCLK clock output (with divide)
26	18	B7	**DSIO**/P41	I/O	I(Pull-UP)	On-chip debugger data I/O*1/I/O common port
*2	19	*2	**VSS**	-	-	Power supply (−)
*4	20	*4	**HVDD**	-	-	I/O Power supply (+)
30	21	C7	**DST2**/P42	I/O	O(L)	On-chip debugger status output*1/I/O common port
*3	22	*3	**LVDD**	-	-	Core power supply (+)
32	23	D6	**DCLK**/P43	I/O	O(H)	On-chip debugger clock output*1/I/O common port
33	24	D7	**TEST**	I	I(Pull-UP)	Test pin (fixed to VSS)
*3	25	*3	**LVDD**	-	-	Core power supply (+)
36	26	E7	**#RESET**	I	I(Pull-UP)	Initial set input (with the noise filter)
-	27	-	**-NC**	-	-	-
38	28	E6	**P00**/REMO	I/O	I(Pull-UP)	I/O common port (with inturrupt) *1/REMC output
39	29	E5	**P01**/REMI	I/O	I(Pull-UP)	I/O common port (with inturrupt)*1/REMC input
40	30	F7	**P02**/EXCL0	I/O	I(Pull-UP)	I/O common port (with inturrupt)*1/T16 Ch0 external clock input
42	31	F6	**P03**/#ADTRG	I/O	I(Pull-UP)	I/O common port (with inturrupt)*1/ A/D convert external trigger
*2	32	*2	**VSS**	-	-	Power supply (−)
46	33	G7	**OSC1**	I	I	OSC1 oscillator input*6
*2	34	*2	**VSS**	-	-	Power supply (−)
49	35	G6	**OSC2**	O	O	OSC1 oscillator output

PAD/Pin/Ball No.			Name	I/O	Default status	Function (Default/Shared by setting)
CHIP	TQFP	WCSP				
*2	36	*2	**VSS**	-	-	Power supply (−)
52	37	G5	**OSC3**	I	I	OSC3 oscillator input*6
-	38	-	**-NC**	-	-	-
*2	39	*2	**VSS**	-	-	Power supply (−)
55	40	G4	**OSC4**	O	O	OSC3 oscillator output
*2	41	*2	**VSS**	-	-	Power supply (−)
58	42	F4	**P04**/SPICLK	I/O	I(Pull-UP)	I/O common port (with inturrupt)*1/SPI clock I/O
59	43	G3	**P05**/SDO	I/O	I(Pull-UP)	I/O common port (with inturrupt)*1/SPI data output
*4	44	*4	**HVDD**	-	-	I/O Power supply (+)
61	45	F3	**P06**/SDI	I/O	I(Pull-UP)	I/O common port (with inturrupt)*1/SPI data input
62	46	G2	**P07**/#SPISS	I/O	I(Pull-UP)	I/O common port (with inturrupt)*1/SPI slave select input
-	47	-	**-NC**	-	-	-
*3	48	*3	**LVDD**	-	-	Core power supply (+)
67	49	G1	**P10**/SCLK	I/O	I(Pull-UP)	I/O common port (with inturrupt)*1/UART Ch0 clock input
*3	50	*3	**LVDD**	-	-	Core power supply (+)
71	51	F1	**P11**/SOUT	I/O	I(Pull-UP)	I/O common port (with inturrupt)*1/UART Ch0 data output
72	52	E3	**P12**/SIN	I/O	I(Pull-UP)	I/O common port (with inturrupt)*1/UART Ch0 data input
73	53	E2	**P13**/EXCL1	I/O	I(Pull-UP)	I/O common port (with inturrupt)*1/T16 Ch1 external clock input
74	54	E1	**P14**/EXCL2	I/O	I(Pull-UP)	I/O common port (with inturrupt)*1/T16 Ch2 external clock input
*4	55	*4	**HVDD**	-	-	I/O Power supply (+)
*2	56	*2	**VSS**	-	-	Power supply (−)
77	57	D1	**P15**/EXCL3	I/O	I(Pull-UP)	I/O common port (with inturrupt)*1/T16E Ch0 external clock input
79	58	D2	**P16**/SCLK1	I/O	I(Pull-UP)	I/O common port (with inturrupt)*1/UART Ch1 clock input
-	59	-	**-NC**	-	-	-
*5	60	D3	**AVDD**	-	-	Analog power supply (+)
83	61	C1	**P17**/AIN3	I	I	I/O common port (with inturrupt)*1/ A/D converter Ch3 input
84	62	C2	**P20**/AIN2	I	I	I/O common port*1/ A/D converter Ch2 input
86	63	B1	**P21**/AIN1	I	I	I/O common port*1/ A/D converter Ch1 input
87	64	B2	**P22**/AIN0	I	I	I/O common port*1/ A/D converter Ch0 input

*1: Default function settings

*2: VSS PAD numbers : 2, 3, 20, 21, 27, 28, 43, 44, 47, 48, 51, 54, 56, 76 78
VSS ball numbers : C6, D4, F5

*3: LVDD PAD numbers : 8, 9, 31, 34, 35, 64, 69, 70
LVDD ball numbers : D5, F2

*4: HVDD PAD numbers : 10, 11, 17, 29, 60, 75
HVDD ball numbers : C4, E4

*5: AVDD PAD numbers : 80, 82, 85

*6: When an external clock is input to the OSC3 or OSC1 pin, the clock signal level must be LVDD.

Note: Do not put bonding on NC pins. (The pins for which “NC” is specified for TQFP, and no number is described for CHIP/WCSP.)

This page intentionally left blank.

2 CPU

The S1C17003 uses an S1C17 core as the core processor.
The S1C17 core is an original Seiko Epson 16-bit RISC processor.
It features low power consumption, high-speed operation, wide address space, main instruction single-clock execution, and gate-saving design. It is ideal for use in controllers or sequencers, in which 8-bit CPUs are widely used.
For detailed information on the S1C17 core, refer to the *S1C17 Family S1C17 Core Manual*.

2.1 S1C17 Core Features

Processor type

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

Instruction set

- Code length — Fixed 16-bit length
- Number of instructions — 111 basic instructions (184 in total)
- Execution cycle — Main instructions executed in one cycle
- Immediate expansion instructions — Expansion of immediate to 24 bits
- Compact, high-speed instruction set optimized for development with C

Register set

- 24-bit general purpose register x 8
- 24-bit special register x 2
- 8-bit special register x 1

Memory space, buses

- Up to 16 Mbytes of memory space (24-bit address)
- Harvard architecture with separate instruction bus (16-bit) and data bus (32-bit)

Interrupt

- Supports reset, NMI, and 32 different types of external interrupt
- Irregular address interrupt
- Debug interrupt
- Reading vector from vector table and direct branching to interrupt processing routines
- Permits software interrupts using vector numbers (all vector numbers can be specified)

Power saving

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

Coprocessor interface

- 16 bits x 16 bits + 32 bits product-sum arithmetic unit
- 16 bits/16 bits division arithmetic unit

2.2 CPU Registers

The S1C17 core contains eight general purpose registers and three special registers.

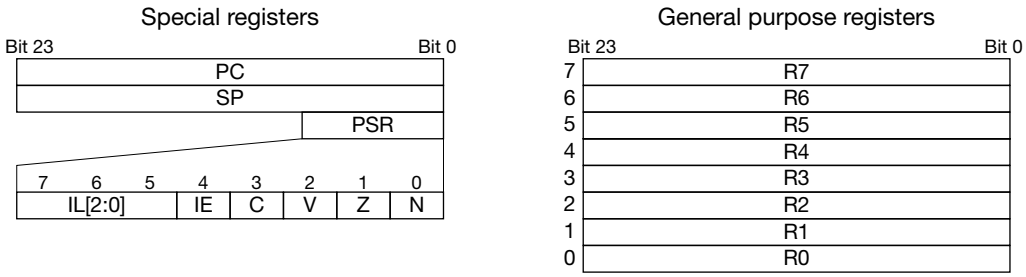


Figure 2.2.1: Registers

2.3 Instruction Set

The S1C17 core instruction codes are all 16-bit and fixed-length. Major instructions are executed in a single cycle using pipeline processing. For more information on the various instructions, refer to the *S1C17 Family S1C17 Core Manual*.

Table 2.3.1: S1C17 core instruction list

Type	Mnemonic		Function
Data transfer	ld.b	%*rd*,%*rs*	General purpose register (byte) ➔ General purpose register (sign extension)
		%*rd*,[%*rb*]	Memory (byte) ➔ General purpose register (sign extension)
		%*rd*,[%*rb*]+	Memory address post-increment/post-decrement
		%*rd*,[%*rb*]-	A pre-decrement function can be used
		%*rd*,-[%*rb*]	
		%*rd*,[%sp+*imm7*]	Stack (byte) ➔ General purpose register (sign extension)
		%*rd*,[*imm7*]	Memory (byte) ➔ General purpose register (sign extension)
		[%*rb*],%*rs*	General purpose register (byte) ➔ Memory
		[%*rb*]+,%*rs*	Memory address post-increment/post-decrement
		[%*rb*]-,%*rs*	A pre-decrement function can be used
		-[%*rb*],%*rs*	
		[%sp+*imm7*],%*rs*	General purpose register (byte) ➔ Stack
		[*imm7*],%*rs*	General purpose register (byte) ➔ Memory
	ld.ub	%*rd*,%*rs*	General purpose register (byte) ➔ General purpose register (zero extension)
		%*rd*,[%*rb*]	Memory (byte) ➔ General purpose register (zero extension)
		%*rd*,[%*rb*]+	Memory address post-increment/post-decrement
		%*rd*,[%*rb*]-	A pre-decrement function can be used
		%*rd*,-[%*rb*]	
		%*rd*,[%sp+*imm7*]	Stack (byte) ➔ General purpose register (zero extension)
		%*rd*,[*imm7*]	Memory (byte) ➔ General purpose register (zero extension)
	ld	%*rd*,%*rs*	General purpose register (16 bits) ➔ General purpose register
		%*rd*,*sign7*	Immediate ➔ General purpose register (sign extension)
		%*rd*,[%*rb*]	Memory (16 bits) ➔ General purpose register
		%*rd*,[%*rb*]+	Memory address post-increment/post-decrement
		%*rd*,[%*rb*]-	A pre-decrement function can be used
		%*rd*,-[%*rb*]	
		%*rd*,[%sp+*imm7*]	Stack (16 bits) ➔ General purpose register
		%*rd*,[*imm7*]	Memory (16 bits) ➔ General purpose register
		[%*rb*],%*rs*	General purpose register (16 bits) ➔ Memory
		[%*rb*]+,%*rs*	Memory address post-increment/post-decrement
		[%*rb*]-,%*rs*	A pre-decrement function can be used
		-[%*rb*],%*rs*	
		[%sp+*imm7*],%*rs*	General purpose register (16 bits) ➔ Stack
		[*imm7*],%*rs*	General purpose register (16 bits) ➔ Memory
	ld.a	%*rd*,%*rs*	General purpose register (24 bits) ➔ General purpose register
		%*rd*,*imm7*	Immediate ➔ General purpose register (zero extension)
		%*rd*,[%*rb*]	Memory (32 bits) ➔ General purpose register (*1)
		%*rd*,[%*rb*]+	Memory address post-increment/post-decrement
		%*rd*,[%*rb*]-	A pre-decrement function can be used
		%*rd*,-[%*rb*]	
		%*rd*,[%sp+*imm7*]	Stack (32 bits) ➔ General purpose register (*1)
		%*rd*,[*imm7*]	Memory (32 bits) ➔ General purpose register (*1)
		[%*rb*],%*rs*	General purpose register (32 bits, zero extension) ➔ Memory (*1)
		[%*rb*]+,%*rs*	Memory address post-increment/post-decrement
		[%*rb*]-,%*rs*	A pre-decrement function can be used
		-[%*rb*],%*rs*	
		[%sp+*imm7*],%*rs*	General purpose register (32 bits, zero extension) ➔ Stack (*1)
		[*imm7*],%*rs*	General purpose register (32 bits, zero extension) ➔ Memory (*1)
		%*rd*,%sp	SP ➔ General purpose register
		%*rd*,%pc	PC ➔ General purpose register
		%*rd*,[%sp]	Stack (32 bits) ➔ General purpose register (*1)
		%*rd*,[%sp]+	Stack pointer post-increment/post-decrement
		%*rd*,[%sp]-	A pre-decrement function can be used
		%*rd*,-[%sp]	

Type	Mnemonic		Function
Data transfer	ld.a	[%sp],%*rs*	General purpose register (32 bits, zero extension) ➔ Stack (*1)
		[%sp]+,%*rs*	Stack pointer post-increment/post-decrement
		[%sp]-,%*rs*	A pre-decrement function can be used
		-[%sp],%*rs*	
		%sp,%*rs*	General purpose register (24 bits) ➔ SP
		%sp,*imm7*	Immediate ➔ SP
Integer arithmetic	add	%*rd*,%*rs*	Adds 16 bits between general purpose registers
	add/c		Supports conditional execution (/c: Executed when C = 1, /nc: Executed when
	add/nc		C = 0)
	add	%*rd*,*imm7*	Adds general purpose register and immediate 16 bits
	add.a	%*rd*,%*rs*	Adds 24 bits between general purpose registers
	add.a/c		Supports conditional execution (/c: Executed when C = 1, /nc: Executed when
	add.a/nc		C = 0)
	add.a	%sp,%*rs*	Adds SP and general purpose register 24 bits
		%*rd*,*imm7*	Adds general purpose register and immediate 24 bits
		%sp,*imm7*	Adds SP and immediate 24 bits
	adc	%*rd*,%*rs*	Adds 16 bits with carry between general purpose registers
	adc/c		Supports conditional execution (/c: Executed when C = 1, /nc: Executed when
	adc/nc		C = 0)
	adc	%*rd*,*imm7*	Adds general purpose register and immediate 16 bits with carry
	sub	%*rd*,%*rs*	Subtracts 16 bits between general purpose registers
	sub/c		Supports conditional execution (/c: Executed when C = 1, /nc: Executed when
	sub/nc		C = 0)
	sub	%*rd*,*imm7*	Subtracts general purpose register and immediate 16 bits
	sub.a	%*rd*,%*rs*	Subtracts 24 bits between general purpose registers
	sub.a/c		Supports conditional execution (/c: Executed when C = 1, /nc: Executed when
	sub.a/nc		C = 0)
	sub.a	%sp,%*rs*	Subtracts SP and general purpose register 24 bits
		%*rd*,*imm7*	Subtracts general purpose register and immediate 24 bits
		%sp,*imm7*	Subtracts SP and immediate 24 bits
	sbc	%*rd*,%*rs*	Subtracts 16 bits with carry between general purpose registers
	sbc/c		Supports conditional execution (/c: Executed when C = 1, /nc: Executed when
	sbc/nc		C = 0)
	sbc	%*rd*,*imm7*	Subtracts general purpose register and immediate 16 bits with carry
	cmp	%*rd*,%*rs*	Compares 16 bits between general purpose registers
	cmp/c		Supports conditional execution (/c: Executed when C = 1, /nc: Executed when
	cmp/nc		C = 0)
	cmp	%*rd*,*sign7*	Compares general purpose registers and immediate 16 bits
	cmp.a	%*rd*,%*rs*	Compares 24 bits between general purpose registers
	cmp.a/c		Supports conditional execution (/c: Executed when C = 1, /nc: Executed when
	cmp.a/nc		C = 0)
	cmp.a	%*rd*,*imm7*	Compares general purpose registers and immediate 24 bits
	cmc	%*rd*,%*rs*	Compares 16 bits with carry between general purpose registers
	cmc/c		Supports conditional execution (/c: Executed when C = 1, /nc: Executed when
	cmc/nc		C = 0)
	cmc	%*rd*,*sign7*	Compares general purpose register and immediate 16 bits with carry
Logic operations	and	%*rd*,%*rs*	AND operation between general purpose registers
	and/c		Supports conditional execution (/c: Executed when C = 1, /nc: Executed when
	and/nc		C = 0)
	and	%*rd*,*sign7*	AND operation for general purpose register and immediate
	or	%*rd*,%*rs*	OR operation between general purpose registers
	or/c		Supports conditional execution (/c: Executed when C = 1, /nc: Executed when
	or/nc		C = 0)
	or	%*rd*,*sign7*	OR operation for general purpose register and immediate
	xor	%*rd*,%*rs*	EXCLUSIVE OR between general purpose registers
	xor/c		Supports conditional execution (/c: Executed when C = 1, /nc: Executed when
	xor/nc		C = 0)
	xor	%*rd*,*sign7*	EXCLUSIVE OR for general purpose register and immediate
	not	%*rd*,%*rs*	NOT operation between general purpose registers (1 complement)
	not/c		Supports conditional execution (/c: Executed when C = 1, /nc: Executed when
	not/nc		C = 0)
	not	%*rd*,*sign7*	NOT operation for general purpose register and immediate (1 complement)

Type	Mnemonic		Function
Shift & swap	sr	%rd,%rs	Right logic shift (shift bit number specified by register)
		%rd,imm7	Right logic shift (shift bit number specified by immediate)
	sa	%rd,%rs	Right operation shift (shift bit number specified by register)
		%rd,imm7	Right operation shift (shift bit number specified by immediate)
	sl	%rd,%rs	Left logic shift (shift bit number specified by register)
		%rd,imm7	Left logic shift (shift bit number specified by immediate)
	swap	%rd,%rs	Byte swap at 16-bit boundary
Immediate extension	ext	imm13	Extend operand for next instruction
Conversion	cv.ab	%rd,%rs	Convert 8-bit coded data to 24 bits
	cv.as	%rd,%rs	Convert 16-bit coded data to 24 bits
	cv.al	%rd,%rs	Convert 32-bit data to 24 bits
	cv.la	%rd,%rs	Convert 24-bit data to 32 bits
	cv.ls	%rd,%rs	Convert 16-bit data to 32 bits
Branch	jpr	sign10	PC-relative jump
	jpr.d	%rb	Allows delayed branching
	jpa	imm7	Absolute jump
	jpa.d	%rb	Allows delayed branching
	jrgt	sign7	Conditional PC-relative jump Branch conditions: !Z & !(N ^ V)
	jrgt.d		Allows delayed branching
	jrge	sign7	Conditional PC-relative jump Branch conditions: !(N ^ V)
	jrge.d		Allows delayed branching
	jrlt	sign7	Conditional PC-relative jump Branch conditions: N ^ V
	jrlt.d		Allows delayed branching
	jrle	sign7	Conditional PC-relative jump Branch conditions: Z | N ^ V
	jrle.d		Allows delayed branching
	jrugt	sign7	Conditional PC-relative jump Branch conditions: !Z & !C
	jrugt.d		Allows delayed branching
	jruge	sign7	Conditional PC-relative jump Branch conditions: !C
	jruge.d		Allows delayed branching
	jrult	sign7	Conditional PC-relative jump Branch conditions: C
	jrult.d		Allows delayed branching
	jrule	sign7	Conditional PC-relative jump Branch conditions: Z | C
	jrule.d		Allows delayed branching
	jreq	sign7	Conditional PC-relative jump Branch conditions: Z
	jreq.d		Allows delayed branching
	jrne	sign7	Conditional PC-relative jump Branch conditions: !Z
	jrne.d		Allows delayed branching
	call	sign10	PC-relative subroutine call
	call.d	%rb	Allows delayed branching
	calla	imm7	Absolute subroutine call
	calla.d	%rb	Allows delayed branching
	ret		Return from subroutine
	ret.d		Allows delayed branching
	int	imm5	Software interrupt
	intl	imm5,imm3	Software interrupt with interrupt level specification
	reti		Return from interrupt
	reti.d		Allows delayed branching
	brk		Debug interrupt
	retd		Return from debug processing
System control	nop		No operation
	halt		HALT
	slp		SLEEP
	ei		Permits interrupt
	di		Prevents interrupt
Coprocessor control	ld.cw	%rd,%rs	Transfer data to coprocessor
		%rd,imm7	
	ld.ca	%rd,%rs	Transfer data to coprocessor and obtain results and flag status
		%rd,imm7	
	ld.cf	%rd,%rs	Transfer data to coprocessor and obtain flag status
		%rd,imm7	

*1 Instruction ld.a accesses 32-bit memory. When data is transferred from register to memory, 32 bits of data with the first 8 bits set to 0 are written to memory. When data is read from memory, the first 8 bits are ignored.

The codes used in this table are explained below.

Table 2.3.2: Code meanings

Code	Description
%*rs*	General purpose source register
%*rd*	General purpose destination register
[%*rb*]	Memory specified indirectly by general purpose register
[%*rb*]+	Memory specified indirectly by general purpose register (with address post-increment)
[%*rb*]-	Memory specified indirectly by general purpose register (with address post-decrement)
-[%*rb*]	Memory specified indirectly by general purpose register (with address pre-decrement)
%sp	Stack pointer
[%sp],[%sp+*imm7*]	Stack
[%sp]+	Stack (with address post-increment)
[%sp]-	Stack (with address post-decrement)
-[%sp]	Stack (with address pre-decrement)
imm3,*imm5*,*imm7*,*imm13*	Immediate without code (number indicates bit length)
sign7,*sign10*	Immediate with code (number indicates bit length)

2.4 Vector Table

The vector table contains the vectors (processing routine start addresses) for interrupt processing routines. When an interrupt occurs, the S1C17 core reads the vector corresponding to the interrupt and executes that processing routine. The boot address for starting program execution must be written at the top of the vector table after resetting.
The S1C17003 vector table starts from address 0x8000. The vector table base address can be read from the TTBR (vector table base register) at address 0xffff80.
For more information on Vector Table, refer to "6. Interrupt Controller"

The base (top) address for the vector table for writing interrupt vectors can be set using the MISC_TTBRL and MISC_TTBRH registers (0x5328 and 0x532a). The MISC_TTBRL and MISC_TTBRH registers are set to the 0x8000 address after initial resetting. This means only the reset vector must be written to the above address, even when changing the vector table location. Bits 7 to 0 in the MISC_TTBRL register are fixed to 0; the initial address of the vector table normally starts from the 256 byte boundary.

0x5328–0x532a: Vector Table Address Low/High Registers (MISC_TTBRL, MISC_TTBRH)

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

Note: The MISC_TTBRL and MISC_TTBRH registers are write-protected. To write to these registers, write-protection must be overridden by writing 0x96 to the MISC Protect Register (0x5324). Normally, the MISC Protect Register (0x5324) should be set to a value other than 0x96, except when writing to the MISC_TTBRL and MISC_TTBRH registers, since unnecessary writes may result in system malfunctions.

2.5 PSR Readout

The S1C17003 incorporates a PSR register (0x532c) for reading out the contents of the PSR (Processor Status Register) in the S1C17 core. Reading out the contents of this register makes it possible to check the contents of the PSR using application software. Note that data cannot be written to the PSR.

0x532c: PSR Register (MISC_PSR)

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

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

Read out the value (interrupt level) of the IL bit of the PSR. (default: 0x0)

D4 PSRIE: PSR Interrup Enable (IE) Bit

Read out the value (interrupt enable) of the PSR IE bit.
1(R): 1 (Interrupt permitted)
0(R): 0 (Interrupt prohibited) (default)

D3 PSRC: PSR Carry (C) Flag

Read out the value of the PSR C (carry) flag.
1(R): 1
0(R): 0 (default)

D2 PSRV: PSR Overflow (V) Flag

Read out the value of the PSR V (overflow) flag.
1(R): 1
0(R): 0 (default)

D1 PSRZ: PSR Zero (Z) Flag

Read out the value of the PSR Z (zero) flag.
1(R): 1
0(R): 0 (default)

D0 PSRN: PSR Negative (N) Flag

Read out the value of the PSR N (negative) flag.
1(R): 1
0(R): 0 (default)

2.6 Processor Information

The S1C17003 contains a processor ID register (0xffff84) to allow specification of the CPU core type by the application software.

0xffff84: Processor ID Register (IDIR)

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

This is the read-only register containing the ID code indicating the processor type. The S1C17 core ID code is 0x10.

3 Memory Map and Bus Control

Figure 3.1 shows the S1C17003 memory map.

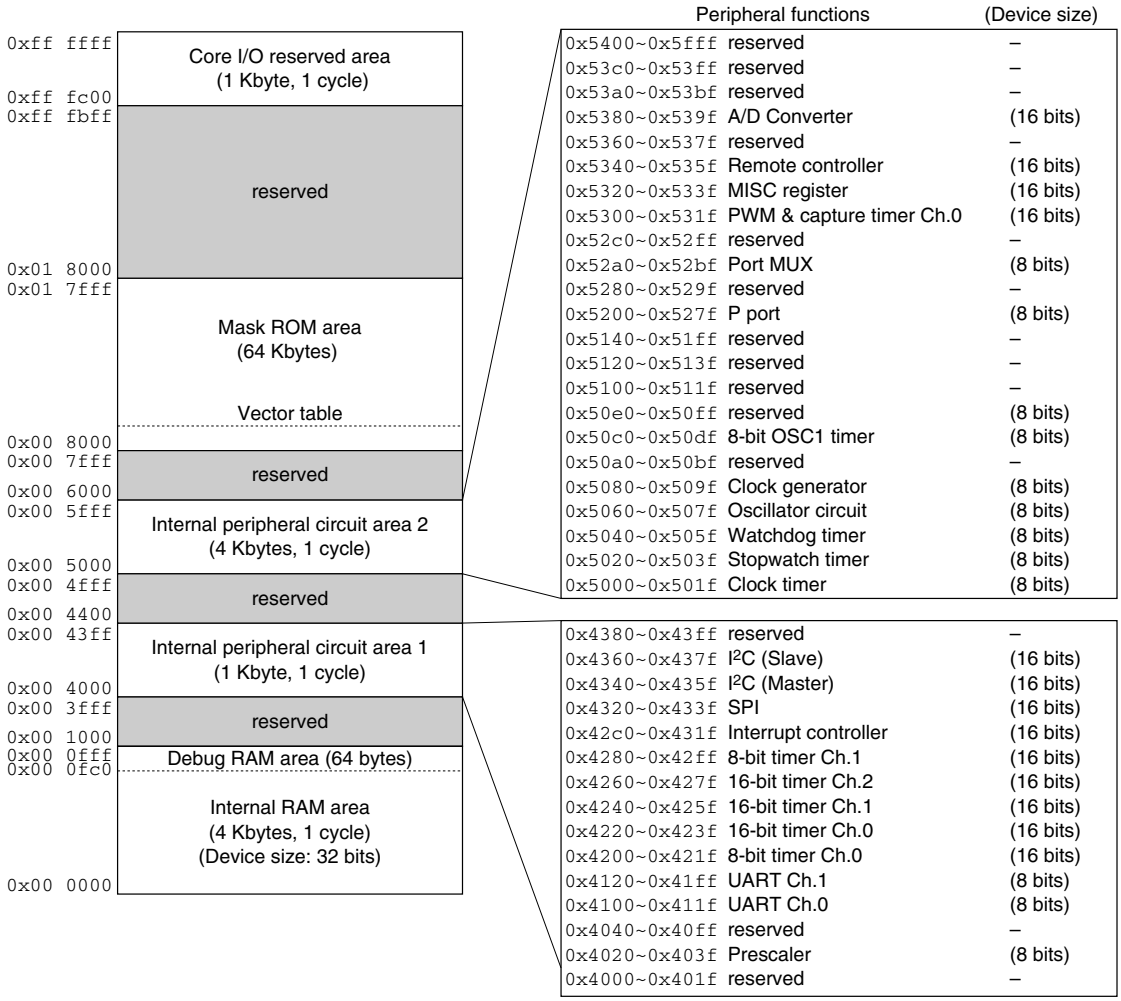


Figure 3.1: S1C17003 memory map

3.1 Bus Cycle

The CPU operates using CCLK as a datum. For more information on CCLK, refer to "8.2 CPU Core Clock (CCLK) Control."
The time from one CCLK rise-up to the next forms 1 CCLK, defined as one bus cycle. As shown in Figure 3.1, the number of cycles required for a single bus access depends on the peripheral circuits and memory. The number of bus accesses also varies and depends on the CPU instruction (access size) and device size.

Table 3.1.1: Bus access numbers

Device size	CPU access size	Bus access number
8 bits	8 bits	1
	16 bits	2
	32 bits*	4
16 bits	8 bits	1
	16 bits	1
	32 bits*	2
32 bits	8 bits	1
	16 bits	1
	32 bits*	1

* First 8 bits of data for 32-bit data access
  The first 8 bits of 32-bit data are written to memory as 0. The first 8 bits are ignored when read from memory. Interrupt processing stack operation involves reading and writing 32 bits with the PSR value in the first 8 bits and the return address in the last 24 bits.

3.1.1 Access Size Restrictions

All modules can be accessed using 8-bit, 16-bit, and 32-bit instructions. Where possible, we recommend matching access to device size. Reading from non-essential registers may alter the state of peripheral circuits and cause problems.

3.1.2 Instruction Execution Cycle Restrictions

In the event of any of the conditions listed below, instruction fetch and data access will not be performed simultaneously, and the instruction fetch cycle will be extended by the amount of access cycles for the areas in which data exists.

- If an instruction is executed for an internal RAM area accessing internal RAM area data

3.2 Mask ROM Area

3.2.1 Mask ROM

The 64 Kbyte area from 0x8000 to 0x17fff contains ROM enabling data or application programs to be written. Address 0x8000 is defined as the vector table base address. The vector table (see "2.4 Vector Table") must be placed at the start of this area. The vector table base address can be modified with the MISC_TTBRL/MISC_TTBRH registers (0x5328 and 0x532a).
The ROM is read in 1 to 5 cycles.

3.2.2 ROM read access cycle settings

To maintain compatibility with the S1C17602, the read access cycle in ROM area can be set with the FLCYC[2:0] (D[2:0]/MISC_FL register). Set the FLCYC[2:0] to 0x4 in usual circumstances.

0x5320: ROM Control Register (MISC_FL)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
ROM Control Register (MISC_FL)	0x5320 (16 bits)	D15–3	–	reserved	–		–	–	0 when being read.
		D2–0	FLCYC[2:0]	FLASHC read access cycle	FLCYC[2:0]	Read cycle	0x3	R/W	
					0x7–0x5	reserved			
					0x4	1 cycles			
					0x3	5 cycles			
					0x2	4 cycles			
					0x1	3 cycles			
					0x0	2 cycles			

D[2:0] FLCYC[2:0]: FLASHC Read Access Cycle Setup Bits

Sets the number of read access cycles for the ROM.

Table 3.2.2.1: ROM read access cycle settings

FLCYC[2:0]	Read access cycles	CCLK frequency
0x7 to 0x5	Reserved	–
0x4	1 cycle	20 MHz max.
0x3	5 cycles	20 MHz max.
0x2	4 cycles	20 MHz max.
0x1	3 cycles	20 MHz max.
0x0	2 cycles	20 MHz max.

(Default: 0x3)

Note: • Do not set the read access cycles to a value exceeding the CCLK maximum permissible frequency. This will cause malfunctions.
• Set FLCYC[2:0]=0x4 in order to maximize the performance.

3.3 Internal RAM Area

3.3.1 Internal RAM

RAM exists in a 4-Kbyte area from address 0x0 to 0xfff. This RAM can be accessed in one cycle for reading or writing. In addition to storing variables, it can also be used to copy instruction codes and execute them rapidly in RAM.

Note: The last 64 bytes of the internal RAM (0xfc0 to 0xfff) are reserved for on-chip debugging. This area should not be accessed by application programs when using debug functions (for example, during application development).
It can be used for applications in mass-produced products that do not require debugging.

The S1C17003 enables the RAM size used to apply restrictions to 4 KB or 2 KB. For example, when using the S1C17003 to develop products with internal ROM, you can set the RAM size to match that of the target product, preventing creating programs that seek to access areas outside the RAM areas of the target product.
The RAM size is selected using IRAMSZ[1:0] (D[1:0]/MISC_IRAMSZ register).

0x5326: IRAM Size Select Register (MISC_IRAMSZ)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
IRAM Size Select Register (MISC_IRAMSZ)	0x5326 (16 bits)	D15–2	–	reserved	–		–	–	0 when being read.
		D1–0	IRAMSZ[1:0]	IRAM size select	IRAMSZ[1:0]	Read cycle	0x2	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	reserved			
					0x0	reserved			

D[1:0] IRAMSZ[1:0]: IRAM Size Select Bits
Select the internal RAM size used.

Table 3.3.1.1: Internal RAM size selection

IRAMSZ[1:0]	Internal RAM size
0x3	reserved
0x2	reserved
0x1	reserved
0x0	reserved

(Default: 0x2)

Notes:
- **The IRAM Size Select Register is write-protected. The write-protection must be overridden by writing 0x96 to the MISC Protect Register (0x5324). Note that MISC Protect Register (0x5324) should normally be set to a value other than 0x96, except when writing to the IRAM Size Select Register. Unnecessary writes may result in system malfunctions.**
- **Please do not change the setting of IRAMSZ[2:0]/MISC_IRAMSZ Register from a default value.**

3.4 Internal Peripheral Circuit Area

The 1 Kbyte area starting at address 0x4000 and the 4 Kbyte area from 0x5000 are assigned for use as internal peripheral circuit I/O and control registers.

3.4.1 Internal Peripheral Circuit Area 1 (0x4000 onward)

The internal peripheral circuit area 1 starting at address 0x4000 is assigned for use as the following internal peripheral function I/O memory and can be accessed in a single cycle.

- Prescaler (PSC, 8-bit device)
- UART (UART, 8-bit device)
- 8-bit timer (T8F, 16-bit device)
- 16-bit timer (T16, 16-bit device)
- Interrupt controller (ITC, 16-bit device)
- SPI (SPI, 16-bit device)
- I^2C master (I^2C, 16-bit device)
- I^2C slave (I^2C, 16-bit device)

3.4.2 Internal Peripheral Circuit Area 2 (0x5000 onward)

The internal peripheral circuit area 2 starting at address 0x5000 is assigned for use as the following internal peripheral function I/O memory, and can be accessed in one cycle.

- Clock timer (CT, 8-bit device)
- Stopwatch timer (SWT, 8-bit device)
- Watchdog timer (WDT, 8-bit device)
- Oscillator circuit (OSC, 8-bit device)
- Clock generator (CLG, 8-bit device)
- 8-bit OSC1 PWM timer (T8OSC1, 8-bit device)
- Input/output port & port MUX (P, 8-bit device)
- PWM timer (T16E, 16-bit device)
- MISC register (MISC, 16-bit device)
- Remote controller (REMC, 16-bit device)
- A/D converter (ADC10, 16-bit device)

3.5 Core I/O Reserved Area

The 1 Kbyte area from 0xfffc00 to 0xffffff is used as the CPU core I/O area, and the following I/O registers are assigned.

Table 3.5.1: I/O map (Core I/O reserved area)

Peripheral circuit	Address	Register name		Function
S1C17 core I/O	0xffff84	IDIR	Processor ID Register	Processor ID display
	0xffff90	DBRAM	Debug RAM Base Register	Debugging RAM base address display
	0xffffa0	DCR	Debug Control Register	Debug control
	0xffffb8	IBAR2	Instruction Break Address Register 2	Instruction break address #2 setting
	0xffffbc	IBAR3	Instruction Break Address Register 3	Instruction break address #3 setting
	0xffffd0	IBAR4	Instruction Break Address Register 4	Instruction break address #4 setting

See “2.6 Processor Information” for more information on IDIR and “24. On-chip Debugger (DBG)” for more information on other registers.

This area incorporates S1C17 core registers, in addition to those described above. For more information on these registers, refer to the *S1C17 Core Manual*.

4 Power Supply Voltage

This section explains the operating voltage of the S1C17003.

4.1 Power Supply Pins

The S1C17003 has the power supply pins shown in Table 4.1.1.

Table 4.1.1 Power Supply Pins

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

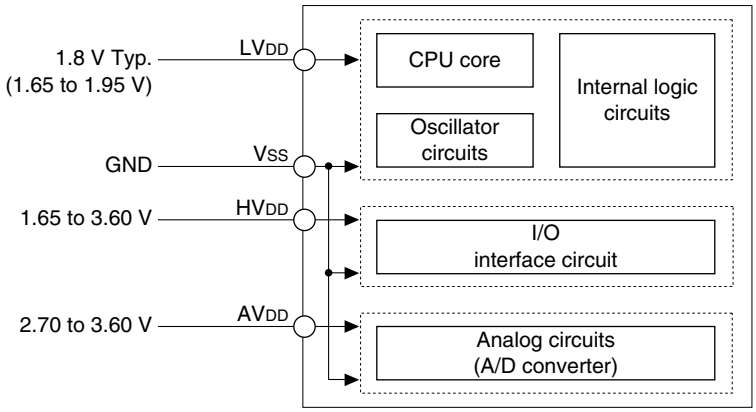


Figure 4.1.1 Power Supply System

4.2 Operating Voltage (LV_{DD}, V_{SS})

The core CPU and internal logic circuits of the S1C17003 operate with a voltage supplied between the LV_{DD} and V_{SS} pins.
The following operating voltage can be used:

LV_{DD} = 1.65 V to 1.95 V (1.80 V ± 0.15 V, V_{SS} = GND)

Note: The S1C17003 TQFP package has five LV_{DD} pins and nine V_{SS} pins; the WCSP package has two LV_{DD} pins and three V_{SS} pins. Be sure to supply the operating voltage to all the pins. Do not open any of them.

4.3 Power Supply for I/O Interface (HV_{DD})

The HV_{DD} voltage is used for interfacing with external I/O signals. For the output interface of the S1C17003, the HV_{DD} voltage is used as high level and the V_{SS} voltage as low level. The V_{SS} pin is used for the ground common with LV_{DD}. The following voltage is enabled for HV_{DD}:

HV_{DD} = 1.65 V to 3.60 V (V_{SS} = GND)

Notes: • The S1C17003 TQFP package has five HV_{DD} pins; the WCSP package has two HV_{DD} 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 LV_{DD}.

4.4 Power Supply for Analog Circuits (AV_{DD})

The analog power supply pin (AV_{DD}) is provided separately from the LV_{DD} and HV_{DD} pins in order that the digital circuits do not affect the analog circuit (A/D converter). The AV_{DD} pin is used to supply an analog power voltage and the V_{SS} pin is used as the analog ground.
The following voltage is enabled for AV_{DD}:

AV_{DD} = 2.70 V to 3.60 V or 1.65 V to 3.60 V (Note) (V_{SS} = GND)

Notes: • Be sure to supply a voltage within the range from 1.65 to 3.60 V to the AV_{DD} pin even if the analog circuit is not used. It is not necessary to supply a voltage same as the HV_{DD} level.

• The AV_{DD} 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 AV_{DD} 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.

4.5 Precautions on Power Supply

Power-on sequence

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

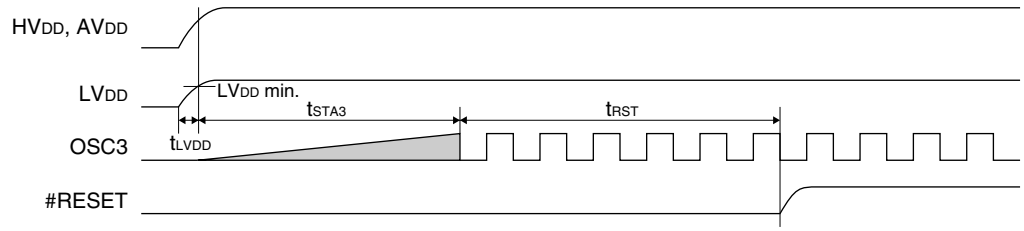


Figure 4.5.1 Power-On Sequence

(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.

5 Initial Reset

5.1 Initial Reset Factors

Shown below are the three different initial reset factors for initializing S1C17003 internal circuits.

(1) External initial reset via #RESET pin
(2) External initial reset via P0 port (pins P00 to P03) key entry (set by software)
(3) Internal initial reset via watchdog timer (set by software)

Figure 5.1.1 illustrates the initial reset circuit configuration.

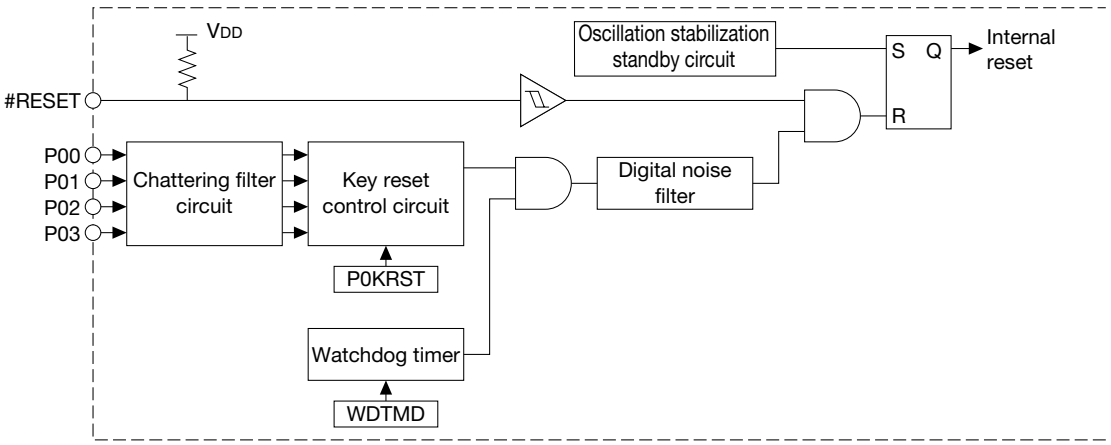


Figure 5.1.1: Initial reset circuit configuration

The CPU and peripheral circuits are initialized by initial reset factors. The CPU begins reset processing once the factors are canceled.
This causes the reset vector to be read from the start of the vector table, and the program (initialization routine) starting at that address to be executed.

5.1.1 #RESET pin

Initial resetting is possible by inputting external Low level to the #RESET pin.
To initialize the S1C17003 reliably, the #RESET pin must be maintained at Low level for at least the specified duration after the power supply voltage rises. (Refer to "26.4 Input/Output Terminal Characteristics")
Initial resetting is canceled if the #RESET input changes from Low to High, and the CPU begins reset interrupt processing.
The #RESET pin incorporates a pull-up resistance.

5.1.2 P0 Port Key-Entry Reset

Initial resetting is possible by inputting external Low level simultaneously to the ports (P00 to P03) selected by software. The ports can be selected by P0KRST[1:0] (D[1:0]/P0_KRST register).

∗ **P0KRST[1:0]**: P0 Port Key-Entry Reset Configuration Bits in the P0 Port Key-Entry Reset Configuration (P0_KRST) Register (D[1:0]/0x5209)

Table 5.1.2.1: P0 port key-entry reset settings

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

For example, initial reset is applied when input to the four ports P00 to P03 is Low level simultaneously if P0KRST[1:0] is set to 0x3.

Note:
- **Make sure the specified ports are not simultaneously switched to Low during normal operations when using the P0 port key-entry reset function.**
- **The P0 port key-entry reset function is enabled by software and cannot be used to perform a reset at power-on.**
- **The P0 port key-entry reset function cannot be used in SLEEP state.**

5.1.3 Reset by Watchdog Timer

The S1C17003 incorporates a watchdog timer to detect runaway CPU. If the watchdog timer is not reset by software every 4 seconds (with this failure indicating a runaway CPU), the timer overflows, generating an NMI or reset. A reset is generated by writing "1" to WDTMD (D1/WDT_ST register). (NMI is generated if WDTMD is 0.)

∗**WDTMD: NMI/Reset Mode Select Bit in the Watchdog Timer Status (WDT_ST) Register (D1/0x5041)**

For detailed information on the watchdog timer, refer to "17 Watchdog Timer (WDT)."

Note:
- **When using the reset function with the watchdog timer, to prevent accidental resetting, take care to program so that the watchdog timer is reset every four seconds.**
- **The watchdog timer reset function is enabled by software and cannot be used to perform a reset at power-on.**

5.2 Initial Reset Sequence

CPU startup waits for the oscillation stabilization standby time to expire after resetting is cancelled via the #RESET pin at power-on. Figure 5.2.1 illustrates the sequence of operations after canceling the initial reset. The CPU starts up in sync with the fosc3 (internal oscillation circuit) clock after the reset is canceled.

*f_{OSC3}: OSC3 clock frequency

Note: The oscillation stabilization standby time does not include the oscillation start time. The time may be longer than that shown between power-on or SLEEP cancellation and instruction execution.

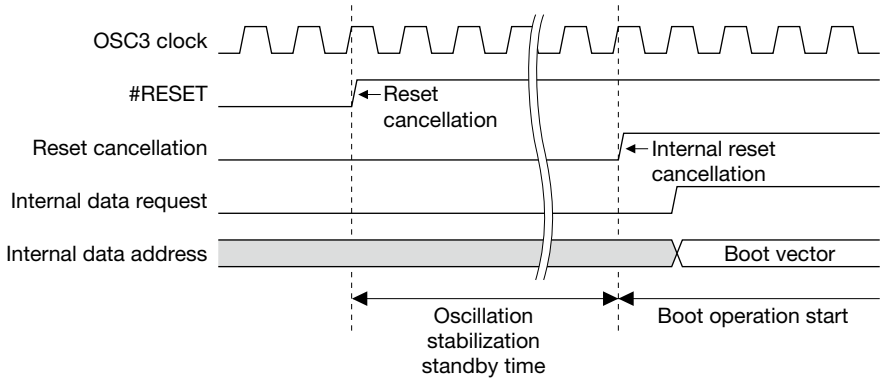


Figure 5.2.1: Sequence of operations after initial reset cancellation

5.3 Initial Settings at Initial Resetting

The CPU internal register is initialized by initial resetting, as shown below.

R0 to R7: 0x0
PSR: 0x0 (interrupt level = 0, interrupt prohibited)
SP: 0x0
PC: Reset vector at start of vector table is loaded by reset processing.

The internal RAM should be initialized via software, since it is not initialized by initial resetting.

The internal peripheral circuits are initialized in accordance with their particular specifications. They should be reset via software, if necessary. For detailed information on initial values after initial resetting, refer to the I/O register list in the Appendix or the respective peripheral circuit descriptions.

6 Interrupt Controller

6.1 ITC Configuration

The ITC enables the interrupt level (priority) for determining the processing sequence when multiple maskable interrupts occur simultaneously to be set for each interrupt type separately. For details on the maskable interrupt types, refer to the vector table shown in the next page.
Each interrupt type has the number of interrupt factors as shown in parentheses in the table mentioned above. Settings to permit or prohibit interrupt for different factors are set by the respective peripheral module registers.
For specific information on interrupt factors and their control, refer to the peripheral module explanations.
Figure 6.1.1 illustrates the interrupt system configuration.

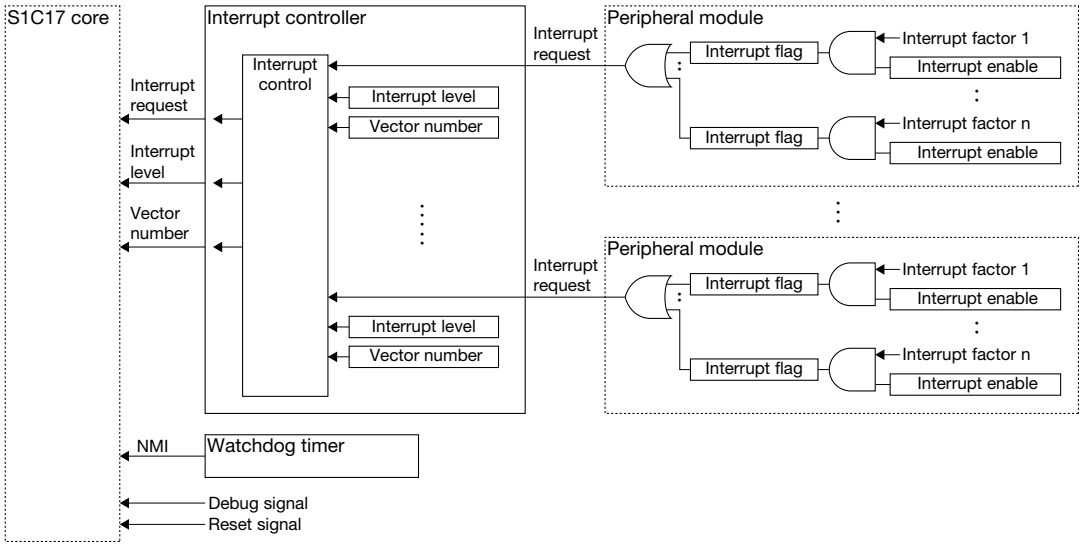


Figure 6.1.1: Interrupt system

6.2 Vector Table

The vector table contains the vectors (processing routine start addresses) for interrupt processing routines. When an interrupt occurs, the S1C17 core reads the vector corresponding to the interrupt and executes that processing routine. The base (top) address for the vector table can be set using the MISC_TTBRL and MISC_TTBRH registers (0x5328 and 0x532a) (See "2.4 Vector Table"). "TTBR" in Table 6.2.1 indicates the values set for these registers. The MISC_TTBRL and MISC_TTBRH registers are set to the 0x8000 address after initial resetting. Table 6.2.1 shows the S1C17003 vector table.

Table 6.2.1: Vector table

Vector No./ Software interrupt No.	Vector address	Hardware interrupt name	Hardware interrupt factor	Priority	Mask
0 (0x00)	TTBR + 0x00	Reset	• Low input to #RESET pin • Watchdog timer overflow *2	1	impossible
1 (0x01)	TTBR + 0x04	Irregular address interrupt	Memory access instruction	2	
–	(0xfffc00)	Debug interrupt	brk instruction etc.	3	
2 (0x02)	TTBR + 0x08	NMI	Watchdog timer overflow *2	4	
3 (0x03)	TTBR + 0x0c	Compiler (reserved)	Use simulation library of C compiler	–	
4 (0x04)	TTBR + 0x10	P0 port interrupt	P00 to P07 port input	High *1	Possible
5 (0x05)	TTBR + 0x14	P1 port interrupt	P10 to P17 port input	↑	
6 (0x06)	TTBR + 0x18	Stopwatch timer interrupt	• Timer 100 Hz signal • Timer 10 Hz signal • Timer 1 Hz signal		
7 (0x07)	TTBR + 0x1c	Clock timer interrupt	• Timer 32 Hz signal • Timer 8 Hz signal • Timer 2 Hz signal • Timer 1 Hz signal		
8 (0x08)	TTBR + 0x20	8-bit OSC1 timer interrupt	Compare match		
9 (0x09)	TTBR + 0x24	reserved	–		
10 (0x0a)	TTBR + 0x28	reserved	–		
11 (0x0b)	TTBR + 0x2c	PWM timer Ch. 0 interrupt	• Compare A • Compare B		
12 (0x0c)	TTBR + 0x30	8-bit timer Ch.0/Ch.1 interrupt	Timer underflow		
13 (0x0d)	TTBR + 0x34	16-bit timer Ch.0 interrupt	Timer underflow		
14 (0x0e)	TTBR + 0x38	16-bit timer Ch.1 interrupt	Timer underflow		
15 (0x0f)	TTBR + 0x3c	16-bit timer Ch.2 interrupt	Timer underflow		
16 (0x10)	TTBR + 0x40	UART Ch.0 interrupt	• Transmit buffer empty • Receive buffer full • Receive error		
17 (0x11)	TTBR + 0x44	UART Ch.1 interrupt/ I^2C (slave)	• UART Ch.1 transmit buffer empty • UART Ch.1 receive buffer full • UART Ch.1 receive error • I^2C (slave) transmit buffer empty • I^2C (slave) receive buffer full • I^2C (slave) bus status change		
18 (0x12)	TTBR + 0x48	SPI interrupt	• Transmit buffer empty • Receive buffer full		
19 (0x13)	TTBR + 0x4c	I²C (master) interrupt	• Transmit buffer empty • Receive buffer full		
20 (0x14)	TTBR + 0x50	Remote controller interrupt	• Data length counter underflow • Input rising edge detection • Input falling edge detection		
21 (0x15)	TTBR + 0x54	reserved	–		
22 (0x16)	TTBR + 0x58	A/D converter interrupt	• Conversion finish • Conversion result override		
23 (0x17) : 31 (0x1f)	TTBR + 0x5c : TTBR + 0x7c	reserved : reserved	– : –	↓ Low *1	

*1: When same interrupt level is set
*2: Watchdog timer interrupt selects reset or NMI using software.

Vector numbers 4 to 8, 11 to 20, 22 are assigned maskable interrupts supported by the S1C17003.

6.3 Maskable Interrupt Control

6.3.1 Peripheral Module Interrupt Control Bit

The peripheral module causing the interrupt includes interrupt enable bits and interrupt flags for each interrupt cause. Setting the interrupt enable bit to 1 (interrupt permitted) sets the interrupt flag to 1, depending on the cause of the interrupt. The flag state is sent to the ITC as an interrupt request signal, generating an interrupt request to the S1C17 core. The corresponding interrupt enable bits should be set to 0 for those causes for which interrupts are not desired. In this case, the interrupt flag will not be set to 1, even if the interrupt cause occurs, and the interrupt request signal will not be activated to the ITC.
Interrupt flags set to 1 must be reset within the interrupt processing routine after the interrupt has occurred. The ITC will generate the same interrupt again once the interrupt processing routine has been ended by the reti instruction with the interrupt flag still set to 1, since it detects interrupt requests using the signal level.
For specific information on interrupt causes, interrupt flags, and interrupt enable bits, refer to the individual peripheral module descriptions.

6.3.2 ITC Interrupt Request Processing

On receiving an interrupt signal from a peripheral module, the ITC sends interrupt request, interrupt level, and vector number signals to the S1C17 core.
Vector numbers are determined by the ITC internal hardware for each interrupt cause, as shown in Table 6.2.1.
The interrupt level is a value used by the S1C17 core to compare with the IL bit (PSR). This interrupt level is used in the S1C17 core to prohibit subsequently occurring interrupts with the same or lower level. (See section 6.3.3.)
The default ITC settings are level 0 for all maskable interrupts. Interrupt requests are not accepted by the S1C17 core if the level is 0.
The ITC includes control bits for selecting the interrupt level, and these can be set to between 0 (low) and 7 (high) interrupt levels for each interrupt type.

Table 6.3.2.1: Interrupt level setting bits

Hardware interrupt	Interrupt level setting bit	Register address
P0 port interrupt	ILV0[2:0] (D[2:0]/ITC_LV0 register)	0x4306
P1 port interrupt	ILV1[2:0] (D[10:8]/ITC_LV0 register)	0x4306
Stopwatch timer interrupt	ILV2[2:0] (D[2:0]/ITC_LV1 register)	0x4308
Clock timer interrupt	ILV3[2:0] (D[10:8]/ITC_LV1 register)	0x4308
8-bit OSC1 timer interrupt	ILV4[2:0] (D[2:0]/ITC_LV2 register)	0x430a
reserved	ILV5[2:0] (D[10:8]/ITC_LV2 register)	0x430a
reserved	ILV6[2:0] (D[2:0]/ITC_LV3 register)	0x430c
PWM timer Ch.0 interrupt	ILV7[2:0] (D[10:8]/ITC_LV3 register)	0x430c
8-bit timer Ch.0/Ch.1 interrupt	ILV8[2:0] (D[2:0]/ITC_LV4 register)	0x430e
16-bit timer Ch.0 interrupt	ILV9[2:0] (D[10:8]/ITC_LV4 register)	0x430e
16-bit timer Ch.1 interrupt	ILV10[2:0] (D[2:0]/ITC_LV5 register)	0x4310
16-bit timer Ch.2 interrupt	ILV11[2:0] (D[10:8]/ITC_LV5 register)	0x4310
UART Ch.0 interrupt	ILV12[2:0] (D[2:0]/ITC_LV6 register)	0x4312
UART Ch.0/I^2C (slave) interrupt	ILV13[2:0] (D[10:8]/ITC_LV6 register)	0x4312
SPI interrupt	ILV14[2:0] (D[2:0]/ITC_LV7 register)	0x4314
I^2C (master) interrupt	ILV15[2:0] (D[10:8]/ITC_LV7 register)	0x4314
Remote controller interrupt	ILV16[2:0] (D[2:0]/ITC_LV8 register)	0x4316
reserved	ILV17[2:0] (D[10:8]/ITC_LV8 register)	0x4316
A/D Conveter interrupt	ILV18[2:0] (D[2:0]/ITC_LV9 register)	0x4318
reserved	ILV19[2:0] (D[10:8]/ITC_LV9 register)	0x4318

If interrupt requests are input to the ITC simultaneously from multiple peripheral modules, the ITC outputs the interrupt request with the highest priority to the S1C17 core in accordance with the following conditions.
1. Interrupts with the highest interrupt level take precedence.
2. If multiple interrupt requests are input with the same interrupt level, the interrupt with the lowest vector number takes precedence.

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

6.3.3 S1C17 Core Interrupt Processing

Maskable interrupts for the S1C17 core occur when all of the following conditions are met:

- Interrupts are permitted by the interrupt control bit inside the peripheral module.
- The PSR (S1C17 core internal processor status register) IE (interrupt enable) bit has been set to 1.
- The interrupt factor has a higher interrupt level set than that set for the PSR IL (interrupt level).
- No other interrupt factors having higher procedence (e.g., NMI) are present.

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

The S1C17 core samples interrupt requests for each cycle. On accepting an interrupt request, the S1C17 core switches to interrupt processing when execution of the current instruction is complete.
Interrupt processing involves the following steps:

(1) The PSR and current program counter (PC) value is moved to the stack.
(2) The PSR IE bit is reset to 0 (preventing subsequent maskable interrupts).
(3) The PSR IL is set to the received interrupt level. (The NMI does not affect interrupt levels.)
(4) The vector for the interrupt factor occurring is loaded to the PC to execute the interrupt processing routine.

When an interrupt is received, (2) prevents subsequent maskable interrupts. Setting the IE bit to 1 within the interrupt processing routine allows handling of multiple interrupts. In this case, IL is changed by (3), and only interrupts with higher levels than those already being processed will be accepted.
Ending interrupt processing routines using a reti instruction returns the PSR to the state before the interrupt. The program resumes processing following the instruction being executed at the time the interrupt occurred via the next branch.

6.4 NMI

The S1C17003 can generate NMIs (non-maskable interrupts) using the watchdog timer. The vector number for NMIs is 2, and the vector address is set in the vector table initial address + 8 bytes. These interrupts take precedence over other interrupt factors and are accepted unconditionally by the S1C17 core.

For detailed information on generating NMIs, refer to “17 Watchdog Timer (WDT).”

6.5 Software Interrupts

Interrupts can be generated via software with S1C17 core int *imm5* or intl *imm5* and *imm3* instructions. The vector table vector number (0 to 31) is specified by the operand immediate *imm5*. With the intl instruction, *imm3* can be used to specify an interrupt level (0 to 7) for the PSR IL fields.
Details of the processor interrupt processing are the same as for when an interrupt generated by hardware occurs.

6.6 HALT and SLEEP Mode Cancellation

HALT or SLEEP mode is released by the following signals, and the CPU starts up.

- Interrupt requests from the ITC to the CPU.
- The NMI from the watchdog timer.
- Device interrupts
- Reset

Note: When HALT or SLEEP mode is released by an interrupt request from the ITC to the CPU, the process branches to an interrupt routine immediately after the release if the CPU can permit interrupts. Otherwise, the process executes an instruction following the halt or slp instruction. The ITC interrupt level setting cannot mask the release of HALT or SLEEP mode.

For details, refer to "B.1 Clock Control Power Saving" in Appendix B.

6.7 Control Register Details

Table 6.7.1: ITC registers

Address	Register name		Function
0x4306	ITC_LV0	Interrupt Level Setup Register 0	P0 and P1 interrupt level setting
0x4308	ITC_LV1	Interrupt Level Setup Register 1	SWT and CT interrupt level setting
0x430a	ITC_LV2	Interrupt Level Setup Register 2	T8OSC1 interrupt level setting
0x430c	ITC_LV3	Interrupt Level Setup Register 3	T16E Ch.0 interrupt level setting
0x430e	ITC_LV4	Interrupt Level Setup Register 4	T8F Ch.0/Ch.1 and T16 Ch.0 interrupt level setting
0x4310	ITC_LV5	Interrupt Level Setup Register 5	T16 Ch.1 and Ch.2 interrupt level setting
0x4312	ITC_LV6	Interrupt Level Setup Register 6	UART Ch.0 and Ch.1/I^2C (slave) interrupt level setting
0x4314	ITC_LV7	Interrupt Level Setup Register 7	SPI and I^2C (master) interrupt level setting
0x4316	ITC_LV8	Interrupt Level Setup Register 8	REMC interrupt level setting
0x4318	ITC_LV9	Interrupt Level Setup Register 9	A/D interrupt level setting

The ITC registers are described in detail below. These are 16-bit registers.

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

0x4306: Interrupt Level Setup Register 0 (ITC_LV0)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Interrupt Level Setup Register 0 (ITC_LV0)	**0x4306** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**ILV1[2:0]**	P1 interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**ILV0[2:0]**	P0 interrupt level	0 to 7	0x0	R/W	

D[15:11] Reserved

D[10:8] ILV1[2:0]: P1 Port Interrupt Level Bits

Set the P1 port interrupt level (0 to 7). (Default: 0)

The S1C17 core does not accept interrupts with levels set lower than the PSR IL value.

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

If multiple interrupts occur at the same time permitted by the interrupt enable bit, the ITC sends the interrupt request with the highest level set by the ITC_LV*x* registers (0x4306 to 0x4316) to the S1C17 core.

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

The other interrupts are held until all have been accepted by the S1C17 core in descending order of priority.

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

D[7:3] Reserved

D[2:0] ILV0[2:0]: P0 Port Interrupt Level Bits

Set the P0 port interrupt level (0 to 7). (Default: 0)

Refer to the ILV1[2:0] (D[10:8]) description.

0x4308: Interrupt Level Setup Register 1 (ITC_LV1)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Interrupt Level Setup Register 1 (ITC_LV1)	**0x4308** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**ILV3[2:0]**	CT interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**ILV2[2:0]**	SWT interrupt level	0 to 7	0x0	R/W	

D[15:11] Reserved

D[10:8] ILV3[2:0]: Clock Timer Interrupt Level Bits

Set the clock timer interrupt level (0 to 7). (Default: 0)
Refer to the ITC_LV0 register (0x4306) ILV1[2:0] (D[10:8]) description.

D[7:3] Reserved

D[2:0] ILV2[2:0]: Stopwatch Timer Interrupt Level Bits

Set the stopwatch timer interrupt level (0 to 7). (Default: 0)
Refer to the ITC_LV0 register (0x4306) ILV1[2:0] (D[10:8]) description.

0x430a: Interrupt Level Setup Register 2 (ITC_LV2)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Interrupt Level Setup Register 2 (ITC_LV2)	**0x430a** (16 bits)	D15–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**ILV4[2:0]**	T8OSC1 interrupt level	0 to 7	0x0	R/W	

D[15:3] Reserved

D[2:0] ILV4[2:0]: 8-bit OSC1 Timer Interrupt Level Bits

Set the 8-bit OSC1 timer interrupt level (0 to 7). (Default: 0)
Refer to the ITC_LV0 register (0x4306) ILV1[2:0] (D[10:8]) description.

0x430c: Interrupt Level Setup Register 3 (ITC_LV3)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Interrupt Level Setup Register 3 (ITC_LV3)	**0x430c** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**ILV7[2:0]**	T16E Ch.0 interrupt level	0 to 7	0x0	R/W	
		D7–0	**–**	reserved	–	–	–	0 when being read.

D[15:11] Reserved

D[10:8] ILV7[2:0]: PWM & Capture Timer Interrupt Level Bits

Set the PWM timer interrupt level (0 to 7). (Default: 0)
Refer to the ITC_LV0 register (0x4306) ILV1[2:0] (D[10:8]) description.

D[7:0] Reserved

0x430e: Interrupt Level Setup Register 4 (ITC_LV4)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Interrupt Level Setup Register 4 (ITC_LV4)	**0x430e** (16 bits)	D15–11	–	reserved	–	–	–	0 when being read.
		D10–8	**ILV9[2:0]**	T16 Ch.0 interrupt level	0 to 7	0x0	R/W	
		D7–3	–	reserved	–	–	–	0 when being read.
		D2–0	**ILV8[2:0]**	T8F Ch.0/Ch.1 interrupt level	0 to 7	0x0	R/W	

D[15:11] Reserved

D[10:8] ILV9[2:0]: 16-bit Timer Ch.0 Interrupt Level Bits

Set the 16-bit timer Ch.0 interrupt level (0 to 7). (Default: 0)
Refer to the discussion of ITC_LV0 register (0x4306) ILV1[2:0] (D[10:8]).

D[7:3] Reserved

D[2:0] ILV8[2:0]: 8-bit Timer Ch.0/Ch.1 Interrupt Level Bits

Set the 8-bit timer interrupt level (0 to 7). (Default: 0)
Refer to the ITC_LV0 register (0x4306) ILV1[2:0] (D[10:8]) description.

0x4310: Interrupt Level Setup Register 5 (ITC_LV5)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Interrupt Level Setup Register 5 (ITC_LV5)	**0x4310** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**ILV11[2:0]**	T16 Ch.2 interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**ILV10[2:0]**	T16 Ch.1 interrupt level	0 to 7	0x0	R/W	

D[15:11] Reserved

D[10:8] ILV11[2:0]: 16-bit Timer Ch.2 Interrupt Level Bits

Set the 16-bit timer Ch.2 interrupt level (0 to 7). (Default: 0)
Refer to the ITC_LV0 register (0x4306) ILV1[2:0] (D[10:8]) description.

D[7:3] Reserved

D[2:0] ILV10[2:0]: 16-bit Timer Ch.1 Interrupt Level Bits

Set the 16-bit timer Ch.1 interrupt level (0 to 7). (Default: 0)
Refer to the ITC_LV0 register (0x4306) ILV1[2:0] (D[10:8]) description.

0x4312: Interrupt Level Setup Register 6 (ITC_LV6)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Interrupt Level Setup Register 6 (ITC_LV6)	**0x4312** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**ILV13[2:0]**	UART Ch.1 (slave) interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**ILV12[2:0]**	UART Ch.0 interrupt level	0 to 7	0x0	R/W	

D[15:11] Reserved

D[10:8] ILV13[2:0]: UART Ch.1/I²C (slave) Interrupt Level Bits

Set the UART Ch.1 or I²C (slave) interrupt level (0 to 7). (Default: 0)
Refer to the ITC_LV0 register (0x4306) ILV1[2:0] (D[10:8]) description.

D[7:3] Reserved

D[2:0] ILV12[2:0]: UART Ch.0 Interrupt Level Bits

Set the UART Ch.0 interrupt level (0 to 7). (Default: 0)
Refer to the ITC_LV0 register (0x4306) ILV1[2:0] (D[10:8]) description.

0x4314: Interrupt Level Setup Register 7 (ITC_LV7)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Interrupt Level Setup Register 7 (ITC_LV7)	**0x4314** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**ILV15[2:0]**	I^2C (master) interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**ILV14[2:0]**	SPI interrupt level	0 to 7	0x0	R/W	

D[15:11] Reserved

D[10:8] ILV15[2:0]: I^2C (master) Interrupt Level Bits

Set the I^2C interrupt level (0 to 7). (Default: 0)
Refer to the ITC_LV0 register (0x4306) ILV1[2:0] (D[10:8]) description.

D[7:3] Reserved

D[2:0] ILV14[2:0]: SPI Interrupt Level Bits

Set the SPI interrupt level (0 to 7). (Default: 0)
Refer to the ITC_LV0 register (0x4306) ILV1[2:0] (D[10:8]) description.

0x4316: Interrupt Level Setup Register 8 (ITC_LV8)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Interrupt Level Setup Register 8 (ITC_LV8)	**0x4316** (16 bits)	D15–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**ILV16[2:0]**	REMC interrupt level	0 to 7	0x0	R/W	

D[15:3] Reserved

D[2:0] ILV16[2:0]: REMC Interrupt Level Bits

Set the remote controller interrupt level (0 to 7). (Default: 0)
Refer to the discussion of ITC_LV0 register (0x4306) ILV1[2:0] (D[10:8]).

0x4318: Interrupt Level Setup Register 9 (ITC_LV9)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Interrupt Level Setup Register 9 (ITC_LV9)	**0x4318** (16 bits)	D15–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**ILV18[2:0]**	A/D converter interrupt level	0 to 7	0x0	R/W	

D[15:3] Reserved

D[2:0] ILV18[2:0]: A/D Convertere Interrupt Level Bits

Set the A/D converter interrupt level (0 to 7). (Default: 0)
Refer to the discussion of ITC_LV0 register (0x4306) ILV1[2:0] (D[10:8]).

6.8 Precautions

To prevent the recurrence of interrupts due to the same interrupt factor, always reset the interrupt flag before permitting interrupts, resetting PSR, or executing the `reti` instruction.

7 Oscillator Circuit (OSC)

7.1 OSC Module Configuration

The S1C17003 contains two internal oscillator circuits (OSC3 and OSC1). The OSC3 oscillator circuit generates the main clock for high-speed operation of the S1C17 core and peripheral circuits. The OSC1 oscillator circuit generates a sub-clock for timer and low-power operations.
The OSC3 clock is selected as the system clock after initial resetting.
Oscillator circuit on/off switching and system clock selection (between OSC3 and OSC1) is controlled by software.
External clock output is also possible.
HSCLK (high-speed clock) can be used hereafter as an equivalent term as OSC3.

Figure 7.1.1 illustrates the clock system and OSC module configuration.

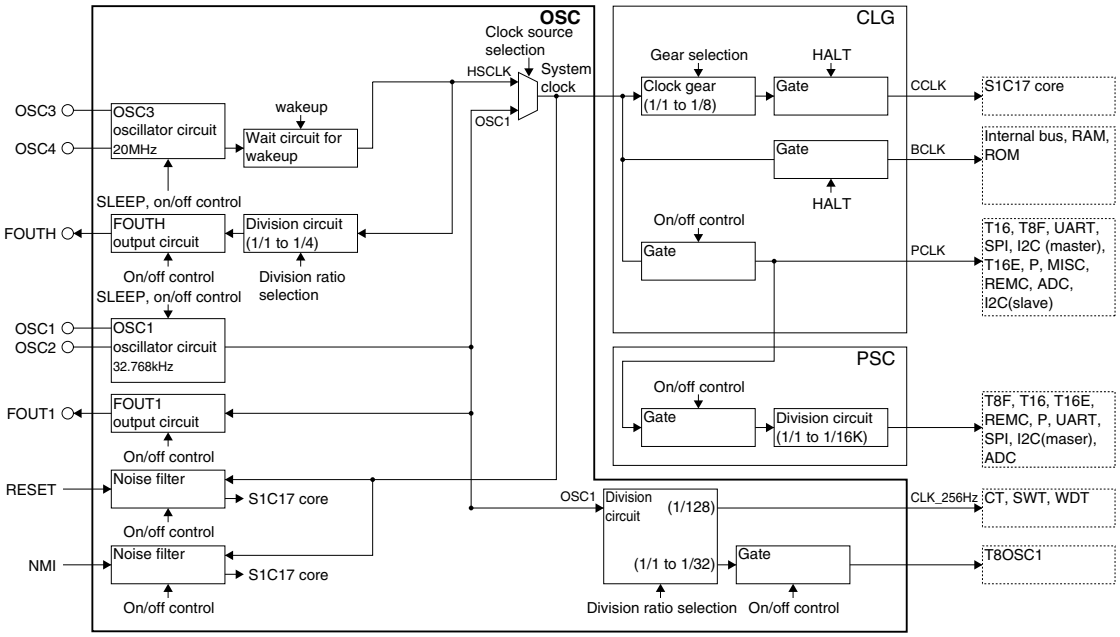


Figure 7.1.1: OSC module configuration

To reduce power consumption, control the clock in conjunction with processing and use standby mode. For more information on reducing power consumption, refer to “Appendix B: Power Saving.”

7.2 OSC3 Oscillator Circuit

OSC3 is a high-precision, high-speed oscillator circuit using crystal or ceramic oscillator. It generates the clock for S1C17 and peripheral circuits during initialization.
Figure 7.2.1 illustrates the OSC3 oscillator circuit configuration.

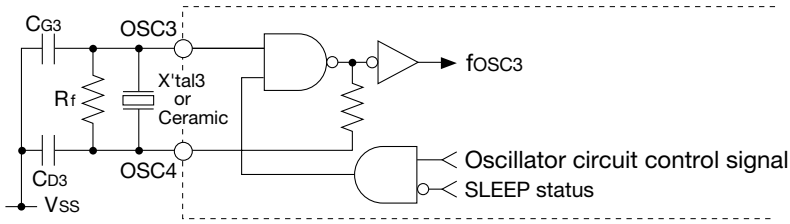


Figure 7.2.1: OSC3 oscillator circuit

A crystal oscillator (X'tal3) or ceramic oscillator (Ceramic) and feedback resistor (R_f) should be connected between the OSC3 and OSC4 pins. Additionally, two capacitors (C_{G3} and C_{D3}) should be connected between the OSC3/OSC4 pins and Vss.

OSC3 oscillation on/off

The OSC3 oscillator circuit stops oscillating if OSC3EN (D0/OSC_CTL register) is set to 0 and starts oscillating if set to 1. The OSC3 oscillator circuit stops oscillating even in SLEEP mode.

∗ **OSC3EN**: OSC3 Enable Bit in the Oscillation Control (OSC_CTL) Register (D0/0x5061)

After the initial resetting, OSC3EN is set to 1 and the OSC3 oscillator circuit is on. To use the OSC3 clock, the clock must also be switched, in addition to the on/off controls described above. For specific information on switching, see "7.4 Clock Switching."

Stabilization wait time at start of OSC3 oscillation

When using the OSC3 clock, the OSC3 oscillator circuit incorporates an oscillation stabilization wait timer to prevent malfunctions due to unstable clock operations at the start of OSC3 oscillation—e.g., when waking from SLEEP, or when the OSC3 oscillation circuit is switched on via software. The OSC3 clock is not fed to the system until the time set for this timer has elapsed.
Use the OSC3WT[1:0] (D[5:4]/OSC_CTL register) to select among four different oscillation stabilization wait times.

∗ **OSC3WT[1:0]**: OSC3 Wait Cycle Select Bits in the Oscillation Control (OSC_CTL) Register (D[5:4]/0x5061)

Table 7.2.1: OSC3 oscillation stabilization wait time settings

OSC3WT[1:0]	Oscillation stabilization wait time
0x3	128 cycles
0x2	256 cycles
0x1	512 cycles
0x0	1,024 cycles

(Default: 0x0)

This is set to 1,024 cycles (OSC3 clock) after initial resetting.

Note: The stability of oscillation depends on the oscillator and external add-on components. Full evaluation is required for configuring shorter stabilization wait time.
OSC3 clock system supply wait time =< OSC3 oscillation start time (max.) + OSC3 oscillation stabilization wait time.

External clock input of OSC3

The clock can be input to the OSC3 pin from external. To stop the external clock, stop it at the VSS level. For information about input clock waveforms, refer to "26 Electrical Characteristics."

7.3 OSC1 Oscillator Circuit

OSC1 is a high-precision, low-speed oscillator circuit using a 32.768 kHz crystal oscillator.
The OSC1 clock is generally used as the timer operation clock (for the clock timer, stopwatch timer, watchdog timer, and 8-bit OSC1 timer). It reduces power consumption and can be used as the system clock instead of the OSC3 clock when no high-speed processing is required.
Figure 7.3.1 illustrates the OSC1 oscillator circuit configuration.

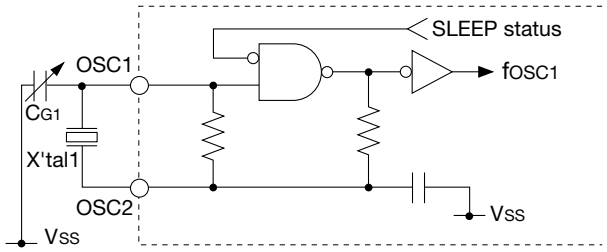


Figure 7.3.1: OSC1 oscillator circuit

A crystal oscillator (X'tal1) (typ. 32.768 kHz) should be connected between the OSC1 and OSC2 pins. Additionally, trimmer capacitor CG1 (0 to 25 pF) should be connected between the OSC1 pin and Vss.

OSC1 oscillation on/off

The OSC1 oscillator circuit stops oscillating if OSC1EN (D1/OSC_CTL register) is set to 0 and starts oscillating if set to 1. The OSC1 oscillator circuit stops oscillating even in SLEEP mode.

* **OSC1EN**: OSC1 Enable Bit in the Oscillation Control (OSC_CTL) Register (D1/0x5061)

Following initial resetting, OSC1EN is set to 0, and the OSC1 oscillator circuit is halted.

Stabilization wait time at start of OSC1 oscillation

The OSC1 oscillator circuit incorporates an oscillation stabilization wait timer to prevent malfunctions due to unstable clock operations at the start of OSC1 oscillation—for example, when power is first turned on, on awaking from SLEEP, or when the OSC1 oscillation circuit is turned on via software. The OSC1 clock does not feed the system for a period of 256 cycles after the start of oscillation.

OSC clock system supply wait time =< IOSC oscillation start time (max.) + OSC1 oscillation stabilization wait time.

Pin settings when OSC1 is not used

Keep the OSC1 and OSC2 pins open.

Note: Set OSC1EN (the D1/OSC_CTL register) to 0 while the OSC1 and OSC2 pins are kept open.

7.4 Clock Switching

The system clock select section of the S1C17003 consists of OSC1-HSCLK select. Figure 7.4.1 shows the configuration of the system clock select section.

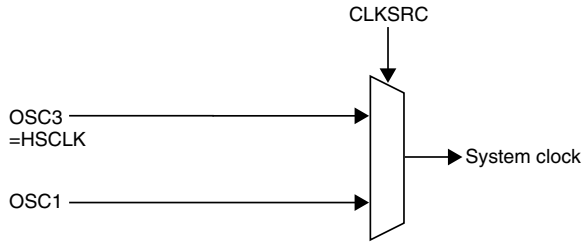


Figure 7.4.1: System clock select section

OSC1 HSCLK selection

The S1C17003 includes the OSC1 oscillator circuit to generate low-speed clocks. Either of the OSC1 or HSCLK can be selected for the system clock. HSCLK is selected when operation starts after the initial reset.
To select OSC1 for the system clock, turn on the OSC1 oscillator circuit (see section 7.4), and then write 1 to CLKSRC (D1/OSC_SRC register). To select HSCLK for the system clock, write 0 to SRCSRC while HSCLK is operating.

∗ **CLKSRC**: System Clock Source Select Bit in the Clock Source Select (OSC_SRC) Register (D0/0x5060)

Oscillator circuits other than selected for the system clock and are not used as the operating clock for peripheral circuits can be stopped to reduce current consumption.

Notes:
- To select OSC1_HSCLK, both of the OSC1 and HSCLK must be operating. Writing to HSCLKSEL while one of them is not operating does not switch the system clock, and does not change the CLKSRC value.
- The oscillator circuit selected for the system clock cannot be turned off.
- Sequential access of write/read to the CLKSRC register is prohibited. Between write and read access instructions to CLKSRC, insert at least one instruction unrelated to access to the CLKSRC register.
- It takes one HSCLK cycle at minimum or one OSC1 cycle at maximum to switch clocks from OSC1 to HSCLK and vice versa.

7.5 8-bit OSC1 Timer Clock Control

The OSC module consists of a division circuit for generating the 8-bit OSC1 timer operation clock and a device for controlling the feed. The 8-bit OSC1 timer is a programmable timer that operates only using the OSC1 division clock. For detailed information, refer to "14 8-bit OSC1 Timer (T8OSC1)."

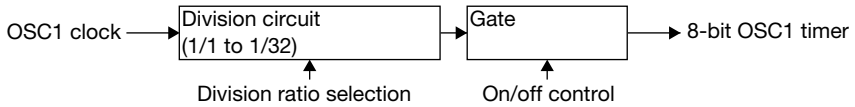


Figure 7.5.1: 8-bit OSC1 timer clock control circuit

Clock division ratio selection

Select the OSC1 clock division ratio using T8O1CK[2:0] (D[3:1]/OSC_T8OSC1 register)

∗ **T8O1CK[2:0]**: T8OSC1 Clock Division Ratio Select Bits in the T8OSC1 Clock Control (OSC_T8OSC1) Register (D[3:1]/0x5065)

Table 7.5.1: T8OSC1 clock division ratio selection

T8O1CK[2:0]	Division ratio
0x7 to 0x6	Reserved
0x5	OSC1-1/32
0x4	OSC1-1/16
0x3	OSC1-1/8
0x2	OSC1-1/4
0x1	OSC1-1/2
0x0	OSC1-1/1

(Default: 0x0)

Clock feed control

The clock feed to the 8-bit OSC1 timer is controlled using T8O1CE (D0/OSC_T8OSC1 register).
The T8O1CE default setting is 0, which stops the clock feed. Setting T8O1CE to 1 sends the clock generated as above to the 8-bit OSC1 timer. Stop the clock feed to reduce power consumption if 8-bit OSC1 timer operation is not required.

∗ **T8O1CE**: T8OSC1 Clock Enable Bit in the T8OSC1 Clock Control (OSC_T8OSC1) Register (D0/0x5065)

Note: Change of clock division ratio selection (T8O1CK [2:0](D[3:1]/0x5063)) should be executed when T8O1CE(D0/0x5065) is 0 and the clock to 8-bit OSC1 timer is in "Stop" state.

7.6 Clock External Output (FOUTH, FOUT1)

The HSCLKdivision clock (FOUTH) and OSC1 clock (FOUT1) can be output to devices outside the chip.

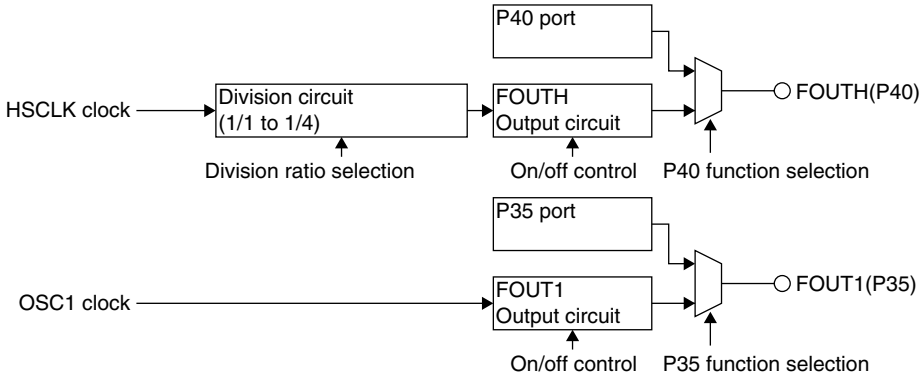


Figure 7.6.1: Clock output circuit

FOUTH output

FOUTH is the HSCLK division clock.

Output pin setting

The FOUTH output pin is combined with the P40 port. This functions as the P40 port pin by default, so the pin function should be changed by writing 1 to P40MUX (D0/P4_PMUX register) if use is required for FOUTH output.

∗ **P40MUX**: P40 Port Function Select Bit in the P4 Port Function Select (P4_PMUX) Register (D0/0x52a8)

FOUTH clock frequency selection

Three different clock output frequencies can be selected. Select the division ratio for the OSC3 clock using FOUTHD[1:0] (D[3:2]/OSC_FOUT register).

∗ **FOUTHD[1:0]**: FOUTH Clock Division Ratio Select Bits in the FOUT Control (OSC_FOUT) Register (D[3:2]/0x5064)

Table 7.6.1: FOUTH clock division ratio selection

FOUTHD[1:0]	Division ratio
0x3	Reserved
0x2	OSC3-1/4
0x1	OSC3-1/2
0x0	OSC3-1/1

(Default: 0x0)

Clock output control

The clock output is controlled using the FOUTHE (D1/OSC_FOUT register). Setting FOUTHE to 1 outputs the FOUTH clock from the FOUTH pin. Setting it to 0 halts output.

∗ **FOUTHE**: FOUTH Output Enable Bit in the FOUT Control (OSC_FOUT) Register (D1/0x5064)

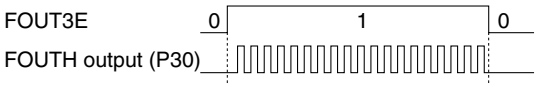


Figure 7.6.2: FOUTH output

Notes:
- Since the FOUTH signal is asynchronized with FOUTHE writing, switching output on or off will generate certain hazards.
- Change of the single selection (FOUTHD [1:0] (D[3:2]/0x5064) of FOUTH clock frequency should be executed when FOUTHE (D1/0x5064) is 0 and clock output is in “Stop” status.

FOUT1 output

FOUT1 is the OSC1 clock.

Output pin setting

The FOUT1 output pin is combined with the P35 port. This functions as the P35 port pin by default, so the pin function should be changed by writing 1 to P35MUX (D3/P1_PMUX register) if use is required for FOUT1 output.

* **P35MUX**: P35 Port Function Select Bit in the P3 Port Function Select (P3_PMUX) Register (D3-2/0x52a7)

Clock output control

The clock output is controlled using the FOUT1E (D0/OSC_FOUT register). Setting FOUT1E to 1 outputs the FOUT1 clock from the FOUT1 pin. Setting it to 0 halts output.

* **FOUT1E**: FOUT1 Output Enable Bit in the FOUT Control (OSC_FOUT) Register (D1/0x5064)

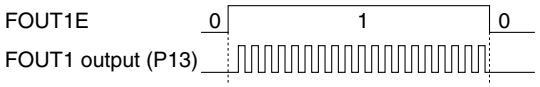


Figure 7.6.3: FOUT1 output

Note: Since the FOUT1 signal is asynchronized with FOUT1E writing, switching output on or off will generate certain hazards.

7.7 RESET and NMI Input Noise Filters

Since accidental activation of RESET or NMI by noise in the S1C17 core input signal will cause unintended resetting or NMI processing, the OSC module incorporates noise filters operated by the system clock. The filters remove noise from these signals before they reach the S1C17 core.
Separate noise filters are used for each signal. You can select to use or bypass them individually. All are active immediately after the initial resetting.

RESET input noise filter: Filters noise when RSTFE (D1/OSC_NFEN register) = 1; bypassed when RSTFE = 0
NMI input noise filter: Filters noise when NMIFE (D0/OSC_NFEN register) = 1; bypassed when NMIFE = 0

* **RSTFE**: Reset Noise Filter Enable Bit in the Noise Filter Enable (OSC_NFEN) Register (D1/0x5062)
* **NMIFE**: NMI Noise Filter Enable Bit in the Noise Filter Enable (OSC_NFEN) Register (D0/0x5062)

Notes:
- All noise filters should normally be enabled.
- The S1C17003 does not feature external NMI input pins, but the watchdog timer NMI request signal passes through these filters.

7.8 Control Register Details

Table 7.8.1: OSC register list

Address	Register name		Function
0x5060	OSC_SRC	Clock Source Select Register	Clock source selection
0x5061	OSC_CTL	Oscillation Control Register	Oscillation control
0x5062	OSC_NFEN	Noise Filter Enable Register	Noise filter on/off
0x5063	reserved	reserved	reserved
0x5064	OSC_FOUT	FOUT Control Register	Clock external output control
0x5065	OSC_T8OSC1	T8OSC1 Clock Control Register	8-bit OSC1 timer clock setting
0x5066	reserved	reserved	reserved
0x5067	reserved	reserved	reserved

The OSC module registers are described in detail below. These are 8-bit registers.

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

0x5060: Clock Source Select Register (OSC_SRC)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Clock Source Select Register (OSC_SRC)	0x5060 (8 bits)	D7–2	–	reserved	–				–	–	0 when being read.
		D1	HSCLKSEL	High-speed clock select	1	OSC3			1	R	1 when being read.
		D0	CLKSRC	System clock source select	1	OSC1	0	HSCLK	0	R/W	

D[7:2] Reserved

D1 HSCLKSEL: High-speed Clock Select Bit

Selects the high-speed clock (HSCLK).
1 (R): OSC3 (fixed)

D0 CLKSRC: System Clock Source Select Bit

Selects the system clock source.
1 (R/W): OSC1
0 (R/W): HSCLK (default)

HSCLK (OSC3) is selected for normal (high-speed) operations. If the HSCLK clock is not required, OSC1 can be set as the system clock and HSCLK (OSC3) stopped to reduce power consumption.

Notes:
- If the system clock is switched from HSCLK to OSC1 immediately after starting OSC1 oscillation, the system clock will stop until the OSC1 clock starts up (for the OSC1 clock 256-cycle period).
- Continuous access of write and read to CLKSRC register (DO/Ox5060) is prohibited. Enter at least one instruction that is not related to access to CLKSRC register between write and read.

0x5061: Oscillation Control Register (OSC_CTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Oscillation Control Register (OSC_CTL)	0x5061 (8 bits)	D7–6	–	reserved	–	–	–	0 when being read.
		D5–4	OSC3WT[1:0]	OSC3 wait cycle select	OSC3WT[1:0] / Wait cycle: 0x3 / 128 cycles; 0x2 / 256 cycles; 0x1 / 512 cycles; 0x0 / 1024 cycles	0x0	R/W	
		D3-2	–	reserved	–	–	–	0 when being read.
		D1	OSC1EN	OSC1 enable	1 Enable 0 Disable	0	R/W	
		D0	OSC3EN	OSC3 enable	1 Enable 0 Disable	1	R/W	

D[5:4] OSC3WT[1:0]: OSC3 Wait Cycle Select Bits

An oscillation stabilization wait timer is set to prevent malfunctions due to unstable clock operation at the start of OSC3 oscillation.

The OSC3 clock is not fed to the system immediately after OSC3 oscillation starts—for example, when power is first turned on, on awaking from SLEEP, or when the OSC3 oscillation circuit is turned on via software—until the time set here has elapsed.

Table 7.8.2: OSC3 oscillation stabilization wait time settings

OSC3WT[1:0]	Oscillation stabilization wait time
0x3	128 cycles
0x2	256 cycles
0x1	512 cycles
0x0	1,024 cycles

(Default: 0x0)

This is set to 1,024 cycles (OSC3 clock) after initial resetting. The CPU does not begin operating immediately after resetting until this time has elapsed.

Note: The OSC3 oscillation start time depends on the oscillator and externally connected components. The time should be set with an adequate oscillation stabilization wait time. Refer to the typical oscillation start times specified in "26 Electrical Characteristics."

D[3:2] Reserved

D1 OSC1EN: OSC1 Enable Bit

Permits or prohibits OSC1 oscillator circuit operation.

1 (R/W): Permitted (on)

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

Notes:
- The OSC1 oscillator circuit cannot be stopped if the OSC1 clock is being used as the system clock.
- The OSC1 clock is not fed to the system for 256 cycles to prevent malfunctions immediately after OSC1 oscillation is started by changing the OSC1EN setting from 0 to 1.

D0 OSC3EN: OSC3 Enable Bit

Permits or prohibits OSC3 oscillator circuit operation.

1 (R/W): Permitted (on) (default)

0 (R/W): Prohibited (off)

Note: The OSC3 oscillator circuit cannot be stopped if the OSC3 clock is being used as the system clock.

0x5062: Noise Filter Enable Register (OSC_NFEN)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Noise Filter Enable Register (OSC_NFEN)	0x5062 (8 bits)	D7–2	–	reserved	–	–	–	0 when being read.
		D1	RSTFE	Reset noise filter enable	1 Enable 0 Disable	1	R/W	
		D0	NMIFE	NMI noise filter enable	1 Enable 0 Disable	0	R/W	

D[7:2] Reserved

D1 RSTFE: Reset Noise Filter Enable Bit

Enables or disables the RESET input noise filter.
1 (R/W): Enabled (noise filtering) (default)
0 (R/W): Disabled (bypass)

This noise filter inputs only RESET pulses of not less than 16 cycles of the system clock (OSC3 or OSC1 clock) to the S1C17 core. This should normally be enabled.

D0 NMIFE: NMI Noise Filter Enable Bit

Enables or disables the NMI input noise filter.
1 (R/W): Enabled (noise filtering)
0 (R/W): Disabled (bypass) (default)

This noise filter inputs only NMI pulses of not less than 16 cycles of the system clock (OSC3 or OSC1 clock) to the S1C17 core. Pulses having widths of less than 16 cycles are filtered out as noise. This should normally be enabled.

Note: The S1C17003 does not feature external NMI input pins, but the watchdog timer NMI request signal passes through these filters.

0x5064: FOUT Control Register (OSC_FOUT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
FOUT Control Register (OSC_FOUT)	0x5064 (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.
		D3–2	FOUTHD[1:0]	FOUTH clock division ratio select	FOUTHD[1:0] / Division ratio 0x3 reserved 0x2 HSCLK-1/4 0x1 HSCLK-1/2 0x0 HSCLK-1/1	0x0	R/W	
		D1	FOUTHE	FOUTH output enable	1 Enable 0 Disable	0	R/W	
		D0	FOUT1E	FOUT1 output enable	1 Enable 0 Disable	0	R/W	

D[7:4] Reserved

D[3:2] FOUTHD[1:0]: FOUTH Clock Division Ratio Select Bits

Select the HSCLK clock division ratio to set the FOUTH clock frequency.

Table 7.8.3: FOUTH clock division ratio selection

FOUTHD[1:0]	Division ratio
0x3	Reserved
0x2	OSC3-1/4
0x1	OSC3-1/2
0x0	OSC3-1/1

(Default: 0x0)

D1 FOUTHE: FOUTH Output Enable Bit

Permits or prohibits FOUTH clock (HSCLK division clock) external output.

1 (R/W): Permitted (on)

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

Setting FOUTHE to 1 outputs the FOUTH clock from the FOUTH pin. Setting it to 0 stops the output.

D0 FOUT1E: FOUT1 Output Enable Bit

Permits or prohibits FOUT1 clock (OSC1 clock) external output.

1 (R/W): Permitted (on)

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

Setting FOUT1E to 1 outputs the FOUT1 clock from the FOUT1 pin. Setting it to 0 stops the output.

Note: Change of the single selection (FOUTHD [1:0] (D[3:2]/0x5064) of FOUTH clock frequency should be executed when FOUTHE (D1/0x5064) is 0 and clock output is in “Stop” status.

0x5065: T8OSC1 Clock Control Register (OSC_T8OSC1)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
T8OSC1 Clock Control Register (OSC_T8OSC1)	0x5065 (8 bits)	D7–4	–	reserved	–		–	–	0 when being read.
		D3–1	T8O1CK[2:0]	T8OSC1 clock division ratio select	T8O1CK[2:0]	Division ratio	0x0	R/W	
					0x7–0x6	reserved			
					0x5	OSC1-1/32			
					0x4	OSC1-1/16			
					0x3	OSC1-1/8			
					0x2	OSC1-1/4			
					0x1	OSC1-1/2			
					0x0	OSC1-1/1			
		D0	T8O1CE	T8OSC1 clock output enable	1 Enable	0 Disable	0	R/W	

D[7:4] Reserved

D[3:1] T8O1CK[2:0]: T8OSC1 Clock Division Ratio Select Bits

Select the OSC1 clock division ratio and set the 8-bit OSC1 timer operation clock.

Table 7.8.4: T8OSC1 clock division ratio selection

T8O1CK[2:0]	Division ratio
0x7 to 0x6	Reserved
0x5	OSC1-1/32
0x4	OSC1-1/16
0x3	OSC1-1/8
0x2	OSC1-1/4
0x1	OSC1-1/2
0x0	OSC1-1/1

(Default: 0x0)

D0 T8O1CE: T8OSC1 Clock Output Enable Bit

Permits or prohibits clock feed to the 8-bit OSC1 timer.

1 (R/W): Permitted (on)

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

The T8O1CE default setting is 0, which stops the clock feed. Setting T8O1CE to 1 sends the clock selected by the above bit to the 8-bit OSC1 timer. Stop the clock feed to reduce power consumption if 8-bit OSC1 timer operation is not required.

Note: Change of clock division ratio selection (T8O1CK [2:0](D[3:1]/0x5063)) should be executed when T8O1CE(D0/0x5065) is 0 and the clock to 8-bit OSC1 timer is in “Stop” state.

7.9 Precautions

- The oscillation start time depends on the oscillator and externally connected components. The time should be set with an adequate OSC3 oscillation stabilization wait time. Refer to the typical oscillation start times specified in "26 Electrical Characteristics."
- Switching the system clock from HSCLK to OSC1 immediately after starting OSC1 oscillation will stop the system clock until the OSC1 clock starts up (for the OSC1 clock 256-cycle period).
- The OSC3 oscillator circuit cannot be stopped if the OSC3 clock is being used as the system clock.
- The OSC1 oscillator circuit cannot be stopped if the OSC1 clock is being used as the system clock.
- Since the FOUTH/FOUT1 signal is asynchronized with FOUTHE/FOUT1E writing, switching output on or off will generate certain hazards.
- Continuous access of write and read to CLKSRC register (D0/0x5060) is prohibited. Enter at least one instruction that is not related to access to CLKSRC register between write and read.
- Change of clock division ratio selection (T8O1CK [2:0](D[3:1]/0x5065)) should be executed when T8O1CE(D0/0x5065) is 0 and the clock to 8-bit OSC1 timer is in "Stop" state.
- Change of the single selection (FOUTHD [1:0] (D[3:2]/0x5064) of FOUTH clock frequency should be executed when FOUTHE (D1/0x5064) is 0 and clock output is in "Stop" status.
- The stability of oscillation depends on the oscillator and external add-on components. Full evaluation is required for configuring shorter stabilization wait time.
  OSC3 clock system supply wait time =< OSC3 oscillation start time (max.) + OSC3 oscillation stabilization wait time.
- Set OSC3EN (the D0/OSC_CTL register) to 0 while the OSC3 and OSC4 pins are kept open.
- Set OSC1EN (the D1/OSC_CTL register) to 0 while the OSC1 and OSC2 pins are kept open.

8 Clock Generator (CLG)

8.1 Clock Generator Configuration

The clock generator controls the system clock feed to the S1C17 core and peripheral modules.
Figure 8.1.1 illustrates the clock system and CLG module configuration.

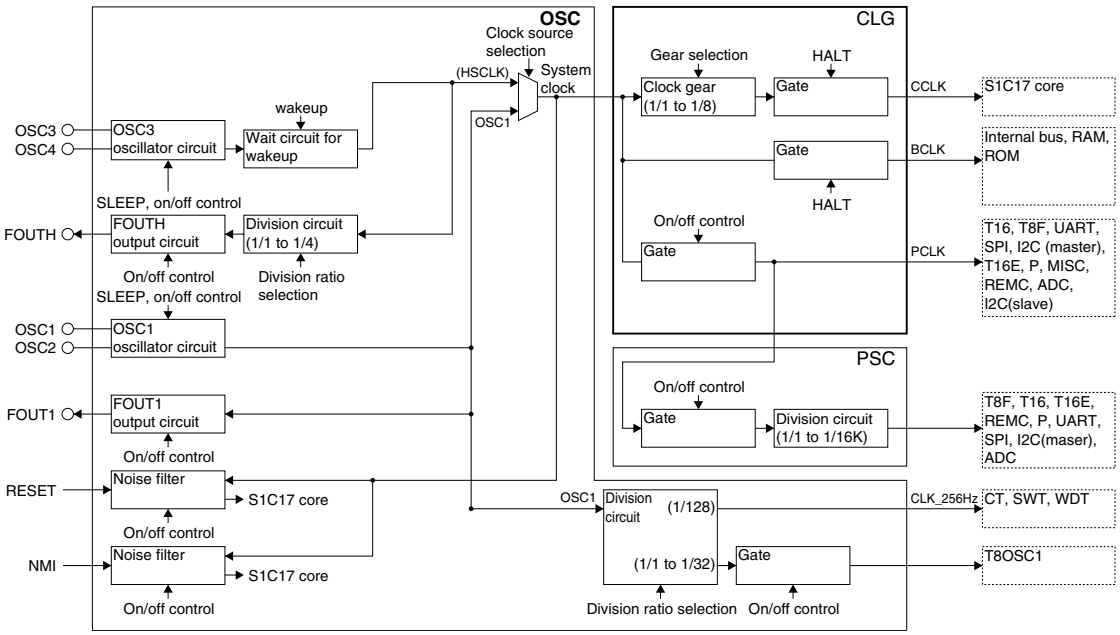


Figure 8.1.1: CLG module configuration

To reduce power consumption, control the clock in conjunction with processing and use standby mode. For more information on reducing power consumption, refer to “Appendix B: Power Saving.”

8.2 CPU Core Clock (CCLK) Control

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

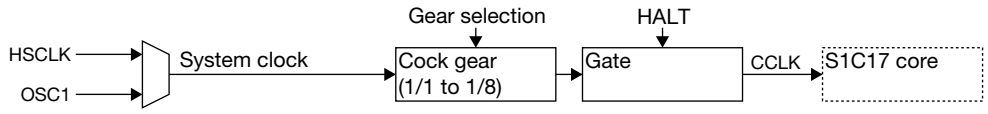


Figure 8.2.1: CCLK feed system

Clock gear settings

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

* **CCLKGR[1:0]**: CCLK Clock Gear Ratio Select Bits in the CCLK Control (CLG_CCLK) Register (D[1:0]/0x5081)

Table 8.2.1: CCLK gear ratio selection

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

(Default: 0x0)

Clock feed control

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

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

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

For more information on system clock control, refer to “7 Oscillator Circuit (OSC).”

8.3 Peripheral Module Clock (PCLK) Control

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

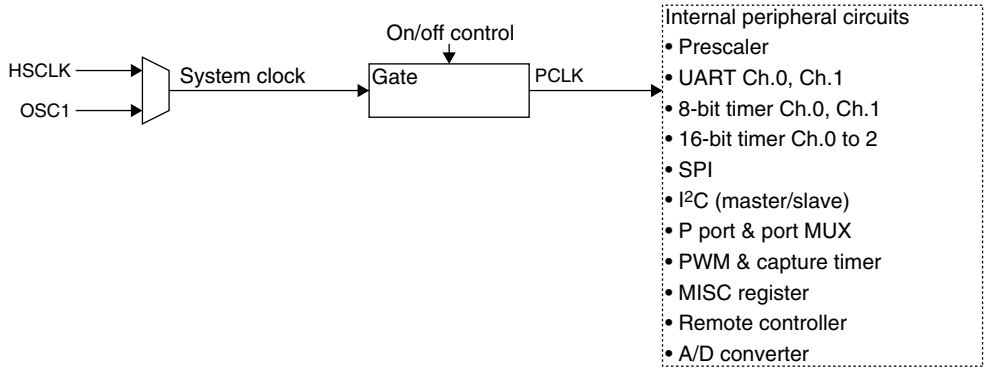


Figure 8.3.1: Peripheral module clock control circuit

Clock feed control

PCLK feed is controlled by PCKEN[1:0] (D[1:0]/CLG_PCLK register).

∗ **PCKEN[1:0]**: PCLK Enable Bits in the PCLK Control (CLG_PCLK) Register (D[1:0]/0x5080)

Table 8.3.1: PCLK control

PCKEN[1:0]	PCLK feed
0x3	Permitted (on)
0x2	Setting prohibited
0x1	Setting prohibited
0x0	Prohibited (off)

(Default: 0x3)

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

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

Peripheral modules not operating on PCLK

The OSC1 peripheral module operates using a clock other than PCLK. Therefore, PCLK is not required.

OSC1 peripheral module

The clock timer, stopwatch timer, watchdog timer, and 8-bit OSC1 timer operate using the OSC1 division clock.

8.4 Control Register Details

Table 8.4.1 CLG register list

Address	Register name		Function
0x5080	CLG_PCLK	PCLK Control Register	PCLK feed control
0x5081	CLG_CCLK	CCLK Control Register	CCLK division ratio setting

The CLG module registers are described in detail below. These are 8-bit registers.

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

0x5080: PCLK Control Register (CLG_PCLK)

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

D[7:2] Reserved

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

Permit or prohibit clock (PCLK) feed to internal peripheral modules.

Table 8.4.2: PCLK control

PCKEN[1:0]	PCLK feed
0x3	Permitted (on)
0x2	Setting prohibited
0x1	Setting prohibited
0x0	Prohibited (off)

(Default: 0x3)

The PCKEN[1:0] default setting is 0x3, which enables clock feed. Stop the clock feed to reduce power consumption if the peripheral modules listed below are not required.

Peripheral modules operated using PCLK
- Prescaler (PWM timer, remote controller, P port)
- UART Ch.0 to 1
- 8-bit timer Ch.0 to 1
- 16-bit timer Ch.0 to 2
- SPI
- I^2C (master/slave)
- P port & port MUX
- PWM & capture timer
- MISC register
- Remote controller
- A/D converter

The following peripheral modules operate, including access to control registers, using a clock other than PCLK. Therefore, PCLK does not need to be turned on.
- Clock timer
- Stopwatch timer
- Watchdog timer
- 8-bit OSC1 timer

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

0x5081: CCLK Control Register (CLG_CCLK)

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

D[7:2] Reserved

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

Select the gear ratio for reducing system clock speed and set the CCLK clock speed for operating the S1C17 core. To reduce power consumption, operate the S1C17 core using the slowest possible clock speed.

Table 8.4.3: CCLK gear ratio selection

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

(Default: 0x0)

8.5 Precautions

(1) The default settings enable PCLK feed to peripheral modules. To reduce power consumption, stop the clock feed if the peripheral modules listed below are not used.

Peripheral modules operated using PCLK
- Prescaler (PWM timer, remote controller, P port)
- UART Ch.0 to 1
- 8-bit timer Ch.0 to 1
- 16-bit timer Ch.0 to 2
- SPI
- I^2C (master/slave)
- P port & port MUX
- PWM & capture timer
- MISC register
- Remote controller
- A/D converter

The following peripheral modules operate, including access to control registers, using a clock other than PCLK. Therefore, PCLK does not need to be turned on.
- Clock timer
- Stopwatch timer
- Watchdog timer
- 8-bit OSC1 timer

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

* **PCKEN[1:0]**: PCLK Enable Bits in the PCLK Control (CLG_PCLK) Register (D[1:0]/0x5080)

9 Prescaler (PSC)

9.1 Prescaler Configuration

The S1C17003 incorporates a prescaler to generate a clock for timer operations. The prescaler generates 15 different frequencies by dividing the PCLK clock fed from the clock generator into 1/1 to 1/16K. The peripheral modules to which the clock is fed include clock selection registers enabling selection of one as a count or operation clock.

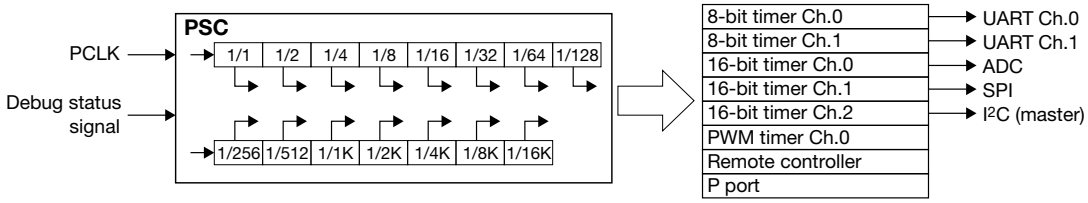


Figure 9.1.1: Prescaler

The prescaler is controlled by the PRUN bit (D0/PSC_CTL register). To operate the prescaler, write 1 to PRUN. Writing 0 to PRUN stops the prescaler. Stopping the prescaler while the timer and interface module are halted enables the current consumption to be reduced. The prescaler is stopped immediately after initial resetting.

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

Note: PCLK must be fed from the clock generator to use the prescaler.

The prescaler features another control bit, PRUND (D1/PSC_CTL register), which specifies prescaler operations in Debug mode. Setting PRUND to 1 also operates the prescaler in Debug mode. Setting it to 0 stops the prescaler once the S1C17 core switches to Debug mode. Set PRUND to 1 if the timer and interface module are to be used during debugging.

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

9.2 Control Register Details

Table 9.2.1: Prescaler register

Address	Register name		Function
0x4020	PSC_CTL	Prescaler Control Register	Prescaler start/stop control

The prescaler register is an 8-bit register.

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

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 Setting Bit for Debug Mode

Selects prescaler operations in Debug mode.
1 (R/W): Operate
0 (R/W): Stop (default)

Setting PRUND to 1 operates the prescaler even in Debug mode. Setting it to 0 stops the prescaler once the S1C17 core switches to Debug mode. Set PRUND to 1 to use the timer and interface module during debugging.

D0 PRUN: Prescaler Run/Stop Control Bit

Starts or stops prescaler operation.
1 (R/W): Start operation
0 (R/W): Stop (default)

Write 1 to PRUN to operate the prescaler. Write 0 to PRUN to stop the prescaler. To reduce current consumption, stop the prescaler if the timer and interface module are already stopped.

9.3 Precautions

PCLK must be fed from the clock generator to use the prescaler.

10 Input/Output Port (P)

10.1 Input/Output Port Configuration

The S1C17003 includes 30 input/output ports (P0[7:0], P1[6:0], P2[7,4,3], P3[7:0], P4[3:0]) and four input only ports (P17, P2[2:0]) to allow software switching of input/output direction. These share internal peripheral module input/output pins (with certain exceptions), but pins not used for peripheral modules can be used as general purpose input/output ports.
Figure 10.1.1 illustrates the input/output port configuration.

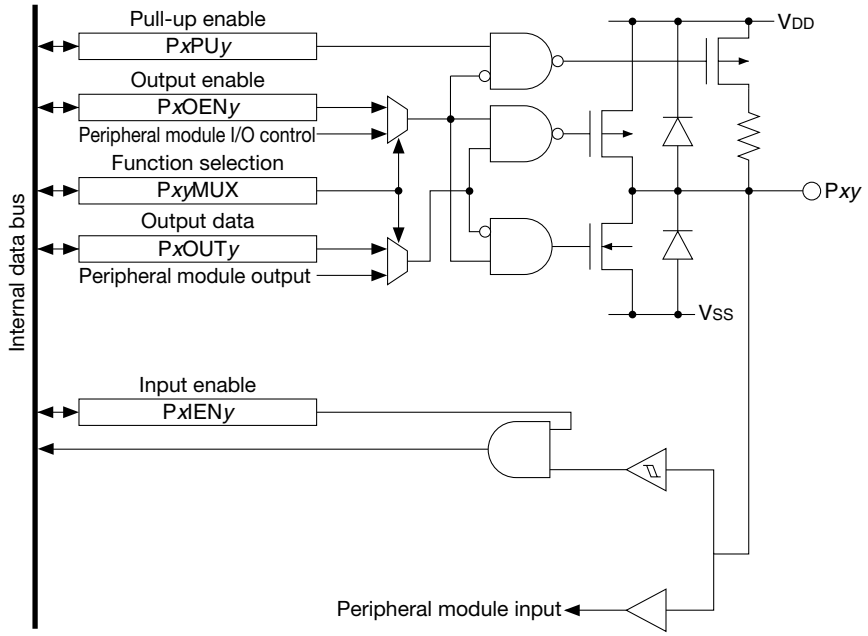


Figure 10.1.1: Input/output port configuration

The P0 and P1 ports can generate input interrupts.
The P0[3:0] port can be used for key entry resets. (For more information, refer to "5.1.2 P0 Port Key Entry Reset.")

Note: The PCLK clock must be fed from the clock generator to access the input/output port. The prescaler output clock is also needed to operate the P0/P1 port chattering filter. Switch on the prescaler when using this function.

10.2 Input/Output Pin Function Selection (Port MUX)

The input/output port pins share peripheral module input/output pins (with certain exceptions). Each pin can be set for use as an input/output port or for peripheral modules via the corresponding port function selection bits for each port. Pins not used for peripheral modules can be used as general purpose input/output ports.

Table 10.2.1: Input/output pin function selection

Pin function 1 P*xx*MUX = 0	Pin function 2 P*xx*MUX = 01	Pin function 3 P*xx*MUX = 10	Pin function 4 P*xx*MUX = 11	Port function selection bit	Control register
P00	REMO (REMC)	—	—	P00MUX (D1-0)	P0 Port Function Select (P0_PMUX) Register (0x52a0)
P01	REMI (REMC)	—	—	P01MUX (D3-2)	
P02/EXCL0 (T16)	—	—	—	P02MUX (D5-4)	
P03	#ADTRG (ADC10SA)	—	—	P03MUX (D7-6)	
P04	SPICLK (SPI)	—	—	P04MUX (D1-0)	P0 Port Function Select (P0_PMUX) Register (0x52a1)
P05	SDO (SPI)	—	—	P05MUX (D3-2)	
P06	SDI (SPI)	—	—	P06MUX (D5-4)	
P07	#SPISS (SPI)	—	—	P07MUX (D7-6)	
P10	SCLK0 (UART)	—	—	P10MUX (D1-0)	P1 Port Function Select (P1_PMUX) Register (0x52a2)
P11	SOUT0 (UART)	—	—	P11MUX (D3-2)	
P12	SIN0 (UART)	—	—	P12MUX (D5-4)	
P13/EXCL1 (T16)	—	—	—	P13MUX (D7-6)	
P14/EXCL2 (T16)	—	—	—	P14MUX (D1-0)	P1 Port Function Select (P1_PMUX) Register (0x52a3)
P15/EXCL3 (T16E)	—	—	—	P15MUX (D3-2)	
P16	SCLK1 (UART)	—	—	P16MUX (D5-4)	
P17	AIN3 (ADC10SA)	—	—	P17MUX (D7-6)	
P20	AIN2 (ADC10SA)	—	—	P20MUX (D1-0)	P2 Port Function Select (P2_PMUX) Register (0x52a4)
P21	AIN1 (ADC10SA)	—	—	P21MUX (D3-2)	
P22	AIN0 (ADC10SA)	—	—	P22MUX (D5-4)	
P23	—	—	—	P23MUX (D7-6)	
P24	—	—	—	P24MUX (D1-0)	P2 Port Function Select (P2_PMUX) Register (0x52a5)
P27	SOUT1(UART)	—	—	P27MUX (D7-6)	
P30	SIN1(UART)	—	—	P30MUX (D1-0)	P3 Port Function Select (P3_PMUX) Register (0x52a6)
P31	SCL0 (I2CM)	—	—	P31MUX (D3-2)	
P32	SDA0 (I2CM)	—	—	P32MUX (D5-4)	
P33	SCL1 (I2CS)	SCL0 (I2CM)	—	P33MUX (D7-6)	
P34	SDA1 (I2CS)	SDA0 (I2CM)	—	P34MUX (D1-0)	P3 Port Function Select (P3_PMUX) Register (0x52a7)
P35	FOUT1 (CLG)	#BFR (I2CS)	—	P35MUX (D3-2)	
P36	TOUT3 (T16E)	—	—	P36MUX (D5-4)	
P37	TOUTN3 (T16E)	—	TOUT4(T8OSC1)	P37MUX (D7-6)	
P40	FOUTH (CLG)	—	—	P40MUX (D1-0)	P4 Port Function Select (P4_PMUX) Register (0x52a8)
DSIO (DBG)	P41	—	—	P41MUX (D3-2)	
DST2 (DBG)	P42	—	—	P42MUX (D5-4)	
DCLK (DBG)	P43	—	—	P43MUX (D7-6)	

Resetting the input/output port pins (P*xx*) resets them to their default functions (pin function 1 in Table 10.2.1).

P02, P13, P14, P15, P16 Pins can be used as external clock input pin of 16 bit timer by setting them to input mode. P17, P20, P21, P22 Pins are input only.

For information on functions other than the input/output ports, refer to the discussion of the peripheral modules indicated in parentheses. The sections below discuss port functions with the pins set as general purpose input/output ports.

10.3 Data Input/Output

The input/output ports permit selection of the data input/output direction for each bit using P*x*OEN[7:0] (P*x*_OEN register) and P*x*IEN[7:0] (P*x*_IEN register). P*x*OEN[7:0] executes on/off control of data output, while P*x*IEN[7:0] executes on/off control of data input.

* **P0OEN[7:0]**: P0[7:0] Port Output Enable Bits in the P0 Port Output Enable (P0_OEM) Register (D[7:0]/0x5202)
* **P1OEN[7:0]**: P1[7:0] Port Output Enable Bits in the P1 Port Output Enable (P1_OEM) Register (D[7:0]/0x5212)
* **P2OEN[7:0]**: P2[7:0] Port Output Enable Bits in the P2 Port Output Enable (P2_OEM) Register (D[7:0]/0x5222)
* **P3OEN[7:0]**: P3[7:0] Port Output Enable Bits in the P3 Port Output Enable (P3_OEM) Register (D[7:0]/0x5232)
* **P4OEN[3:0]**: P4[3:0] Port Output Enable Bits in the P4 Port Output Enable (P4_OEM) Register (D[3:0]/0x5242)

* **P0IEN[7:0]**: P0[7:0] Port Input Enable Bits in the P0 Port Input Enable (P0_IEN) Register (D[7:0]/0x520a)
* **P1IEN[7:0]**: P1[7:0] Port Input Enable Bits in the P1 Port Input Enable (P1_IEN) Register (D[7:0]/0x521a)
* **P2IEN[7:0]**: P2[7:0] Port Input Enable Bits in the P2 Port Input Enable (P2_IEN) Register (D[7:0]/0x522a)
* **P3IEN[7:0]**: P3[7:0] Port Input Enable Bits in the P3 Port Input Enable (P3_IEN) Register (D[7:0]/0x523a)
* **P4IEN[3:0]**: P4[3:0] Port Input Enable Bits in the P4 Port Input Enable (P4_IEN) Register (D[3:0]/0x524a)

Table 10.3.1: Data Input/Output list

P*x*OEN[7:0] Output control	P*x*IEN[7:0] Input control	P*x*PU[7:0] Pull-up control	Port status
0	1	0	Functions as input port (with pull-up off). Port pin (external input signal) value can be read from P*x*IN[7:0] (input data). Output is disabled.
0	1	1	Functions as input port (with pull-up on). (Default) port pin (external input signal) value can be read from P*x*IN[7:0] (input data). Output is disabled.
1	0	1 or 0	Functions as output port (with pull-up off). Input is disabled, and the value read from P*x*IN[7:0] (input data) is 0.
1	1	1 or 0	Functions as output port (with pull-up off). Input is also enabled, and the port pin value (output value) can be read from P*x*IN[7:0] (input data).
0	0	0	The pin is in high impedance state (with pull-up off), causing the I/O buffer to be in floating gate state. Therefore, this setting is prohibited.
0	0	1	The pin is in high impedance state (with pull-up on). Output is disabled, and the value read from P*x*IN[7:0] (input data) is 0.

The input/output direction for the port selecting the peripheral module function is controlled by the peripheral module. The P*x*IO[7:0] setting is ignored.

Data input

When set to input mode, P*x*IO[7:0] is set to 0 (default). The input/output port set to input mode switches to high-impedance state, and functions as the input port. If pull-up is enabled by the P*x*_PU register, the port will be pulled up.

In input mode, the input pin state can be read out directly from P*x*IN[7:0] (P*x*_IN register). The value read will be 1 when the input pin is at High (V_{DD}) level and 0 when it is at Low (V_{SS}) level.

* **P0IN[7:0]**: P0[7:0] Port Input Data Bits in the P0 Port Input Data (P0_IN) Register (D[7:0]/0x5200)
* **P1IN[7:0]**: P1[7:0] Port Input Data Bits in the P1 Port Input Data (P1_IN) Register (D[7:0]/0x5210)
* **P2IN[7:0]**: P2[7:0] Port Input Data Bits in the P2 Port Input Data (P2_IN) Register (D[7:0]/0x5220)
* **P3IN[7:0]**: P3[7:0] Port Input Data Bits in the P3 Port Input Data (P3_IN) Register (D[7:0]/0x5230)
* **P4IN[3:0]**: P4[7:0] Port Input Data Bits in the P4 Port Input Data (P4_IN) Register (D[3:0]/0x5240)

Data output

When set to output mode, P*x*OEN[7:0] is set to 1. The input/output port set to output mode functions as the output port, while the port pin outputs High (V_{DD}) level if P*x*OUT[7:0] (P*x*_OUT register) is written as 1 and outputs Low (V_{SS}) level if written as 0. Note that the port will not be pulled up in output mode even if pull-up is enabled by the P*x*_PU register.

* **P0OUT[7:0]**: P0[7:0] Port Output Data Bits in the P0 Port Output Data (P0_OUT) Register (D[7:0]/0x5201)
* **P1OUT[7:0]**: P1[7:0] Port Output Data Bits in the P1 Port Output Data (P1_OUT) Register (D[7:0]/0x5211)
* **P2OUT[7:0]**: P2[7:0] Port Output Data Bits in the P2 Port Output Data (P2_OUT) Register (D[7:0]/0x5221)
* **P3OUT[7:0]**: P3[7:0] Port Output Data Bits in the P3 Port Output Data (P3_OUT) Register (D[7:0]/0x5231)
* **P4OUT[3:0]**: P4[3:0] Port Output Data Bits in the P4 Port Output Data (P4_OUT) Register (D[3:0]/0x5241)

Writing to P*x*OUT[7:0] is possible without affecting pin status, even in input mode.

10.4 Pull-up Control

The input/output port contains a pull-up resistor, which you can choose to use or not use individually for each bit using the P*x*PU[7:0] (P*x*_PU register).

* **P0PU[7:0]**: P0[7:0] Port Pull-up Enable Bits in the P0 Port Pull-up Control (P0_PU) Register (D[7:0]/0x5203)
* **P1PU[7:0]**: P1[7:0] Port Pull-up Enable Bits in the P1 Port Pull-up Control (P1_PU) Register (D[7:0]/0x5213)
* **P2PU[7:0]**: P2[7:0] Port Pull-up Enable Bits in the P2 Port Pull-up Control (P2_PU) Register (D[7:0]/0x5223)
* **P3PU[7:0]**: P3[7:0] Port Pull-up Enable Bits in the P3 Port Pull-up Control (P3_PU) Register (D[7:0]/0x5233)
* **P4PU[3:0]**: P4[3:0] Port Pull-up Enable Bits in the P4 Port Pull-up Control (P4_PU) Register (D[3:0]/0x5243)

Setting P*x*PU[7:0] to 1 (default) enables the pull-up resistor and pulls up the port pin in input mode. It will not be pulled up if set to 0.
The P*x*PU[7:0] setting is disabled in output mode, and the pin is not pulled up.

Input/output ports that are not used should be set with pull-up enabled.
This pull-up setting is also enabled for ports for which the peripheral module function has been selected.

A delay will occur in the waveform rise-up depending on time constants such as pull-up resistance and pin load capacitance if the port pin is switched from Low level to High level by the internal pull-up resistor. Therefore, an appropriate wait time must be set for loading an input/output port.

10.5 P0 and P1 Port Chattering Filter Function

The P0 and P1 port include a chattering filter circuit for key entry, which you can select to use or not use (and for which you can select a verification time if used) individually for the four P0[3:0] and P0[7:4], P1 [3:0], P1 [7:4] ports using P*x*CF1[2:0] (D[2:0]/P*x*_CHAT register), P*x*CF2[2:0] (D[6:4]/P*x*_CHAT register).

- **P0CF1[2:0]**: P0[3:0] Chattering Filter Time Select Bits in the P0 Port Chattering Filter Control (P0_CHAT) Register (D[2:0]/0x5208)
- **P0CF2[2:0]**: P0[7:4] Chattering Filter Time Select Bits in the P0 Port Chattering Filter Control (P0_CHAT) Register (D[6:4]/0x5208)
- **P1CF1[2:0]**: P1[3:0] Chattering Filter Time Select Bits in the P1 Port Chattering Filter Control (P1_CHAT) Register (D[2:0]/0x5218)
- **P1CF2[2:0]**: P1[7:4] Chattering Filter Time Select Bits in the P1 Port Chattering Filter Control (P1_CHAT) Register (D[6:4]/0x5218)

Table 10.5.1: Chattering filter function settings

P*x*CF1[2:0]/P*x*CF2[2:0]	Verification time *
0x7	16384/f_{PCLK} (8ms)
0x6	8192/f_{PCLK} (4ms)
0x5	4096/f_{PCLK} (2ms)
0x4	2048/f_{PCLK} (1ms)
0x3	1024/f_{PCLK} (512µs)
0x2	512/f_{PCLK} (256µs)
0x1	256/f_{PCLK} (128µs)
0x0	No verification time (Off)

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

Note:
- **The chattering filter verification time refers to the maximum pulse width that can be filtered. Generating an input interrupt requires a minimum input time of the verification time and a maximum input time of twice the verification time.**
- **Input interrupts will not be accepted for a transition into SLEEP mode with the chattering filter left on. The chattering filter should be set off (no verification time) before executing the slp instruction.**
- **P0/P1 port interrupts must be blocked when P*x*_CHAT register (0x5208/0x5218) settings are being changed. Changing the setting while interrupts are permitted may generate inadvertent P0/P1 interrupts.**
- **A phenomenon may occur in which the internal signal oscillates due to the time elapsed until the signal reaches the threshold value if the input signal rise-up/drop-off time is delayed. Since input interrupts will malfunction under these conditions, the input signal rise-up/drop-off time should normally be set to 25 ns or less.**

10.6 Port Input Interrupt

Ports P0 and P1 include input interrupt functions.
Select which of the 16 ports are to be used for interrupts based on requirements. You can also select whether interrupts are generated for either the rising edge or falling edge of input signals.
Figure 10.6.1 illustrates the port input interrupt circuit configuration.

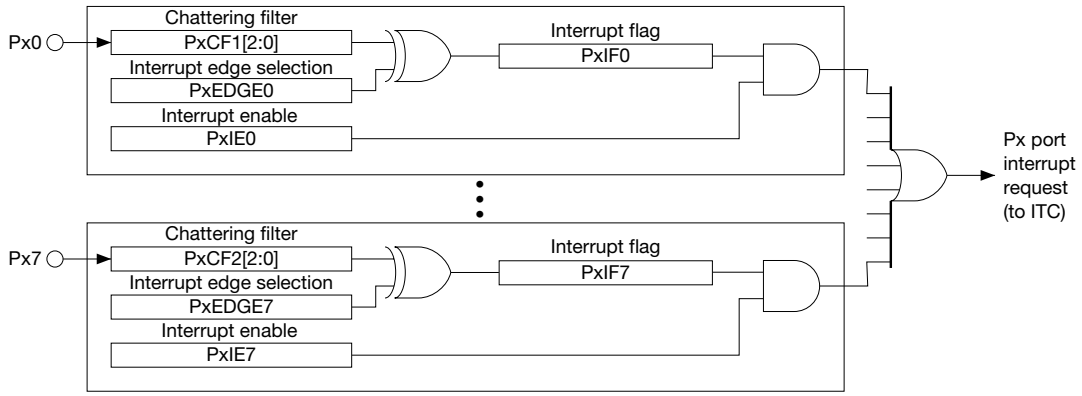


Figure 10.6.1: Port input interrupt circuit configuration

Interrupt port selection

Select the port generating an interrupt using P*x*IE[7:0] (P*x*_IMSK register).

- **P0IE[7:0]**: P0[7:0] Port Interrupt Enable Bits in the P0 Port Interrupt Mask (P0_IMSK) Register (D[7:0]/0x5205)
- **P1IE[7:0]**: P1[7:0] Port Interrupt Enable Bits in the P1 Port Interrupt Mask (P1_IMSK) Register (D[7:0]/0x5215)

Setting P*x*IE[7:0] to 1 enables interrupt generation by the corresponding port. Setting to 0 (default) disables interrupt generation.

Interrupt edge selection

Port input interrupts can be generated at either the rising edge or falling edge of the input signal. Select the edge used to generate interrupts using P*x*EDGE[7:0] (P*x*_EDGE register).

- **P0EDGE[7:0]**: P0[7:0] Port Interrupt Edge Select Bits in the P0 Port Interrupt Edge Select (P0_EDGE) Register (D[7:0]/0x5206)
- **P1EDGE[7:0]**: P1[7:0] Port Interrupt Edge Select Bits in the P1 Port Interrupt Edge Select (P1_EDGE) Register (D[7:0]/0x5216)

Setting P*x*EDGE[7:0] to 1 generates port input interrupts at the input signal falling edge. Setting it to 0 (default) generates interrupts at the rising edge.

Interrupt flags

The ITC is able to accept interrupt requests for both P0 and P1 port interrupts, and the P port module contains interrupt flags P*x*IF[7:0] corresponding to the individual 16 ports to enable individual control of the 16 P0[7:0] and P1[7:0] port interrupts. P*x*IF[7:0] will be set to 1 at the specified edge (rising or falling edge) of the input signal. A P0 or P1 port interrupt request signal is also output to the ITC at the same time if the corresponding P*x*IE[7:0] is set to 1. Meeting the ITC and S1C17 core interrupt conditions generates an interrupt.

* **P0IF[7:0]**: P0[7:0] Port Interrupt Flags in the P0 Port Interrupt Flag (P0_IFLG) Register (D[7:0]/0x5207)
* **P1IF[7:0]**: P1[7:0] Port Interrupt Flags in the P1 Port Interrupt Flag (P1_IFLG) Register (D[7:0]/0x5217)

P*x*IF[7:0] is reset by writing as 1.

Note: • The P port module interrupt flag P*x*IF[7:0] must be reset within the interrupt processing routine following a port interrupt to prevent recurring interrupts.
• To prevent generating unnecessary interrupts, reset the relevant P*x*IF[7:0] before permitting interrupts for the required port using P*x*IE[7:0] (P*x*_IMSK register).

Interrupt vector

The port interrupt vector numbers and vector addresses are as shown below.

Table 10.6.1: Port interrupt vectors

Port	Vector number	Vector address
P0	4 (0x04)	TTBR + 0x10
P1	5 (0x05)	TTBR + 0x14

Other interrupt settings

The ITC allows the precedence of P0 and P1 port interrupts to be set between level 0 (default) and level 7. The PSR (S1C17 core internal processor status register) IE (interrupt enable) bit must be set to 1 to generate actual interrupts.

For specific information on interrupt processing, see “6 Interrupt Controller (ITC).”

10.7 Control Register Details

Table 10.7.1: Input/output port control register list

Address	Register name		Function
0x5200	P0_IN	P0 Port Input Data Register	P0 port input data
0x5201	P0_OUT	P0 Port Output Data Register	P0 port output data
0x5202	P0_OEN	P0 Port Output Enable Register	P0 port output enable
0x5203	P0_PU	P0 Port Pull-up Control Register	P0 port pull-up control
0x5205	P0_IMSK	P0 Port Interrupt Mask Register	P0 port interrupt mask setting
0x5206	P0_EDGE	P0 Port Interrupt Edge Select Register	P0 port interrupt edge selection
0x5207	P0_IFLG	P0 Port Interrupt Flag Register	P0 port interrupt occurrence status display/reset
0x5208	P0_CHAT	P0 Port Chattering Filter Control Register	P0 port chattering filter control
0x5209	P0_KRST	P0 Port Key-Entry Reset Configuration Register	P0 port key entry reset setting
0x520a	P0_IEN	P0 Port Input Enable Register	P0 port input enable
0x5210	P1_IN	P1 Port Input Data Register	P1 port input data
0x5211	P1_OUT	P1 Port Output Data Register	P1 port output data
0x5212	P1_OEN	P1 Port Output Enable Register	P1 port output enable
0x5213	P1_PU	P1 Port Pull-up Control Register	P1 port pull-up control
0x5215	P1_IMSK	P1 Port Interrupt Mask Register	P1 port interrupt mask setting
0x5216	P1_EDGE	P1 Port Interrupt Edge Select Register	P1 port interrupt edge selection
0x5217	P1_IFLG	P1 Port Interrupt Flag Register	P1 port interrupt occurrence status display/reset
0x5218	P1_CHAT	P1 Port Chattering Filter Control Register	P1 port chattering filter control
0x521a	P1_IEN	P1 Port Input Enable Register	P1 port input enable
0x5220	P2_IN	P2 Port Input Data Register	P2 port input data
0x5221	P2_OUT	P2 Port Output Data Register	P2 port output data
0x5222	P2_OEN	P2 Port Output Enable Register	P2 port output enable
0x5223	P2_PU	P2 Port Pull-up Control Register	P2 port pull-up control
0x522a	P2_IEN	P2 Port Input Enable Register	P2 port input enable
0x5230	P3_IN	P3 Port Input Data Register	P3 port input data
0x5231	P3_OUT	P3 Port Output Data Register	P3 port output data
0x5232	P3_OEN	P3 Port Output Enable Register	P3 port output enable
0x5233	P3_PU	P3 Port Pull-up Control Register	P3 port pull-up control
0x523a	P3_IEN	P3 Port Input Enable Register	P3 port input enable
0x5240	P4_IN	P4 Port Input Data Register	P4 port input data
0x5241	P4_OUT	P4 Port Output Data Register	P4 port output data
0x5242	P4_OEN	P4 Port Output Enable Register	P4 port output enable
0x5243	P4_PU	P4 Port Pull-up Enable Register	P4 port pull-up control enable
0x524a	P4_IEN	P4 Port Input Enable Register	P4 port input enable
0x52a0	P0_PMUX	P0 Port Function Select Register	P0 port function selection
0x52a1	P0_PMUX	P0 Port Function Select Register	P0 port function selection
0x52a2	P1_PMUX	P1 Port Function Select Register	P1 port function selection
0x52a3	P1_PMUX	P1 Port Function Select Register	P1 port function selection
0x52a4	P2_PMUX	P2 Port Function Select Register	P2 port function selection
0x52a5	P2_PMUX	P2 Port Function Select Register	P2 port function selection
0x52a6	P3_PMUX	P3 Port Function Select Register	P3 port function selection
0x52a7	P3_PMUX	P3 Port Function Select Register	P3 port function selection
0x52a8	P4_PMUX	P4 Port Function Select Register	P4 port function selection

The input/output port registers are described in detail below. These are 8-bit registers.

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

0x5200/0x5210/0x5220/0x5230/0X5240: P*x* Port Input Data Registers (P*x*_IN)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
P0 Port Input Data Register (P0_IN)	**0x5200** (8 bits)	D7–0	**P0IN[7:0]**	P0[7:0] port input data	1	1 (H)	0	0 (L)	×	R	
P1 Port Input Data Register (P1_IN)	**0x5210** (8 bits)	D7–0	**P1IN[7:0]**	P1[7:0] port input data	1	1 (H)	0	0 (L)	×	R	
P2 Port Input Data Register (P2_IN)	**0x5220** (8 bits)	D7–0	**P2IN[7:0]**	P2[7:0] port input data	1	1 (H)	0	0 (L)	×	R	P25, P26 : 0 when being read.
P3 Port Input Data Register (P3_IN)	**0x5230** (8 bits)	D7–0	**P3IN[7:0]**	P3[7:0] port input data	1	1 (H)	0	0 (L)	×	R	
P4 Port Input Data Register (P4_IN)	**0x5240** (8 bits)	D7–4	**–**	reserved	–				–	–	x when being read.
		D3–0	**P4IN[3:0]**	P4[3:0] port input data	1	1 (H)	0	0 (L)	×	R	

Note: The "*x*" in the bit names indicates the port number (0 to 4).

D[7:0] P*x*IN[7:0]: P*x*[7:0] Port Input Data Bits (P4 port is P4IN[3:0])

Read out the P port pin status. (Default: external pin status)

1(R): High level

0(R): Low level

P*x*IN[7:0] correspond directly to the P*x*[7:0] pins and read the pin voltage level regardless of input/output mode. 1 is read when the pin voltage is High; 0 is read when the voltage is Low.

Writing operations to the read-only P*x*IN[7:0] are disabled.

The read-out value of P2IN[6:5] is fixed to 0.

0x5201/0x5211/0x5221/0x5231/0X5241: P*x* Port Output Data Registers (P*x*_OUT)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
P0 Port Output Data Register (P0_OUT)	**0x5201** (8 bits)	D7–0	**P0OUT[7:0]**	P0[7:0] port output data	1	1 (H)	0	0 (L)	0	R/W	
P1 Port Output Data Register (P1_OUT)	**0x5211** (8 bits)	D7–0	**P1OUT[7:0]**	P1[7:0] port output data	1	1 (H)	0	0 (L)	0	R/W	
P2 Port Output Data Register (P2_OUT)	**0x5221** (8 bits)	D7–0	**P2OUT[7:0]**	P2[7:0] port output data	1	1 (H)	0	0 (L)	0	R/W	
P3 Port Output Data Register (P3_OUT)	**0x5231** (8 bits)	D7–0	**P3OUT[7:0]**	P3[7:0] port output data	1	1 (H)	0	0 (L)	0	R/W	
P4 Port Output Data Register (P4_OUT)	**0x5241** (8 bits)	D7–4	–	reserved	–				–	–	0 when being read.
		D3–0	**P4OUT[3:0]**	P4[3:0] port output data	1	1 (H)	0	0 (L)	0	R/W	

Note: The "*x*" in the bit names indicates the port number (0 to 4).

D[7:0] P*x*OUT[7:0]: P*x*[7:0] Port Output Data Bits (P4 port is P4OUT[3:0])

Set the data to be output from the port pin.

1(R/W): High level

0(R/W): Low level (default)

P*x*OUT[7:0] correspond directly to the P*x*[7:0] pins and output data from the port pin as written. Setting the data bit to 1 sets the port pin to High; setting it to 0 sets it to Low.

Port data can also be written in input mode.

The P25 and P26 pins are not present. Settings onto the register are therefore disabled.

The P17, and P20 to P22 pins are assigned exclusively for input. Settings onto the register are therefore disabled.

0x5202/0x5212/0x5222/0x5232/0x5242: P*x* Port Output Enable Registers (P*x*_OEN)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
P0 Port Output Enable Register (P0_OEN)	**0x5202** (8 bits)	D7–0	**P0OEN[7:0]**	P0[7:0] port output enable select	1	Output Enable	0	Output Disable	0	R/W	
P1 Port Output Enable Register (P1_OEN)	**0x5212** (8 bits)	D7–0	**P1OEN[7:0]**	P1[7:0] port output enable select	1	Output Enable	0	Output Disable	0	R/W	
P2 Port Output Enable Register (P2_OEN)	**0x5222** (8 bits)	D7–0	**P2OEN[7:0]**	P2[7:0] port output enable select	1	Output Enable	0	Output Disable	0	R/W	
P3 Port Output Enable Register (P3_OEN)	**0x5232** (8 bits)	D7–0	**P3OEN[7:0]**	P3[7:0] port output enable select	1	Output Enable	0	Output Disable	0	R/W	
P4 Port Output Enable Register (P4_OEN)	**0x5242** (8 bits)	D7–4	–	reserved	–				–	–	0 when being read.
		D3–0	**P4OEN[3:0]**	P4[3:0] port output enable select	1	Output Enable	0	Output Disable	0	R/W	

Note: The "*x*" in the bit names indicates the port number (0 to 3).

D[7:0] P*x*IO[7:0]: P*x*[7:0] Port Output Enable Select Bits (P3 port is P3IN[3:0])

Set Port Output to enable/disable
1(R/W): Enable
0(R/W): Disable (default)

P*x*IO[7:0] are the output enable bits corresponding directly to the P*x*[7:0] ports. Setting to 1 selects output mode, while setting to 0 selects high inpedence. The peripheral module function determines the input/output direction for when a pin is used for peripheral modules.
For the input/output control by each register, refer to "Table 10.3.1 Data input/output list."
The P25 and P26 pins are not present. Settings onto the register are therefore disabled.
The P17, and P20 to P22 pins are assigned exclusively for input. Settings onto the register are therefore disabled.

0x5203/0x5213/0x5223/0x5233/0x5234: P*x* Port Pull-up Control Registers (P*x*_PU)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
P0 Port Pull-up Control Register (P0_PU)	**0x5203** (8 bits)	D7–0	**P0PU[7:0]**	P0[7:0] port pull-up enable	1	Enable	0	Disable	1 (0xff)	R/W	
P1 Port Pull-up Control Register (P1_PU)	**0x5213** (8 bits)	D7–0	**P1PU[7:0]**	P1[7:0] port pull-up enable	1	Enable	0	Disable	1 (0xff)	R/W	
P2 Port Pull-up Control Register (P2_PU)	**0x5223** (8 bits)	D7–0	**P2PU[7:0]**	P2[7:0] port pull-up enable	1	Enable	0	Disable	1 (0xff)	R/W	
P3 Port Pull-up Control Register (P3_PU)	**0x5233** (8 bits)	D7–0	**P3PU[7:0]**	P3[7:0] port pull-up enable	1	Enable	0	Disable	1 (0xff)	R/W	
P4 Port Pull-up Control Register (P4_PU)	**0x5243** (8 bits)	D7–4	–	reserved	–				–	–	1 when being read.
		D3–0	**P4PU[3:0]**	P4[3:0] port pull-up enable	1	Enable	0	Disable	1 (0xff)	R/W	

Note: The "*x*" in the bit names indicates the port number (0 to 4).

D[7:0] P*x*PU[7:0]: P*x*[7:0] Port Pull-up Enable Bits (P4 port is P4PU[3:0])

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

P*x*PU[7:0] are the pull-up control bits that correspond directly to the P*x*[7:0] ports. Setting to 1 enables the pull-up resistor and pulls up the port pin in input mode. It will not be pulled up if set to 0.
The P*x*PU[7:0] setting is disabled in output mode, and the pin is not pulled up.
Input/output ports that are not used should be set with pull-up enabled.
This pull-up setting is also enabled for ports for which the peripheral module function has been selected.

A delay will occur in the waveform rise-up depending on time constants such as pull-up resistance and pin load capacitance if the port pin is switched from Low level to High level by the internal pull-up resistor. Therefore, an appropriate wait time must be set for loading an input/output port.

The P25 and P26 pins are not present. Settings onto the register are therefore disabled.

0x5205/5215: P*x* Port Interrupt Mask Registers (P*x*_IMSK)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
P0 Port Interrupt Mask Register (P0_IMSK)	**0x5205** (8 bits)	D7–0	**P0IE[7:0]**	P0[7:0] port interrupt enable	1	Enable	0	Disable	0	R/W	
P1 Port Interrupt Mask Register (P1_IMSK)	**0x5215** (8 bits)	D7–0	**P1IE[7:0]**	P1[7:0] port interrupt enable	1	Enable	0	Disable	0	R/W	

Note: The "*x*" in the bit names indicates the port number (0 or 1).

D[7:0] P*x*IE[7:0]: P*x*[7:0] Port Interrupt Enable Bits

Permit or prohibit P0[7:0] and P1[7:0] port interrupt.
1 (R/W): Interrupt permitted
0 (R/W): Interrupt prohibited (default)

Setting P*x*IE[7:0] to 1 permits the corresponding interrupt, while setting to 0 blocks interrupts. Status changes for the input pin with interrupt blocked do not affect interrupt occurrence.

0x5206/5216: P*x* Port Interrupt Edge Select Registers (P*x*_EDGE)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
P0 Port Interrupt Edge Select Register (P0_EDGE)	**0x5206** (8 bits)	D7–0	**P0EDGE[7:0]**	P0[7:0] port interrupt edge select	1	Falling edge	0	Rising edge	0	R/W	
P1 Port Interrupt Edge Select Register (P1_EDGE)	**0x5216** (8 bits)	D7–0	**P1EDGE[7:0]**	P1[7:0] port interrupt edge select	1	Falling edge	0	Rising edge	0	R/W	

Note: The "*x*" in the bit names indicates the port number (0 or 1).

D[7:0] P*x*EDGE[7:0]: P*x*[7:0] Port Interrupt Edge Select Bits

Select the input signal edge for generating P0[7:0] and P1[7:0] port interrupts.
1 (R/W): Falling edge
0 (R/W): Rising edge (default)

Port interrupts are generated at the input signal falling edge if P*x*EDGE[7:0] are set to 1 and at the rising edge if set to 0.

0x5207/5217: P*x* Port Interrupt Flag Registers (P*x*_IFLG)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
P0 Port Interrupt Flag Register (P0_IFLG)	**0x5207** (8 bits)	D7–0	**P0IF[7:0]**	P0[7:0] port interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.
P1 Port Interrupt Flag Register (P1_IFLG)	**0x5217** (8 bits)	D7–0	**P1IF[7:0]**	P1[7:0] port interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.

Note: The "*x*" in the bit names indicates the port number (0 or 1).

D[7:0] P*x*IF[7:0]: P*x*[7:0] Port Interrupt Flags

These are interrupt flags indicating the interrupt factor occurrence status.

1(R): Interrupt factor present
0(R): No interrupt factor (default)
1(W): Reset flag
0(W): Disabled

P*x*IF[7:0] are interrupt flags corresponding to the individual 16 ports of P0[7:0] and P1[7:0]. P*x*IF[7:0] will be set to 1 at the specified edge (rising or falling edge) of the input signal. A P0 or P1 port interrupt request signal is also output to the ITC at the same time if the corresponding P*x*IE[7:0] is set to 1. This interrupt request signal causes the P0/P1 port interrupt flag inside the ITC to be set to 1. Meeting the ITC and S1C17 core interrupt conditions generates an interrupt.
P*x*IF[7:0] is reset by writing as 1.

Note: • The P port module interrupt flag P*x*IF[7:0] must be reset within the interrupt processing routine following a port interrupt to prevent recurring interrupts.

• To prevent genarating unnecessary interrupts, reset the relevant P*x*IF[7:0] before permitting interrupts for the required port using P*x*IE[7:0] (P*x*_IMSK register).

- **P0IE[7:0]**: P0[7:0] Port Interrupt Enable Bits in the P0 Port Interrupt Mask (P0_IMSK) Register (D[7:0]/0x5205)
- **P1IE[7:0]**: P1[7:0] Port Interrupt Enable Bits in the P1 Port Interrupt Mask (P1_IMSK) Register (D[7:0]/0x5215)

0x5208/0x5218: P*x* Port Chattering Filter Control Register (P*x*_CHAT)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
P0 Port Chattering Filter Control Register (P0_CHAT)	0x5208 (8 bits)	D7	–	reserved	–		–	–	0 when being read.
		D6–4	P0CF2[2:0]	P0[7:4] chattering filter time select	P0CF2[2:0]	Filter time	0	R/W	
					0x7	16384/fPCLK	0x0	R/W	
					0x6	8192/fPCLK			
					0x5	4096/fPCLK			
					0x4	2048/fPCLK			
					0x3	1024/fPCLK			
					0x2	512/fPCLK			
					0x1	256/fPCLK			
					0x0	None			
		D3	–	reserved	–		–	–	0 when being read.
		D2–0	P0CF1[2:0]	P0[3:0] chattering filter time select	P0CF1[2:0]	Filter time	0x0	R/W	
					0x7	16384/fPCLK			
					0x6	8192/fPCLK			
					0x5	4096/fPCLK			
					0x4	2048/fPCLK			
					0x3	1024/fPCLK			
					0x2	512/fPCLK			
					0x1	256/fPCLK			
					0x0	None			
P1 Port Chattering Filter Control Register (P1_CHAT)	0x5218 (8 bits)	D7	–	reserved	–		–	–	0 when being read.
		D6–4	P1CF2[2:0]	P1[7:4] chattering filter time select	P0CF2[2:0]	Filter time	0	R/W	
					0x7	16384/fPCLK	0x0	R/W	
					0x6	8192/fPCLK			
					0x5	4096/fPCLK			
					0x4	2048/fPCLK			
					0x3	1024/fPCLK			
					0x2	512/fPCLK			
					0x1	256/fPCLK			
					0x0	None			
		D3	–	reserved	–		–	–	0 when being read.
		D2–0	P1CF1[2:0]	P1[3:0] chattering filter time select	P0CF1[2:0]	Filter time	0x0	R/W	
					0x7	16384/fPCLK			
					0x6	8192/fPCLK			
					0x5	4096/fPCLK			
					0x4	2048/fPCLK			
					0x3	1024/fPCLK			
					0x2	512/fPCLK			
					0x1	256/fPCLK			
					0x0	None			

Note: The "*x*" in the bit names indicates the port number (0 or 1).

D7 Reserved

D[6:4] P*x*CF2[2:0]: P*x*[7:4] Chattering Filter Time Select Bits

Set the chattering filter circuit included in the P0[7:4] or P1[7:4] ports.

D3 Reserved

D[2:0] P*x*CF1[2:0]: P*x*[3:0] Chattering Filter Time Select Bits

Set the chattering filter circuit included in the P0[3:0] or P1[3:0]ports.

The P0 or P1 port includes a chattering filter circuit for key entry or port interrupt. You can select whether to use this function respectively for P0[3:0], P0[7:4], P1[3:0] and P1[7:4] ports using P*x*CF1/2[2:0]. You can also select relevant verification time accordingly.

Table 10.7.2: Chattering filter function settings

P*x*CF1[2:0], P*x*CF2[2:0]	Verification time *
0x7	16384/fPCLK (8ms)
0x6	8192/fPCLK (4ms)
0x5	4096/fPCLK (2ms)
0x4	2048/fPCLK (1ms)
0x3	1024/fPCLK (512µs)
0x2	512/fPCLK (256µs)
0x1	256/fPCLK (128µs)
0x0	**No verification time (Off)**

(Default: 0x0, *when OSC3 = 2 MHz and PCLK = OSC3)

Note:
- The chattering filter verification time refers to the maximum pulse width that can be filtered. Generating an input interrupt requires a minimum input time of the verification time and a maximum input time of twice the verification time.
- Input interrupts will not be accepted for a transition into SLEEP mode with the chattering filter left on. The chattering filter should be set off (no verification time) before executing the slp instruction.
- P0/P1 port interrupts must be blocked when P*x*_CHAT register settings are being changed. Changing the setting while interrupts are permitted may generate inadvertent P0/P1 interrupts.
- A phenomenon may occur in which the internal signal oscillates due to the time elapsed until the signal reaches the threshold value if the input signal rise-up/drop-off time is delayed. Since input interrupts will malfunction under these conditions, the input signal rise-up/drop-off time should normally be set to 25 ns or less.

0x5209: P0 Port Key-Entry Reset Configuration Register (P0_KRST)

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

D[7:2] Reserved

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

Select the port combination used for P0 port key entry resetting.

Table 10.7.3: P0 port key entry input reset settings

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

(Default: 0x0)

The key entry reset function performs an initial reset by inputting Low level simultaneously from externally to the port selected here.

For example, if P0KRST[1:0] is set to 0x3, an initial reset is performed when the four ports P00 to P03 are simultaneously set to Low level.

Set P0KRST[1:0] to 0x0 when this reset function is not used.

Note:
- **Make sure the specified ports are not simultaneously switched to Low during normal operations when using the P0 port key-entry reset function.**
- **The P0 port key entry reset function is disabled on initial resetting and cannot be used for resetting at power-on.**
- **The P0 port key-entry reset function cannot be used in SLEEP state.**

0x520a/0x521a/0x522a/0x523a/0x524a: P*x* Port Input Enable Registers (P*x*_IEN)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
P0 Port Input Enable Register (P0_IEN)	**0x520a** (8 bits)	D7–0	**P0IEN[7:0]**	P0[7:0] port input enable	1	Enable	0	Disable	0xff	R/W	
P1 Port Input Enable Register (P1_IEN)	**0x521a** (8 bits)	D7–0	**P1IEN[7:0]**	P1[7:0] port input enable	1	Enable	0	Disable	0xff	R/W	
P2 Port Input Enable Register (P2_IEN)	**0x522a** (8 bits)	D7–0	**P2IEN[7:0]**	P2[7:0] port input enable	1	Enable	0	Disable	0xff	R/W	
P3 Port Input Enable Register (P3_IEN)	**0x523a** (8 bits)	D7–0	**P3IEN[7:0]**	P3[7:0] port input enable	1	Enable	0	Disable	0xff	R/W	
P4 Port Input Enable Register (P4_IEN)	**0x524a** (8 bits)	D7–4	**–**	reserved	–				–	–	1 when being read.
		D3–0	**P4IEN[3:0]**	P4[3:0] port input enable	1	Enable	0	Disable	0xff	R/W	

Note: The "*x*" in the bit names indicates the port number (0 to 4).

D[7:0] P*x*IEN[7:0]: P*x*[7:0] Port Input Enable Bits (P4 port is P4IEN[3:0])

Permits or prevents port input.
1(R/W): Permit (Default)
0(R/W): Prohibit

PxIEN[7:0] are input enable bits that correspond directly to the Px[7:0] ports. Setting to 1 enables the input signal level to be read from the Px_IN register, while setting to 0 prevents signal input and fixes the input data values read out to 0.

0x52a0: P0 Port Function Select Register (P0_PMUX)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
P0 Port Function Select Register (P0_PMUX)	0x52a0 (8 bits)	D7	–	reserved	–	–	–	0 when being read.
		D6	P03MUX	P03 port function select	1 #ADTRG 0 P03	0	R/W	
		D5	–	reserved	–	–	–	0 when being read.
		D4	P02MUX	P02 port function select	1 Reserved 0 P02/EXCL0	0	R/W	
		D3	–	reserved	–	–	–	0 when being read.
		D2	P01MUX	P01 port function select	1 REMI 0 P01	0	R/W	
		D1	–	reserved	–	–	–	0 when being read.
		D0	P00MUX	P00 port function select	1 REMO 0 P00	0	R/W	

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

D7 Reserved

D6 P03MUX: P03 Port Function Select Bit

1 (R/W): #ADTRG（ADC10SA）
0 (R/W): P03 port (default)

D5 Reserved

D4 P02MUX: P02 Port Function Select Bit

1 (R/W): Reserved
0 (R/W): P02 port / EXCL0 (T16 Ch. 0) (default)
*For EXCL0, input status can be selected as P*x*OEN[7:0]=0, P*x*IEN=1.

D3 Reserved

D2 P01MUX: P01 Port Function Select Bit

1 (R/W): REMI (REMC)
0 (R/W): P01 port (default)

D1 Reserved

D0 P00MUX: P00 Port Function Select Bit

1 (R/W): REMO (REMC)
0 (R/W): P00 port (default)

0x52a1: P0 Port Function Select Register (P0_PMUX)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
P0 Port Function Select Register (P0_PMUX)	0x52a1 (8 bits)	D7	–	reserved	–	–	–	0 when being read.
		D6	P07MUX	P07 port function select	1 #SPISS 0 P07	0	R/W	
		D5	–	reserved	–	–	–	0 when being read.
		D4	P06MUX	P06 port function select	1 SDI 0 P06	0	R/W	
		D3	–	reserved	–	–	–	0 when being read.
		D2	P05MUX	P05 port function select	1 SDO 0 P05	0	R/W	
		D1	–	reserved	–	–	–	0 when being read.
		D0	P04MUX	P04 port function select	1 SPICLK 0 P04	0	R/W	

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

D7 Reserved

D6 P07MUX: P07 Port Function Select Bit

1 (R/W): #SPISS（SPI slave）
0 (R/W): P07 port (default)

D5 Reserved

D4 P06MUX: P06 Port Function Select Bit

1 (R/W): SDI (SPI)
0 (R/W): P06 port (default)

D3 Reserved

D2 P05MUX: P05 Port Function Select Bit

1 (R/W): SDO (SPI)
0 (R/W): P05 port (default)

D1 Reserved

D0 P04MUX: P04 Port Function Select Bit

1 (R/W): SPICLK (REMC)
0 (R/W): P04 port (default)

0x52a2: P1 Port Function Select Register (P1_PMUX)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
P1 Port Function Select Register (P1_PMUX)	**0x52a2** (8 bits)	D7-6	**P13MUX [1:0]**	P13 port function select	P13MUX[1:0] / Port 0x3 / Reserved 0x2 / Reserved 0x1 / Reserved 0x0 / P13/EXCL1	0	R/W	
		D5	**–**	reserved	–	–	–	0 when being read.
		D4	**P12MUX**	P12 port function select	1 SIN / 0 P12	0	R/W	
		D3	**–**	reserved	–	–	–	0 when being read.
		D2	**P11MUX**	P11 function select	1 SOUT / 0 P11	0	R/W	
		D1	**–**	reserved	–	–	–	0 when being read.
		D0	**P10MUX**	P10 port function select	1 SCLK / 0 P10	0	R/W	

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

D[7:6] P13MUX: P13 Port Function Select Bit

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): Reserved
0x0 (R/W): P13 port / EXCL1 (T16 Ch.1) (default)
*For EXCL1, input status can be selected as P*x*OEN[7:0]=0, P*x*IEN=1.

D5 Reserved

D4 P12MUX: P12 Port Function Select Bit

1 (R/W): SIN (UART Ch.0)
0 (R/W): P12 port (default)

D3 Reserved

D2 P11MUX: P11 Port Function Select Bit

1 (R/W): SOUT (UART Ch.0)
0 (R/W): P11 port (default)

D1 Reserved

D0 P10MUX: P10 Port Function Select Bit

1 (R/W): SCLK (UART Ch.0)
0 (R/W): P10 port (default)

0x52a3: P1 Port Function Select Register (P1_PMUX)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
P1 Port Function Select Register (P1_PMUX)	0x52a3 (8 bits)	D7	–	reserved	–		–	–	0 when being read.
		D6	P17MUX	P17 port function select	1 AIN3	0 P17	0	R/W	
		D5-4	P16MUX [1:0]	P16 port function select	P16MUX[1:0]	Port	0	R/W	
					0x3	Reserved			
					0x2	Reserved			
					0x1	SCLK1			
					0x0	P16/EXCL4			
		D3-2	P15MUX [1:0]	P15 port function select	P15MUX[1:0]	Port	0	R/W	
					0x3	Reserved			
					0x2	Reserved			
					0x1	Reserved			
					0x0	P15/EXCL3			
		D1-0	P14MUX [1:0]	P14 port function select	P14MUX[1:0]	Port	0	R/W	
					0x3	Reserved			
					0x2	Reserved			
					0x1	Reserved			
					0x0	P14/EXCL2			

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

D7 Reserved

D6 P17MUX: P17 Port Function Select Bit

1 (R/W): AIN3 (ADC10SA Ch.3)
0 (R/W): P17 port (default)

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

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): SCLK1 (UART Ch.1)
0x0 (R/W): P16 port (default)

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

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): Reserved
0x0 (R/W): P15 port / EXCL3 (T16E Ch.0) (default)
*For EXCL3, input status can be selected as P*x*OEN[7:0]=0, P*x*IEN=1.

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

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): Reserved
0x0 (R/W): P14 port / EXCL2 (T16 Ch.2) (default)
*For EXCL2, input status can be selected as P*x*OEN[7:0]=0, P*x*IEN=1.

0x52a4: P2 Port Function Select Register (P2_PMUX)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
P2 Port Function Select Register (P2_PMUX)	0x52a4 (8 bits)	D7	–	reserved	–	–	–	0 when being read.
		D6	P23MUX	P23 port function select	1 SENB0 0 P23	0	R/W	
		D5	–	reserved	–	–	–	0 when being read.
		D4	P22MUX	P22 port function select	1 AIN0 0 P22	0	R/W	
		D3	–	reserved	–	–	–	0 when being read.
		D2	P21MUX	P21 port function select	1 AIN1 0 P21	0	R/W	
		D1	–	reserved	–	–	–	0 when being read.
		D0	P20MUX	P20 port function select	1 AIN2 0 P20	0	R/W	

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

D7 Reserved

D6 P23MUX: P23 Port Function Select Bit

1 (R/W): Reserved
0 (R/W): P23 port (default)

D5 Reserved

D4 P22MUX: P22 Port Function Select Bit

1 (R/W): AIN0 (ADC Ch.0)
0 (R/W): P22 port (default)

D3 Reserved

D2 P21MUX: P21 Port Function Select Bit

1 (R/W): AIN1 (ADC Ch.1)
0 (R/W): P21 port (default)

D1 Reserved

D0 P20MUX: P20 Port Function Select Bit

1 (R/W): AIN2 (ADC Ch.2)
0 (R/W): P20 port (default)

0x52a5: P2 Port Function Select Register (P2_PMUX)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
P2 Port Function Select Register (P2_PMUX)	**0x52a5** (8 bits)	D7-6	**P27MUX [1:0]**	P27 port function select	P27MUX[1:0]	Port	0	R/W	
					0x3	Reserved			
					0x2	Reserved			
					0x1	SOUT1			
					0x0	P27			
		D5-1	–	reserved	–		–	–	0 when being read.
		D0	**P24MUX**	P24 port function select	1 SENA0	0 P24	0	R/W	

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

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

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): SOUT1 (UART Ch.1)
0x0 (R/W): P27 port (default)

D[5:1] Reserved

D0 P24MUX: P24 Port Function Select Bit

1 (R/W): Reserved
0 (R/W): P24 port (default)

0x52a6: P3 Port Function Select Register (P3_PMUX)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
P3 Port Function Select Register (P3_PMUX)	**0x52a6** (8 bits)	D7-6	**P33MUX [1:0]**	P33 port function select	P33MUX[1:0]	Port	0	R/W	
					0x3	Reserved			
					0x2	SCL0			
					0x1	SCL1			
					0x0	P33			
		D5-4	**P32MUX [1:0]**	P32 port function select	P32MUX[1:0]	Port	0	R/W	
					0x3	Reserved			
					0x2	Reserved			
					0x1	SDA0			
					0x0	P32			
		D3-2	**P31MUX [1:0]**	P31 port function select	P31MUX[1:0]	Port	0	R/W	
					0x3	Reserved			
					0x2	Reserved			
					0x1	SCL0			
					0x0	P31			
		D1-0	**P30MUX [1:0]**	P30 port function select	P30MUX[1:0]	Port	0	R/W	
					0x3	Reserved			
					0x2	Reserved			
					0x1	SIN1 (UART)			
					0x0	P30			

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

D[7:6] P33MUX: P33 Port Function Select Bit

0x3 (R/W): Reserved
0x2 (R/W): SCL0 (I^2C master)
0x1 (R/W): SCL1 (I^2C slave)
0x0 (R/W): P33 port (default)

D[5:4] P32MUX: P32 Port Function Select Bit

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): SDA0 (I^2C master)
0x0 (R/W): P32 port (default)

D[3:2] P31MUX: P31 Port Function Select Bit

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): SCL0 (I^2C master)
0x0 (R/W): P31 port (default)

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

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): SIN1 (UART Ch.1)
0x0 (R/W): P30 port (default)

0x52a7: P3 Port Function Select Register (P3_PMUX)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
P3 Port Function Select Register (P3_PMUX)	**0x52a7** (8 bits)	D7-6	**P37MUX [1:0]**	P37 port function select	P37MUX[1:0]	Port	0	R/W	
					0x3	TOUT4			
					0x2	Reserved			
					0x1	TOUTN3			
					0x0	P37			
		D5-4	**P36MUX [1:0]**	P36 port function select	P36MUX[1:0]	Port	0	R/W	
					0x3	Reserved			
					0x2	Reserved			
					0x1	TOUT3			
					0x0	P36			
		D3-2	**P35MUX [1:0]**	P35 port function select	P35MUX[1:0]	Port	0	R/W	
					0x3	Reserved			
					0x2	#BFR			
					0x1	FOUT1			
					0x0	P35			
		D1-0	**P34MUX [1:0]**	P34 port function select	P34MUX[1:0]	Port	0	R/W	
					0x3	Reserved			
					0x2	SDA0			
					0x1	SDA1			
					0x0	P34			

The P34 to P37 input/output port pins are shared with the peripheral module pins. This register is used to select how the pins are used.

D[7:6] P37MUX: P37 Port Function Select Bit

0x3 (R/W): TOUT4 (T8OSC1)
0x2 (R/W): Reserved
0x1 (R/W): TOUTN3 (T16E Ch.0)
0x0 (R/W): P37 port (default)

D[5:4] P36MUX: P36 Port Function Select Bit

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): TOUT3 (T16E Ch.0)
0x0 (R/W): P36 port (default)

D[3:2] P31MUX: P31 Port Function Select Bit

0x3 (R/W): Reserved
0x2 (R/W): #BFR (I^2C slave)
0x1 (R/W): FOUT1 (OSC1)
0x0 (R/W): P35 port (default)

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

0x3 (R/W): Reserved
0x2 (R/W): SDA0 (I^2C master)
0x1 (R/W): SDA1 (I^2C slave)
0x0 (R/W): P34 port (default)

0x52a8: P4 Port Function Select Register (P4_PMUX)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
P4 Port Function Select Register (P4_PMUX)	0x52a8 (8 bits)	D7	–	reserved	–	–	–	0 when being read.
		D6	P43MUX	P43 port function select	1 P43 0 DCLK	0	R/W	
		D5	–	reserved	–	–	–	0 when being read.
		D4	P42MUX	P42 port function select	1 P42 0 DST2	0	R/W	
		D3	–	reserved	–	–	–	0 when being read.
		D2	P41MUX	P41 port function select	1 P41 0 DSIO	0	R/W	
		D1	–	reserved	–	–	–	0 when being read.
		D0	P40MUX	P40 port function select	1 FOUTH 0 P40	0	R/W	

The P40 to P43 input/output port pins are shared with the peripheral module pins. This register is used to select how the pins are used.

D7 Reserved

D6 P43MUX: P43 Port Function Select Bit
1 (R/W): P43 port
0 (R/W): DCLK (DBG) (default)

D5 Reserved

D4 P42MUX: P42 Port Function Select Bit
1 (R/W): P42 port
0 (R/W): DST2 (DBG) (default)

D3 Reserved

D2 P41MUX: P41 Port Function Select Bit
1 (R/W): P41 port
0 (R/W): DSIO (OBG) (default)

D1 Reserved

D0 P40MUX: P40 Port Function Select Bit
1 (R/W): FOUTH (HSCLK)
0 (R/W): P40 port (default)

10.8 Precautions

Operation clock

- The PCLK clock must be fed from the clock generator to access the input/output port.
  The prescaler output clock is also needed to operate the P0 and P1 port chattering filter. Switch on the prescaler when using this function.

Pull-up

- A delay will occur in the waveform rise-up depending on time constants such as pull-up resistance and pin load capacitance if the port pin is switched from Low level to High level by the internal pull-up resistor. Therefore, an appropriate wait time must be set for loading an input/output port.
- Input/output ports that are not used should be set with pull-up resistance enabled.

P0 and P1 port interrupts

- Reset the corresponding interrupt flags P0IF[7:0] (0x5207) and P1IF[7:0] (0x5217) within the interrupt processing routine following a port interrupt to prevent recurring interrupts.
- To prevent generating unnecessary interrupts, reset the corresponding interrupt flag—P0IF[7:0] (0x5207) or P1IF[7:0] (0x5217)—before permitting interrupts for the required port with the P0_IMSK register (0x5205) or P1_IMSK register (0x5215).

P0/P1 Port chattering filter circuit

- Input interrupts will not be accepted for a transition into SLEEP mode with the chattering filter left on. The chattering filter should be set off (no verification time) before executing the slp instruction.
- P0/P1 port interrupts must be blocked when P*x*_CHAT register (0x5208/0x5218) settings are being changed. Changing the setting while interrupts are permitted may generate inadvertent P0/P1 interrupts.
- The chattering filter verification time refers to the maximum pulse width that can be filtered. Generating an input interrupt requires a minimum input time of the verification time and a maximum input time of twice the verification time.
- A phenomenon may occur in which the internal signal oscillates due to the time elapsed until the signal reaches the threshold value if the input signal rise-up/drop-off time is delayed. Since input interrupts will malfunction under these conditions, the input signal rise-up/drop-off time should normally be set to 25 ns or less.

P0 port key-entry reset

- Make sure the specified ports are not simultaneously switched to Low during normal operations when using the P0 port key-entry reset function.
- The P0 port key entry reset function is disabled on initial resetting and cannot be used for resetting at power-on.
- The P0 port key-entry reset function cannot be used in SLEEP state.

11 16-bit Timer (T16)

11.1 16-bit Timer Overview

The S1C17003 incorporates a 3-channel 16-bit timer (T16).
The 16-bit timer consists of a 16-bit presettable down counter and a 16-bit reload data register holding the preset values. The timer counts down from the initial value set in the reload data register and outputs an underflow signal when the counter underflows. The underflow signal is used to generate an interrupt and an internal serial interface clock. The underflow cycle can be programmed by selecting the prescaler clock and reload data, enabling the application program to obtain time intervals and serial transfer speeds as required.
The timer also combines an event counter function via the input/output port pins and the external input signal pulse width measurement function.
Figure 11.1.1 illustrates the 16-bit timer configuration.

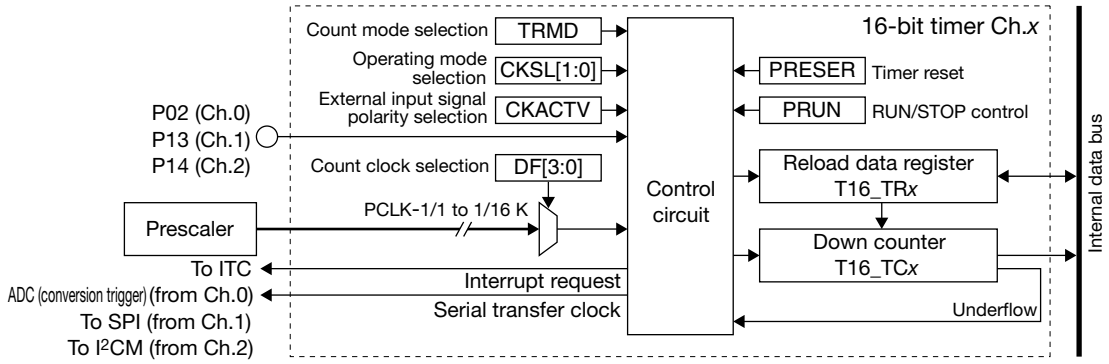


Figure 11.1.1: 16-bit timer configuration (1-channel)

Note: The 3-channel 16-bit timer module has the same functions except for the control register address. The description in this section applies to all channels of the 16-bit timer. The "*x*" in the register name refers to the channel number (0 to 2). The register addresses are referenced as "*Ch.0*," "*Ch.1*," and "*Ch.2*."
Example: T16_CTL*x* register (0x4226/0x4246/0x4266)
Ch.0: T16_CTL0 register (0x4226)
Ch.1: T16_CTL1 register (0x4246)
Ch.2: T16_CTL2 register (0x4266)

11.2 16-bit Timer Operating Modes

The 16-bit timer has the following three operating modes:
1. Internal clock mode (Normal timer counting internal clock)
2. External clock mode (Functions as event counter)
3. Pulse width measurement mode (Counts external input pulse width using internal clock)

The operating mode is selected using CKSL[1:0] (D[9:8]/T16_CTL*x* register).

∗ **CKSL[1:0]**: Input Clock and Pulse Width Count Mode Select Bits in the 16-bit Timer Ch.*x* Control (T16_CTL*x*) Register (D[9:8]/0x4226/0x4246/0x4266)

Table 11.2.1: Operating mode selection

CKSL[1:0]	Operating mode
0x3	Reserved
0x2	Pulse width measurement mode
0x1	External clock mode
0x0	Internal clock mode

(Default: 0x0)

11.2.1 Internal Clock Mode

Internal clock mode uses the prescaler output clock as the count clock.
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 time until underflow occurs can be finely programmed by selecting the prescaler clock and initial counter value, making it useful for serial transfer clock generation and sporadic time measurement.

Count clock selection

The count clock is selected by the DF[3:0] (D[3:0]/T16_CLK*x* register) from the 15 types generated by the prescaler dividing the PCLK clock into 1/1 to 1/16 K divisions.

∗ **DF[3:0]**: Timer Input Clock Select Bits in the 16-bit Timer Ch.*x* Input Clock Select (T16_CLK*x*) Register (D[3:0]/0x4220/0x4240/0x4260)

Table 11.2.1.1: Count clock selection

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: • The prescaler must run before operating the 16-bit timer in internal clock mode.

• Make sure the 16-bit timer count is halted before changing count clock settings.

For detailed information on the prescaler control, see "9 Prescaler (PSC)."

11.2.2 External Clock Mode

External clock mode uses the clock and pulses input via the input/output port as a count clock. These inputs can also be used as an event counter. Timer operations other than the input clock are the same as for internal clock mode.

External clock input port

The following input ports are used for external clock or pulse input.

Table 11.2.2.1: External clock input port

Timer channel	Input signal name	Input/output port pin
Ch.0	EXCL0	P02
Ch.1	EXCL1	P13
Ch.2	EXCL2	P14

Confirm that the input/output ports used for external clock or pulse input are set to input mode (the default setting). No pin function selection is needed. While the input/output ports function as general purpose inputs, the input signal is also sent to the 16-bit timer.

The P02, P13, and P14 ports used by 16-bit timer Ch.0, Ch.1 and Ch.2 incorporate chattering filter circuits and can also be used as EXCL*x* inputs. For instructions on controlling chattering filter circuits, see "10.5 P0 and P1 Port Chattering Filter Function."

Signal polarity selection

CKACTV (D10/T16_CTL*x* register) is used in this mode to select the falling edge or rising edge of the input signal for counting.

* **CKACTV**: External Clock Active Level Select Bit in the 16-bit Timer Ch.*x* Control (T16_CTL*x*) Register (D10/0x4226/0x4246/0x4266)

Counting down uses the rising edge when CKACTV is 1 (default) and uses the falling edge when set to 0.

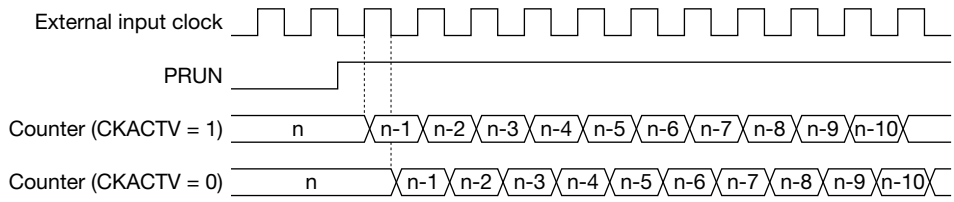


Figure 11.2.2.1: External clock mode count

The 16-bit timer does not use the prescaler in this mode. If no other peripheral modules use the prescaler clock, the prescaler can be stopped to reduce current consumption. (The prescaler clock is used for P0, P1 port chattering filtering.)

11.2.3 Pulse Width Measurement Mode

In pulse width measurement mode, when pulses with the specified polarity are input from the external clock port, the internal clock is fed only while the signal is active, enabling counting. This enables interrupt generation and input pulse width measurements for pulse inputs of the specified width or greater.

Pulse input port

The Input/output port used for external pulse input is the same as for external clock mode (see Table 11.2.2.1). Input pulses using the input/output port corresponding to the timer channel in input mode.

Count clock selection

Counting uses the prescaler output clock selected by DF[3:0] (D[3:0]/T16_CLK*x* register) in the same way as for internal clock mode. Select the clock to suit approximate input pulse widths and counting accuracy.

Signal polarity selection

CKACTV (D10/T16_CTL*x* register) is used to select the active level for the pulses counted. The High period is measured when CKACTV is 1 (default) and the Low period is measured when CKACTV is set to 0.

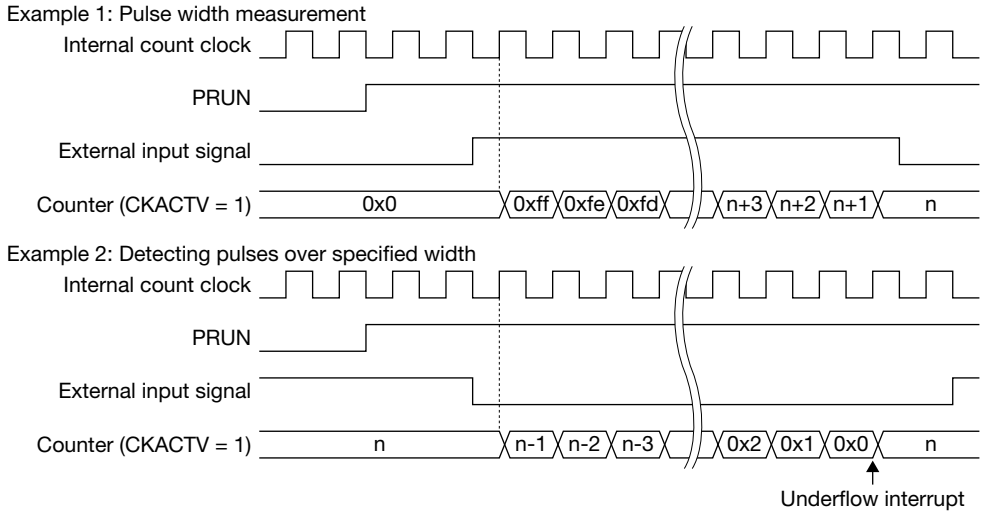


Figure 11.2.3.1: Pulse width measurement mode count operation

11.3 Count Mode

The 16-bit timer features two count modes: Repeat mode and One-shot mode. These modes are selected using the TRMD (D4/T16_CTL*x* register).

∗ **TRMD**: Count Mode Select Bit in the 16-bit Timer Ch.*x* Control (T16_CTL*x*) Register (D4/0x4226/0x4246/0x4266)

Repeat mode (TRMD = 0, default)

Setting TRMD to 0 sets the 16-bit timer to Repeat mode.

In this mode, once the count starts, the 16-bit timer continues running until stopped by the application program. If the counter underflows, the timer presets the reload data register value into the counter and continues the count. Thus, the timer periodically outputs an underflow pulse. The 16-bit timer should be set to this mode to generate periodic interrupts at desired intervals or to generate a serial transfer clock.

One-shot mode (TRMD = 1)

Setting TRMD to 1 sets the 16-bit timer to One-shot mode.

In this mode, the 16-bit timer stops automatically as soon as the counter underflows. This means only one interrupt can be generated after the timer starts. Note that the timer presets the reload data register value to the counter, then stops after an underflow has occurred. The 16-bit timer should be set to this mode to set a specific wait time or for pulse width measurement.

11.4 16-bit Timer Reload Register and Underflow Cycle

The reload data register T16_TR*x* (0x4222/0x4242/0x4262) is used to set the initial value for the down counter. The initial counter value set in the reload data register is preset to the down counter if the 16-bit timer is reset or the counter underflows. If the 16-bit timer is started after resetting, the timer counts down from the reload value (initial value). This means this reload value and the input clock frequency. determines the time elapsed from the point at which the timer starts until the underflow occurs (or between underflows). The time determined is used to obtain the specified wait time, the intervals between periodic interrupts, and the programmable serial interface transfer clock.

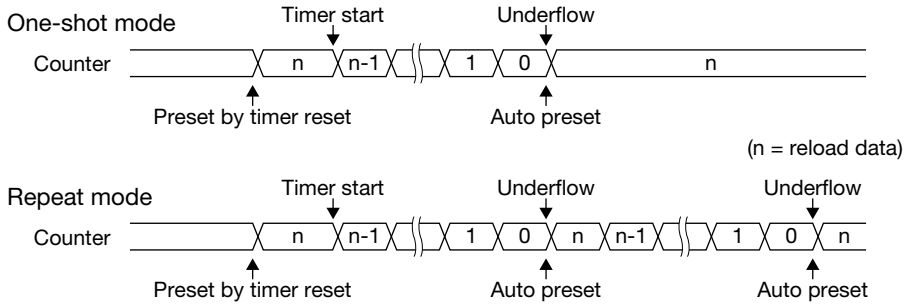


Figure 11.4.1: Preset timing

The underflow cycle can be calculated as follows:

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

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

11.5 16-bit Timer Reset

The 16-bit timer is reset by writing 1 to PRESER (D1/T16_CTL*x* register). The reload data is preset and the counter is initialized.

* **PRESER**: Timer Reset Bit in the 16-bit Timer Ch.*x* Control (T16_CTL*x*) Register (D1/0x4226/0x4246/0x4266)

11.6 16-bit Timer RUN/STOP Control

Make the following settings before starting the 16-bit timer.
(1) Select the operating mode (Internal clock, External clock, or Pulse width measurement). See Section 11.2.
(2) For Internal clock or Pulse width measurement mode, select the count clock (prescaler output clock). See Section 11.2.1.
(3) Set the count mode (One-shot or Repeat). See Section 11.3.
(4) Calculate the initial counter value and set the reload data register. See Section 11.4.
(5) Reset the timer and preset the counter to the initial value. See Section 11.5.
(6) If using timer interrupts, set the interrupt level and allow interrupts for the relevant timer channel. See Section 11.8.

To start the 16-bit timer, write 1 to PRUN (D0/T16_CTL*x* register).

∗ **PRUN**: Timer Run/Stop Control Bit in the 16-bit Timer Ch.*x* Control (T16_CTL*x*) Register (D0/0x4226/0x4246/0x4266)

The timer starts counting down from the initial value or from the current counter value if no initial value was preset. When the counter underflows, the timer outputs an underflow pulse and presets the counter to the initial value. An interrupt request is sent simultaneously to the interrupt controller (ITC).
If One-shot mode is set, the timer stops the count.
If Repeat mode is set, the timer continues to count from the reloaded initial value.

Write 0 to PRUN to stop the 16-bit timer via the application program. The counter stops counting and retains the current counter value until either the timer is reset or restarted. To restart the count from the initial value, the timer should be reset before writing 1 to PRUN.

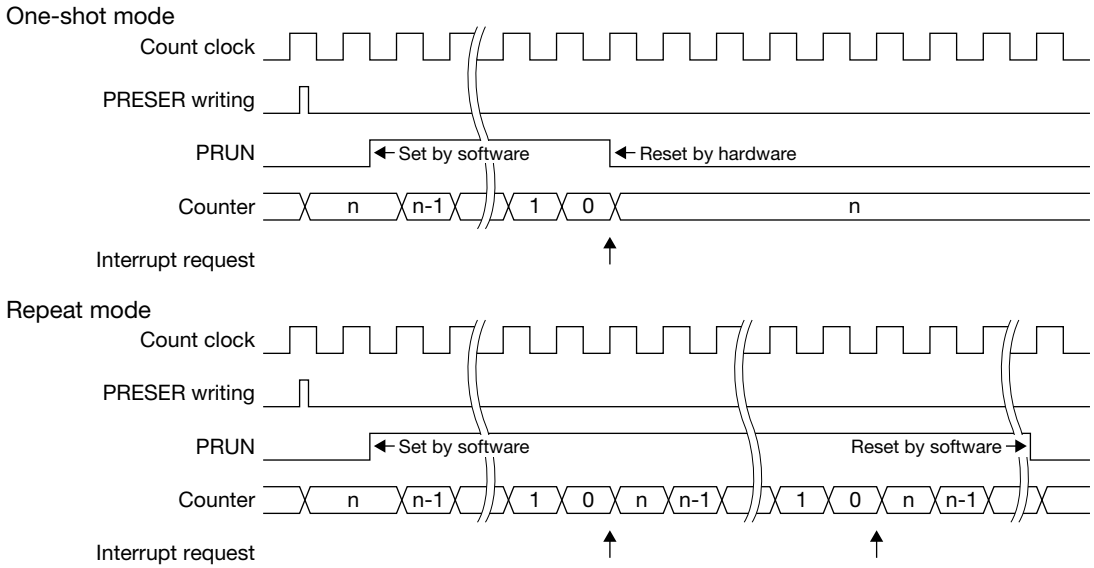


Figure 11.6.1: Count operation

In Pulse width measurement mode, the timer counts only while PRUN is set to 1 and the external input signal is at the specified active level. When the external input signal becomes inactive, the 16-bit timer stops counting and retains the counter value until the next active level input. (See Figure 11.2.3.1.)

11.7 16-bit Timer Output Signal

The 16-bit timer outputs underflow pulses when the counter underflows.
These pulses are used for timer interrupt requests.
These pulses are also used to generate the internal serial interface serial transfer clock.

The clock generated and underflow signal are sent to the internal serial interface, as shown below.
16-bit timer Ch.0 output underflow signal → ADC/10SA (conversion trigger)
16-bit timer Ch.1 output clock → SPI
16-bit timer Ch.2 output clock → I²C

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

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

I²CM $$TR = \frac{clk\_in}{bps \times 4} - 1$$

clk_in: Count clock (prescaler output clock) frequency [Hz]
TR: Reload data (0 to 65535)
bps: Transfer rate (bit/s)

11.8 16-bit Timer Interrupts

The 16-bit timer outputs interrupt requests to the interrupt controller (ITC) when the counter underflows.

Underflow interrupt

Generated by a counter underflow, this interrupt request sets the interrupt flag T16IF (D0/T16_INTx register) to 1 inside the T16 module provided for each channel.

* **T16IF**: 16-bit Timer Interrupt Flag in the 16-bit Timer Ch.*x* Interrupt Control (T16_INT*x*) Register (D0/0x4228/0x4248/0x4268)

To use this interrupt, set T16IE (D8/T16_INTx register) to 1. If T16IE is set to 0 (default), T16IF will not be set to 1, and the interrupt request for this cause will not be sent to the ITC.

* **T16IE**: 16-bit Timer Interrupt Enable Bit in the 16-bit Timer Ch.*x* Interrupt Control (T16_INT*x*) Register (D8/0x4228/0x4248/0x4268)

If T16IF is set to 1, the T16 module outputs an interrupt request to the ITC. An interrupt is generated if interrupt conditions are satisfied for the ITC and S1C17 core.

Note: • The T16 module interrupt flag T16IF must be reset within the interrupt processing routine following a 16-bit timer interrupt to prevent recurring interrupts.
• Reset T16IF before permitting 16-bit timer interrupts with T16IE to prevent unwanted interrupts occurring.

Interrupt vectors

The timer interrupt vector numbers and vector addresses are listed below.

Table 11.8.1: Timer interrupt vectors

Timer channel	Vector number	Vector address
16-bit Timer Ch.0	13 (0x0d)	TTBR + 0x34
16-bit Timer Ch.1	14 (0x0e)	TTBR + 0x38
16-bit Timer Ch.2	15 (0x0f)	TTBR + 0x3c

Other interrupt settings

The ITC allows the precedence of 16-bit timer interrupts to be set between level 0 (default) and level 7 for each channel. The PSR (S1C17 core internal processor status register) IE (interrupt enable) bit must be set to 1 to generate actual interrupts.

For specific information on interrupt processing, see "6 Interrupt Controller (ITC)."

11.9 Control Register Details

Table 11.9.1: 16-bit timer register list

Address	Register name		Function
0x4220	T16_CLK0	16-bit Timer Ch.0 Input Clock Select Register	Prescaler output clock selection
0x4222	T16_TR0	16-bit Timer Ch.0 Reload Data Register	Reload data setting
0x4224	T16_TC0	16-bit Timer Ch.0 Counter Data Register	Counter data
0x4226	T16_CTL0	16-bit Timer Ch.0 Control Register	Timer mode setting and timer RUN/STOP
0x4228	T16_INT0	16-bit Timer Ch.0 Interrupt Control Register	Interrupt Control
0x4240	T16_CLK1	16-bit Timer Ch.1 Input Clock Select Register	Prescaler output clock selection
0x4242	T16_TR1	16-bit Timer Ch.1 Reload Data Register	Reload data setting
0x4244	T16_TC1	16-bit Timer Ch.1 Counter Data Register	Counter data
0x4246	T16_CTL1	16-bit Timer Ch.1 Control Register	Timer mode setting and timer RUN/STOP
0x4248	T16_INT1	16-bit Timer Ch.1 Interrupt Control Register	Interrupt Control
0x4260	T16_CLK2	16-bit Timer Ch.2 Input Clock Select Register	Prescaler output clock selection
0x4262	T16_TR2	16-bit Timer Ch.2 Reload Data Register	Reload data setting
0x4264	T16_TC2	16-bit Timer Ch.2 Counter Data Register	Counter data
0x4266	T16_CTL2	16-bit Timer Ch.2 Control Register	Timer mode setting and timer RUN/STOP
0x4268	T16_INT2	16-bit Timer Ch.2 Interrupt Control Register	Interrupt Control

The 16-bit timer registers are described in detail below. These are 16-bit registers.

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

0x4220/0x4240/0x4260: 16-bit Timer Ch.*x* Input Clock Select Registers (T16_CLK*x*)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
16-bit Timer	0x4220	D15–4	–	reserved	–		–	–	0 when being read.
Ch.*x* Input	0x4240	D3–0	DF[3:0]	Timer input clock select	DF[3:0]	Clock	0x0	R/W	
Clock Select	0x4260			(Prescaler output clock)	0xf	reserved			
Register	(16 bits)				0xe	PCLK-1/16384			
(T16_CLK*x*)					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 "*x*" in the register names indicates the channel number (0 to 2).

D[15:4] Reserved

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

Select the 16-bit timer count clock from the 15 different prescaler output clocks.

Table 11.9.2: Count clock selection

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: Make sure the 16-bit timer count is halted before changing count clock settings.

0x4222/0x4242/0x4262: 16-bit Timer Ch.*x* Reload Data Registers (T16_TR*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
16-bit Timer Ch.*x* Reload Data Register (T16_TR*x*)	**0x4222** **0x4242** **0x4262** (16 bits)	D15–0	**TR[15:0]**	16-bit timer reload data TR15 = MSB TR0 = LSB	0x0 to 0xffff	0x0	R/W	

Note: The "*x*" in the register names indicates the channel number (0 to 2).

0x4222: 16-bit Timer Ch.0 Reload Data Register (T16_TR0)
0x4242: 16-bit Timer Ch.1 Reload Data Register (T16_TR1)
0x4262: 16-bit Timer Ch.2 Reload Data Register (T16_TR2)

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

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

The reload data set in this register is preset to the counter if the timer is reset or the counter underflows. If the 16-bit timer is started after resetting, the timer counts down from the reload value (initial value). This means this reload value and the input clock frequency determine the time elapsed from the point at which the timer starts until the underflow occurs (or between underflows). The time determined is used to obtain the desired wait time, the intervals between periodic interrupts, and the programmable serial interface transfer clock.

0x4224/0x4244/0x4264: 16-bit Timer Ch.*x* Counter Data Registers (T16_TC*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
16-bit Timer Ch.*x* Counter Data Register (T16_TC*x*)	**0x4224** **0x4244** **0x4264** (16 bits)	D15–0	**TC[15:0]**	16-bit timer counter data TC15 = MSB TC0 = LSB	0x0 to 0xffff	0xffff	R	

Note: The "*x*" in the register names indicates the channel number (0 to 2).

0x4224: 16-bit Timer Ch.0 Counter Data Register (T16_TC0)
0x4244: 16-bit Timer Ch.1 Counter Data Register (T16_TC1)
0x4264: 16-bit Timer Ch.2 Counter Data Register (T16_TC2)

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

Reads out the counter data. (Default: 0xffff)
This register is read-only and cannot be written to.

0x4226/0x4246/0x4266: 16-bit Timer Ch.*x* Control Registers (T16_CTL*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
16-bit Timer Ch.*x* Control Register (T16_CTL*x*)	0x4226 0x4246 0x4266 (16 bits)	D15–11	–	reserved	–	–	–	0 when being read.
		D10	CKACTV	External clock active level select	1 High 0 Low	1	R/W	
		D9–8	CKSL[1:0]	Input clock and pulse width measurement mode select	CKSL[1:0] Mode 0x3 reserved 0x2 Pulse width 0x1 External clock 0x0 Internal clock	0x0	R/W	
		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	

Note: The "*x*" in the register names indicates the channel number (0 to 2).

0x4226: 16-bit Timer Ch.0 Control Register (T16_CTL0)
0x4246: 16-bit Timer Ch.1 Control Register (T16_CTL1)
0x4266: 16-bit Timer Ch.2 Control Register (T16_CTL2)

D[15:11] Reserved

D10 CKACTV: External Clock Active Level Select Bit

Selects the external input pulse polarity or external clock counting edge.
1 (R/W): Active High/Rising edge (default)
0 (R/W): Active Low/Falling edge

This setting determines whether the external input clock rising edge or falling edge is used for counting in external clock mode (when CKSL[1:0] = 0x1). In pulse width measurement mode (when CKSL[1:0] = 0x2), this setting determines external input pulse polarity.

D[9:8] CKSL[1:0]: Input Clock and Pulse Width Measurement Mode Select Bits

Select the 16-bit timer operating mode.

Table 11.9.3: Operating mode selection

CKSL[1:0]	Operating mode
0x3	Reserved
0x2	Pulse width measurement mode
0x1	External clock mode
0x0	Internal clock mode

(Default: 0x0)

Internal clock mode uses the prescaler output clock as the count clock. 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 time until underflow occurs can be finely programmed by selecting the prescaler clock and initial counter value, allowing its use for serial transfer clock generation and sporadic time measurement.

External clock mode uses the clock and pulses input via the input/output ports (Ch.0: P02, Ch.1: P13, Ch.2: P14) as a count clock and can also be used as an event counter. Timer operations other than the input clock are the same as for internal clock mode.

In pulse width measurement mode, when pulses with the specified polarity are input from the external clock port, the internal clock is fed only while the signal is active, enabling counting. This enables interrupt generation and input pulse width measurements for pulse inputs of the specified width or greater.

D[7:5] Reserved

D4 TRMD: Count Mode Select Bit

Selects the 16-bit timer count mode.
1 (R/W): One-shot mode
0 (R/W): Repeat mode (default)

Setting TRMD to 0 sets the 16-bit timer to Repeat mode. In this mode, once the count starts, the 16-bit timer continues to run until stopped by the application. If the counter underflows, the timer presets the counter to the reload data register value and continues the count. Thus, the timer periodically outputs an underflow pulse. Set the 16-bit timer to this mode to generate periodic interrupts at desired intervals or to generate a serial transfer clock.

Setting TRMD to 1 sets the 16-bit timer to One-shot mode. In this mode, the 16-bit timer stops automatically as soon as the counter underflows. This means only one interrupt can be generated after the timer starts. Note that the timer presets the counter to the reload data register value, then stops when an underflow occurs. Set the 16-bit timer to this mode to set a specific wait time or for pulse width measurement.

D[3:2] Reserved

D1 PRESER: Timer Reset Bit

Resets the 16-bit timer.
1 (W): Reset
0 (W): Disabled
0 (R): Normally 0 when read out (default)

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

D0 PRUN: Timer Run/Stop Control Bit

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

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

0x4228/0x4248/0x4268: 16-bit Timer Ch.*x* Interrupt Control Registers (T16_INT*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
16-bit Timer Ch.*x* Interrupt Control Register (T16_INT*x*)	0x4228 0x4248 0x4268 (16 bits)	D15–9	–	reserved	–	–	–	0 when being read.
		D8	T16IE	16-bit timer interrupt enable	1 Enable 0 Disable	0	R/W	
		D7–1	–	reserved	–	–	–	0 when being read.
		D0	T16IF	16-bit timer interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.

Note: The "*x*" in register names indicates the channel number (0 to 2).

0x4228: 16-bit Timer Ch.0 Interrupt Control Register (T16_INT0)
0x4248: 16-bit Timer Ch.1 Interrupt Control Register (T16_INT1)
0x4268: 16-bit Timer Ch.2 Interrupt Control Register (T16_INT2)

D[15:9] Reserved

D8 T16IE: 16-bit Timer Interrupt Enable Bit

Permits or prevents interrupts caused by counter underflows for each channel.
1 (R/W): Permit interrupt
0 (R/W): Prevent interrupt (default)

Setting T16IE to 1 enables 16-bit timer interrupt requests to the ITC; setting to 0 prevents interrupts.

D[7:1] Reserved

D0 T16IF: 16-bit Timer Interrupt Flag

Interrupt flag indicating the counter underflow interrupt cause occurrence status for each channel.
1 (R): Interrupt cause present
0 (R): No interrupt cause (default)
1 (W): Reset flag
0 (W): Disable

T16IF is the T16 module interrupt flag. Setting T16IE (D8) to 1 sets the counter to 1 if an underflow occurs during counting. A 16-bit timer interrupt request signal is output to the ITC at the same time. An interrupt is generated if interrupt conditions are satisfied for the ITC and S1C17 core.
Writing 1 to this bit resets T16IF.

Note:
- **To prevent interrupt recurrences, the T16 module interrupt flag T16IF must be reset within the interrupt processing routine following a 16-bit timer interrupt.**
- **To prevent unwanted interrupts, reset T16IF before permitting 16-bit timer interrupts with T16IE.**

11.10 Precautions

- The prescaler must run before the 16-bit timer.
- Set the count clock and count mode only while the 16-bit timer count is stopped.
- To prevent interrupt recurrences, the T16 module interrupt flag T16IF (D0/T16_INTx register) must be reset within the interrupt processing routine following a 16-bit timer interrupt.

  * **T16IF**: 16-bit Timer Interrupt Flag in 16-bit Timer Ch.x Interrupt Control (T16_INTx) Register (D0/0x4228/0x4248/0x4268)

- To prevent unwanted interrupts, reset T16IF before permitting 16-bit timer interrupts with T16IE (D8/T16_INTx register).

  * **T16IE**: 16-bit Timer Interrupt Enable Bit in 16-bit Timer Ch.x Interrupt Control (T16_INTx) Register (D8/0x4228/0x4248/0x4268)

12 8-bit Timer (T8F)

12.1 8-bit Timer Overview

The S1C17003 incorporates a Dual channel 8-bit timer with Fine mode.
The 8-bit timer consists of an 8-bit presettable down counter and an 8-bit reload data register holding the preset values. 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 UART clock. The underflow cycle can be programmed by selecting the prescaler clock and reload data, enabling the application program to obtain time intervals and serial transfer speeds as required. Fine mode provides a function that minimizes transfer rate errors.
Figure 12.1.1 illustrates the 8-bit timer configuration.

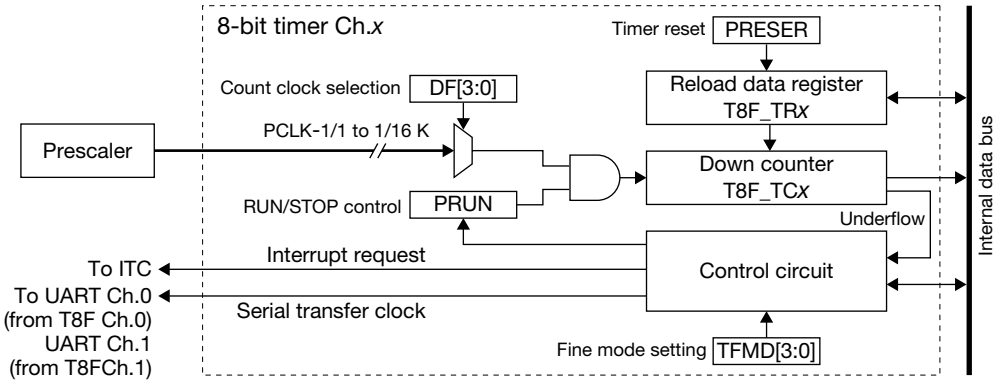


Figure 12.1.1: 8-bit timer configuration (Single channel)

Note: The 2-channel 8-bit timer modules have the same functions for both channels. Only the control register addresses are different. The description in this section applies to all 8-bit timer channels. The "*x*" in the register name indicates the channel number (0 or 1). Register addresses are given in the format (Ch.0/Ch.1).
Example: T8F_CTL*x* register (0x4206/0x4286)
Ch.0: T8F_CTL0 register (0x4206)
Ch.1: T8F_CTL1 register (0x4286)

12.2 8-bit Timer Count Mode

The 8-bit timer features two count modes: Repeat mode and One-shot mode. These modes are selected using the TRMD bit (D4/T8F_CTL register).

∗ **TRMD**: Count Mode Select Bit in the 8-bit Timer Ch.*x* Control (T8F_CTL*x*) Register (D4/0x4206/0x4286))

Repeat mode (TRMD = 0, default)

Setting TRMD to 0 sets the 8-bit timer to Repeat mode.

In this mode, once the count starts, the 8-bit timer continues running until stopped by the application program. If the counter underflows, the timer presets the reload data register value into the counter and continues the count. Thus, the timer periodically outputs an underflow pulse. The 8-bit timer should be set to this mode to generate periodic interrupts at desired intervals or to generate a serial transfer clock.

One-shot mode (TRMD = 1)

Setting TRMD to 1 sets the 8-bit timer to One-shot mode.

In this mode, the 8-bit timer stops automatically as soon as the counter underflows. This means only one interrupt can be generated after the timer starts. Note that the timer presets the reload data register value to the counter, then stops after an underflow has occurred. The 8-bit timer should be set to this mode to set a specific wait time.

Note: Make sure the 8-bit timer count is halted before changing count mode settings.

12.3 Count Clock

The 8-bit timer uses the prescaler output clock as the count clock. The prescaler generates 15 different clocks by dividing the PCLK clock into 1/1 to 1/16 K divisions. One of these is selected by the DF[3:0] bit (D[3:0]/T8F_CLK register).

* **DF[3:0]**: Timer Input Clock Select Bits in the 8-bit Timer Ch.*x* Input Clock Select (T8F_CLK*x*) Register (D[3:0]/ 0x4200/0x4280)

Table 12.3.1: Count clock selection

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: • **The prescaler must run before the 8-bit timer.**

• **Make sure the 8-bit timer count is halted before changing count clock settings.**

For detailed information on the prescaler control, see "9 Prescaler (PSC)."

12.4 8-bit Timer Reload Register and Underflow Cycle

The reload data register T8F_TR (0x4202/0x4282) is used to set the initial value for the down counter.
The initial counter value set in the reload data register is preset to the down counter if the 8-bit timer is reset or the counter underflows. If the 8-bit timer is started after resetting, the timer counts down from the reload value (initial value). This means this reload value and the input clock frequency. determines the time elapsed from the point at which the timer starts until the underflow occurs (or between underflows). The time determined is used to obtain the specified wait time, the intervals between periodic interrupts, and the programmable serial interface transfer clock.

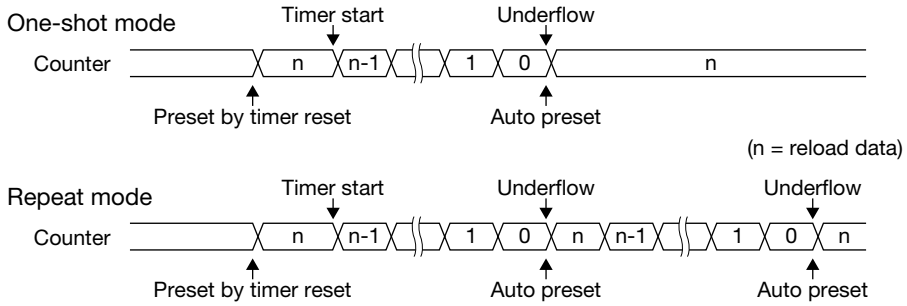


Figure 12.4.1: Preset timing

The underflow cycle can be calculated as follows:

$$\text{Underflow interval} = \frac{\text{T8F\_TR} + 1}{\text{clk\_in}} \text{ [s]} \qquad \text{Underflow cycle} = \frac{\text{clk\_in}}{\text{T8F\_TR} + 1} \text{ [Hz]}$$

clk_in: Count clock (prescaler output clock) frequency [Hz]
T8F_TR: Reload data (0 to 255)

Note: The UART generates a sampling clock that divides the 8-bit timer output into 1/16 divisions. Be careful when setting the transfer rate.

12.5 8-bit Timer Reset

The 8-bit timer is reset by writing 1 to PRESER bit (D1/T8F_CTL*x* register). The reload data is preset and the counter is initialized.

* **PRESER**: Timer Reset Bit in the 8-bit Timer Ch.*x* Control (T8F_CTL*x*) Register (D1/0x4206/0x4286)

12.6 8-bit Timer RUN/STOP Control

Make the following settings before starting the 8-bit timer:
(1) Set the count mode (One-shot or Repeat). See Section 12.2.
(2) Select the count clock (prescaler output clock). See Section 12.3.
(3) Calculate the initial counter value and set it to the reload data register. See Section 12.4.
(4) Reset the timer and preset the initial value to the counter. See Section 12.5.
(5) If using timer interrupts, set the interrupt level and permit interrupts. See Section 12.9.

To start the 8-bit timer, write 1 to PRUN (D0/T8F_CTL*x* register).

* **PRUN**: Timer Run/Stop Control Bit in the 8-bit Timer Ch.*x* Control (T8F_CTL*x*) Register (D0/0x4206/0x4286)

The timer starts counting down from the initial value or from the current counter value if no initial value was preset. When the counter underflows, the timer outputs an underflow pulse and presets the counter to the initial value. An interrupt request is sent simultaneously to the interrupt controller (ITC).
If One-shot mode is set, the timer stops the count.
If Repeat mode is set, the timer continues to count from the reloaded initial value.

Write 0 to PRUN bit to stop the 8-bit timer via the application program. The counter stops counting and retains the current counter value until either the timer is reset or restarted. To restart the count from the initial value, the timer should be reset before writing 1 to PRUN.

Resetting the timer while counting is underway sets the counter to the reload register value and continues the count.

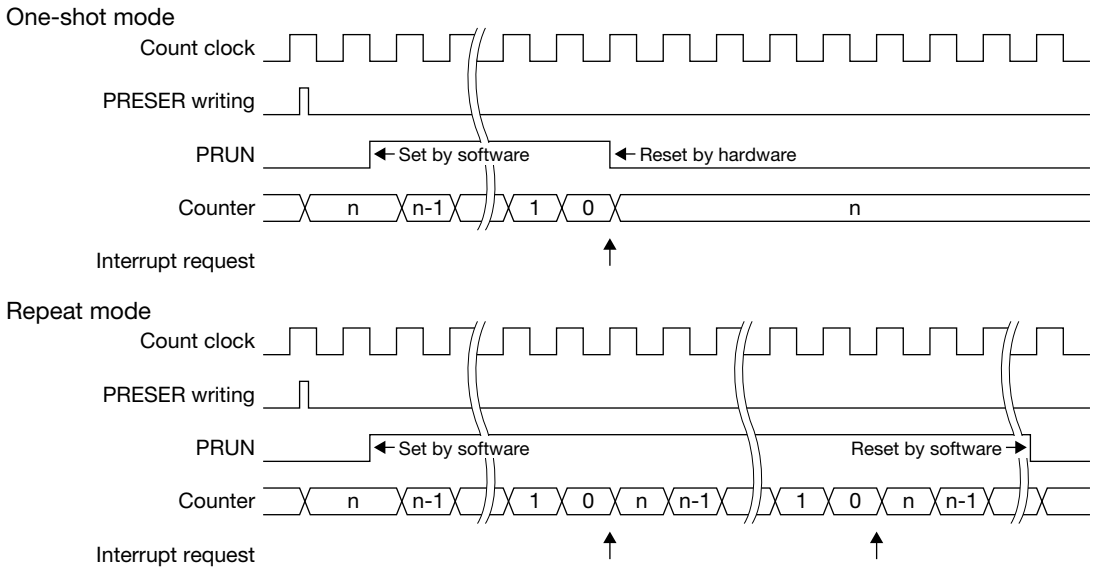


Figure 12.6.1: Count operation

12.7 8-bit Timer Output Signal

The 8-bit timer outputs underflow pulses when the counter underflows.
These pulses are used for timer interrupt requests.

The underflow pulses are also used to generate the serial transfer clock and are transmitted to the UART.

8-bit timer Ch.0 output clock → UART Ch.0
8-bit timer Ch.1 output clock → UART Ch.1

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

$$bps = \frac{clk\_in}{\{(T8F\_TR + 1) \times 16 + TFMD\}}$$

$$T8F\_TR = \left(\frac{clk\_in}{bps} - TFMD - 16\right) \div 16$$

clk_in: Count clock (prescaler output clock) frequency [Hz]
T8F_TR: Reload data (0 to 255)
bps: Transfer rate (bit/s)
TFMD: Fine mode setting (0 to15)

12.8 Fine Mode

Fine mode provides a function that minimizes transfer rate errors.
The 8-bit timer can output a programmable clock signal for use as the UART Ch.0 serial transfer clock. The timer output clock can be set to the required frequency by selecting the appropriate prescaler output clock and reload data. Note that errors may occur, depending on the transfer rate. Fine mode extends the output clock cycle by delaying the underflow pulse from the counter. This delay can be specified with the TFMD[3:0] bit (D[11:8]/ T8F_CTL register).

* **TFMD[3:0]**: Fine Mode Setup Bits in the 8-bit Timer Ch*x* Control (T8F_CTL*x*) Register (D[11:8]/0x4206/0x4286)

The TFMD[3:0] bit specifies the delay pattern to be inserted into the 16 underflow intervals. Inserting one delay extends the output clock cycle by one count clock cycle. This setting delays the interrupt timing in the same way.

Table 12.8.1: Delay patterns specified by 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 the insertion of a delay cycle.

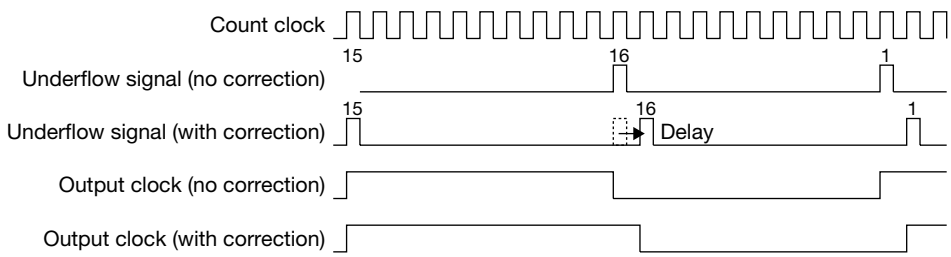


Figure 12.8.1: Delay cycle insertion in Fine mode

After the initial resetting, TFMD[3:0] is set to 0x0, preventing insertion of delay cycles.

12.9 8-bit Timer Interrupts

The 8-bit timer outputs interrupt requests to the interrupt controller (ITC) when the counter underflows.

Underflow interrupt

This interrupt request generated by a counter underflow sets the interrupt flag T8IF (D0/T8F_INT register) to 1 within the T8F module.

* **T8IF:** 8-bit Timer Interrupt Flag in the 8-bit Timer Ch*x* Interrupt Control (T8F_INT*x*) Register (D0/0x4208/0x4288))

To use this interrupt, set T8IE (D8/T8F_INT register) to 1. If T8IE is set to 0 (the default value), T8IF will not be set to 1, and interrupt request for this interrupt cause will not be sent to the ITC.

* **T8IE**: 8-bit Timer Interrupt Enable Bit in the 8-bit Timer Ch*x* Interrupt Control (T8F_INT*x*) Register (D8/0x4208/0x4288)

If T8IF is set to 1, the T8F module outputs an interrupt request to the ITC. An interrupt is generated if interrupt conditions are satisfied for the ITC and S1C17 core.

Note:
- **To prevent interrupt recurrences, the T8F module interrupt flag T8IF must be reset within the interrupt processing routine following an 8-bit timer interrupt.**
- **To prevent unwanted interrupts, reset T8IF before permitting 8-bit timer interrupts with T8IE.**
- **The 8-bit timer uses one interrupt signal for Ch.0 and Ch.1 interrupt requests to the ITC. The same interrupt processing routine is performed regardless of which interrupt is generated. When using both channel interrupts, read out the interrupt flag in the T8F module as part of the interrupt processing routine and check which channel generates the interrupt.**

Interrupt vectors

The 8-bit timer interrupt vector numbers and vector addresses are listed below.

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

Other interrupt settings

The ITC allows the priority of 8-bit timer interrupts to be set between level 0 (the default value) and level 7 for each channel. To generate actual interrupts, the PSR (S1C17 core internal processor status register) IE (interrupt enable) bit must be set to 1.
For more information on interrupt processing, see "6 Interrupt Controller (ITC)."

12.10 Control Register Details

Table 12.10.1: 8-bit timer register list

Address	Register name		Function
0x4280	T8F_CLK1	8-bit Timer Ch.1 Input Clock Select Register	Prescaler output clock selection
0x4282	T8F_TR1	8-bit Timer Ch.1 Reload Data Register	Reload data setting
0x4284	T8F_TC1	8-bit Timer Ch.1 Counter Data Register	Counter data
0x4286	T8F_CTL1	8-bit Timer Ch.1 Control Register	Timer mode setting and timer RUN/STOP
0x4288	T8F_INT1	8-bit Timer Ch.1 Interrupt Control Register	Interrupt control

The 8-bit timer registers are described in detail below. These are 16-bit registers.

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

0x4200/0x4280: 8-bit Timer Ch.*x* Input Clock Select Register (T8F_CLK*x*)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
8-bit Timer Ch*x* Input Clock Select Register (T8F_CLK*x*)	**0x4200** **0x4280** (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 indication "*x*" in the register name indicates the channel number (0 or 1).

0x4200: 8-bit Timer Ch.0 Input Clock Select Register (T8F_CLK0)
0x4280: 8-bit Timer Ch.1 Input Clock Select Register (T8F_CLK1)

D[15:4] Reserved

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

Select the 8-bit timer count clock from the 15 different prescaler output clocks.

Table 12.10.2: Count clock selection

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: Make sure the 8-bit timer count is halted before changing count clock settings.

0x4202/0x4282: 8-bit Timer Ch.*x* Reload Data Register (T8F_TR*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
8-bit Timer Ch*x* Reload Data Register (T8F_TR*x*)	**0x4202** (16 bits) / **0x4282**	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 indication "*x*" in the register name indicates the channel number (0 or 1).

0x4202: 8-bit Timer Ch.0 Reload Data Register (T8F_TR0)
0x4282: 8-bit Timer Ch.1 Reload Data Register (T8F_TR1)

D[15:8] Reserved

D[7:0] TR[7:0]: 8-bit Timer Reload Data

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

The reload data set in this register is preset to the counter if the timer is reset or the counter underflows. If the 8-bit timer is started after resetting, the timer counts down from the reload value (initial value). This means this reload value and the input clock frequency determine the time elapsed from the point at which the timer starts until the underflow occurs (or between underflows). The time determined is used to obtain the desired wait time, the intervals between periodic interrupts, and the programmable serial interface transfer clock.

0x4204/0x4284: 8-bit Timer Ch.*x* Counter Data Register (T8F_TC*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
8-bit Timer Ch*x* Counter Data Register (T8F_TC*x*)	**0x4204** (16 bits)	D15–8	**–**	reserved	–	–	–	0 when being read.
	0x4284	D7–0	**TC[7:0]**	8-bit timer counter data TC7 = MSB TC0 = LSB	0x0 to 0xff	0xff	R	

Note: The indication "*x*" in the register name indicates the channel number (0 or 1).

0x4204: 8-bit Timer Ch.0 Counter Data Register (T8F_TC0)
0x4284: 8-bit Timer Ch.1 Counter Data Register (T8F_TC1)

D[15:8] Reserved

D[7:0] TC[7:0]: 8-bit Timer Counter Data
Reads out the counter data. (Default: 0xff)
This register is read-only and cannot be written to.

0x4206/0x4286: 8-bit Timer Ch.*x* Control Register (T8F_CTL*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
8-bit Timer Ch*x* Control Register (T8F_CTL*x*)	**0x4206** (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	

Note: The indication "*x*" in the register name indicates the channel number (0 or 1).

0x4206: 8-bit Timer Ch.0 Control Register (T8F_CTL0)
0x4286: 8-bit Timer Ch.1 Control Register (T8F_CTL1)

D[15:12] Reserved

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

Correct the transfer rate error. (Default: 0x0)

The TFMD[3:0] bit specifies the delay pattern to be inserted into the 16 underflow intervals. Inserting one delay extends the output clock cycle by one count clock cycle. This setting delays the interrupt timing in the same way.

Table 12.10.3: Delay patterns specified by 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 the insertion of a delay cycle.

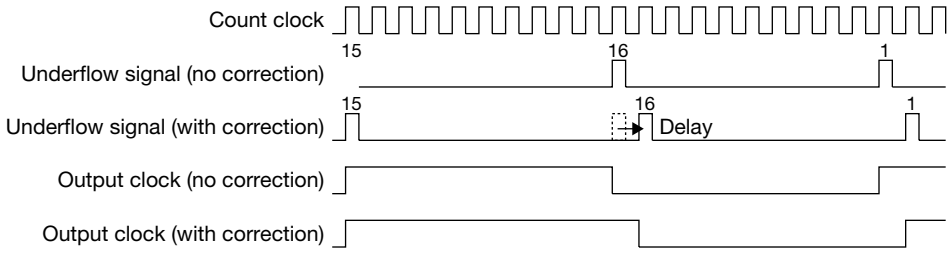


Figure 12.10.1: Delay cycle insertion in Fine mode

D[7:5] Reserved

D4 TRMD: Count Mode Select Bit

Selects the 8-bit timer count mode.
1 (R/W): One-shot mode
0 (R/W): Repeat mode (default)

Setting TRMD to 0 sets the 8-bit timer to Repeat mode. In this mode, once the count starts, the 8-bit timer continues to run until stopped by the application. If the counter underflows, the timer presets the counter to the reload data register value and continues the count. Thus, the timer periodically outputs an underflow pulse. Set the 8-bit timer to this mode to generate periodic interrupts at desired intervals or to generate a serial transfer clock.

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

Note: Make sure the 8-bit timer count is halted before changing count mode settings.

D[3:2] Reserved

D1 PRESER: Timer Reset Bit

Resets the 8-bit timer.
1 (W): Reset
0 (W): Disabled
0 (R): Normally 0 when read out (default)

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

D0 PRUN: Timer Run/Stop Control Bit

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

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

0x4208/0x4288: 8-bit Timer Ch.*x* Interrupt Control Register (T8F_INT*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
8-bit Timer Ch*x* Interrupt Control Register (T8F_INT*x*)	0x4208 0x4288 (16 bits)	D15–9	–	reserved	–	–	–	0 when being read.
		D8	T8IE	8-bit timer interrupt enable	1 Enable 0 Disable	0	R/W	
		D7–1	–	reserved	–	–	–	0 when being read.
		D0	T8IF	8-bit timer interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.

Note: The indication "*x*" in the register name indicates the channel number (0 or 1).

0x4208: 8-bit Timer Ch.0 Interrupt Control Register (T8F_INT0)
0x4288: 8-bit Timer Ch.1 Interrupt Control Register (T8F_INT1)

D[15:9] Reserved

D8 T8IE: 8-bit Timer Interrupt Enable Bit

Permits or prevents interrupts caused by counter underflows for each channel.
1 (R/W): Permit interrupt
0 (R/W): Prevent interrupt (default)

Setting T8IE to 1 permits 8-bit timer interrupt requests to the ITC; setting to 0 prevents interrupts.

D[7:1] Reserved

D0 T8IF: 8-bit Timer Interrupt Flag

Interrupt flag indicating the counter underflow interrupt cause occurrence status for each channel.
1 (R): Interrupt cause present
0 (R): No interrupt cause (default)
1 (W): Reset flag
0 (W): Disable

T8IF is the T8F module interrupt flag. Setting T8IE (D8) to 1 sets the counter to 1 if an underflow occurs during counting. An 8-bit timer interrupt request signal is output to the ITC at the same time. An interrupt is generated if interrupt conditions are satisfied for the ITC and S1C17 core.
Writing 1 to this bit resets T8IF.

Note:
- To prevent interrupt recurrences, the T8 module interrupt flag T8IF must be reset within the interrupt processing routine following an 8-bit timer interrupt.
- To prevent unwanted interrupts, reset T8IF before permitting 8-bit timer interrupts with T8IE.
- The 8-bit timer uses one interrupt signal for Ch.0 and Ch.1 interrupt requests to the ITC. The same interrupt processing routine is performed regardless of which interrupt is generated. When using both channel interrupts, read out the interrupt flag in the T8F module as part of the interrupt processing routine and check which channel generates the interrupt.

12.11 Precautions

- The prescaler must run before the 8-bit timer.
- Set the count clock and count mode only while the 8-bit timer count is stopped.
- To prevent interrupt recurrences, the T8F module interrupt flag T8IF (D0/T8F_INT register) must be reset within the interrupt processing routine following an 8-bit timer interrupt.

  * **T8IF**: 8-bit Timer Ch*x* Interrupt Flag in the 8-bit Timer Ch*x* Interrupt Control (T8F_INT*x*) Register (D0/0x4208/0x4288)

- To prevent unwanted interrupts, reset T8IF before permitting 8-bit timer interrupts with T8IE (D8/T8F_INT register).

  * **T8IE**: 8-bit Timer Ch*x* Interrupt Enable Bit in the 8-bit Timer Ch*x* Interrupt Control (T8F_INT*x*) Register (D8/0x4208/0x4288)

- The 8-bit timer uses one interrupt signal for Ch.0 and Ch.1 interrupt requests to the ITC. The same interrupt processing routine is performed regardless of which interrupt is generated. When using both channel interrupts, read out the interrupt flag in the T8F module as part of the interrupt processing routine and check which channel generates the interrupt.

13 PWM Timer (T16E)

13.1 PWM Timer Overview

The S1C17003 incorporates a channel PWM Timer.
Figure 13.1.1 illustrates the PWM Timer configuration.

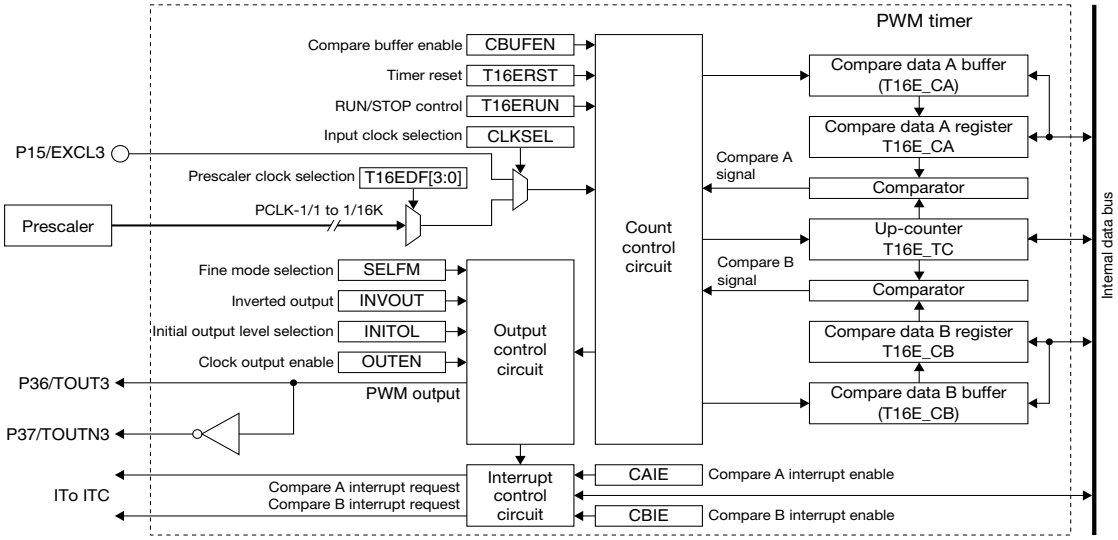


Figure 13.1.1: PWM Timer configuration

The PWM Timer includes a 16-bit up-counter (T16E_TC register), two 16-bit compare data registers (T16E_CA and T16E_CB registers), and the corresponding buffers.

Software can configure the count value of the 16-bit counter, and reset it to 0, while an external signal from the input/output port pin (EXCL3) or the Prescaler output clock counts up the 16-bit counter. Software can read the count value.

The compare data A and B registers hold data for comparison against the up-counter contents. Data can be read or written directly to or from the compare data registers. The compare data buffers enables loading to the compare data registers of comparison values set when the counter is reset by software or by a compare B match signal. Software can be used to set which of the compare data register and buffer the comparison values are written to.

If the counter value matches the contents of each compare data register, the comparator outputs a signal to control interrupts and output signals. These registers can be used to program the interrupt occurrence cycle and output clock frequency and duty ratio.

13.2 PWM Timer Operating Modes

The PWM Timer has the following two operating modes:
1. Internal clock mode (Timer counting internal clock)
2. External clock mode (Functions as event counter)

The operating mode is selected using CLKSEL (D3/T16E_CTL register).

∗ **CLKSEL**: Input Clock Select Bit in the PWM Timer Control (T16E_CTL) Register (D3/0x5306)

Setting CLKSEL to 0 (default) selects internal clock mode, while setting to 1 selects external clock mode.

Internal clock mode

Internal clock mode uses the prescaler output clock as the count clock.
The count clock is selected by the T16EDF[3:0] (D[3:0]/T16E_CLK register) from the 15 types generated by the prescaler dividing the PCLK clock into 1/1 to 1/16 K divisions.

∗ **T16EDF[3:0]**: Timer Input Clock Select Bits in the PWM Timer Input Clock Select (T16E_CLK) Register (D[3:0]/0x5308)

Table 13.2.1: Prescaler clock selection

T16EDF[3:0]	Prescaler output clock	T16EDF[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:
- **The prescaler must run before operating the PWM Timer in internal clock mode.**
- **Make sure the PWM Timer count is halted before changing count clock settings.**

For detailed information on the prescaler control, see "9 Prescaler (PSC)."

External clock mode

In external clock mode, channel 0 uses a clock or pulse input via the P15(EXCL3) port for the count clock. Therefore it can be used as an event counter. Timer operations other than input clock are the same as those in the internal clock mode.
To input the EXCL3 clock via the P15 port, write 0 to the P15MUX (D3-2/P1_PMUX register) to change the pin function, and set it to the input mode.

∗ **T15MUX**: P15 Port Function Select bit in the P0 Port Function Select (P1_PMUX) Register (D3-2/0x52a3)

The PWM Timer increments counts based on the input signal rising edge.

The PWM Timer does not use the prescaler in this mode. If no other peripheral modules are using the prescaler clock, the prescaler can be stopped to reduce current consumption.

13.3 Setting and Resetting Counter Value

The PWM Timer counter can be reset to 0 by writing 1 to the T16ERST bit (D1/T16E_CTL register).

* **T16ERST**: Timer Reset Bit in the PWM Timer Control (T16E_CTL) Register (D1/0x5306)

Normally, the counter should be reset by writing 1 to this bit before starting the count.
The counter is reset by hardware if the counter matches compare data B after the count starts.

The counter can also be set to any desired value by writing data to T16ETC[15:0] (D[15:0]/T16E_TC register).

* **T16ETC[15:0]**: Counter Data in the PWM Timer Counter Data (T16E_TC) Register (D[15:0]/0x5304)

13.4 Compare Data Settings

Compare data register/buffer selection

The PWM Timer incorporates a data comparator allowing comparison of counter data against any desired value. This comparison data is stored in the compare data A and B registers. Data can be read or written directly to or from the compare data registers.
The compare data buffers enable automatic loading to the compare data registers of the comparison values set in the buffers when the counter is reset by software (writing 1 to T16ERST) or by a compare B match signal. The CBUFEN (D5/T16E_CTL register) is used to set which of the compare data register and buffer the comparison values are written to.

* **CBUFEN**: Comparison Buffer Enable Bit in the PWM Timer Control (T16E_CTL) Register (D5/0x5306)

Writing 1 to CBUFEN selects the compare data buffer. Writing 0 to it selects the compare data register. The compare data register is selected after initial resetting.

Compare data writing

Compare data A is written to T16ECA[15:0] (D[15:0]/T16E_CA register). Compare data B is written to T16ECB[15:0] (D[15:0]/T16E_CB register).

* **T16ECA[15:0]**: Compare Data A in the PWM Timer Compare Data A (T16E_CA) Register (D[15:0]/0x5300)
* **T16ECB[15:0]**: Compare Data B in the PWM Timer Compare Data B (T16E_CB) Register (D[15:0]/0x5302)

When CBUFEN is set to 0, the compare data register values can be read or written directly by these registers.
When CBUFEN is set to 1, data is read from and written to these registers via the compare data buffers. The buffer contents are loaded into the compare data registers when the counter is reset.
The compare data registers and buffers are set to 0x0 after initial resetting.

The timer compares the count data against the compare data registers and generates a compare match signal if the values are equal. This compare match signal generates an interrupt and controls the clock (TOUT3/TOUTN3 signal) output externally.

Compare data B also determines the counter reset cycle.
The counter reset cycle can be calculated as follows:

$$\text{Counter reset interval} = \frac{CB + 1}{clk\_in} \text{ [s]}$$

$$\text{Counter reset cycle} = \frac{clk\_in}{CB + 1} \text{ [Hz]}$$

CB: Compare data B (T16E_CB register value)
clk_in: Prescaler output clock frequency

13.5 PWM Timer RUN/STOP Control

Set the following before starting the PWM Timer.
(1) Set the operating mode (input clock). See Section 13.2.
(2) Set the clock output. See Section 13.6.
(3) If using interrupts, set the interrupt level and permit interrupts for the PWM Timer. See Section 13.7.
(4) Set the counter value or reset to 0. See Section 13.3.
(5) Set the compare data. See Section 13.4.

The PWM Timer includes T16ERUN (D0/T16E_CTL register) to control Run/Stop.

* **T16ERUN**: Timer Run/Stop Control Bit in the PWM Timer Control (T16E_CTL) Register (D0/0x5306)

The timer starts counting when T16ERUN is written as 1. Writing 0 to T16ERUN prevents clock input and stops the count.
This control does not affect the counter data. The counter data is retained even when the count is halted, allowing resumption of the count from that data.
If T16ERUN and T16ERST are written as 1 simultaneously, the timer starts counting after the reset.

If the counter matches the compare data A register setting during counting, a compare A match signal is output and a compare A interrupt factor generated.
Likewise, if the counter matches the compare data B register setting, a compare B match signal is output and a compare B interrupt factor generated. The counter is reset to 0 at the same time. If CBUFEN is set to 1, the value set in the compare data buffers is loaded into the compare data registers. If interrupts are permitted, an interrupt request is sent to the interrupt controller (ITC).
In either case, counting continues unaffected. For compare B, counting starts from the counter value 0.

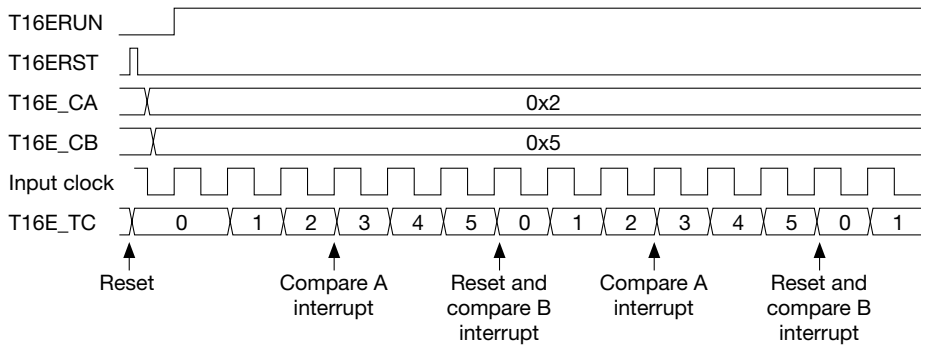


Figure 13.5.1: Basic counter operation timing

13.6 Clock Output Control

The PWM Timer can generate a TOUT3/TOUTN3 signal using the compare match signal.
Figure 13.6.1 shows the PWM Timer clock output circuit.

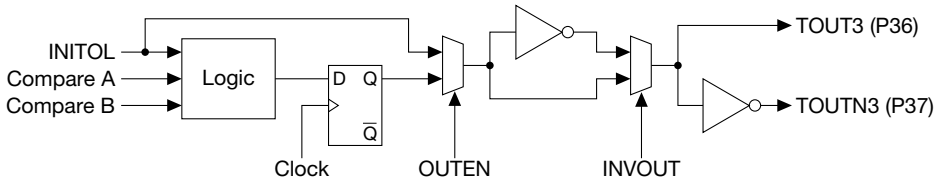


Figure 13.6.1: PWM Timer clock output circuit

Initial output level settings

The default output level is 0 (Low level) while the TOUT3 clock output is Off (TOUTN3 output is High level). This can be changed to 1 (TOUT3 = High level, TOUTN3 = Low level) with INITOL (D8/T16E_CTL register).

∗ **INITOL**: Initial Output Level Select Bit in the PWM Timer Ch.*x* Control (T16E_CTL) Register (D8/0x5306)

When INITOL is 0 (default), TOUT3 initial output level is low (TOUT3 output is High). When INITOL is set to 1, the initial output level should be high (TOUT3 output is Low).

Output signal polarity selection

By default, an active High (normal Low) TOUT3 output signal is generated (TOUTN3 output signal is active Low). This logic can be inverted by INVOUT (D4/T16E_CTL register). Writing 1 to INVOUT causes the timer to generate an active Low (normal High) TOUT3 signal (TOUTN3 signal is active High).

∗ **INVOUT**: Inverse Output Control Bit in the PWM Timer Control (T16E_CTL) Register (D4/0x5306)

Setting INVOUT to 1 also inverts the initial output level set for INITOL.
See Figure 13.6.2 for more information on output waveforms.

Output pin settings

The TOUT3/TOUTN3 signal generated here can be output from the following pins and can provide a programmable clock and PWM signal to external devices.
TOUT3 output → TOUT3 (P36) pin, TOUTN3 output → TOUTN3 (P37) pin

The pin used for output is set for input/output port use after initial resetting and switches to input mode. The pin then becomes high-impedance.
Switching the pin function to TOUT3/TOUTN3 output outputs the level set by INITOL and INVOUT. After the timer output starts, the output is maintained at this level until changed by the counter value.

Table 13.6.1: Initial output level

INITOL	INVOUT	Initial output level
1	1	Low
1	0	High
0	1	High
0	0	Low

Clock output start

To output the TOUT3 clock, write 1 to OUTEN (D2/T16E_CTL register). Writing 0 to OUTEN switches the output to the initial output level as set by INITOL and INVOUT.

* **OUTEN**: Clock Output Enable Bit in the PWM Timer Control (T16E_CTL) Register (D2/0x5306)

Figure 13.6.2 illustrates the output waveform.

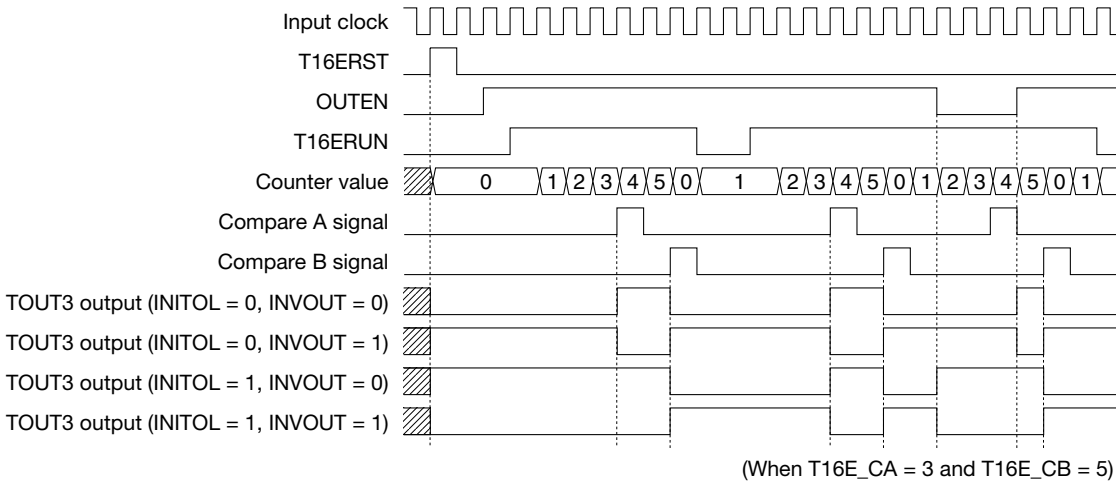


Figure 13.6.2: PWM Timer output waveform

TOUT3 output when INVOUT = 0 (Active High)

The timer outputs Low level (initial output level at output start) until the counter matches the compare data A set in the T16E_CA register (0x5300). When the counter reaches the next compare data A value, the output pin switches to High level, and a compare A interrupt factor is generated. If the counter subsequently counts up to compare data B set in the T16E_CB register (0x5302), the counter is reset and the output pin is returned to the Low level. A compare B interrupt factor is also generated at the same time.
The TOUTN3 pins output the inverted signals described above.

TOUT3 output when INVOUT = 1 (Active High)

The timer outputs High level (inverted value of the initial output level at output start) until the counter matches the compare data A set in the T16E_CA register (0x5300). When the counter reaches the next compare data A value, the output pin switches to Low level, and a compare A interrupt factor is generated. If the counter subsequently counts up to compare data B set in the T16E_CB register (0x5302), the counter is reset and the output pin is returned to the High level. A compare B interrupt factor is also generated at the same time.
The TOUTN3 pins output the inverted signals described above.

Clock output Fine mode settings

With the default settings, the clock output changes at the input clock rise-up if the counter value matches the compare data A.

If the counter data register T16ETC[14:0] matches the compare data A register T16ECA0[15:1], the Fine mode clock output changes in accordance with the compare data A bit 0 (T16ECA0) value.
When T16ECA0 is 0: Changes at input clock rise-up.
When T16ECA0 is 1: Changes at half-cycle delayed input clock drop-off.

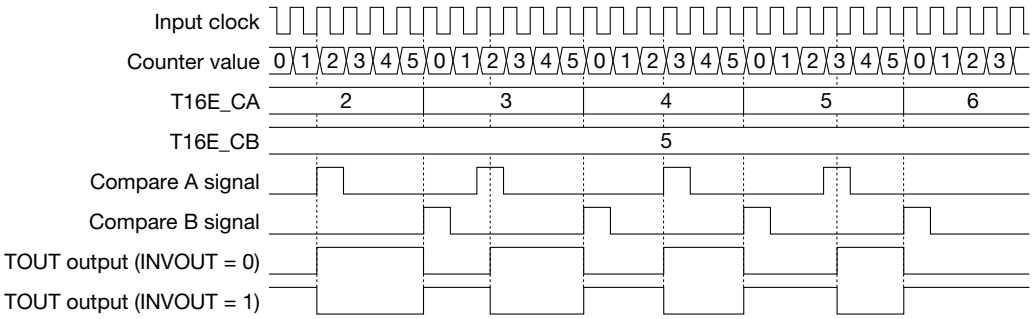


Figure 13.6.3: Fine mode clock output

The output duty can thus be adjusted in Fine mode in input clock half-cycle steps. Note that a pulse will be output with an input clock 1-cycle width when compare data A = 0 (same as for default). The maximum value for compare data B in Fine mode is $2^{15} - 1 = 32{,}767$, and the compare data A range will be 0 to (2 x compare data B - 1).

Fine mode is set by SELFM (D6/T16E_CTL register).

∗ **SELFM**: Fine Mode Select Bit in the PWM Timer Control (T16E_CTL) Register (D6/0x5306)

Writing 1 to SELFM sets Fine mode. Fine mode is disabled after initial resetting.

Precautions

(1) Compare data should be set with A ≥ 0 and B ≥ 1 when using the timer output. The minimum settings are A = 0 and B = 1, and the timer output cycle is half the input clock.

(2) Setting compare data with A > B (A > B × 2 for Fine mode) generates a compare B match signal only. It does not generate a compare A match signal. In this case, the TOUT3 output is fixed at Low (High when INVOUT = 1), and the TOUTN3 output is fixed at High (Low when INVOUT = 1).

13.7 PWM Timer Interrupts

The T16E module includes functions for generating the following two kinds of interrupts:
• Compare A match interrupt
• Compare B match interrupt

The T16E module outputs a single interrupt signal shared by the above two interrupt factors to the interrupt controller (ITC). The interrupt flag within the T16E module should be read to identify the interrupt factor that occurred.

Compare A match interrupt

This interrupt request is generated when the counter matches the compare data A register setting during counting. It sets the interrupt flag CAIF (D0/T16E_INT register) within the T16E module to 1.

∗ **CAIF**: Compare A Interrupt Flag in the PWM Timer Interrupt Flag (T16E_IFLG) Register (D0/0x530c)

To use this interrupt, set CAIE (D0/T16E_IMSK register) to 1. If CAIE is set to 0 (default), CAIF is not set to 1, and the interrupt request for this factor is not sent to the ITC.

∗ **CAIE**: Compare A Interrupt Enable Bit in the PWM Timer Interrupt Mask (T16E_IMSK) Register (D0/0x530a)

If CAIF is set to 1, the T16E module outputs an interrupt request to the ITC. An interrupt is generated if the ITC and S1C17 core interrupt conditions are satisfied.
CAIF should be read and checked within the PWM Timer interrupt processing routine to determine whether the PWM Timer interrupt is attributable to compare A matching.

Compare B match interrupt

This interrupt request is generated when the counter matches the compare data B register setting during counting. It sets the interrupt flag CBIF (D1/T16E_INT register) within the T16E module to 1.

∗ **CBIF**: Compare B Interrupt Flag in the PWM Timer Interrupt Flag (T16E_IFLG) Register (D1/0x530c)

To use this interrupt, set CBIE (D1/T16E_INT register) to 1. If CBIE is set to 0 (default), CBIF is not set to 1, and the interrupt request for this factor is not sent to the ITC.

∗ **CBIE**: Compare B Interrupt Enable Bit in the PWM Timer Interrupt Mask (T16E_IMSK) Register (D1/0x530a)

If CAIF is set to 1, the T16E module outputs an interrupt request to the ITC. An interrupt is generated if the ITC and S1C17 core interrupt conditions are satisfied.
CAIF should be read and checked within the PWM Timer interrupt processing routine to determine whether the PWM Timer interrupt is attributable to compare A matching.

Note: • To prevent interrupt recurrences, the T16E module interrupt flags CAIF and CBIF must be reset within the interrupt processing routine following a PWM Timer interrupt.

• To prevent generating unnecessary interrupts, reset the corresponding CAIF or CBIF before permitting compare A or compare B interrupts from CAIE or CBIE.

Interrupt vectors

The PWM Timer interrupt vector numbers and vector addresses are listed below.

Table 13.7.1: PWM Timer interrupt vectors

Timer channel	Vector number	Vector address
T16E	11 (0x0b)	TTBR + 0x2c

Other interrupt settings

The ITC allows the priority of PWM Timer interrupts to be set between level 0 (the default value) and level 7 for each channel. To generate actual interrupts, the PSR (S1C17 core internal processor status register) IE (interrupt enable) bit must be set to 1.

For more information on interrupt processing, see "6 Interrupt Controller (ITC)."

13.8 Control Register Details

Table 13.8.1: PWM Timer register list

Address	Register name		Function
0x5300	T16E_CA	PWM Timer Ch.0 Compare Data A Register	Compare data A setting
0x5302	T16E_CB	PWM Timer Ch.0 Compare Data B Register	Compare data B setting
0x5304	T16E_TC	PWM Timer Ch.0 Counter Data Register	Counter data
0x5306	T16E_CTL	PWM Timer Ch.0 Control Register	Timer mode setting and timer RUN/STOP
0x5308	T16E_CLK	PWM Timer Ch.0 Input Clock Select Register	Prescaler output clock selection
0x530a	T16E_IMSK	PWM Timer Ch.0 Interrupt Mask Register	Interrupt factor mask selection
0x530c	T16E_IFLG	PWM Timer Ch.0 Interrupt Flag Register	Interrupt factor checking

The PWM Timer registers are described in detail below. These are 16-bit registers.

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

0x5300: PWM Timer Compare Data A Register (T16E_CA)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
PWM Timer Compare Data A Register (T16E_CA)	**0x5300** (16 bits)	D15–0	**T16ECA[15:0]**	Compare data A T16ECA15 = MSB T16ECA0 = LSB	0x0 to 0xffff	0x0	R/W	

D[15:0] T16ECA[15:0]: Compare Data A

Sets the PWM Timer compare data A. (Default: 0x0)

When CBUFEN (D5/T16E_CTL register) is set to 0, this register can be used to directly read from or directly write to the compare data A register.

When CBUFEN is set to 1, data is read from and written to these registers via the compare data A buffer. The buffer contents are loaded into the compare data A register when the counter is reset.

The data set is compared against the counter data, and a compare A interrupt factor is generated if the contents match. The timer output waveform changes at the same time (rising when INVOUT (D4/T16E_CTL register) = 0 and trailing when INVOUT = 1). These processes do not affect the counter data or the count process.

0x5302: PWM Timer Compare Data B Register (T16E_CB)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
PWM Timer Compare Data B Register (T16E_CB)	**0x5302** (16 bits)	D15–0	**T16ECB[15:0]**	Compare data B T16ECB15 = MSB T16ECB0 = LSB	0x0 to 0xffff	0x0	R/W	

D[15:0] T16ECB[15:0]: Compare Data B

Sets the PWM Timer compare data B. (Default: 0x0)

When CBUFEN (D5/T16E_CTL register) is set to 0, this register can be used to directly read from or directly write to the compare data B register.

When CBUFEN is set to 1, data is read from and written to these registers via the compare data B buffer. The buffer contents are loaded into the compare data B register when the counter is reset.

The data set is compared against the counter data, and a compare B interrupt factor is generated if the contents match. The timer output waveform changes at the same time (rising when INVOUT (D4/T16E_CTL register) = 0 and trailing when INVOUT = 1). The counter is reset to 0.

0x5304: PWM Timer Counter Data Register (T16E_TC)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
PWM Timer Counter Data Register (T16E_TC)	**0x5304** (16 bits)	D15–0	**T16ETC[15:0]**	Counter data T16ETC15 = MSB T16ETC0 = LSB	0x0 to 0xffff	0x0	R/W	

D[15:0] T16ETC[15:0]: Counter Data

Counter data can be read out. (Default: 0x0)

The counter value can also be set by writing data to this register.

0x5306: PWM Timer Control Register (T16E_CTL)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
PWM Timer Control Register (T16E_CTL)	0x5306 (16 bits)	D15–9	–	reserved	–				–	–	0 when being read.
		D8	INITOL	Initial output level	1	High	0	Low	0	R/W	
		D7	–	reserved	–				–	–	0 when being read.
		D6	SELFM	Fine mode select	1	Fine mode	0	Normal mode	0	R/W	
		D5	CBUFEN	Comparison buffer enable	1	Enable	0	Disable	0	R/W	
		D4	INVOUT	Inverse output	1	Invert	0	Normal	0	R/W	
		D3	CLKSEL	Input clock select	1	External	0	Internal	0	R/W	
		D2	OUTEN	Clock output enable	1	Enable	0	Disable	0	R/W	
		D1	T16ERST	Timer reset	1	Reset	0	Ignored	0	W	0 when being read.
		D0	T16ERUN	Timer run/stop control	1	Run	0	Stop	0	R/W	

D[15:9] Reserved

D8 INITOL: Initial Output Level Bit

Sets the timer output initial output level.
1 (R/W): TOUT3 = High, TOUTN3 = Low
0 (R/W): TOUT3 = Low, TOUTN3 = High (default)

The timer output pin switches to the initial output level set here when the clock output is switched off by writing 0 to OUTEN (D2). Note that this level will be inverted when INVOUT (D4) is 1.

D7 Reserved

D6 SELFM: Fine Mode Select Bit

Sets the clock output to Fine mode.
1 (R/W): Fine mode
0 (R/W): Normal output (default)

When SELFM is set to 1, the clock output is set to Fine mode, and the output clock duty becomes adjustable in input clock half-cycle steps.
When SELFM is set to 0, normal clock output is used.

D5 CBUFEN: Comparison Buffer Enable Bit

Permits and prevents writing to the compare data buffer.
1 (R/W): Permitted
0 (R/W): Prohibited (default)

When CBUFEN is set to 1, compare data is read and written via the compare data buffer. The buffer contents are loaded into the compare data register when the counter is reset by software or compare B signal.
When CBUFEN is set to 0, compare data is read and written directly to and from the compare data register.

D4 INVOUT: Inverse Output Control Bit

Selects the timer output signal polarity.
1 (R/W): Inverted (TOUT3 = active Low, TOUTN3 = active High)
0 (R/W): Normal (TOUT3 = active High, TOUTN3 = active Low) (default)

Writing 1 to INVOUT generates a TOUT3 output active Low signal (Off level = High). When INVOUT is 0, an active High signal (Off level = Low) is generated.
Writing 1 to this bit also inverts the initial output level set by INITOL (D8). The signal level above is inverted for TOUTN3 output.

D3 CLKSEL: Input Clock Select Bit

Selects the timer input clock.
1 (R/W): External clock
0 (R/W): Internal clock (default)

Writing 0 to CLKSEL selects the internal clock (Prescaler output) for the timer input clock, while writing 1 selects the external clock (a clock input via the EXCL3 (P15) pin) and it functions as an event counter.

D2 OUTEN: Clock Output Enable Bit

Controls the TOUT3/TOUTN3 signal (timer output clock) output.
1 (R/W): Permitted
0 (R/W): Prohibited (default)

Writing 1 to OUTEN outputs the TOUT3/TOUTN3 signal from the corresponding output pin.
TOUT3 output → TOUT3 (P36) pin, TOUTN3 output → TOUTN3 (P37) pin

Writing 0 to OUTEN stops the output, and switches to the Off level corresponding to the settings for INVOUT (D4). The above pins must be set to TOUT3/TOUTN3 output using the port function selection register before outputting the TOUT3/TOUTN3 signals.

D1 T16ERST: Timer Reset Bit

Resets the counter.
1 (W): Reset
0 (W): Disabled
0 (R): Normally 0 when read out (default)

Writing 1 to T16ERST resets the PWM Timer counter.

D0 T16ERUN: Timer Run/Stop Control Bit

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

The PWM Timer starts the count when T16ERUN is written as 1 and stops when written as 0. The counter data is retained when stopped until the subsequent reset or run. Counting can be resumed when switched from Stop to Run from the data retained.

0x5308: PWM Timer Input Clock Select Register (T16E_CLK)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
PWM Timer Input Clock Select Register (T16E_CLK)	0x5308 (16 bits)	D15–4	–	reserved	–		–	–	0 when being read.
		D3–0	T16EDF[3:0]	Timer input clock select (Prescaler output clock)	T16EDF[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] T16EDF[3:0]: Timer Input Clock Select Bits

Select the PWM Timer count clock from the 15 different prescaler output clocks.

Table 13.8.2: Count clock selection

T16EDF[3:0]	Prescaler output clock	T16EDF[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: Make sure the PWM Timer count is halted before changing count clock settings.

0x530a: PWM Timer Interrupt Mask Registers (T16E_IMSK)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
PWM Timer Interrupt Mask Register (T16E_IMSK)	0x530a (16 bits)	D15–2	–	reserved	–	–	–	0 when being read.
		D1	CBIE	Compare B interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	CAIE	Compare A interrupt enable	1 Enable 0 Disable	0	R/W	

D[15:2] Reserved

D1 CBIE: Compare B Interrupt Enable Bit

Permits or prohibits compare B match interrupts.
1 (R/W): Interrupt permitted
0 (R/W): Interrupt prohibited (default)

Setting CBIE to 1 permits compare B interrupt requests to the ITC. Setting it to 0 prohibits interrupts.

D0 CAIE: Compare A Interrupt Enable Bit

Permits or prohibits compare A match interrupts.
1 (R/W): Interrupt permitted
0 (R/W): Interrupt prohibited (default)

Setting CAIE to 1 permits compare A interrupt requests to the ITC. Setting it to 0 prohibits interrupts.

0x530c: PWM Timer Interrupt Flag Registers (T16E_IFLG)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
PWM Timer Interrupt Flag Register (T16E_IFLG)	0x530c (16 bits)	D15–2	–	reserved	–				–	–	0 when being read.
		D1	CBIF	Compare B interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D0	CAIF	Compare A interrupt flag					0	R/W	

D[15:2] Reserved

D1 CBIF: Compare B Interrupt Flag

Interrupt flag indicating the compare B interrupt factor occurrence status.

1(R): Interrupt factor present
0(R): No interrupt factor (default)
1(W): Reset flag
0(W): Disabled

CBIF is the interrupt flag corresponding to compare B interrupts. Setting CBIE (D1/T16E_IMSK register) to 1 sets this to 1 when the counter matches the compare data B register setting during counting. A PWM Timer interrupt request signal is output to the ITC at the same time. This interrupt request signal generates an interrupt if the ITC and S1C17 core interrupt conditions are satisfied.
CBIF is reset by writing 1.

D0 CAIF: Compare A Interrupt Flag

Interrupt flag indicating the compare A interrupt factor occurrence status.

1(R): Interrupt factor present
0(R): No interrupt factor (default)
1(W): Reset flag
0(W): Disabled

CAIF is the interrupt flag corresponding to compare A interrupts. Setting CAIE (D0/T16E_IMSK register) to 1 sets this to 1 when the counter matches the compare data A register setting during counting. A PWM Timer interrupt request signal is output to the ITC at the same time. This interrupt request signal generates an interrupt if the ITC and S1C17 core interrupt conditions are satisfied.
CAIF is reset by writing 1.

Note:
- **To prevent interrupt recurrences, T16E module interrupt flags CAIF and CBIF must be reset within the interrupt processing routine following a PWM Timer interrupt.**
- **To prevent generating unnecessary interrupts, reset the corresponding CAIF or CBIF before permitting compare A or compare B interrupts from CAIE (D0/T16E_IMSK register) or CBIE (D1/T16E_IMSK register).**

13.9 Precautions

- The prescaler must run before operating the PWM Timer.
- Make sure the PWM Timer count is halted before changing count clock settings.
- Compare data should be set with A ≥ 0 and B ≥ 1 when using the timer output. The minimum settings are A = 0 and B = 1, and the timer output cycle is half the input clock.
- Setting compare data with A > B (A > B x 2 for Fine mode) generates a compare B match signal only. It does not generate a compare A match signal. In this case, the timer output is fixed at Low (High when INVOUT = 1).
- To prevent interrupt recurrences, the T16E module interrupt flags CAIF (D0/T16E_IFLG register) and CBIF (D1/T16E_IFLG register) must be reset within the interrupt processing routine following a PWM Timer interrupt.
- To prevent generating unnecessary interrupts, reset the corresponding CAIF (D0/T16E_IFLG register) or CBIF (D1/T16E_IFLG register) before permitting compare A or compare B interrupts from CAIE (D0/T16E_IMSK register) or CBIE (D1/T16E_IMSK register).

14 8-bit OSC1 Timer (T8OSC1)

14.1 8-bit OSC1 Timer Overview

The S1C17003 incorporates a single-channel 8-bit OSC1 timer that uses the OSC1 clock as its oscillation source. Figure 14.1.1 illustrates the 8-bit OSC1 timer configuration.

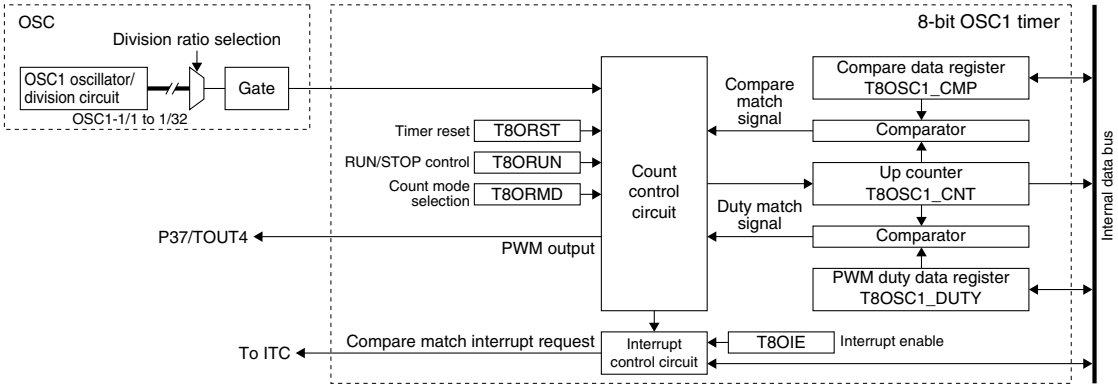


Figure 14.1.1: 8-bit OSC1 timer configuration

The 8-bit OSC1 timer includes an 8-bit up-counter (T8OOSC1_CNT register), an 8-bit compare data register (T8OSC1_CMP register), and an 8-bit PWM duty data register (T8OSC1_DUTY register).

The 8-bit counter can be reset to 0 by software and counts up using the OSC1 division clock (OSC1-1/1 to OSC1-1/32). The count value can be read by software.

The compare data and PWM duty registers store the data used for comparisons against up-counter contents.
If the counter values match the contents of each data register, the comparator outputs a signal to control the interrupts and the PWM output signal. The compare data register can be used to set the interrupt generating and PWM output clock frequencies. The PWM duty data register can be used to set the PWM output clock duty ratio.

14.2 8-bit OSC1 Timer Count Mode

The 8-bit OSC1 timer features two count modes: Repeat mode and One-shot mode. These modes are selected using the T8ORMD bit (D1/T8OSC1_CT register).

∗ **T8ORMD**: Count Mode Select Bit in the 8-bit OSC1 Timer Control (T8OSC1_CTL) Register (D1/0x50c0)

Repeat mode (T8ORMD = 0, default)

Setting T8ORMD to 0 sets the 8-bit OSC1 timer to Repeat mode.

In this mode, once the count starts, the 8-bit OSC1 timer continues running until stopped by the application program. If the counter matches the compare data, the timer resets the counter and continues counting. The interrupt signal is output at the same time. The 8-bit OSC1 timer should be set to this mode to generate periodic interrupts at desired intervals or to perform PWM output.

One-shot mode (T8ORMD = 1)

Setting T8ORMD to 1 sets the 8-bit OSC1 timer to One-shot mode.

In this mode, the 8-bit OSC1 timer stops automatically as soon as the counter matches the compare data.

This means only one interrupt can be generated after the timer starts. Note that the timer resets the counter, then stops after a complete match has occurred. The 8-bit OSC1 timer should be set to this mode to set a specific wait time.

Note:
- **Make sure the 8-bit OSC1 timer count is halted before changing count mode settings.**
- **If count operation is activated while the count mode is set to one-shot mode, and the CPU enters halt state, the counter does not stop even when a compare match occurs, disabling one-shot operation.**

14.3 Count Clock

The 8-bit OSC1 timer uses the OSC1 division clock output by the OSC module as the count clock. The OSC module generates 6 different clocks by dividing the OSC1 clock into 1/1 to 1/32 divisions. One of these is selected by T8O1CK[2:0] (D[3:1]/OSC_T8OSC1 register).

* **T8O1CK[2:0]**: T8OSC1 Clock Division Ratio Select Bits in the T8OSC1 Clock Control (OSC_T8OSC1) Register (D[3:1]/0x5065)

Table 14.3.1: Count clock selection

T8O1CK[2:0]	Division ratio
0x7 to 0x6	Reserved
0x5	OSC1-1/32
0x4	OSC1-1/16
0x3	OSC1-1/8
0x2	OSC1-1/4
0x1	OSC1-1/2
0x0	OSC1-1/1

(Default: 0x0)

The clock feed to the 8-bit OSC1 timer is controlled using T8O1CE (D0/OSC_T8OSC1 register). The T8O1CE default setting is 0, which stops the clock feed. Setting T8O1CE to 1 sends the clock generated as above to the 8-bit OSC1 timer. If 8-bit OSC1 timer operation is not required, the clock feed should be stopped to reduce power consumption.

* **T8O1CE**: T8OSC1 Clock Enable Bit in the T8OSC1 Clock Control (OSC_T8OSC1) Register (D0/0x5065)

Note: Make sure the 8-bit OSC1 timer count is halted before changing count clock settings.

For detailed information on clock control, refer to "7 Oscillator Circuit (OSC)."

14.4 Resetting 8-bit OSC1 Timer

The 8-bit OSC1 Timer can be reset to 0 by writing 1 to the T8ORS bit (D4/T8OSC1_CTL register).

* **T8ORST**: Timer Reset Bit in the 8-bit OSC1 Timer Control (T8OSC1_CTL) Register (D4/0x50c0)

Normally, the counter should be reset by writing 1 to this bit before starting the count.
The counter is reset by hardware if the counter matches compare data after the count starts.

14.5 Compare Data Settings

Compare data is written to T8OCMP[7:0] (D[7:0]/T8OSC1_CMP register).

* **T8OCMP[7:0]**: Compare Data Bits in the 8-bit OSC1 Timer Compare Data (T8OSC1_CMP) Register (D[7:0]/0x50c2)

After initial resetting, the compare data register is set to 0x0.

The timer compares the count data against the compare data register and generates a compare match signal as well as resets the counter if the values are equal. This compare match signal can generate an interrupt.
The compare match cycle can be calculated as follows:

$$\text{Compare match interval} = \frac{\text{CMP} + 1}{\text{clk\_in}} \text{ [s]}$$

$$\text{Compare match cycle} = \frac{\text{clk\_in}}{\text{CMP} + 1} \text{ [Hz]}$$

CMP: Compare data (T8OSC1_CMP register value)
clk_in: 8-bit OSC1 timer count clock frequency

When the 8-bit OSC1 timer is used to generate a PWM signal, the compare data determines the frequency of the output signal. (For a discussion of PWM output, refer to Section 14.8.)

14.6 8-bit OSC1 Timer RUN/STOP Control

Set the following items before starting the 8-bit OSC1 timer.
(1) Set the count mode (One-shot or Repeat). See Section 14.2.
(2) Select the operation clock. See Section 14.3.
(3) If using interrupts, set the interrupt level and permit interrupts for the 8-bit OSC1 timer. See Section 14.7.
(4) Reset the timer. See Section 14.4.
(5) Set the compare data. See Section 14.5.
(6) To output PWM signals, set the PWM duty data. See Section 14.8.

The 8-bit OSC1 timer includes T8ORUN (D0/T8OSC1_CTL register) to control Run/Stop.

* **T8ORUN**: Timer Run/Stop Control Bit in the 8-bit OSC1 Timer Control (T8OSC1_CTL) Register (D0/0x50c0)

The timer starts counting when T8ORUN is written as 1. Writing 0 to T8ORUN prevents clock input and stops the count.
This control does not affect the counter data. The counter data is retained even when the count is halted, allowing resumption of the count from that data.
If T8ORUN and T8ORST are written as 1 simultaneously, the timer starts counting after the reset.

If the counter matches the compare data register setting during counting, a compare match signal is output and a compare interrupt factor generated.
Likewise, if the counter matches the compare data B register setting, a compare B match signal is output and a compare B interrupt factor generated. The counter is reset to 0 at the same time.
If interrupts are permitted, an interrupt request is sent to the interrupt controller (ITC).
If One-shot mode is set, the timer stops the count.
If Repeat mode is set, the timer continues to count from 0.

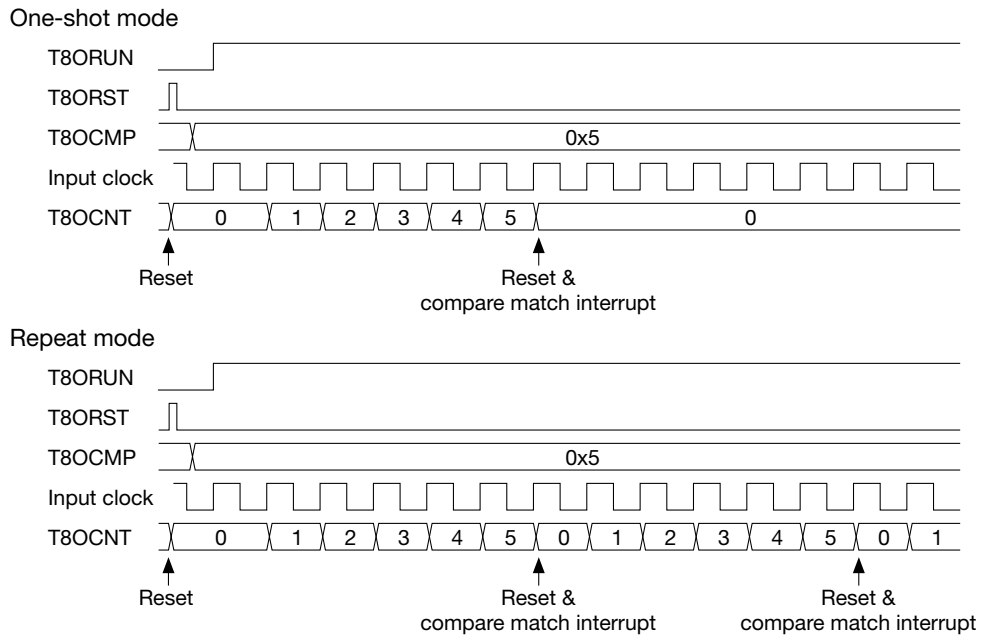


Figure 14.6.1: Basic counter operation timing

14.7 8-bit OSC1 Timer Interrupts

The T8OSC1 module outputs an interrupt request to the interrupt controller (ITC) by compare match.

Compare match interrupt

This interrupt request is generated when the counter matches the compare data register setting during counting. It sets the interrupt flag T8OIF (D0/T8OSC1_IFLG register) within the T8OSC1 module to 1.

∗ **T8OIF**: 8-bit OSC1 Timer Interrupt Flag in the 8-bit OSC1 Timer Interrupt Flag (T8OSC1_IFLG) Register (D0/0x50c4)

To use this interrupt, set T8OIE (D0/T8OSC1_IMSK register) to 1. If T8OIE is set to 0 (default), T8OIE is not set to 1, and the interrupt request for this factor is not sent to the ITC.

∗ **T8OIE**: 8-bit OSC1 Timer Interrupt Enable Bit in the 8-bit OSC1 Timer Interrupt Mask (T8OSC1_IMSK) Register (D0/0x50c3)

If T8OIF is set to 1, the T8OSC1 module outputs an interrupt request to the ITC. This interrupt request signal generates an interrupt if the ITC and S1C17 core interrupt conditions are satisfied.

Note: • To prevent interrupt recurrences, the T8OSC1 module interrupt flag T8OIF must be reset within the interrupt handler routine following an 8-bit OSC1 timer interrupt.

• To prevent generating unnecessary interrupts, reset the corresponding T8OIF before permitting compare 8-bit OSC1 interrupts from T8OIE.

Interrupt vectors

The 8-bit OSC timer interrupt vector numbers and vector addresses are listed below.

Vector number: 8 (0x08)
Vector address: TTBR + 0x20

Other interrupt settings

The ITC allows the priority of 8-bit OSC1 timer interrupts to be set between level 0 (the default value) and level 7. To generate actual interrupts, the PSR (S1C17 core internal processor status register) IE (interrupt enable) bit must be set to 1.
For more information on interrupt processing, see “6 Interrupt Controller (ITC).”

14.8 PWM output

The 8-bit OSC1 timer can generate a PWM signal in accordance with the compare data and PWM duty data settings and output it from the TOUT4 (P37) pin.

Output pin setting

The PWM output pin (TOUT4) also acts as a pin (P37) for a general-purpose input/output port. In the default state, this pin is set as a general-purpose input/output port pin. To use it as a PWM output pin, change the function by setting the value 3 in the P37MUX (D7-6/P3_PMUX register).

∗ **P37MUX**: P37 Port Function Select Bit in the P3 Port Function Select (P3_PMUX) Register (D7-6/0x52a7)

PWM waveform control

The PWM waveform frequency can be set by the compare data register (0x50c2) (see Section 14.5). The duty ratio can be adjusted by the PWM duty data register (0x50c5).
The timer outputs a Low level signal until the counter value matches the value of the PWM duty data register. When the counter value exceeds the value of the PWM duty data, the output pin changes to High. Once the counter counts up to the compare data register value, the counter is reset and the output pin returns to Low.
Figure 14.8.1 shows the output waveform.

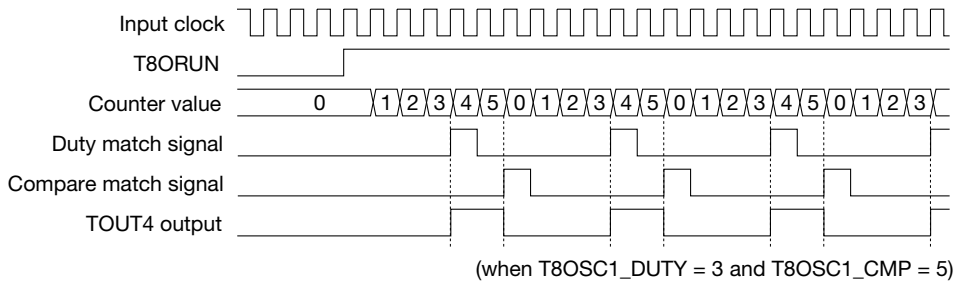


Figure 14.8.1 PWM output waveform

Precautions

(1) When using the timer output, set the following: PWM duty data ≥ 0, compare data ≥ 1. The minimum setting value is 0 for PWM duty data and 1 for compare data. The timer output cycle is 1/2 of the input clock.
(2) When the PWM duty data is set greater than the compare data, only the compare match signal will be generated. No duty match signal will be generated. In that case, the TOUT4 output is fixed to Low.

14.9 Control Register Details

Table 14.9.1: 8-bit OSC1 timer register list

Address	Register name		Function
0x50c0	T8OSC1_CTL	8-bit OSC1 Timer Control Register	Timer mode setting and timer RUN/STOP
0x50c1	T8OSC1_CNT	8-bit OSC1 Timer Counter Data Register	Counter data
0x50c2	T8OSC1_CMP	8-bit OSC1 Timer Compare Data Register	Compare data setting
0x50c3	T8OSC1_IMSK	8-bit OSC1 Timer Interrupt Mask Register	Interrupt mask setting
0x50c4	T8OSC1_IFLG	8-bit OSC1 Timer Interrupt Flag Register	Interrupt occurrence status display/resetting
0x50c5	T8OSC1_DUTY	8-bit OSC1 Timer PWM Duty Data Register	PWM output data setting

The 8-bit OSC1 timer registers are described in detail below. These are 8-bit registers.

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

0x50c0: 8-bit OSC1 Timer Control Register (T8OSC1_CTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
8-bit OSC1 Timer Control Register (T8OSC1_CTL)	0x50c0 (8 bits)	D7–5	–	reserved	–	–	–	0 when being read.
		D4	T8ORST	Timer reset	1 Reset 0 Ignored	0	W	
		D3–2	–	reserved	–	–	–	
		D1	T8ORMD	Count mode select	1 One shot 0 Repeat	0	R/W	
		D0	T8ORUN	Timer run/stop control	1 Run 0 Stop	0	R/W	

D[7:5] Reserved

D4 T8ORST: Timer Reset Bit

Resets the 8-bit OSC1 timer.

1 (W): Reset

0 (W): Disabled

0 (R): Normally 0 when read out (default)

Writing 1 to this bit resets the counter to 0.

D[3:2] Reserved

D1 T8ORMD: Count Mode Select Bit

Selects the 8-bit OSC1 timer count mode.

1 (R/W): One-shot mode

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

Setting T8ORMD to 0 sets the 8-bit OSC1 timer to Repeat mode. In this mode, once the count starts, the 8-bit timer continues to run until stopped by the application. If the counter matches the compare data register value, the timer resets the counter and continues counting. This means the timer periodically outputs a compare match signal. Set the 8-bit OSC1 timer to this mode to generate periodic interrupts at the desired interval or to perform PWM output.

Setting T8ORMD to 1 sets the 8-bit OSC1 timer to One-shot mode. In this mode, the 8-bit OSC1 timer stops automatically when the counter matches the compare data register value. This means an interrupt can be generated only once after the timer has been started. Note that the timer resets the counter and then stops after a compare match has occurred. Set the 8-bit OSC1 timer to this mode to create a specific wait time.

Note: Set the count mode only while the 8-bit OSC1 timer count is stopped.

D0 T8ORUN: Timer Run/Stop Control Bit

Controls the timer RUN/STOP.

1 (R/W): Run

0 (R/W): Stop (default)

The timer starts counting when T8ORUN is written as 1 and stops when written as 0. When the timer is stopped, the counter data is retained until reset or until the next RUN state.

0x50c1: 8-bit OSC1 Timer Counter Data Register (T8OSC1_CNT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
8-bit OSC1 Timer Counter Data Register (T8OSC1_CNT)	**0x50c1** (8 bits)	D7–0	**T8OCNT[7:0]**	Timer counter data T8OCNT7 = MSB T8OCNT0 = LSB	0x0 to 0xff	0x0	R	

D[7:0] T8OCNT[7:0]: Counter Data

Reads out the counter data. (Default: 0x0)

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

Note: The correct counter value may not be read out (reading is unstable) if the register is read while counting is underway.

Obtain the counter value by one of the following methods:

- Read the counter value while the counter is halted.
- Read the counter twice in succession. Treat the value as valid if the values read are identical.

0x50c2: 8-bit OSC1 Timer Compare Data Register (T8OSC1_CMP)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
8-bit OSC1 Timer Compare Data Register (T8OSC1_CMP)	**0x50c2** (8 bits)	D7–0	**T8OCMP[7:0]**	Compare data T8OCMP7 = MSB T8OCMP0 = LSB	0x0 to 0xff	0x0	R/W	

D[7:0] T8OCMP[7:0]: Compare Data

Sets the 8-bit OSC1 timer compare data. (Default: 0x0)

The data set is compared against the counter data, and a compare match interrupt factor is generated if the contents match. And the counter is reset to 0.

0x50c3: 8-bit OSC1 Timer Interrupt Mask Register (T8OSC1_IMSK)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
8-bit OSC1	0x50c3	D7–1	–	reserved	–				–	–	0 when being read.
Timer Interrupt Mask Register (T8OSC1_IMSK)	(8 bits)	D0	T8OIE	8-bit OSC1 timer interrupt enable	1	Enable	0	Disable	0	R/W	

D[7:1] Reserved

D0 T8OIE: 8-bit OSC1 Timer Interrupt Enable Bit

Permits or prohibits compare match interrupts.

1 (R/W): Interrupt permitted

0 (R/W): Interrupt prohibited (default)

Setting T8OIE to 1 permits 8-bit OSC1 timer interrupt requests to the ITC. Setting it to 0 prohibits interrupts.

0x50c4: 8-bit OSC1 Timer Interrupt Flag Register (T8OSC1_IFLG)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
8-bit OSC1	0x50c4	D7–1	–	reserved	–				–	–	0 when being read.
Timer Interrupt Flag Register (T8OSC1_IFLG)	(8 bits)	D0	T8OIF	8-bit OSC1 timer interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.

D[7:1] Reserved

D0 T8OIF: 8-bit OSC1 Timer Interrupt Flag

Interrupt flag indicating the compare match interrupt factor occurrence status.

1(R): Interrupt factor present
0(R): No interrupt factor (default)
1(W): Reset flag
0(W): Disabled

T8OIF is the T8OSC1 module interrupt flag. Setting T8OIE (D0/T8OSC1_IMSK register) to 1 sets this to 1 when the counter matches the compare data register setting during counting. An 8-bit OSC1 timer interrupt request signal output simultaneously to the ITC generates an interrupt if the ITC and S1C17 core interrupt conditions are met.

T8OIF is reset by writing as 1.

Note:
- **To prevent interrupt recurrences, the T8OSC1 module interrupt flag T8OIF must be reset within the interrupt handler routine following an 8-bit OSC1 timer interrupt.**
- **To prevent generating unnecessary interrupts, reset T8OIF before permitting compare match interrupts using T8OIE (D0/T8OSC1_IMSK register).**

0x50c5: 8-bit OSC1 Timer PWM Duty Data Register (T8OSC1_DUTY)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
8-bit OSC1 Timer PWM Duty Data Register (T8OSC1_DUTY)	**0x50c5** (8 bits)	D7–0	**T8ODTY[7:0]**	PWM output duty data T8ODTY7 = MSB T8ODTY0 = LSB	0x0 to 0xff	0x0	R/W	

D[7:0] T8ODTY[7:0]: PWM Output Duty Data

Sets the data that determines the duty ratio of PWM waveform. (default: 0x0)

The set data is compared against the counter data. If the contents match, the timer output waveform rises. If the counter data matches the compare data, the timer output waveform falls. These processes do not affect the counter data or count process.

14.10 Precautions

- The 8-bit OSC1 timer clock must be output from the OSC module before the 8-bit OSC1 timer begins running.
- Set the count clock and count mode only while the 8-bit OSC1 timer count is stopped.
- To prevent interrupt recurrences, the T8OSC1 module interrupt flag T8OIF (D0/T8OSC1_IFLG register) must be reset within the interrupt handler routine following an 8-bit OSC1 timer interrupt.
- To prevent generating unnecessary interrupts, reset T8OIF (D0/T8OSC1_IFLG register) before permitting compare match interrupts using T8OIE (D0/T8OSC1_IMSK register).
- The correct counter value may not be read out (reading is unstable) if the counter data register is read while counting is underway.
  To obtain the counter value, read the counter data register while the counter is halted or read the counter data register twice in succession. Treat the value as valid if the values read are identical.
- When using the PWM output, set the following: PWM duty data ≥ 0, compare data ≥ 1. The minimum setting value is 0 for PWM duty data and 1 for compare data. The timer output cycle is 1/2 of the input clock.
- When the PWM duty data is set greater than the compare data, only the compare match signal is generated. No duty match signal is generated. In that case, the TOUT4 output is fixed to Low.

15 Clock Timer (CT)

15.1 Clock Timer Overview

The S1C17003 incorporates a single-channel clock timer that uses the OSC1 clock as its oscillation source.
The clock timer consists of an 8-bit binary counter that uses the 256 Hz signal divided from the OSC1 clock as the input clock and allows data for each bit (128 Hz to 1 Hz) to be read out by software.
The clock timer can also generate interrupts using the 32 Hz, 8 Hz, 2 Hz, and 1 Hz signals.
This clock timer is normally used for various timing functions, such as clocks.
Figure 15.1.1 illustrates the clock timer configuration.

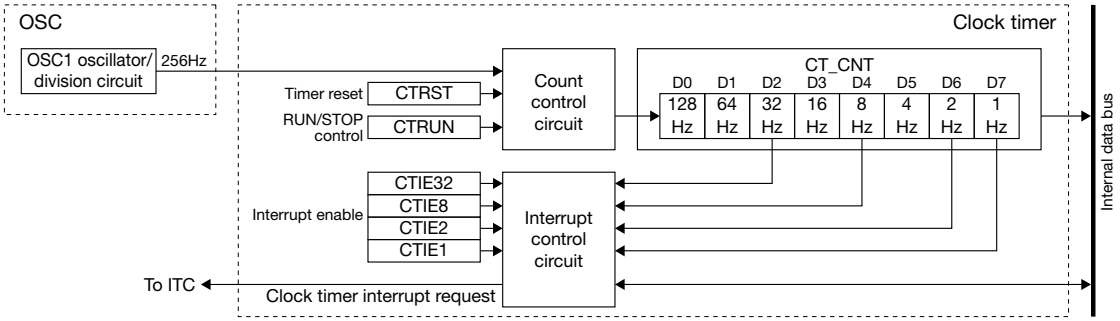


Figure 15.1.1: Clock timer configuration

15.2 Operation Clock

The clock timer uses the 256 Hz clock output by the OSC module as the operation clock.
The OSC module generates this operation clock by dividing the OSC1 clock into 1/128, resulting in a frequency of 256 Hz when the OSC1 clock frequency is 32.768 kHz. The frequency described in this section will vary accordingly for other OSC1 clock frequencies.
The OSC module does not include a 256 Hz clock output control bit. The 256 Hz clock is normally fed to the clock timer when the OSC1 oscillation is on.
For detailed information on OSC1 oscillator circuit control, refer to "7 Oscillator Circuit (OSC)."

15.3 Clock Timer Resetting

Reset the clock timer by writing 1 to the CTRST bit (D4/CT_CTL register). This clears the counter to 0.

* **CTRST**: Clock Timer Reset Bit in the Clock Timer Control (CT_CTL) Register (D4/0x5000)

Apart from this operation, the counter is also cleared by initial resetting.

15.4 Clock Timer RUN/STOP Control

Set the following items before starting the clock timer.
(1) If using interrupts, set the interrupt level and permit interrupts for the clock timer. See Section 15.5.
(2) Reset the timer. See Section 15.3.

The clock timer includes CTRUN (D0/CT_CTL register) to control Run/Stop.

* **CTRUN**: Clock Timer Run/Stop Control Bit in the Clock Timer Control (CT_CTL) Register (D0/0x5000)

The clock timer starts operating when CTRUN is written as 1. Writing 0 to CTRUN prevents clock input and stops the operation.
This control does not affect the counter (CT_CNT register) data. The counter data is retained even when the count is halted, allowing resumption of the count from that data.
If CTRUN and CTRST are written as 1 simultaneously, the clock timer starts counting after the reset.

Interrupt factors are generated during counting at the corresponding 32 Hz, 8 Hz, 2 Hz, and 1 Hz signal falling edges. If interrupts are permitted, interrupt requests are sent to the interrupt controller (ITC).

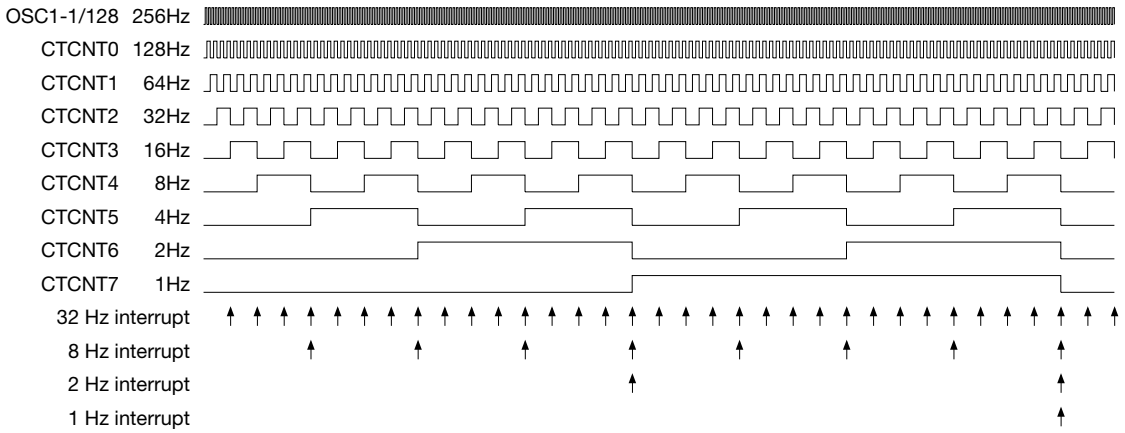


Figure 15.4.1: Clock timer timing chart

Note: The clock timer switches to Run/Stop mode when data is written to CTRUN synchronized with the 256 Hz signal falling edge. When 0 is written to CTRUN, the timer switches to Stop state after counting an additional "+1." 1 is retained for CTRUN reading until the timer actually stops.
Figure 15.4.2 shows the Run/Stop control timing chart.

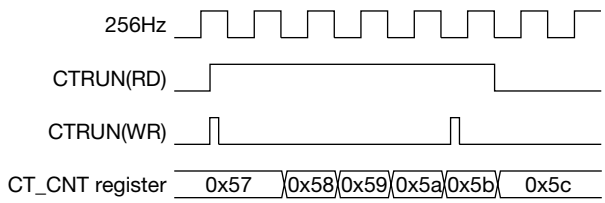


Figure 15.4.2: Run/Stop control timing chart

15.5 Clock Timer Interrupts

The CT module includes functions for generating the following four kinds of interrupts:
32 Hz, 8 Hz, 2 Hz, 1 Hz interrupts

The CT module outputs a single interrupt signal shared by the above four interrupt factors to the interrupt controller (ITC). The interrupt flag within the CT module should be read to identify the interrupt factor that occurred.

32 Hz, 8 Hz, 2 Hz, 1 Hz interrupts

Generated at the 32 Hz, 8 Hz, 2 Hz, and 1 Hz signal falling edges, these interrupt requests set the following interrupt flags in the CT module to 1.

- **CTIF32**: 32 Hz Interrupt Flag in the Clock Timer Interrupt Flag (CT_IFLG) Register (D3/0x5003)
- **CTIF8**: 8 Hz Interrupt Flag in the Clock Timer Interrupt Flag (CT_IFLG) Register (D2/0x5003)
- **CTIF2**: 2 Hz Interrupt Flag in the Clock Timer Interrupt Flag (CT_IFLG) Register (D1/0x5003)
- **CTIF1**: 1 Hz Interrupt Flag in the Clock Timer Interrupt Flag (CT_IFLG) Register (D0/0x5003)

To use these interrupts, set the following interrupt enable bits to 1 for the corresponding interrupt flags. If the interrupt enable bits are set to 0 (default), the interrupt flag will not be set to 1, and the interrupt requests for this factor will not be sent to the ITC.

- **CTIE32**: 32 Hz Interrupt Enable Bit in the Clock Timer Interrupt Mask (CT_IMSK) Register (D3/0x5002)
- **CTIE8**: 8 Hz Interrupt Enable Bit in the Clock Timer Interrupt Mask (CT_IMSK) Register (D2/0x5002)
- **CTIE2**: 2 Hz Interrupt Enable Bit in the Clock Timer Interrupt Mask (CT_IMSK) Register (D1/0x5002)
- **CTIE1**: 1 Hz Interrupt Enable Bit in the Clock Timer Interrupt Mask (CT_IMSK) Register (D0/0x5002)

The CT module outputs an interrupt request to the ITC if the CTIF* is set to 1. This interrupt request signal sets the clock timer interrupt flag inside the ITC to 1 and generates an interrupt if the ITC and S1C17 core interrupt conditions are met.
Check the frequency of a clock timer interrupt by reading CTIF* as part of the clock timer interrupt processing routine.

Note: • To prevent interrupt recurrences, the CT module interrupt flag CTIF*must be reset within the interrupt processing routine following a clock timer interrupt.

• To prevent generating unnecessary interrupts, reset the corresponding CTIF* before permitting clock timer interrupts from CTIE*.

Interrupt vectors

The clock timer interrupt vector numbers and vector addresses are listed below.

Vector number: 7 (0x07)
Vector address: TTBR + 0x1c

Other interrupt settings

The ITC allows the priority of clock timer interrupts to be set between level 0 (the default value) and level 7. To generate actual interrupts, the PSR (S1C17 core internal processor status register) IE (interrupt enable) bit must be set to 1.
For more information on interrupt processing, see “6 Interrupt Controller (ITC).”

15.6 Control Register Details

Table 15.6.1: Clock timer registers list

Address	Register name		Function
0x5000	CT_CTL	Clock Timer Control Register	Timer resetting and Run/Stop control
0x5001	CT_CNT	Clock Timer Counter Register	Counter data
0x5002	CT_IMSK	Clock Timer Interrupt Mask Register	Interrupt mask setting
0x5003	CT_IFLG	Clock Timer Interrupt Flag Register	Interrupt occurrence status display/resetting

The clock timer registers are described in detail below. These are 8-bit registers.

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

0x5000: Clock Timer Control Register (CT_CTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Clock Timer Control Register (CT_CTL)	**0x5000** (8 bits)	D7–5	**–**	reserved	–	–	–	0 when being read.
		D4	**CTRST**	Clock timer reset	1 Reset 0 Ignored	0	W	
		D3–1	**–**	reserved	–	–	–	
		D0	**CTRUN**	Clock timer run/stop control	1 Run 0 Stop	0	R/W	

D[7:5] Reserved

D4 CTRST: Clock Timer Reset Bit

Resets the clock timer.

1 (W): Reset
0 (W): Disabled
0 (R): Normally 0 when read out (default)

Writing 1 to this bit resets the counter to 0x0. When reset in Run state, the clock timer restarts immediately after resetting. The reset data 0x0 is retained when in Stop state.

D[3:1] Reserved

D0 CTRUN: Clock Timer Run/Stop Control Bit

Controls the clock timer Run/Stop.

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

The clock timer starts counting when CTRUN is written as 1 and stops when written as 0. The counter data is retained at Stop state until a reset or the next Run state.

0x5001: Clock Timer Counter Register (CT_CNT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Clock Timer Counter Register (CT_CNT)	**0x5001** (8 bits)	D7–0	**CTCNT[7:0]**	Clock timer counter value	0x0 to 0xff	0	R	

D[7:0] CTCNT[7:0]: Clock Timer Counter Value

Reads out the counter data. (Default: 0xff)
This register is read-only and cannot be written to.
The bits correspond to various frequencies, as follows:
D7: 1Hz
D6: 2Hz
D5: 4Hz
D4: 8Hz
D3: 16Hz
D2: 32Hz
D1: 64Hz
D0: 128Hz

Note: The correct counter value may not be read out (reading is unstable) if the register is read while counting is underway.
Obtain the counter value by one of the following methods:
- **Read the counter value while the counter is halted.**
- **Read the counter twice in succession. Treat the value as valid if the values read are identical.**

0x5002: Clock Timer Interrupt Mask Register (CT_IMSK)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Clock Timer Interrupt Mask Register (CT_IMSK)	**0x5002** (8 bits)	D7–4	**–**	reserved	–				–	–	0 when being read.
		D3	**CTIE32**	32 Hz interrupt enable	1	Enable	0	Disable	0	R/W	
		D2	**CTIE8**	8 Hz interrupt enable	1	Enable	0	Disable	0	R/W	
		D1	**CTIE2**	2 Hz interrupt enable	1	Enable	0	Disable	0	R/W	
		D0	**CTIE1**	1 Hz interrupt enable	1	Enable	0	Disable	0	R/W	

This register permits or prohibits interrupt requests individually for the clock timer 32 Hz, 8 Hz, 2 Hz, and 1 Hz signals. Setting the CTIE*bit to 1 permits clock timer interrupts for the corresponding frequency signal falling edge, while setting to 0 prohibits interrupts.
To enable interrupt generation, the ITC clock timer interrupt enable bits must also be set to permit interrupts.

D[7:4] Reserved

D3 CTIE32: 32 Hz Interrupt Enable Bit
Permits or prohibits 32 Hz signal interrupts.
1 (R/W): Interrupt permitted
0 (R/W): Interrupt prohibited (default)

D2 CTIE8: 8 Hz Interrupt Enable Bit
Permits or prohibits 8 Hz signal interrupts.
1 (R/W): Interrupt permitted
0 (R/W): Interrupt prohibited (default)

D1 CTIE2: 2 Hz Interrupt Enable Bit
Permits or prohibits 2 Hz signal interrupts.
1 (R/W): Interrupt permitted
0 (R/W): Interrupt prohibited (default)

D0 CTIE1: 1 Hz Interrupt Enable Bit
Permits or prohibits 1 Hz signal interrupts.
1 (R/W): Interrupt permitted
0 (R/W): Interrupt prohibited (default)

0x5003: Clock Timer Interrupt Flag Register (CT_IFLG)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Clock Timer Interrupt Flag Register (CT_IFLG)	0x5003 (8 bits)	D7–4	–	reserved	–				–	–	0 when being read.
		D3	CTIF32	32 Hz interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D2	CTIF8	8 Hz interrupt flag					0	R/W	
		D1	CTIF2	2 Hz interrupt flag					0	R/W	
		D0	CTIF1	1 Hz interrupt flag					0	R/W	

This register indicates the occurrence state of interrupt factors due to clock timer 32 Hz, 8 Hz, 2 Hz, and 1 Hz signals. If a clock timer interrupt occurs, identify the interrupt factor (frequency) by reading the interrupt flag in this register. CTIF* are CT module interrupt flags corresponding to the individual 32 Hz, 8 Hz, 2 Hz, and 1 Hz interrupts. It is set to 1 at the falling edge of each signal if CTIE* (CT_IMSK register) is set to 1. The clock timer interrupt request signal is output to the ITC at the same time. This interrupt request signal generates an interrupt if the ITC and S1C17 core interrupt conditions are met.
CTIF* is reset by writing as 1.

Note:
- **To prevent interrupt recurrences, the CT module interrupt flag CTIF*must be reset within the interrupt processing routine following a clock timer interrupt.**
- **To prevent generating unnecessary interrupts, CTIF* must be reset before permitting clock timer interrupts using CTIE.***

D[7:4] Reserved

D3 CTIF32: 32 Hz Interrupt Flag

Interrupt flag indicating the 32 Hz interrupt factor occurrence status.
1(R): Interrupt factor present
0(R): No interrupt factor (default)
1(W): Reset flag
0(W): Disabled

Setting CTIE32 (D3/CT_IMSK register) to 1 sets CTIF32 to 1 at the 32 Hz signal falling edge.

D2 CTIF8: 8 Hz Interrupt Flag

Interrupt flag indicating the 8 Hz interrupt factor occurrence status.
1(R): Interrupt factor present
0(R): No interrupt factor (default)
1(W): Reset flag
0(W): Disabled

Setting CTIE8 (D2/CT_IMSK register) to 1 sets CTIF8 to 1 at the 8 Hz signal falling edge.

D1 CTIF2: 2 Hz Interrupt Flag

Interrupt flag indicating the 2 Hz interrupt factor occurrence status.
1(R): Interrupt factor present
0(R): No interrupt factor (default)
1(W): Reset flag
0(W): Disabled

Setting CTIE2 (D1/CT_IMSK register) to 1 sets CTIF2 to 1 at the 2 Hz signal falling edge.

D0 CTIF1: 1 Hz Interrupt Flag

Interrupt flag indicating the 1 Hz interrupt factor occurrence status.
1(R): Interrupt factor present
0(R): No interrupt factor (default)
1(W): Reset flag
0(W): Disabled

Setting CTIE1 (D0/CT_IMSK register) to 1 sets CTIF1 to 1 at the 1 Hz signal falling edge.

15.7 Precautions

- The OSC1 oscillator circuit must be set to On before operating the clock timer.
- To prevent generating unnecessary interrupts, reset the CT_IFLG register interrupt flag before permitting clock timer interrupts by the CT_IMSK register.
- The clock timer switches to Run/Stop mode when data is written to CTRUN synchronized with the 256 Hz signal falling edge. When 0 is written to CTRUN (D0/CT_CTL register), the timer switches to Stop state after counting an additional "+1." 1 is retained for CTRUN reading until the timer actually stops.
  Figure 15.7.1 shows the Run/Stop control timing chart.

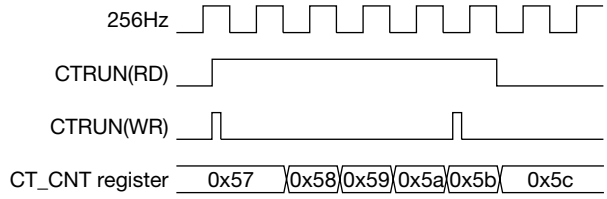


Figure 15.7.1: Run/Stop control timing chart

- Executing the slp instruction will destabilize a running clock timer (CTRUN = 1) during recovery from SLEEP state. When switching to SLEEP state, set the clock timer to STOP (CTRUN = 0) before executing the slp instruction.
- The correct counter value may not be read out (reading is unstable) if the counter register is read while counting is underway.
  Read the counter register while the counter is halted or read the counter register twice in succession. Treat the value as valid if the values read are identical.

This page intentionally left blank.

16 Stopwatch Timer (SWT)

16.1 Stopwatch Timer Overview

The S1C17003 incorporates a 1/100-second and 1/10-second stopwatch timer. The stopwatch timer consists of a 4-bit 2-stage BCD counter (1/100 and 1/10 second) that uses the 256 Hz signal divided from the OSC1 clock as the input clock and allows count data to be read out by software.
The stopwatch timer can also generate interrupts using the 100 Hz (approximately 100 Hz), 10 Hz (approximately 10 Hz), and 1 Hz signals.
Figure 16.1.1 illustrates the stopwatch timer configuration.

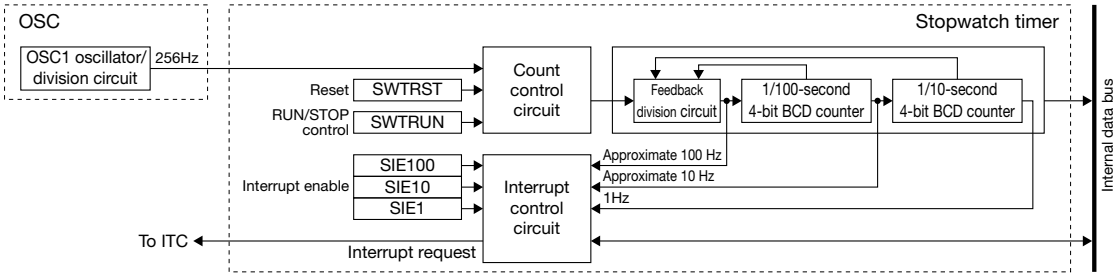


Figure 16.1.1: Stopwatch timer configuration

16.2 BCD Counters

The stopwatch counter consists of 1/100-second and 1/10-second 4-bit BCD counters.
The count value can be read from the SWT_BCNT register.

1/100-second counter

* **BCD100[3:0]**: 1/100 Sec. BCD Counter Value in the Stopwatch Timer BCD Counter (SWT_BCNT) Register (D[3:0]/0x5021)

1/10-second counter

* **BCD10[3:0]**: 1/10 Sec. BCD Counter Value in the Stopwatch Timer BCD Counter (SWT_BCNT) Register (D[7:4]/0x5021)

Count-up Pattern

A feedback division circuit is used to generate 100 Hz, 10 Hz, and 1 Hz signals from the 256 Hz clock. The counter count-up pattern varies as shown in Figure 16.2.1.

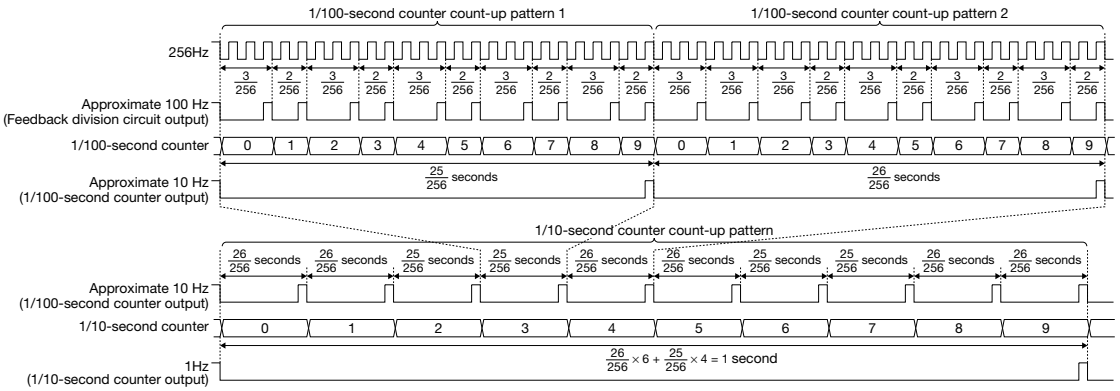


Figure 16.2.1: Stopwatch timer count-up patterns

The feedback division circuit generates an approximate 100 Hz signal at 2/256-second and 3/256-second intervals from the 256 Hz signal fed from the OSC module.

The 1/100-second counter counts the approximate 100 Hz signal output by the feedback division circuit and generates an approximate 10 Hz signal at 25/256-second and 26/256-second intervals.
Count-up will be pseudo 1/100-second counting at 2/256-second and 3/256-second intervals.

The 1/10-second counter counts the approximate 10 Hz signal generated by the 1/100-second counter at a ratio of 4:6, and generates a 1 Hz signal.
Count-up will be pseudo 1/10-second counting at 25/256-second and 26/256-second intervals.

16.3 Operation Clock

The stopwatch timer uses the 256 Hz clock output by the OSC module as the operation clock.
The OSC module generates this operation clock by dividing the OSC1 clock into 1/128, resulting in a frequency of 256 Hz when the OSC1 clock frequency is 32.768 kHz. The frequency described in this section will vary accordingly for other OSC1 clock frequencies.
The OSC module does not include a 256 Hz clock output control bit. The 256 Hz clock is normally fed to the stopwatch timer when the OSC1 oscillation is on.
For detailed information on OSC1 oscillator circuit control, refer to "7 Oscillator Circuit (OSC)."

16.4 Stopwatch Timer Resetting

Reset the stopwatch timer by writing 1 to the SWTRST bit (D4/SWT_CTL register). This clears the counter to 0.

* **SWTRST**: Stopwatch Timer Reset Bit in the Stopwatch Timer Control (SWT_CTL) Register (D4/0x5020)

Apart from this operation, the counter is also cleared by initial resetting.

16.5 Stopwatch Timer RUN/STOP Control

Set the following items before starting the stopwatch timer.
(1) If using interrupts, set the interrupt level and permit interrupts for the stopwatch timer. See Section 16.6.
(2) Reset the timer. See Section 16.4.

The stopwatch timer includes SWTRUN (D0/SWT_CTL register) to control Run/Stop.

∗ **SWTRUN**: Stopwatch Timer Run/Stop Control Bit in the Stopwatch Timer Control (SWT_CTL) Register (D0/0x5020)

The stopwatch timer starts counting when SWTRUN is written as 1. Writing 0 to SWTRUN prevents clock input and stops the count.
This control does not affect the counter (SWT_BCNT register) data. The counter data is retained even when the count is halted, allowing resumption of the count from that data.
If SWTRUN and SWTRST are written as 1 simultaneously, the stopwatch timer starts counting after the reset.

Interrupt factors are generated during counting at the corresponding 100 Hz (approximate 100 Hz), 10 Hz (approximate 10 Hz), and 1 Hz signal falling edges. If interrupts are permitted, interrupt requests are sent to the interrupt controller (ITC).

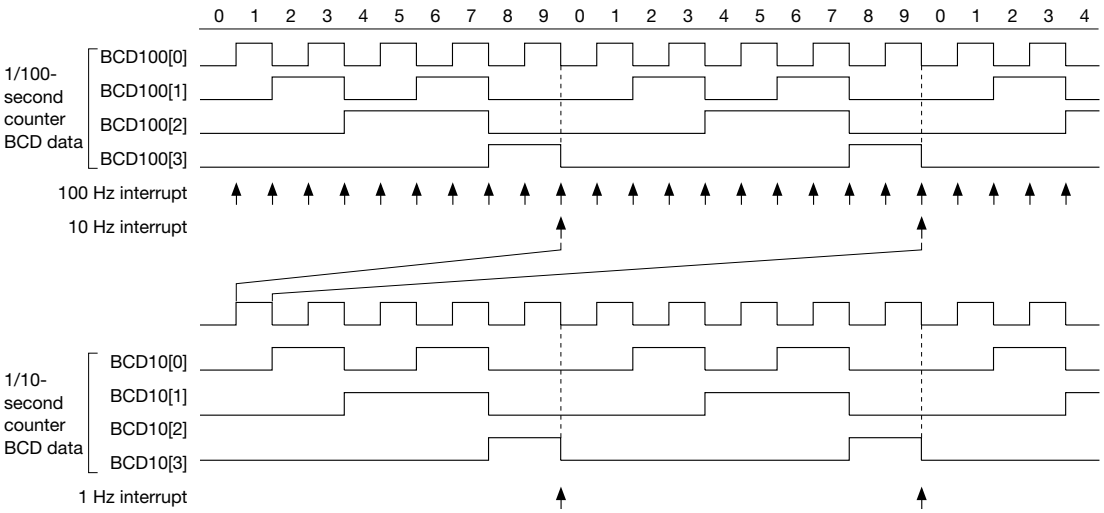


Figure 16.5.1: Stopwatch timer timing chart

Note: The stopwatch timer switches to Run/Stop mode when data is written to SWTRUN synchronized with the 256 Hz signal falling edge. When 0 is written to SWTRUN, the timer switches to Stop state after counting an additional "+1." 1 is retained for SWTRUN reading until the timer actually stops.
Figure 16.5.2 shows the Run/Stop control timing chart.

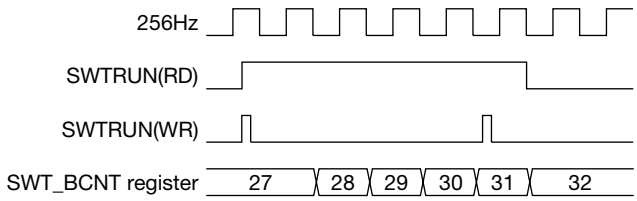


Figure 16.5.2: Run/Stop control timing chart

16.6 Stopwatch Timer Interrupts

The SWT module includes functions for generating the following three kinds of interrupts:
- 100 Hz interrupt
- 10 Hz interrupt
- 1 Hz interrupt

The SWT module outputs a single interrupt signal shared by the above three interrupt factors to the interrupt controller (ITC). The interrupt flag within the SWT module should be read to identify the interrupt factor that occurred.

100 Hz, 10 Hz, 1 Hz interrupts

Generated at the 100 Hz (approximate 100 Hz), 10 Hz (approximate 10 Hz), and 1 Hz signal falling edges, these interrupt requests set the following interrupt flags in the SWT module to 1.

* **SIF1**: 1 Hz Interrupt Flag in the Stopwatch Timer Interrupt Flag (SWT_IFLG) Register (D2/0x5023)
* **SIF10**: 10 Hz Interrupt Flag in the Stopwatch Timer Interrupt Flag (SWT_IFLG) Register (D1/0x5023)
* **SIF100**: 100 Hz Interrupt Flag in the Stopwatch Timer Interrupt Flag (SWT_IFLG) Register (D0/0x5023)

To use these interrupts, set the following interrupt enable bits to 1 for the corresponding interrupt flags. If the interrupt enable bits are set to 0 (default), the interrupt flag will not be set to 1, and the interrupt requests for this factor will not be sent to the ITC.

* **SIE1**: 1 Hz Interrupt Enable Bit in the Stopwatch Timer Interrupt Mask (SWT_IMSK) Register (D2/0x5022)
* **SIE10**: 10 Hz Interrupt Enable Bit in the Stopwatch Timer Interrupt Mask (SWT_IMSK) Register (D1/0x5022)
* **SIE100**: 100 Hz Interrupt Enable Bit in the Stopwatch Timer Interrupt Mask (SWT_IMSK) Register (D0/0x5022)

The SWT module outputs an interrupt request to the ITC if the SIF* is set to 1. This interrupt request signal generates an interrupt if the ITC and S1C17 core interrupt conditions are met.
Check the frequency of a stopwatch timer interrupt by reading SIF* as part of the stopwatch timer interrupt processing routine.

Note:
- **To prevent interrupt recurrences, the SWT module interrupt flag SIF*must be reset within the interrupt processing routine following a stopwatch timer interrupt.**
- **To prevent generating unnecessary interrupts, reset the corresponding SIF* before permitting stopwatch timer interrupt from SIE*.**

Interrupt vectors

The stopwatch timer interrupt vector numbers and vector addresses are listed below.

Vector number: 6 (0x06)
Vector address: TTBR + 0x18

Other interrupt settings

The ITC allows the priority of stopwatch timer interrupts to be set between level 0 (the default value) and level 7. To generate actual interrupts, the PSR (S1C17 core internal processor status register) IE (interrupt enable) bit must be set to 1.
For more information on interrupt processing, see “6 Interrupt Controller (ITC).”

16.7 Control Register Details

Table 16.7.1 Stopwatch timer register list

Address	Register name		Function
0x5020	SWT_CTL	Stopwatch Timer Control Register	Timer resetting and Run/Stop control
0x5021	SWT_BCNT	Stopwatch Timer BCD Counter Register	BCD counter data
0x5022	SWT_IMSK	Stopwatch Timer Interrupt Mask Register	Interrupt mask setting
0x5023	SWT_IFLG	Stopwatch Timer Interrupt Flag Register	Interrupt occurrence status display/resetting

The stopwatch timer registers are described in detail below. These are 8-bit registers.

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

0x5020: Stopwatch Timer Control Register (SWT_CTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Stopwatch Timer Control Register (SWT_CTL)	**0x5020** (8 bits)	D7–5	**–**	reserved	–	–	–	0 when being read.
		D4	**SWTRST**	Stopwatch timer reset	1 Reset 0 Ignored	0	W	
		D3–1	**–**	reserved	–	–	–	
		D0	**SWTRUN**	Stopwatch timer run/stop control	1 Run 0 Stop	0	R/W	

D[7:5] Reserved

D4 SWTRST: Stopwatch Timer Reset Bit

Resets the stopwatch timer.

1 (W): Reset

0 (W): Disabled

0 (R): Normally 0 when read out (default)

Writing 1 to this bit resets the counter to 0x0. When reset in Run state, the stopwatch timer restarts immediately after resetting. The reset data 0x0 is retained when in Stop state.

D[3:1] Reserved

D0 SWTRUN: Stopwatch Timer Run/Stop Control Bit

Controls the stopwatch timer Run/Stop.

1 (R/W): Run

0 (R/W): Stop (default)

The stopwatch timer starts counting when SWTRUN is written as 1 and stops when written as 0. The counter data is retained at Stop state until a reset or the next Run state.

0x5021: Stopwatch Timer BCD Counter Register (SWT_BCNT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Stopwatch Timer BCD Counter Register (SWT_BCNT)	0x5021 (8 bits)	D7–4	BCD10[3:0]	1/10 sec. BCD counter value	0 to 9	0	R	
		D3–0	BCD100[3:0]	1/100 sec. BCD counter value	0 to 9	0	R	

D[7:4] BCD10[3:0]: 1/10 Sec. BCD Counter Value

Read the 1/10-second counter BCD data. (Default: 0)
This register is read-only and cannot be written to.

D[3:0] BCD100[3:0]: 1/100 Sec. BCD Counter Value

Read the 1/100-second counter BCD data. (Default: 0)
This register is read-only and cannot be written to.

Note: The correct counter value may not be read out (reading is unstable) if the register is read while counting is underway.
Obtain the counter value by one of the following methods:
- Read the counter value while the counter is halted.
- Read the counter twice in succession. Treat the value as valid if the values read are identical.

0x5022: Stopwatch Timer Interrupt Mask Register (SWT_IMSK)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Stopwatch Timer Interrupt Mask Register (SWT_IMSK)	**0x5022** (8 bits)	D7–3	**–**	reserved	–				–	–	0 when being read.
		D2	**SIE1**	1 Hz interrupt enable	1	Enable	0	Disable	0	R/W	
		D1	**SIE10**	10 Hz interrupt enable	1	Enable	0	Disable	0	R/W	
		D0	**SIE100**	100 Hz interrupt enable	1	Enable	0	Disable	0	R/W	

This register permits or prohibits interrupt requests individually for the stopwatch timer 100 Hz, 10 Hz, and 1 Hz signals. Setting the SIE*bit to 1 permits stopwatch timer interrupts for the corresponding frequency signal falling edge, while setting to 0 prohibits interrupts.
To enable interrupt generation, the ITC stopwatch timer interrupt enable bits must also be set to permit interrupts.

D[7:3] Reserved

D2 SIE1: 1 Hz Interrupt Enable Bit

Permits or prohibits 1 Hz signal interrupts.
1 (R/W): Interrupt permitted
0 (R/W): Interrupt prohibited (default)

D1 SIE10: 10 Hz Interrupt Enable Bit

Permits or prohibits 10 Hz signal interrupts.
1 (R/W): Interrupt permitted
0 (R/W): Interrupt prohibited (default)

D0 SIE100: 100 Hz Interrupt Enable Bit

Permits or prohibits 100 Hz signal interrupts.
1 (R/W): Interrupt permitted
0 (R/W): Interrupt prohibited (default)

0x5023: Stopwatch Timer Interrupt Flag Register (SWT_IFLG)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Stopwatch Timer Interrupt Flag Register (SWT_IFLG)	0x5023 (8 bits)	D7–3	–	reserved	–				–	–	0 when being read.
		D2	SIF1	1 Hz interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D1	SIF10	10 Hz interrupt flag					0	R/W	
		D0	SIF100	100 Hz interrupt flag					0	R/W	

This register indicates the occurrence state of interrupt factors due to stopwatch timer 100 Hz, 10 Hz, and 1 Hz signals. If a stopwatch timer interrupt occurs, identify the interrupt factor (frequency) by reading the interrupt flag in this register.
SIF* are SWT module interrupt flags corresponding to the individual 100 Hz, 10 Hz, and 1 Hz interrupts. It is set to 1 at the falling edge of each signal if SIE* (SWT_IMSK register) is set to 1. The stopwatch timer interrupt request signal is output to the ITC at the same time. This interrupt request signal generates an interrupt if the ITC and S1C17 core interrupt conditions are met.
SIF* is reset by writing as 1.

Note:
- **To prevent interrupt recurrences, the SWT module interrupt flag SIF*must be reset within the interrupt processing routine following a stopwatch timer interrupt.**
- **To prevent generating unnecessary interrupts, SIF* must be reset before permitting clock timer interrupts using SIE.***

D[7:3] Reserved

D2 SIF1: 1 Hz Interrupt Flag

Interrupt flag indicating the 1 Hz interrupt factor occurrence status.
1(R): Interrupt factor present
0(R): No interrupt factor (default)
1(W): Reset flag
0(W): Disabled

Setting SIE1 (D2/SWT_IMSK register) to 1 sets SIF1 to 1 at the 1 Hz signal falling edge.

D1 SIF10: 10 Hz Interrupt Flag

Interrupt flag indicating the 10 Hz interrupt factor occurrence status.
1(R): Interrupt factor present
0(R): No interrupt factor (default)
1(W): Reset flag
0(W): Disabled

Setting SIE10 (D1/SWT_IMSK register) to 1 sets SIF10 to 1 at the 10 Hz signal falling edge.

D0 SIF100: 100 Hz Interrupt Flag

Interrupt flag indicating the 100 Hz interrupt factor occurrence status.
1(R): Interrupt factor present
0(R): No interrupt factor (default)
1(W): Reset flag
0(W): Disabled

Setting SIE100 (D0/SWT_IMSK register) to 1 sets SIF100 to 1 at the 100 Hz signal falling edge.

16.8 Precautions

- The OSC1 oscillator circuit must be set to On before operating the stopwatch timer.
- To prevent interrupt recurrences, the SWT_IFLG register interrupt flag must be reset within the interrupt processing routine following a stopwatch timer interrupt.
- To prevent generating unnecessary interrupts, reset the SWT_IFLG register interrupt flag before permitting stopwatch timer interrupts by the SWT_IMSK register.
- The stopwatch timer switches to Run/Stop mode when data is written to SWTRUN (D0/SWT_CTL resister) synchronized with the 256 Hz signal falling edge. When 0 is written to SWTRUN, the timer switches to Stop state after counting an additional "+1." 1 is retained for SWTRUN reading until the timer actually stops.
  Figure 16.8.1 shows the Run/Stop control timing chart.

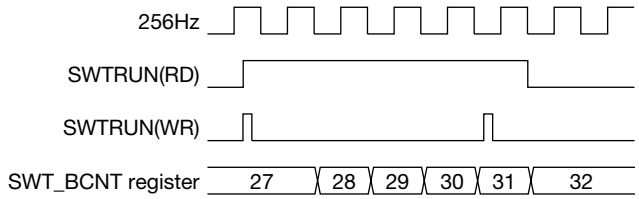


Figure 16.8.1: Run/Stop control timing chart

- Executing the slp instruction will destabilize a running stopwatch timer (SWTRUN = 1) during recovery from SLEEP state. When switching to SLEEP state, set the stopwatch timer to STOP (SWTRUN = 0) before executing the slp instruction.
- The correct counter value may not be read out (reading is unstable) if the counter register is read while counting is underway.
  To obtain the counter value, read the counter register while the counter is halted or read the counter register twice in succession. Treat the value as valid if the values read are identical.

17 Watchdog Timer (WDT)

17.1 Watchdog Timer Overview

The S1C17003 incorporates a watchdog timer that uses the OSC1 oscillator circuit as its oscillation source. The watchdog timer generates an NMI or reset (selectable via software) to the CPU if not reset within 131,072/fosc1 seconds (4 seconds when fosc1 = 32.768 kHz).
Reset the watchdog timer via software within this cycle to prevent NMI/resets, which in turn enables runaway detection for programs that do not pass through the processing routine.
Figure 17.1.1 illustrates the watchdog timer block diagram.

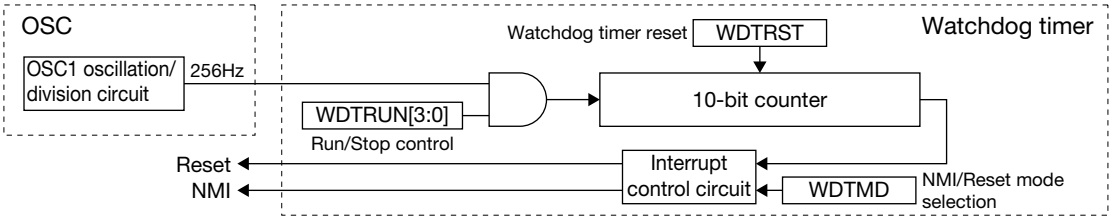


Figure 17.1.1: Watchdog timer block diagram

17.2 Operation Clock

The watchdog timer uses the 256 Hz clock output by the OSC module as the operation clock.
The OSC module generates this operation clock by dividing the OSC1 clock into 1/128, resulting in a frequency of 256 Hz when the OSC1 clock frequency is 32.768 kHz. The frequency described in this section will vary accordingly for other OSC1 clock frequencies.
The OSC module does not include a 256 Hz clock output control bit. The 256 Hz clock is normally fed to the watchdog timer when the OSC1 oscillation is on.
For detailed information on OSC1 oscillator circuit control, refer to "7 Oscillator Circuit (OSC)."

17.3 Watchdog Timer Control

17.3.1 NMI/Reset Mode Selection

WDTMD (D1/WDT_ST register) is used to select whether an NMI signal or a reset signal is output when the watchdog timer has not been reset within the NMI/Reset occurrence cycle.

* **WDTMD**: NMI/Reset Mode Select Bit in the Watchdog Timer Status (WDT_ST) Register (D1/0x5041)

To generate an NMI, set WDTMD to 0 (default). Set to 1 to generate a reset.

17.3.2 Watchdog Timer Run/Stop Control

The watchdog timer starts counting when a value other than 0b1010 is written to WDTRUN[3:0] (D[3:0]/WDT_CTL register) and stops when 0b1010 is written.

* **WDTRUN[3:0]**: Watchdog Timer Run/Stop Control Bits in the Watchdog Timer Control (WDT_CTL) Register (D[3:0]/0x5040)

Initial resetting sets WDTRUN[3:0] to 0b1010 and stops the watchdog timer.
Since an NMI or Reset may be generated immediately after running depending on the counter value, the watchdog timer should also be reset concurrently (before running the watchdog timer), as explained in the following section.

17.3.3 Watchdog Timer Resetting

To reset the watchdog timer, write 1 to WDTRST (D4/WDT_CTL register).

* **WDTRST**: Watchdog Timer Reset Bit in the Watchdog Timer Control (WDT_CTL) Register (D4/0x5040)

A location should be provided for periodically processing the routine for resetting the watchdog timer before an NMI or Reset is generated when using the watchdog timer. Process this routine within 131,072/f_{OSC1} second (4 seconds when f_{OSC1} = 32.768 kHz) cycle.
After resetting, the watchdog timer starts counting with a new NMI/Reset generation cycle.

If the watchdog timer is not reset within the NMI/Reset generation cycle for any reason, the CPU is switched to interrupt processing by NMI or resetting, an interrupt vector is read out, and an interrupt processing routine is executed.
The reset and NMI vector addresses are TTBR + 0x0 and TTBR + 0x08.

If the counter overflows and generates an NMI without the watchdog timer being reset, WDTST (D0/WDT_ST register) is set to 1.

* **WDTST**: NMI Status Bit in the Watchdog Timer Status (WDT_ST) Register (D0/0x5041)

This bit is provided to confirm that the watchdog timer was the source of the NMI.
The WDTST set to 1 is cleared to 0 by resetting the watchdog timer.

17.3.4 Operation in Standby Mode

HALT mode

The watchdog timer operates in HALT mode, as the clock is fed. HALT mode is therefore canceled by an NMI or Reset if it continues for more than the NMI/Reset cycle. To disable the watchdog timer while in HALT mode, stop the watchdog timer by writing 0b1010 to WDTRUN[3:0] before executing the halt instruction. Reset the watchdog timer before resuming operations after HALT mode is canceled.

SLEEP mode

The clock fed from the OSC module is stopped in SLEEP mode, which also stops the watchdog timer. To prevent generation of an unnecessary NMI or Reset after canceling SLEEP mode, reset the watchdog timer before executing the slp instruction. The watchdog should also be stopped as required using WDTRUN[3:0]. WDTRUN[3:0].

17.4 Control Register Details

Table 17.4.1 Watchdog timer register list

Address	Register name		Function
0x5040	WDT_CTL	Watchdog Timer Control Register	Timer reset and Run/Stop control
0x5041	WDT_ST	Watchdog Timer Status Register	Timer mode setting and NMI status display

The watchdog timer registers are described in detail below. These are 8-bit registers.

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

0x5040: Watchdog Timer Control Register (WDT_CTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Watchdog Timer Control Register (WDT_CTL)	0x5040 (8 bits)	D7–5	–	reserved	–	–	–	0 when being read.
		D4	WDTRST	Watchdog timer reset	1 Reset 0 Ignored	0	W	
		D3–0	WDTRUN[3:0]	Watchdog timer run/stop control	Other than 1010 Run 1010 Stop	1010	R/W	

D[7:5] Reserved

D4 WDTRST: Watchdog Timer Reset Bit

Resets the watchdog timer.

1 (W): Reset
0 (W): Disabled
0 (R): Normally 0 when read out (default)

To use the watchdog timer, it must be reset by writing 1 to this bit within the NMI/Reset generation cycle (4 seconds when $fosc1$ = 32.768 kHz).
This resets the up-counter to 0 and starts counting with a new NMI/Reset generation cycle.

D[3:0] WDTRUN[3:0]: Watchdog Timer Run/Stop Control Bits

Controls the watchdog timer Run/Stop.
Values other than 0b1010 (R/W): Run
0b1010 (R/W): Stop (default)

The watchdog timer must also be reset to prevent generation of an unnecessary NMI or Reset while the watchdog timer operates.

0x5041: Watchdog Timer Status Register (WDT_ST)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Watchdog Timer Status Register (WDT_ST)	**0x5041** (8 bits)	D7–2	**–**	reserved	–				–	–	0 when being read.
		D1	**WDTMD**	NMI/Reset mode select	1	Reset	0	NMI	0	R/W	
		D0	**WDTST**	NMI status	1	NMI occurred	0	Not occurred	0	R	

D[7:2] Reserved

D1 WDTMD: NMI/Reset Mode Select Bit

Selects NMI or Reset generation on counter overflow.
1 (R/W): Reset
0 (R/W): NMI (default)

Setting this bit to 1 outputs a reset signal when the counter overflows. Setting to 0 outputs an NMI signal.

D0 WDTST: NMI Status Bit

Indicates a counter overflow and NMI occurrence.
1 (R): NMI occurred (counter overflow)
0 (R): NMI did not occur (default)

This bit confirms that the watchdog timer was the source of the NMI.
The WDTST set to 1 is cleared to 0 by resetting the watchdog timer.
This is also set by a counter overflow if reset output is selected, but is cleared by initial resetting and cannot be confirmed.

17.5 Precautions

- When the watchdog timer is running, this must be reset by software within a 131,072 f_{OSC1} seconds (4 seconds when f_{OSC1} = 32.768 kHz) cycle.
- The watchdog timer must also be reset to prevent generation of an unnecessary NMI or Reset while the watchdog timer operates.

18 UART

18.1 UART Configuration

The S1C17003 includes dual channel UART. The UART transfers data asynchronously with external serial devices at a rate of 150 to 460800bps. It includes 2-byte receive data buffer and one-byte transmit data buffer enabling full-duplex communication. For the transfer clock, either a clock internally generated by the timer module or an external clock input via the SCLKx can be used. Software should be used to select the data length (7 or 8 bits), stop bit length (1 or 2 bits) and parity mode (even, odd, or no parity). The start bit is fixed to 1 bit. Overrun errors, flaming errors and parity errors are detectable during data reception. The UART generates 3 types of interrupts, i.e., transmit buffer empty, receive buffer full, and receive error for each channel, enabling the interrupt handling to process serial data transfer efficiently.
This UART module also incorporates an RZI modulation/demodulation circuit that enables IrDA 1.0-compatible infrared communications simply by adding basic external circuits.
Figure 18.1.1 illustrates the UART configuration.

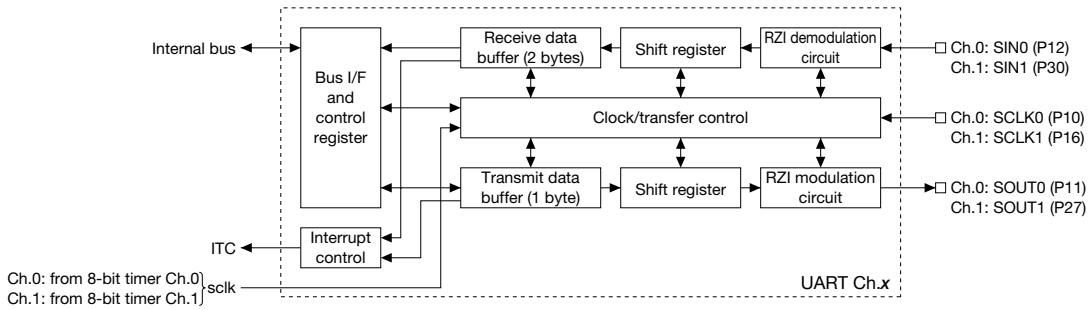


Figure 18.1.1: UART configuration

Note: The UART modules for the two channels have the same functions except for control register addresses. For this reason, the description in this section applies to all UART channels. The "x" in the register name indicates the channel number (0 or 1). Register addresses are indicated either as Ch.0" or "Ch.1".

E.g.: UART_CTLx register (0x4104/0x4124)
Ch.0:UART_CTL0 register (0x4104)
Ch.1:UART_CTL1 register (0x4124)

18.2 UART Pin

Table 18.2.1 lists the UART input/output pins.

Table 18.2.1: UART pin list

Pin name	I/O	Qty	Function
SIN0 (P12)	I	1	UART Ch.0 data input pin Inputs serial data sent from an external device.
SOUT0 (P11)	O	1	UART Ch.0 data output pin Outputs serial data sent to an external device.
SCLK0 (P10)	I	1	UART Ch.0 clock input pin Inputs the external clock when used for the transfer clock.
SIN1 (P30)	I	1	UART Ch.1 data input pin Inputs serial data sent from an external device.
SOUT1 (P27)	O	1	UART Ch.1 data output pin Outputs serial data sent to an external device.
SCLK1 (P16)	I	1	UART Ch.1 clock input pin Inputs the external clock when used for the transfer clock.

The UART input/output pins (SIN*x*, SOUT*x*, SCLK*x*) are shared with general purpose input/output port pins (P1[2:0], P30, P27, P16) and are initially set as general purpose input/output port pins. The function must be switched using the P3_PMUX, P2_PMUX, P1_PMUX register setting to use general purpose input/output port pins as UART input/output pins. Switch the pins to serial interface mode by setting the following control bits to 1.

UART Ch.0

P12 → SIN0

- **P12MUX**: P12 Port Function Select Bit in the P1 Port Function Select (P1_PMUX) Register (D4/0x52a2)

P11 → SOUT0

- **P11MUX**: P11 Port Function Select Bit in the P1 Port Function Select (P1_PMUX) Register (D2/0x52a2)

P10 → SCLK0 (only when using external clock)

- **P10MUX**: P10 Port Function Select Bit in the P1 Port Function Select (P1_PMUX) Register (D0/0x52a2)

UART Ch.1

P30 → SIN1

- **P30MUX**: P30 Port Function Select Bit in the P3 Port Function Select (P3_PMUX) Register (D1-0/0x52a6)

P27 → SOUT1

- **P27MUX**: P27 Port Function Select Bit in the P2 Port Function Select (P2_PMUX) Register (D7-6/0x52a5)

P16 → SCLK1 (only when using external clock)

- **P16MUX**: P16 Port Function Select Bit in the P1 Port Function Select (P1_PMUX) Register (D5-4/0x52a3)

For detailed information on pin function switching, refer to "10.2 Input/output Pin Function Selection (Port MUX)."

18.3 Transfer Clock

The UART transfer clock can be set to internal or external using SSCK (D0/UART_MOD*x* register).

* **SSCK**: Input Clock Select Bit in the UART Ch.*x* Mode (UART_MOD*x*) Register (D0/0x4103/0x4123)

Note: Make sure the UART is halted (when RXEN/UART_CTL*x* register = 0) before changing SSCK.

* **RXEN**: UART Enable Bit in the UART Control (UART_CTL*x*) Register (D0/0x4104/0x4124)

Internal clock

Setting SSCK to 0 (the default value) selects the internal clock. UART Ch.0 uses the 8-bit timer Ch.0 output clock as the transfer timer, while UART Ch.1 uses the 8-bit timer Ch.1 output clock. Thus, bit timers must be programmed to output a clock suited to the transfer rate.
For more information on 8-bit timer control, see "12 8-bit Timer (T8F)."

External clock

Setting SSCK to 1 selects the external clock. In this case, set P10 (Ch.0), P16 (Ch.1) to the SCLK0, SCLK1 pin (see Section 18.2) to input the external clock.

Note:
- **The UART generates a sampling clock that divides the 8-bit timer output into 1/16 divisions. Be careful when setting the transfer rate.**
- **To input the external clock via the SCLK pin, the clock frequency must be less than half of the PCLK and have a duty ratio of 50%.**

18.4 Transfer Data Settings

Set the following conditions to set the transfer data format.
- Data length: 7 or 8 bits
- Start bit: Fixed at 1 bit
- Stop bit: 1 or 2 bits
- Parity bit: Even, odd, no parity

Note: Make sure the UART is halted (when RXEN/UART_CTL*x* register = 0) before changing transfer data format settings.

∗ **RXEN**: UART Enable Bit in the UART Ch.*x* Control (UART_CTL*x*) Register (D0/0x4104/0x4124)

Data length

The data length is selected by CHLN (D4/UART_MOD*x* register). Setting CHLN to 0 (default) sets the data length to 7 bits. Setting CHLN to 1 sets the data length to 8 bits.

∗ **CHLN**: Character Length Select Bit in the UART Ch.*x* Mode (UART_MOD*x*) Register (D4/0x4103/0x4123)

Stop bit

The stop bit length is selected by STPB (D1/UART_MOD*x* register). Setting STPB to 0 (default) sets the stop bit length to 1 bit. Setting STPB to 1 sets the stop bit length to 2 bits.

∗ **STPB**: Stop Bit Select Bit in the UART Ch.*x* Mode (UART_MOD*x*) Register (D1/0x4103/0x4123)

Parity bit

Whether the parity function is enabled or disabled is selected by PREN (D3/UART_MOD*x* register). Setting PREN to 0 (default) disables the parity function. In this case, no parity bit is added to the transfer data and the data is not checked for parity when received. Setting PREN to 1 enables the parity function. In this case, a parity bit is added to the transfer data and the data is checked for parity when received.
When the parity function is enabled, the parity mode is selected by PMD (D2/UART_MOD*x* register). Setting PMD to 0 (default) adds a parity bit and checks for even parity. Setting PMD to 1 adds a parity bit and checks for odd parity.

∗ **PREN**: Parity Enable Bit in the UART Ch.*x* Mode (UART_MOD*x*) Register (D3/0x4103/0x4123)
∗ **PMD**: Parity Mode Select Bit in the UART Ch.*x* Mode (UART_MOD*x*) Register (D2/0x4103/0x4123)

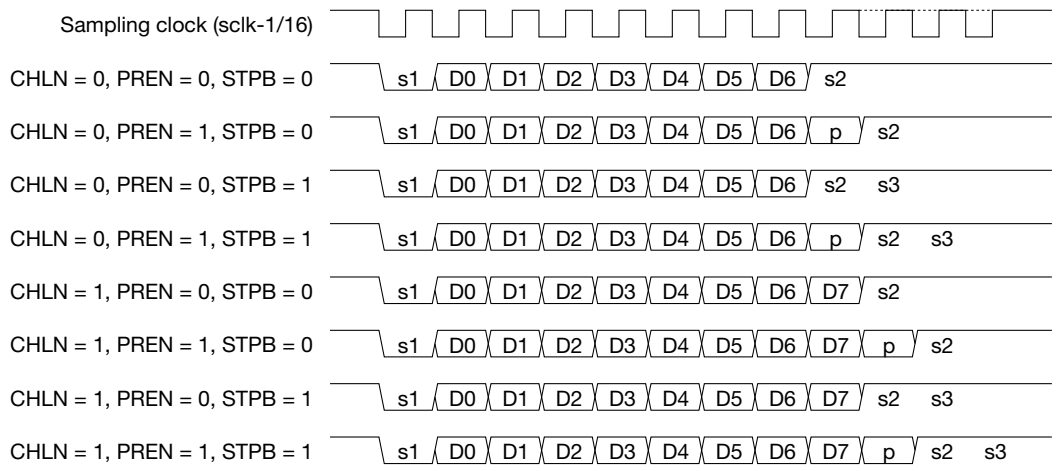


Figure 18.4.1: Transfer data format

18.5 Data Transfer Control

Make the following settings before starting data transfers.

(1) Select input clock. (See Section 18.3.)
    To use the internal clock, program the 8-bit timer to output the transfer clock. See Section 12.
(2) Set the transfer data format. (See Section 18.4.)
(3) To use the IrDA interface, set IrDA mode. (See Section 18.8.)
(4) Set interrupt conditions to use UART interrupts. (See Section 18.7.)

Note: Make sure the UART is halted (when RXEN/UART_CTL*x* register = 0) before changing the above settings.

∗ **RXEN**: UART Enable Bit in the UART Ch.*x* Control (UART_CTL*x*) Register (D0/0x4104/0x4124)

Permitting data transfers

Set the RXEN bit (D0/UART_CTL*x* register) to 1 to permit data transfers. This switches transfer circuits to enable transfers.

Note: Do not set the RXEN bit to 0 while the UART is sending or receiving data.

Data transfer control

To start data transmission, program the transmission data to the UART_TXD*x* register (0x4101/0x4121).

∗ **UART_TXD*x***: UART Ch.*x* Transmit Data Register (0x4101/0x4121)

The data is written to the transmit data buffer, and the transmission circuit starts sending data.
The buffer data is sent to the transmit shift register, and the start bit is output from the SOUT pin. The data in the shift register is then output from the LSB. The transfer data bit is shifted in sync with the sampling clock rising edge and output in sequence via the SOUT pin. Following output of MSB, the parity bit (if parity is enabled) and stop bit are output.

The transmission circuit includes the TDBE (D0/UART_ST*x* register) and TRBS (D2/UART_ST*x* register) status flags.

∗ **TDBE**: Transmit Data Buffer Empty Flag in the UART Ch.*x* Status (UART_ST*x*) Register (D0/0x4100/0x4120)
∗ **TRBS**: Transmit Busy Flag in the UART Ch.*x* Status (UART_ST*x*) Register (D2/0x4100/0x4120)

The TDBE flag indicates the transmit data buffer status. This flag switches to 0 when the application program programs data to the transmit data buffer and reverts to 1 when the buffer data is sent to the transmit shift register. Interrupts can be generated when this flag is 1 (see Section 18.7). Subsequent data is sent after confirming that the transmit data buffer is empty either by using this interrupt or by inspecting the TDBE flag. The transmission buffer size is 1 byte, but a shift register is provided separately to allow data to be written while the previous data is being sent. Always confirm that the transmit data buffer is empty before writing transmission data. Writing data while the TDBE flag is 0 will overprogram earlier transmission data inside the transmit data buffer.

The TRBS flag indicates the shift register status. This flag switches to 1 when transmission data is loaded from the transmit data buffer to the shift register and reverts to 0 once the data is sent. Read this flag to check whether the transmission circuit is operating or at standby.

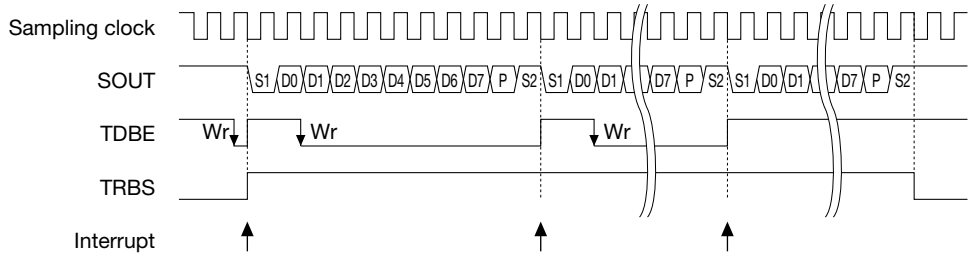


Figure 18.5.1: Data transmission timing chart

Data reception control

The receiving circuit is launched by setting the RXEN bit to 1, enabling data to be received from an external serial device.

When the external serial device sends the start bit, the receiving circuit detects its Low level and starts sampling the following data bits. The data bits are sampled at the sampling clock rising edge, and the lead bit is loaded into the receive shift register as LSB. Once the MSB has been received into the shift register, the received data is loaded into the receive data buffer. If parity checking is enabled, the receiving circuit checks parity at the same time by checking the parity bit received immediately after the MSB.

The receive data buffer, a 2-byte FIFO, receives data until full.

Received data in the buffer can be read from the UART_RXD*x* register (0x4102/0x4122). The oldest data is read out first, clearing the register.

∗ **UART_RXD*x***: UART Ch.x Receive Data Register (0x4102/0x4122)

The receiving circuit includes the RDRY (D1/UART_ST*x* register) and RD2B (D3/UART_ST*x* register) buffer status flags.

∗ **RDRY**: Receive Data Ready Flag in the UART Ch.x Status (UART_ST*x*) Register (D1/0x4100/0x4120)
∗ **RD2B**: Second Byte Receive Flag in the UART Ch.x Status (UART_ST*x*) Register (D3/0x4100/0x4120)

The RDRY flag indicates that the receive data buffer still contains data. The RD2B flag indicates that the receive data buffer is full.

(1) RDRY = 0, RD2B = 0
The receive data buffer contents need not be read, since no data has been received.

(2) RDRY = 1, RD2B = 0
One data has been received. Read the receive data buffer once. This reading 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.

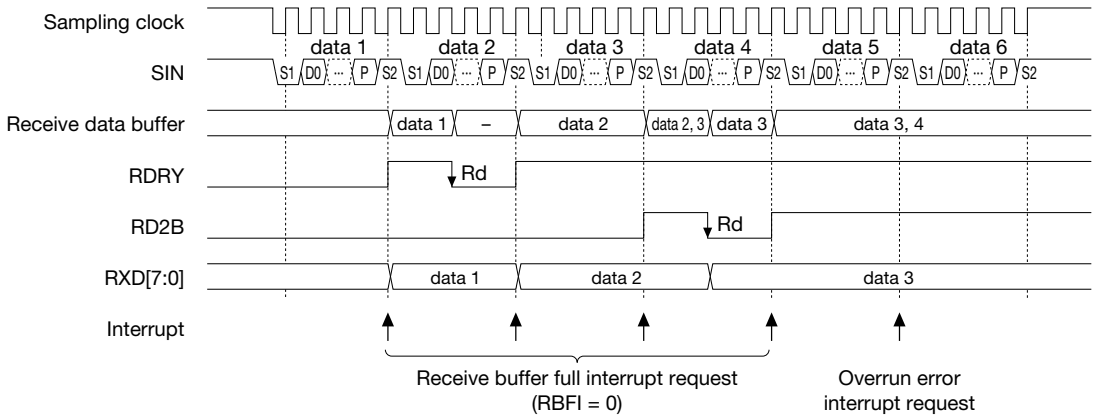
(3) RDRY = 1, RD2B = 1
Two data items have been received. Read the receive data buffer contents twice. The receive data buffer outputs the oldest data first. This reading resets the RD2B flag. The buffer then reverts to the state in (2) above. The second read outputs the most recent received data, after which the buffer reverts to the state in (1) above.
Even when the receive data buffer is full, the shift register can start receiving one more 8-bit data. An overrun error will occur if receiving is finished before the receive data buffer has been read. In this case, the last received data cannot be read. The contents of the receive data buffer must be read out before an overrun error occurs. For detailed information on overrun errors, refer to Section 18.6.

The volume of data received can be checked by reading these flags.

The UART allows receive buffer full interrupts to be generated once data has been received in the receive data buffer. These interrupts can be used to read the receive data buffer. With default settings, a receive buffer full interrupt occurs when the receive data buffer receives one item of data (status (2) above). This can be changed by setting the RBFI bit (D1/UART_CTL*x* register) to 1 so that an interrupt occurs when the receive data buffer receives two items of data.

∗ **RBFI**: Receive Buffer Full Interrupt Condition Setup Bit in the UART Ch.*x* Control (UART_CTL*x*) Register (D1/0x4104/0x4124)

Three error flags are also provided in addition to the flags previously mentioned. See Section 18.6 for detailed information on flags and receive errors.



S1: Start bit, S2: Stop bit, P: Parity bit, Rd: Data bits from RXD[7:0]

Figure 18.5.2: Data receiving timing chart

Blocking data transfers

After a data transfer is completed (both transmission and reception), data transfers are blocked by writing 0 to the RXEN bit. Confirm that the TDBE flag is 1 and the TRBS and RDRY flags are both 0 before blocking data transfer.

Setting the RXEN bit to 0 empties the transmission data buffers, clearing any remaining data. The data being transferred cannot be guaranteed if RXEN is set to 0 while data is being sent or received.

18.6 Receive Errors

Three different receive errors may be detected while receiving data.
Since receive errors are interrupt factors, they can be processed by generating interrupts. For more information on UART interrupt control, refer to Section 18.7.

Parity error

If PREN (D3/UART_MOD*x* register) has been set to 1 (parity enabled), data received is checked for parity.
Data received in the shift register is checked for parity when sent to the receive data buffer. The matching is checked against the PMD (D2/UART_MOD*x* register) setting (odd or even parity). If the result is a non-match, a parity error is issued, and the parity error flag PER (D5/UART_ST*x* register) is set to 1.
Even if this error occurs, the data received is sent to the receive data buffer, and the receiving operation continues. However, the received data cannot be guaranteed if a parity error occurs.
The PER flag (D5/UART_ST*x* register) is reset to 0 by writing as 1.

- **PREN**: Parity Enable Bit in the UART Ch.*x* Mode (UART_MOD*x*) Register (D3/0x4103/0x4123)
- **PMD**: Parity Mode Select Bit in the UART Ch.*x* Mode (UART_MOD*x*) Register (D2/0x4103/0x4123)
- **PER**: Parity Error Flag in the UART Ch.*x* Status (UART_ST*x*) Register (D5/0x4100/0x4120)

Framing error

A framing error occurs if the stop bit is received as 0 and the UART determines sync offset. If the stop bit is set to two bits, only the first bit is checked.
The framing error flag FER (D6/UART_ST*x* register) is set to 1 if this error occurs. The received data is still transferred to the receive data buffer if this error occurs and the receiving operation continues, but the data cannot be guaranteed, even if no framing error occurs for subsequent data receiving.
The FER flag (D6/UART_ST*x* register) is reset to 0 by writing as 1.

- **FER**: Framing Error Flag in the UART Ch.*x* Status (UART_ST*x*) Register (D6/0x4100/0x4120)

Overrun error

Even if the receive data buffer is full (two data items already received), a third item of data can be received in the shift register. However, if the receive data buffer is not emptied (by reading out data received) by the time this data has been received, the third data received in the shift register will not be sent to the buffer and generate an overrun error.
If an overrun error occurs, the overrun error flag OER (D4/UART_ST*x* register) is set to 1.
The receiving operation continues even if this error occurs.
The OER flag (D4/UART_ST*x* register) is reset to 0 by writing as 1.

- **OER**: Overrun Error Flag in the UART Ch.*x* Status (UART_ST*x*) Register (D4/0x4100/0x4120)

18.7 UART Interrupts

The UART includes a function for generating the following three different interrupt types.
- Transmit buffer empty interrupt
- Receive buffer full interrupt
- Receive error interrupt

The UART outputs one interrupt signal shared by the three above interrupt factor types to the interrupt controller (ITC). Inspect the status flag or error flag to determine the interrupt factor occurring.

Transmit buffer empty interrupt

To use this interrupt, set TIEN (D4/UART_CTL*x* register) to 1. If TIEN is set to 0 (default), interrupt requests for this factor will not be sent to the ITC.

∗ **TIEN**: Transmit Buffer Empty Interrupt Enable Bit in the UART Ch.*x* Control (UART_CTL*x*) Register (D4/0x4104/0x4124)

When transmission data written to the transmit data buffer is transferred to the shift register, the UART sets the TDBE bit (D0/UART_ST*x* register) to 1, indicating that the transmit data buffer is empty. If transmit buffer empty interrupts are permitted (TIEN = 1), an interrupt request pulse is sent simultaneously to the ITC.

∗ **TDBE**: Transmit Data Buffer Empty Flag in the UART Ch.*x* Status (UART_ST*x*) Register (D0/0x4100/0x4120)

An interrupt occurs if other interrupt conditions are met.
You can inspect the TDBE flag in the UART interrupt handler routine to determine whether the UART interrupt is attributable to a transmit buffer empty. If TDBE is 0, the next transmission data can be written to the transmit data buffer by the interrupt handler routine.

Receive buffer full interrupt

To use this interrupt, set RIEN (D5/UART_CTL*x* register) to 1. If RIEN is set to 0 (default), interrupt requests for this factor will not be sent to the ITC.

∗ **RIEN**: Receive Buffer Full Interrupt Enable Bit in the UART Ch.*x* Control (UART_CTL*x*) Register (D5/0x4104/0x4124)

If the specified volume of received data is loaded into the receive data buffer when a receive buffer full interrupt is permitted (RIEN = 1), the UART outputs an interrupt request pulse to the ITC. If RBFI (D1/UART_CTL*x* register) is 0, an interrupt request pulse is output as soon as one item of received data is loaded into the receive data buffer (RDRY flag (D1/UART_ST*x* register) is set to 1). If RBFI (D1/UART_CTL*x* register) is 1, an interrupt request pulse is output as soon as two items of received data are loaded into the receive data buffer (RD2B flag (D3/UART_ST*x* register) is set to 1).

∗ **RBFI**: Receive Buffer Full Interrupt Condition Ch.*x* Setup Bit in the UART Control (UART_CTL*x*) Register (D1/0x4104/0x4124)
∗ **RDRY**: Receive Data Ready Flag in the UART Ch.*x* Status (UART_ST*x*) Register (D1/0x4100/0x4120)
∗ **RD2B**: Second Byte Receive Flag in the UART Ch.*x* Status (UART_ST*x*) Register (D3/0x4100/0x4120)

An interrupt occurs if other interrupt conditions are met.
You can inspect the RDRY and RD2B flags in the UART interrupt handler routine to determine whether the UART interrupt is attributable to a receive buffer full. If RDRY or RD2B is 1, the received data can be read from the receive data buffer by the interrupt handler routine.

Receive error interrupt

To use this interrupt, set REIEN (D6/UART_CTL register) to 1. If REIEN is set to 0 (default), interrupt requests will not be sent to the ITC for this factor.

* **REIEN**: Receive Error Interrupt Enable Bit in the UART Control (UART_CTL) Register (D6/0x4104)

The UART sets the error flags shown below to 1 if a parity error, framing error, or overrun error is detected when receiving data. If receive error interrupts are permitted (REIEN = 1), an interrupt request pulse is output at the same time to the ITC.

* **PER**: Parity Error Flag in the UART Ch.*x* Status (UART_ST*x*) Register (D5/0x4100/0x4120)
* **FER**: Framing Error Flag in the UART Ch.*x* Status (UART_ST*x*) Register (D6/0x4100/0x4120)
* **OER**: Overrun Error Flag in the UART Ch.*x* Status (UART_ST*x*) Register (D4/0x4100/0x4120)

If other interrupt conditions are satisfied, an interrupt occurs.
Inspect the error flags above as part of the UART interrupt handler routine to determine whether the UART interrupt was caused by a receive error. If any of the error flags has the value 1, the interrupt handler routine will proceed with error recovery.

Interrupt vectors

The UART interrupt vector numbers and vector addresses are as listed below.

Table 18.7.1: UART interrupt vector

Channel	Vector number	Vector address
Ch.0	16 (0x10)	TTBR + 0x40
Ch.1	17 (0x11)	TTBR + 0x44

Other interrupt settings

The ITC allows the priority of UART interrupts to be set between level 0 (the default value) and level 7 for each channel. To generate actual interrupts, the PSR (S1C17 core internal processor status register) IE (interrupt enable) bit must be set to 1.
For more information on interrupt processing, see "6 Interrupt Controller (ITC)."

18.8 IrDA Interface

This UART module incorporates an RZI modulation/demodulation circuit enabling implementation of IrDA 1.0-compatible infrared communication simply by adding basic external circuits.

The transmission data output from the UART transmit shift register is input to the modulation circuit and output from the SOUT pin after the Low pulse has been modulated to a 3/16 sclk cycle.

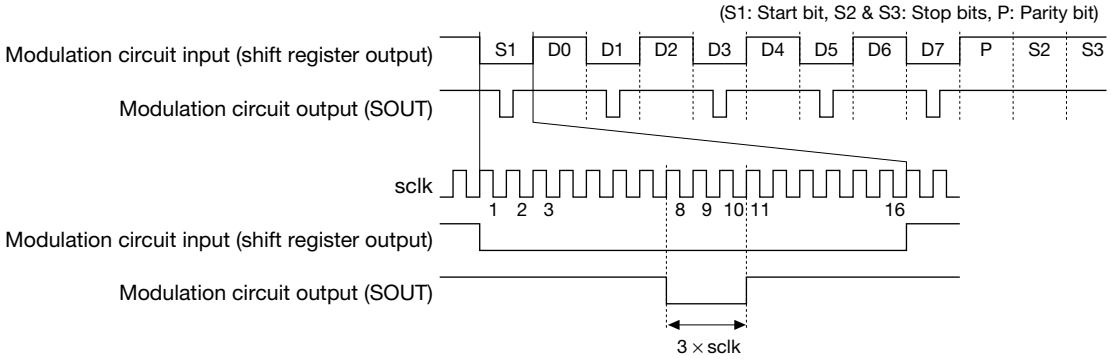


Figure 18.8.1: Transmission signal waveform

The received IrDA signal is input to the demodulation circuit and the Low pulse width is converted to 16 sclk cycles before entry to the receive shift register. The demodulation circuit uses the pulse detection clock selected from the prescaler output clock separately from the transfer cock to detect Low pulses input (when minimum pulse width = 1.41 μs/115,200 bps).

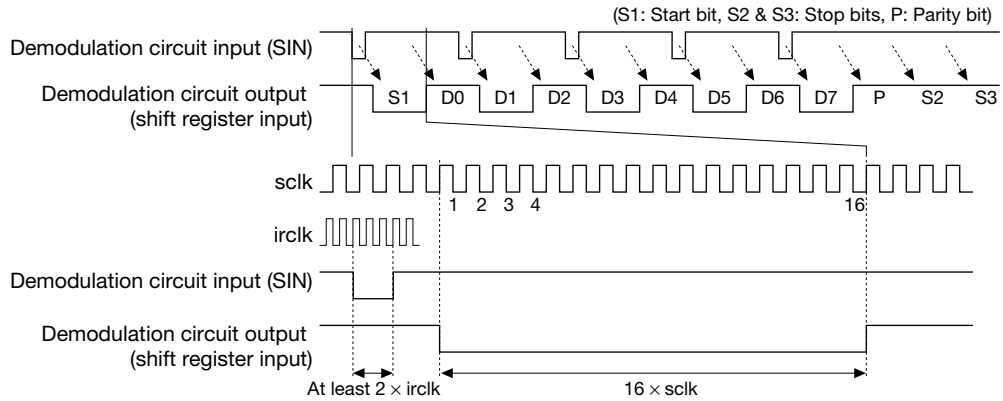


Figure 18.8.2: Receive signal waveform

IrDA enable

To use the IrDA interface function, set IRMD (D0/UART_EXP*x* register) to 1. This enables the RZI modulation/demodulation circuit.

∗ **IRMD**: IrDA Mode Select Bit in the UART Ch.*x* Expansion (UART_EXP*x*) Register (D0/0x4105/0x4125)

Note: This must be set before setting other UART conditions.

IrDA receive detection clock selection

The input pulse detection clock is selected from among the prescaler output clock PCLK-1/1 to PCLK-1/128 using IRCLK[2:0] (D[6:4]/UART_EXP*x* register).

* **IRCLK[2:0]**: IrDA Receive Detection Clock Select Bits in the UART Ch. Expansion (UART_EXP*x*) Register (D[6:4]/0x4105/0x4125)

Table 18.8.1: IrDA receive detection clock selection

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 selected as a clock faster than the 8-bit timer or transfer clock sclk input via the SCLK*x* pin. The demodulation circuit treats Low pulses with a width of at least 2 IrDA receive detection clock cycles as valid and converts them to 16 sclk cycle width Low pulses. Select the prescaler output clock to enable detection of input pulses with a minimum width of 1.41 μs.

Serial data transfer control

Data transfer control in IrDA mode is identical to that for normal interfaces. For detailed information on data format settings and data transfer and interrupt control methods, refer to the previous discussions.

18.9 Control Register Details

Table 18.9.1: UART register list

Address	Register name		Function
0x4100	UART_ST0	UART Ch.0 Status Register	Transfer, buffer, error status display
0x4101	UART_TXD0	UART Ch.0 Transmit Data Register	Transmission data
0x4102	UART_RXD0	UART Ch.0 Receive Data Register	Received data
0x4103	UART_MOD0	UART Ch.0 Mode Register	Transfer data format setting
0x4104	UART_CTL0	UART Ch.0 Control Register	Data transfer control
0x4105	UART_EXP0	UART Ch.0 Expansion Register	IrDA mode setting
0x4120	UART_ST1	UART Ch.1 Status Register	Transfer, buffer, error status display
0x4121	UART_TXD1	UART Ch.1 Transmit Data Register	Transmission data
0x4122	UART_RXD1	UART Ch.1 Receive Data Register	Received data
0x4123	UART_MOD1	UART Ch.1 Mode Register	Transfer data format setting
0x4124	UART_CTL1	UART Ch.1 Control Register	Data transfer control
0x4125	UART_EXP1	UART Ch.1 Expansion Register	IrDA mode setting

The UART registers are described in detail below. These are 8-bit registers.

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

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	

The "*x*" in register names indicates the channel number (0 or 1).

0x4100: UART Ch.0 Status Register (UART_ST0)
0x4120: UART Ch.1 Status Register (UART_ST1)

D7 Reserved

D6 FER: Framing Error Flag

Indicates whether a framing error has occurred.

1 (R): Error occurred
0 (R): No error (default)
1 (W): Reset to 0
0 (W): Disabled

FER is set to 1 when a framing error occurs. Framing errors occur when data is received with the stop bit set to 0.
FER is reset by writing 1.

D5 PER: Parity Error Flag

Indicates whether a parity error has occurred.

1 (R): Error occurred
0 (R): No error (default)
1 (W): Reset to 0
0 (W): Disabled

PER is set to 1 when a parity error occurs. Parity checking is enabled only when PREN (D3/ UART_MOD*x* register) is set to 1 and is performed when received data is transferred from the shift register to the receive data buffer.
PER is reset by writing 1.

D4 OER: Overrun Error Flag

Indicates whether an overrun error has occurred.

1 (R): Error occurred
0 (R): No error (default)
1 (W): Reset to 0
0 (W): Disabled

OER is set to 1 when an overrun error occurs. Overrun errors occur when data is received in the shift register when the receive data buffer is already full and additional data is sent. The receive data buffer is not overwritten if this error occurs. The shift register is overwritten as soon as the error occurs.
OER is reset by writing 1.

D3 RD2B: Second Byte Received Flag

Indicates that the receive data buffer contains two items of received data.

1 (R): Second byte can be read
0 (R): Second byte not received (default)

RD2B is set to 1 when the second byte of data is loaded into the receive data buffer and is reset to 0 when the first data is read from the receive data buffer.

D2 TRBS: Transmit Busy Flag

Indicates the transmit shift register status.

1 (R): Operating

0 (R): Standby (default)

TRBS is set to 1 when transmission data is loaded from the transmit data buffer into the shift register and is reset to 0 when the data transfer is complete. Inspect TRBS to determine whether the transmit circuit is operating or at standby.

D1 RDRY: Receive Data Ready Flag

Indicates that the receive data buffer contains valid received data.

1 (R): Data can be read

0 (R): Buffer empty (default)

RDRY is set to 1 when received data is loaded into the receive data buffer and is reset to 0 when all data has been read from the receive data buffer.

D0 TDBE: Transmit Data Buffer Empty Flag

Indicates the state of the transmit data buffer.

1 (R): Buffer empty (default)

0 (R): Data exists

TDBE is reset to 0 when transmit data is written to the transmit data buffer and is set to 1 when the data is transferred to the shift register.

0x4101/0x4121: UART Ch.*x* Transmit Data Registers (UART_TXD*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
UART Ch.*x* Transmit Data Register (UART_TXD*x*)	**0x4101 0x4121** (8 bits)	D7–0	**TXD[7:0]**	Transmit data TXD7(6) = MSB TXD0 = LSB	0x0 to 0xff (0x7f)	0x0	R/W	

The "*x*" in register names indicates the channel number (0 or 1).

0x4101: UART Ch.0 Transmit Data Register (UART_TXD0)
0x4121: UART Ch.1 Transmit Data Register (UART_TXD1)

D[7:0] TXD[7:0]: Transmit Data

Program transmit data to be set in the transmit data buffer. (Default: 0x0)
The UART begins transmitting when data is written to this register. Data written to TXD[7:0] is retained until sent to the transmit data buffer.
Transmitting data from within the transmit data buffer generates a transmit buffer empty interrupt factor.
TXD7 (MSB) is invalid in 7-bit mode.
Serial converted data is output from the SOUT pin, with the LSB first bits set to 1 as High level and bits set to 0 as Low level.
This register can also be read from.

0x4102/0x4122: UART Ch.*x* Receive Data Registers (UART_RXD*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
UART Ch.*x* Receive Data Register (UART_RXD*x*)	**0x4102 0x4122** (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.

The "*x*" in register names indicates the channel number (0 or 1).

0x4102: UART Ch.0 Receive Data Register (UART_RXD0)
0x4122: UART Ch.1 Receive Data Register (UART_RXD1)

D[7:0] RXD[7:0]: Receive Data

Data in the receive data buffer is read out in sequence, starting with the oldest. Received data is placed in the receive data buffer. The receive data buffer is a 2-byte FIFO that allows proper data receipt until it fills, even if data is not read out. If the buffer is full and the shift register also contains received data, an overrun error will occur, unless the data is read out before receipt of the subsequent data starts.

The receive circuit includes two receive buffer status flags: RDRY (D1/UART_ST*x* register) and RD2B (D3/UART_ST*x* register). The RDRY flag indicates the presence of valid received data in the receive data buffer, while RD2B flag indicates the presence of two items of received data in the receive data buffer.

A receive buffer full interrupt occurs when the received data in the receive data buffer reaches the number specified by RBFI (D1/UART_CTL*x* register).

0 is loaded into RXD7 in 7-bit mode.

Serial data input via the SIN pin is converted to parallel, with the initial bit as LSB, the High level bit as 1, and the Low level bit as 0. This data is then loaded into the receive data buffer.

This register is read-only. (Default: 0x0)

0x4103/0x4123: UART Ch.*x* Mode Registers (UART_MOD*x*)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
UART Ch.*x*	**0x4103**	D7–5	**–**	reserved	–				–	–	0 when being read.
Mode Register	**0x4123**	D4	**CHLN**	Character length	1	8 bits	0	7 bits	0	R/W	
(UART_MOD*x*)	(8 bits)	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	

The "*x*" in register names indicates the channel number (0 or 1).

0x4103: UART Ch.0 Mode Register (UART_MOD0)
0x4123: UART Ch.1 Mode Register (UART_MOD1)

D[7:5] Reserved

D4 CHLN: Character Length Select Bit

Selects the serial transfer data length.
1 (R/W): 8 bits
0 (R/W): 7 bits (default)

D3 PREN: Parity Enable Bit

Enables the parity function.
1 (R/W): With parity
0 (R/W): No parity (default)

PREN is used to select receive data parity checking and to determine whether a parity bit is added to transmitted data. Setting PREN to 1 parity-checks the received data. A parity bit is automatically added to the transmitted data. If PREN is set to 0, no parity bit is checked or added.

D2 PMD: Parity Mode Select Bit

Selects the parity mode.
1 (R/W): Odd parity
0 (R/W): Even parity (default)

Writing 1 to PMD selects odd parity; writing 0 to it selects even parity. Parity checking and parity bit addition are enabled only when PREN (D3) is set to 1. The PMD setting is disabled if PREN (D3) is 0.

D1 STPB: Stop Bit Select Bit

Selects the stop bit length.
1 (R/W): 2 bits
0 (R/W): 1 bit (default)

Writing 1 to STPB selects two stop bits; writing 0 to it selects one bit. The start bit is fixed at one bit.

D0 SSCK: Input Clock Select Bit

Selects the input clock.
1 (R/W): External clock (SCLK*x*)
0 (R/W): Internal clock (default)

Selects whether the internal clock (8-bit timer output clock) or external clock (input via SCLK*x* pin) is used. Writing 1 to SSCK selects the external clock; Writing 0 to it selects the internal clock.

0x4104/0x4124: UART Ch.*x* Control Registers (UART_CTL*x*)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
UART Ch.*x* Control Register (UART_CTL*x*)	0x4104 0x4124 (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	

The "*x*" in register names indicates the channel number (0 or 1).

0x4104: UART Ch.0 Control Register (UART_CTL0)
0x4124: UART Ch.1 Control Register (UART_CTL1)

D7 **Reserved**

D6 **REIEN: Receive Error Interrupt Enable Bit**

Permits interrupt requests to the ITC when a receive error occurs.
1 (R/W): Permitted
0 (R/W): Prohibited (default)

Set this bit to 1 to process receive errors using interrupts.

D5 **RIEN: Receive Buffer Full Interrupt Enable Bit**

Permits interrupt requests to the ITC caused when the received data quantity in the receive data buffer reaches the quantity specified in RBFI (D1).
1 (R/W): Permitted
0 (R/W): Prohibited (default)

Set this bit to 1 to read receive data using interrupts.

D4 **TIEN: Transmit Buffer Empty Interrupt Enable Bit**

Permits interrupt requests to the ITC caused when transmission data in the transmit data buffer is sent to the shift register (i.e. when data transmission begins).
1 (R/W): Permitted
0 (R/W): Prohibited (default)

Set this bit to 1 to program data to the transmit data buffer using interrupts.

D[3:2] **Reserved**

D1 **RBFI: Receive Buffer Full Interrupt Condition Setup Bit**

Sets the quantity of data in the receive buffer to generate a receive buffer full interrupt.
1 (R/W): 2 bytes
0 (R/W): 1 byte (default)

If receive buffer full interrupts are permitted (RIEN = 1), the UART outputs an interrupt request pulse to the ITC when the quantity of received data specified by RBFI is loaded into the receive data buffer. If the RBFI bit is 0, an interrupt request pulse is output as soon as one item of received data is loaded into the receive data buffer (when the RDRY flag (D1/UART_ST*x* register) is set to 1). If RBFI is 1, an interrupt request pulse is output as soon as two items of received data are loaded into the receive data buffer (when the RD2B flag (D3/UART_ST*x* register) is set to 1).

D0 **RXEN: UART Enable Bit**

Permits data transfer by the UART.
1 (R/W): Permitted
0 (R/W): Prohibited (default)

Set RXEN to 1 before starting UART transfers. Setting RXEN to 0 will stop data transfers. Set the transfer conditions while RXEN is 0.
Preventing transfers by writing 0 to RXEN also clears transmit data buffer.

0x4105/0x4125: UART Ch.*x* Expansion Registers (UART_EXP*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
UART Ch.*x* Expansion Register (UART_EXP*x*)	**0x4105** **0x4125** (8 bits)	D7	**–**	reserved	–	–	–	0 when being read.
		D6–4	**IRCLK[2:0]**	IrDA receive detection clock select	IRCLK[2:0] / 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	0x0	R/W	
		D3–1	**–**	reserved	–	–	–	0 when being read.
		D0	**IRMD**	IrDA mode select	1 On 0 Off	0	R/W	

The "*x*" in register names indicates the channel number (0 or 1).

0x4105: UART Ch.0 Expansion Register (UART_EXP0)
0x4125: UART Ch.1 Expansion Register (UART_EXP1)

D7 Reserved

D[6:4] IRCLK[2:0]: IrDA Receive Detection Clock Select Bits

Select the prescaler output clock used as the IrDA input pulse detection clock.

Table 18.9.2: IrDA receive detection clock selection

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 selected as a clock faster than the 8-bit timer or transfer clock sclk input via the SCLK pin.

The demodulation circuit treats Low pulses with a width of at least 2 IrDA receive detection clock cycles as valid. Select the appropriate prescaler output clock to enable detection of input pulses with a minimum width of 1.41 µs.

D[3:1] Reserved

D0 IRMD: IrDA Mode Select Bit

Switches the IrDA interface function on and off.

1 (R/W): On

0 (R/W): Off (default)

Set this to 1 to use the IrDA interface. When this bit is set to 0, this module functions as a normal UART, with no IrDA functions.

18.10 Precautions

- The following UART bits should be set with transfers blocked (RXEN = 0).
  - All UART_MOD*x* register (0x4103/0x4123) bits (SSCK, STPB, PMD, PREN, CHLN)
  - RBFI bit in the UART_CTL*x* register
  - All UART_EXP*x* register (0x4105/0x4125) bits (IRMD, IRCLK[2:0])

  ∗ **RXEN**: UART Enable Bit in the UART Ch.x Control (UART_CTLx) Register (D0/0x4104/0x4124)
- Do not set RXEN to 0 while the UART is transmitting or receiving data.
- The UART transfer rate is capped at 460,800 bps. Do not set faster transfer rates.
- Preventing transfer by setting RXEN to 0 clears (initializes) transfer data buffers. Before writing 0 to RXEN, confirm the absence of data in the buffers awaiting transmission.
- The IrDA receive detection clock must be selected as a clock faster than the 8-bit timer or transfer clock sclk input via the SCLK pin.
- The IrDA interface demodulation circuit treats Low pulses with a width of at least 2 IrDA receive detection clock cycles as valid. Select the appropriate prescaler output clock to enable detection of input pulses with a minimum width of 1.41 μs as a 2 IrDA receive detection clock.

19 SPI

19.1 SPI Configuration

The S1C17003 incorporates a synchronized serial interface module (SPI). This SPI module supports both Master and Slave modes and is used for 8-bit data transfers. Four different data transfer timing patterns (clock phase and polarity) can be selected.
The SPI module includes a transmit data buffer and receive data buffer separate from the shift register, and is capable of generating two different interrupt types (transmit buffer empty and receive buffer full). This allows easy processing of continuous serial data transfer using interrupts.
Figure 19.1.1 illustrates the SPI module configuration.

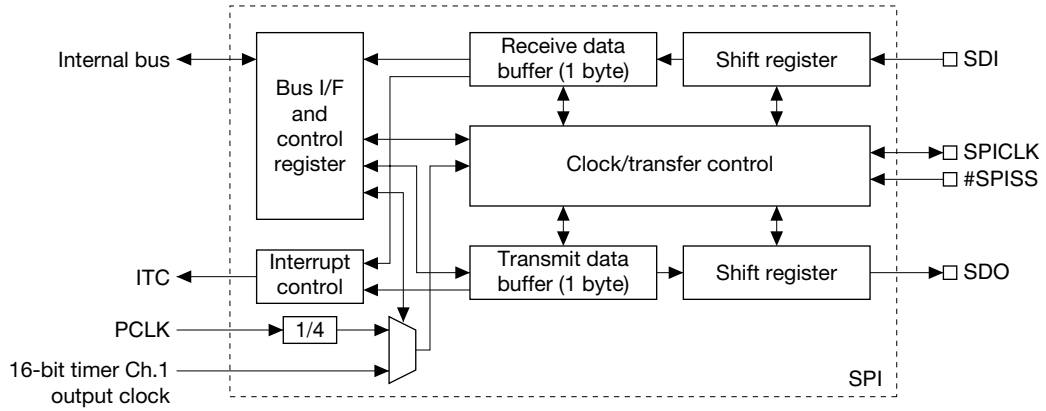


Figure 19.1.1: SPI module configuration

19.2 SPI Input/Output Pins

Table 19.2.1 lists the SPI pins.

Table 19.2.1: SPI pin list

Pin name	I/O	Qty	Function
SDI (P06)	I	1	SPI data input pin Inputs serial data from SPI bus.
SDO (P05)	O	1	SPI data output pin Outputs serial data to SPI bus.
SPICLK (P04)	I/O	1	SPI external clock input/output pin Outputs SPI clock when SPI is in Master mode. Inputs external clock when SPI is used in Slave mode.
#SPISS (P07)	I	1	SPI slave selection signal (active Low) input pin SPI (Slave mode) is selected as slave device by Low input to this pin.

The SPI input/output pins (SDI, SDO, SPICLK, #SPISS) are shared with general purpose input/output port pins (P06, P05, P04, P07) and are initially set as general purpose input/output port pins. The function must be switched using the P0_PMUX register settings to use general purpose input/output port pins as SPI input/output pins. Switch the pins to SPI mode by setting the following control bits to 1.

P06 → SDI
- ∗ **P06MUX**: P06 Port Function Select Bit in the P0 Port Function Select (P0_PMUX) Register (D4/0x52a1)

P05 → SDO
- ∗ **P05MUX**: P05 Port Function Select Bit in the P0 Port Function Select (P0_PMUX) Register (D2/0x52a1)

P04 → SPICLK
- ∗ **P04MUX**: P04 Port Function Select Bit in the P0 Port Function Select (P0_PMUX) Register (D0/0x52a1)

P07 → #SPISS
- ∗ **P07MUX**: P07 Port Function Select Bit in the P0 Port Function Select (P0_PMUX) Register (D6/0x52a1)

For detailed information on pin function switching, refer to “10.2 Input/Output Pin Function Selection (Port MUX).”

19.3 SPI Clock

The Master mode SPI uses the internal clock output by the 16-bit timer Ch.1 as the SPI clock. This clock is output from the SPICLK pin to the slave device while also driving the shift register.
Use the MCLK (D9/SPI_CTL register) to select to use the 16-bit timer Ch.1 output clock or PCLK-1/4 clock is used. Setting MCLK to 1 selects the 16-bit timer Ch.1 output clock; setting to 0 selects the PCLK-1/4 clock.

* **MCLK**: SPI Clock Source Select Bit in the SPI Control (SPI_CTL) Register (D9/0x4326)

Using the 16-bit timer Ch.1 output clock enables programmable transfer rates. For more information on 16-bit timer control, see "11 16-bit Timer (T16)."

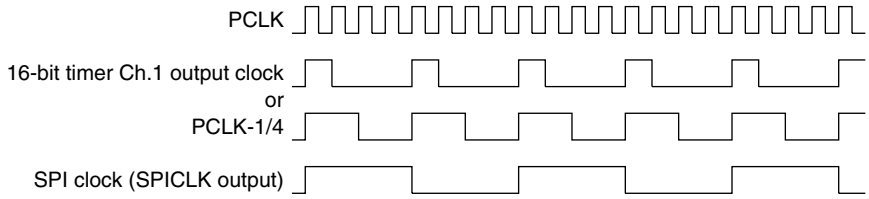


Figure 19.3.1: Master mode SPI clock

In Slave mode, the SPI clock is input via the SPICLK pin.

Note: The duty ratio of the clock input via the SPICLK pin must be 50%.

19.4 Data Transfer Condition Settings

The SPI module can be set to Master or Slave modes. The SPI clock polarity and phase can also be set via the SPI_CTL register.
The data length is fixed at 8 bits.

Note: Make sure the SPI module is halted (when SPEN/SPI_CTL register = 0) before Master/Slave mode selection and clock condition settings.

∗ **SPEN**: SPI Enable Bit in the SPI Control (SPI_CTL) Register (D0/0x4326)

Master/Slave mode selection

MSSL (D1/SPI_CTL register) is used to set the SPI module to Master mode or Slave mode. Setting MSSL to 1 sets Master mode; setting it to 0 (default) sets Slave mode. In Master mode, data is transferred using the internal clock. In Slave mode, data is transferred by inputting the master device clock.

∗ **MSSL**: Master/Slave Mode Select Bit in the SPI Control (SPI_CTL) Register (D1/0x4326)

SPI clock polarity and phase settings

The SPI clock polarity is selected by CPOL (D2/SPI_CTL register). Setting CPOL to 1 treats the SPI clock as active Low; setting it to 0 (default) treats it as active High.

∗ **CPOL**: Clock Polarity Select Bit in the SPI Control (SPI_CTL) Register (D2/0x4326)

The SPI clock phase is selected by CPHA (D3/SPI_CTL register).

∗ **CPHA**: Clock Phase Select Bit in the SPI Control (SPI_CTL) Register (D3/0x4326)

As shown below, these control bits set transfer timing.

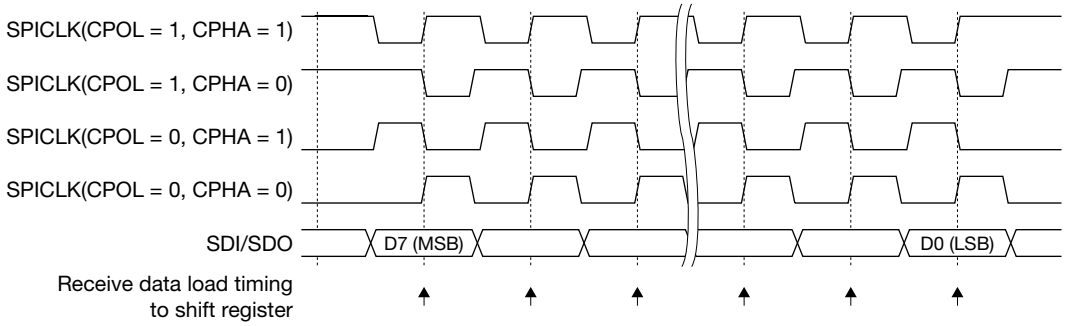


Figure 19.4.1: Clock and data transfer timing

Note: When the SPI module is used in master mode with CPHA set to 0, the clock may change a minimum of one system clock cycle time from change of the first transmit data bit.

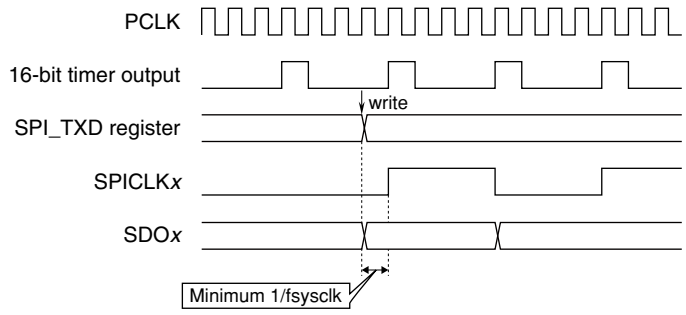


Figure 19.4.2 SDO*x* and SPICLK*x* Change Timings when CPHA = 0

The half SPICLK*x* cycle will be secured from change of data to change of the clock for the second and following transmit data bits and the second and following bytes during continuous transfer.

MSB initial/LSB initial settings

Use MLSB (D8/SPI_CTL register) to select whether the data MSB or LSB is input or output first.
MSB initial is set when MLSB is 0 (the default value); LSB initial is set when MLSB is 1.

* **MLSB**: LSB/MSB First Mode Select Bit in the SPI Control (SPI_CTL) Register (D8/0x4326)

19.5 Data Transfer Control

Make the following settings before starting data transfers.
(1) Set the 16-bit timer Ch.1 to output the SPI clock. (See Section 11.)
(2) Select Master mode or Slave mode. (See Section 19.4.)
(3) Set clock conditions. (See Section 19.4.)
(4) Set the interrupt conditions to use SPI interrupts. (See Section 19.6.)

Note: Make sure the SPI is halted (when SPEN/SPI_CTL register = 0) before changing the above settings.

* **SPEN**: SPI Enable Bit in the SPI Control (SPI_CTL) Register (D0/0x4326)

Permitting data transfers

Set the SPEN bit (D0/SPI_CTL register) to 1 to permit SPI operations. This enables SPI transfers and permits clock input/output.

Note: Do not set SPEN to 0 when the SPI module is transferring data.

Data transfer control

To start data transmission, write the transmission data to the SPI_TXD register (0x4322).

* **SPI_TXD**: SPI Transmit Data Register (0x4322)

The data is written to the transmit data buffer, and the SPI module begins sending data. The buffer data is sent to the transmit shift register. In Master mode, the module starts clock output from the SPICLK pin. In Slave mode, the module awaits clock input from the SPICLK pin. The data in the shift register is shifted in sequence at the clock rising or falling edge, as determined by CPHA (D3/SPI_CTL register) and CPOL (D2/SPI_CTL register) (see Figure 19.4.1) and sent from the SDO pin with MSB leading.

* **CPHA**: Clock Phase Select Bit in the SPI Control (SPI_CTL) Register (D3/0x4326)
* **CPOL**: Clock Polarity Select Bit in the SPI Control (SPI_CTL) Register (D2/0x4326)

The SPI module includes the SPTBE (D0/SPI_ST register) and SPBSY (D2/SPI_ST register) status flags for transfer control.

* **SPTBE**: Transmit Data Buffer Empty Flag in the SPI Status (SPI_ST) Register (D0/0x4320)
* **SPBSY**: Transfer Busy Flag in the SPI Status (SPI_ST) Register (D2/0x4320)

The SPTBE flag indicates the transmit data buffer status. This flag switches to 0 when the application program writes data to the SPI_TXD register (transmit data buffer) and reverts to 1 when the buffer data is sent to the transmit shift register. Interrupts can be generated when this flag is 1 (see Section 19.6). Subsequent data is sent after confirming that the transmit data buffer is empty either by using this interrupt or by inspecting the SPTBE flag. The transmission buffer size is 1 byte, but a shift register is provided separately to allow data to be written while the previous data is being sent. Always confirm that the transmit data buffer is empty before writing transmission data. Writing data while the SPTBE flag is 0 will overwrite earlier transmission data inside the transmit data buffer.

In Master mode, the SPBSY flag indicates the shift register status. This flag switches to 1 when transmission data is loaded from the transmit data buffer to the shift register and reverts to 0 once the data is sent. Read this flag to check whether the SPI module is operating or at standby.
The Slave mode SPBSY flag indicates the SPI slave selection signal (#SPISS pin) status. The flag has the value 1 when the SPI module is selected in Slave mode and the value 0 when the module is not selected.

Data receipt control

In Master mode, dummy data is written to the SPI_TXD register (0x4322). Writing to the SPI TXD register creates the trigger for receipt as well as transmission start. Writing actual transmission data enables simultaneous transfers.
This starts the SPI clock output from SPICLK.

In Slave mode, the module waits until the clock is input from SPICLK. Slave mode involves only data receipt. There is no need to write to the SPI_TXD register if no transmission is required. The receiving operation is started by clock input from the master device. If data is transferred simultaneously, the transmission data is written to the SPI_TXD register before the clock is input.

The data is contained in sequence in the shift register at the rising or falling edge for the clock determined by CPHA (D3/SPI_CTL register) and CPOL (D2/SPI_CTL register). (See Figure 19.4.1.)
The received data is loaded into the receive data buffer once the 8 bits of data are received in the shift register.

Received data in the buffer can be read from the SPI_RXD register (0x4324)

* **SPI_RXD**: SPI Receive Data Register (0x4324)

The SPI module includes an SPRBF flag (D1/SPI_ST register) for receipt control.

* **SPRBF**: Receive Data Buffer Full Flag in the SPI Status (SPI_ST) Register (D1/0x4320)

The SPRBF flag indicates the receive data buffer status. This flag is set to 1 when the data received in the shift register is loaded into the receive data buffer, indicating that the receive data can be read out. It reverts to 0 when the buffer data is read out from the SPI_RXD register. An interrupt can be generated as soon as the flag is set to 1 (see Section 19.6). The received data should be read out either by using this interrupt or by inspecting the SPRBF flag to confirm that the receive data buffer contains valid receive data. The receive data buffer is 1 byte in size, but a shift register is also provided, enabling received data to be retained in the buffer even while the subsequent data is being received. Note that the receive data buffer should be read out before receiving the subsequent data is complete. If receiving the subsequent data is complete before the receive data buffer contents are read out, the newly received data will overwrite the previous received data in the buffer.

In Master mode, the SPBSY flag indicating the shift register state can be used in the same way while transferring data.

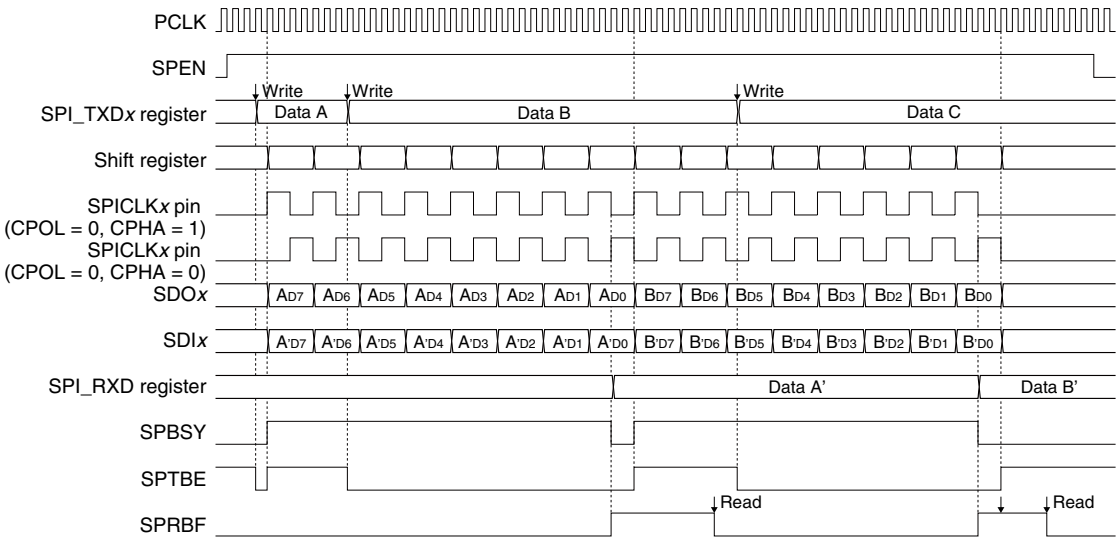


Figure 19.5.1: Data Transmission/Receiving Timing Chart (MSB first)

Blocking data transfers

After a data transfer is completed (both transmission and reception), data transfers are blocked by writing 0 to the SPEN bit. Confirm that the SPTBE flag is 1 and the SPBSY flag is 0 before blocking data transfer.

The data being transferred cannot be guaranteed if SPEN is set to 0 while data is being sent or received.

19.6 SPI Interrupts

The SPI module includes a function for generating the following two different interrupt types.
- Transmit buffer empty interrupt
- Receive buffer full interrupt

The SPI module outputs one interrupt signal shared by the three above interrupt factor types to the interrupt controller (ITC). Inspect the status flag to determine the interrupt factor occurring.

Transmit buffer empty interrupt

To use this interrupt, set SPTIE (D4/SPI_CTL register) to 1. If SPTIE is set to 0 (default), interrupt requests for this factor will not be sent to the ITC.

∗ **SPTIE**: Transmit Data Buffer Empty Interrupt Enable Bit in the SPI Control (SPI_CTL) Register (D4/0x4326)

When transmission data written to the transmit data buffer is transferred to the shift register, the SPI module sets the SPTBE bit (D0/SPI_ST register) to 1, indicating that the transmit data buffer is empty. If transmit buffer empty interrupts are permitted (SPTIE = 1), an interrupt request pulse is sent simultaneously to the ITC.

∗ **SPTBE**: Transmit Data Buffer Empty Flag in the SPI Status (SPI_ST) Register (D0/0x4320)

An interrupt occurs if other interrupt conditions are met.
You can inspect the SPTBE flag in the SPI interrupt processing routine to determine whether the SPI interrupt is attributable to a transmit buffer empty. If SPTBE is 1, the next transmission data can be written to the transmit data buffer by the interrupt processing routine.

Receive buffer full interrupt

To use this interrupt, set SPRIE (D5/SPI_CTL register) to 1. If SPRIE is set to 0 (default), interrupt requests for this factor will not be sent to the ITC.

∗ **SPRIE**: Receive Data Buffer Full Interrupt Enable Bit in the SPI Control (SPI_CTL) Register (D5/0x4326)

When data received in the shift register is loaded into the receive data buffer, the SPI module sets the SPRBF bit (D1/SPI_ST register) to 1, indicating that the receive data buffer contains readable received data. If receive buffer full interrupts are permitted (SPRIE = 1), an interrupt request pulse is output to the ITC at the same time.

∗ **SPRBF**: Receive Data Buffer Full Flag in the SPI Status (SPI_ST) Register (D1/0x4320)

An interrupt occurs if other interrupt conditions are met.
You can inspect the SPRBF flag in the SPI interrupt processing routine to determine whether the SPI interrupt is attributable to a receive buffer full. If SPRBF is 1, the received data can be read from the receive data buffer by the interrupt processing routine.

Interrupt vectors

The SPI interrupt vector numbers and vector addresses are as listed below.

Vector number: 18 (0x12)
Vector address: TTBR + 0x48

Other interrupt settings

The SPI interrupt priority can be set for the ITC between level 0 (default) and level 7. The PSR (S1C17 core internal processor status register) IE (interrupt enable) bit must be set to 1 to generate actual interrupts.
For specific information on interrupt processing, refer to “6 Interrupt Controller (ITC).”

19.7 Control Register Details

Table 19.7.1: SPI register list

Address	Register name		Function
0x4320	SPI_ST	SPI Status Register	Transfer, buffer status display
0x4322	SPI_TXD	SPI Transmit Data Register	Transmission data
0x4324	SPI_RXD	SPI Receive Data Register	Received data
0x4326	SPI_CTL	SPI Control Register	SPI mode and data transfer permission setting

The SPI registers are described in detail below. These are 16-bit registers.

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

0x4320: SPI Status Register (SPI_ST)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
SPI Status Register (SPI_ST)	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 Mode)/ss Signal Low Flag (Slave Mode)

Master mode

Indicates the SPI transfer status.

1 (R): Operating

0 (R): Standby (default)

SPBSY is set to 1 when the SPI starts data transfer in Master mode and is maintained at 1 while transfer is underway.

It is cleared to 0 once the transfer is complete.

Slave mode

Indicates the slave selection (#SPISS) signal status.

1 (R): Low level (this SPI is selected)

0 (R): High level (this SPI is not selected) (default)

SPBSY is set to 1 when the master device sets the #SPISS signal to active to select this SPI module (slave device). It is returned to 0 when the master device clears the SPI module selection by returning the #SPISS signal to inactive.

D1 SPRBF: Receive Data Buffer Full Flag

Indicates the receive data buffer status.

1 (R): Data full

0 (R): No data (default)

SPRBF is set to 1 when data received in the shift register is sent to the receive data buffer (when receiving is complete), indicating that the data can be read. It reverts to 0 once the buffer data is read from the SPI_RXD register (0x4324).

D0 SPTBE: Transmit Data Buffer Empty Flag

Indicates the state of the transmit data buffer.

1 (R): Empty (default)

0 (R): Data exists

SPTBE is set to 0 when transmit data is written to the SPI_TXD register (transmit data buffer, 0x4322), and is set to 1 when the data is transferred to the shift register (when transmission starts).

Transmission data is written to the SPI_TXD register when this bit is 1.

0x4322: SPI Transmit Data Register (SPI_TXD)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SPI Transmit Data Register (SPI_TXD)	**0x4322** (16 bits)	D15–8	**–**	reserved	–	–	–	0 when being read.
		D7–0	**SPTDB[7:0]**	SPI transmit data buffer SPTDB7 = MSB SPTDB0 = LSB	0x0 to 0xff	0x0	R/W	

D[15:8] Reserved

D[7:0] SPTDB[7:0]: SPI Transmit Data Buffer Bits

Set the transmission data to be written to the transmit data buffer. (Default: 0x0)

In Master mode, transmission is started by writing data to this register. In Slave mode, the contents of this register are sent to the shift register and transmission begins when the clock is input from the master.

SPTBE (D0/SPI_ST register) is set to 1 (empty) as soon as data written to this register has been transferred to the shift register. A transmit buffer empty interrupt is generated at the same time. The subsequent transmit data can then be written, even while data is being transmitted.

Serial converted data is output from the SDO pin with MSB leading, with the bit set to 1 as High level and the bit set to 0 as Low level.

Note: Make sure that SPEN is set to 1 before writing data to the SPI_TXD register to start data transmission/reception.

0x4324: SPI Receive Data Register (SPI_RXD)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SPI Receive Data Register (SPI_RXD)	**0x4324** (16 bits)	D15–8	**–**	reserved	–	–	–	0 when being read.
		D7–0	**SPRDB[7:0]**	SPI receive data buffer SPRDB7 = MSB SPRDB0 = LSB	0x0 to 0xff	0x0	R	

D[15:8] Reserved

D[7:0] SPRDB[7:0]: SPI Receive Data Buffer Bits

Contain the received data. (Default: 0x0)

SPRBF (D1/SPI_ST register) is set to 1 (data full) as soon as data is received and the shift register data has been transferred to the receive data buffer. A receive buffer full interrupt is generated at the same time. Data can then be read until subsequent data is received. If receiving the subsequent data is complete before the register has been read out, the new received data overwrites the contents.

Serial data input from the SDI pin with MSB leading is converted to parallel, with the High level bit set to 1 and the Low level bit set to 0. The data is the loaded into this register.

This register is read-only.

0x4326: SPI Control Register (SPI_CTL)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
SPI Control Register (SPI_CTL)	0x4326 (16 bits)	D15–10	–	reserved	–				–	–	0 when being read.
		D9	MCLK	SPI clock source select	1	T16 Ch.1	0	PCLK-1/4	0	R/W	
		D8	MLSB	LSB/MSB first mode select	1	LSB	0	MSB	0	R/W	
		D7–6	–	reserved	–				–	–	0 when being read.
		D5	SPRIE	Receive data buffer full int. enable	1	Enable	0	Disable	0	R/W	
		D4	SPTIE	Transmit data buffer empty int. enable	1	Enable	0	Disable	0	R/W	
		D3	CPHA	Clock phase select	1	Data out	0	Data in	0	R/W	These bits must be set before setting SPEN to 1.
		D2	CPOL	Clock polarity select	1	Active L	0	Active H	0	R/W	
		D1	MSSL	Master/slave mode select	1	Master	0	Slave	0	R/W	
		D0	SPEN	SPI enable	1	Enable	0	Disable	0	R/W	

D[15:10] Reserved

D9 MCLK: SPI Clock Source Select Bit

Selects the SPI clock source.
1 (R/W): 16-bit timer Ch.1
0 (R/W): PCLK-1/4 (default)

D8 MLSB: LSB/MSB First Mode Select Bit

Selects whether data is transferred with MSB first or LSB first.
1 (R/W): LSB first
0 (R/W): MSB first (default)

D[7:6] Reserved

D5 SPRIE: Receive Data Buffer Full Interrupt Enable Bit

Permits or prohibits receive data buffer full SPI interrupts.
1 (R/W): Permitted
0 (R/W): Prohibited (default)

Setting SPRIE to 1 permits the output of SPI interrupt requests to the ITC due to a receive data buffer full. These interrupt requests are generated when the data received in the shift register is transferred to the receive data buffer (when receipt is complete).
SPI interrupts are not generated by receive data buffer full if SPRIE is set to 0.

D4 SPTIE: Transmit Data Buffer Empty Interrupt Enable Bit

Permits or prohibits transmit data buffer empty SPI interrupts.
1 (R/W): Permitted
0 (R/W): Prohibited (default)

Setting SPTIE to 1 permits the output of SPI interrupt requests to the ITC due to a transmit data buffer empty. These interrupt requests are generated when the data written to the transmit data buffer is transferred to the shift register (when transmission starts).
SPI interrupts are not generated by transmit data buffer empty if SPTIE is set to 0.

D3 CPHA: SPI Clock Phase Select Bit

Selects the SPI clock phase. (Default: 0)
Sets the data transfer timing together with CPOL (D2). (See Figure 19.7.1.)

D2 CPOL: SPI Clock Polarity Select Bit

Selects the SPI clock polarity.
1 (R/W): Active Low
0 (R/W): Active High (default)

Sets the data transfer timing together with CPHA (D3). (See Figure 19.7.1.)

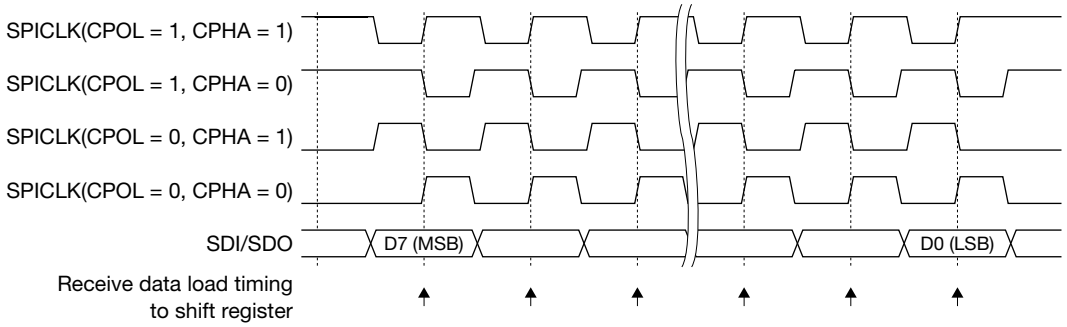


Figure 19.7.1: Clock and data transfer timing

D1 **MSSL: Master/Slave Mode Select Bit**

Sets the SPI module to Master or Slave mode.
1 (R/W): Master mode
0 (R/W): Slave mode (default)

Setting MSSL to 1 selects Master mode; setting it to 0 selects Slave mode. Master mode performs data transfer with the clock generated by the 16-bit timer Ch.1. In Slave mode, data is transferred by inputting the clock from the master device.

D0 **SPEN: SPI Enable Bit**

Permits or prohibits SPI module operation.
1 (R/W): Permitted
0 (R/W): Prohibited (default)

Setting SPEN to 1 starts the SPI module operation, enabling data transfer.
Setting SPEN to 0 stops the SPI module operation.

Note: The SPEN bit should be set to 0 before setting the CPHA, CPOL, and MSSL bits.

19.8 Precautions

- Do not access the SPI_CTL register (0x4326) while the SPBY flag (D2/SPI_ST register) is set to 1, or the SPRBF flag (D1/SPI_ST register) is set to 1 (while sending or receiving data).
  - * **SPBSY**: Transfer Busy Flag in the SPI Status (SPI_ST) Register (D2/0x4320)
  - * **SPRBF**: Receive Data Buffer Full Flag in the SPI Status (SPI_ST) Register (D1/0x4320)
- Do not gain write access to read registers (the SPI_ST and SPI_RXD registers) while sending/receiving data via SPI.

20 I²C Master (I²CM)

20.1 I²C Master Configuration

The S1C17003 incorporates an I²C bus interface module for high-speed synchronized serial communications. The I²C master module operates as a master device (as single master only) using the clock fed from the 16-bit timer Ch.2. It supports standard (100 kbps) and fast (400 kbps) modes as well as 7-bit/10-bit slave address mode. It incorporates a noise filter function to help improve the reliability of data transfers.
This module is capable of generating two different types of interrupts (transmit buffer empty and receive buffer full interrupts) for easy and continuous processing of serial data transfers with interrupts.
Figure 20.1.1 shows the I²C master module configuration.

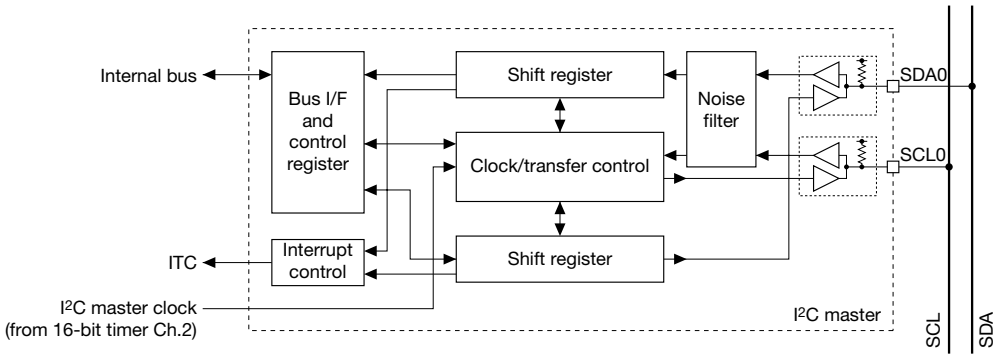


Figure 20.1.1: I²C master module configuration

20.2 I²C Master Input/Output Pins

Table 20.2.1 lists the I²C master pins.

Table 20.2.1: I²C master pin list

Pin name	I/O	Qty	Function
SDA0 (P32 or P34)	I/O	1	I²C master data input/output pin Inputs serial data from the I²C bus. Also outputs serial data to the I²C bus.
SCL0 (P31 or P33)	I/O	1	I²C master clock input/output pin Inputs SCL line status. Also outputs a serial clock.

I²C master input/output pins (SDA0 and SCL0) are shared with general purpose input/output pins (P32 and P31, or P34 and P33), and initially set as general purpose input/output pins. To use them as I²C master input/output pins, the P3_PMUX register must be set to change the function. Set the following control bit to 1 to switch the pins function to I²C master mode. Only 1 channel of I²C master is included. Therefore either SDA0(P32)/SCL0(P31) or SDA0(P34)/ SCL0(P33) combination must be selected.

P32 → SDA0

∗ **P32MUX**: P32 Port Function Select Bit in the P3 Port Function Select (P3_PMUX) Register (D5-4/0x52a6)

P31 → SCL0

∗ **P31MUX**: P31 Port Function Select Bit in the P3 Port Function Select (P3_PMUX) Register (D3-2/0x52a6)

P34 → SDA0

∗ **P34MUX**: P34 Port Function Select Bit in the P3 Port Function Select (P3_PMUX) Register (D1-0/0x52a7)

P33 → SCL0

∗ **P33MUX**: P33 Port Function Select Bit in the P3 Port Function Select (P3_PMUX) Register (D3-2/0x52a6)

For detailed information on pin function switching, refer to “10.2 Input/Output Pin Function Selection (Port MUX).”

20.3 I²C Master Clock

The I²C master module uses the internal clock output by the 16-bit timer Ch.2 as the synchronizing clock. This clock is output from the SCL0 pin to the slave device while also driving the shift register. The clock should be programmed to output a signal matching the transfer rate from the 16-bit timer Ch.2. For more information on 16-bit timer control, refer to "11 16-bit Timer (T16)."

If the I²C master module communicates with a slave device which has clock stretching, Transfer rates are limited up to 50 kbits/s in the Standard-mode, up to 200 kbits in the Fast-mode.

The I²C master module does not function as a slave device. The SCL0 input pin is used to check the I²C bus SCL signal status. It is not used for synchronization clock input.

20.4 Settings Before Data Transfer

The I²C master module includes an optional noise filter function that can be selected via the application program.

Noise filter function

The I²C master module incorporates a function for filtering noise from the SDA0 and SCL0 pin input signals. This function is enabled by setting NSERM (D4/I2C_CTL register) to 1.
Note that using this function requires setting the I²C master clock (16-bit timer Ch.2 output clock) frequency to 1/6 or less of PCLK.

* **NSERM**: Noise Remove On/Off Bit in the I²C Control (I2C_CTL) Register (D4/0x4342)

20.5 Data Transfer Control

Make the following settings before starting data transfers.
(1) Set the 16-bit timer Ch.2 to output the I²C master clock. (See Section 11.)
(2) Select the option function. (See section 20.4.)
(3) Set the interrupt conditions to use I²C master interrupts. (See Section 20.6.)

Note: Make sure the I²C module is halted (when I2CEN/I2C_EN register = 0) before changing the above settings.

∗ **I2CEN**: I²C Enable Bit in the I²C Enable (I2C_EN) Register (D0/0x4340)

Permitting data transfers

Set the I2CEN (D0/I2C_EN register) to 1 to permit I²C operations. This enables I²C master transfers and permits clock input/output.

Note: Do not set I2CEN to 0 when the I²C master module is transferring data.

Data transfer start

To start data transfers, the I²C master (this module) must generate the start condition. The slave address is then sent to establish communications.

(1) Generate start condition

The start condition applies when the SCL line is maintained at High and the SDA line is maintained at Low.

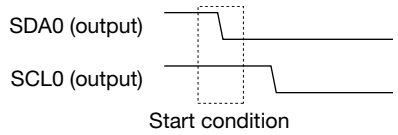


Figure 20.5.1: Start condition

The start condition is generated by setting STRT (D0/I2C_CTL register) to 1.

∗ **STRT**: Start Control Bit in the I²C Control (I2C_CTL) Register (D0/0x4342)

STRT is automatically reset to 0 once the start condition is generated. The I²C bus is busy from this point on.

(2) Slave address transmission

Once the start condition has been generated, the I²C master (this module) sends a bit indicating the slave address and transfer direction for communications. I²C slave addresses are either 7-bit or 10-bit. This module uses an 8-bit transfer data register to send the slave address and transfer direction bit, enabling single transfers in 7-bit address mode. In 10-bit mode, data is sent twice under software control. Figure 20.5.2 gives the configuration of the address data.

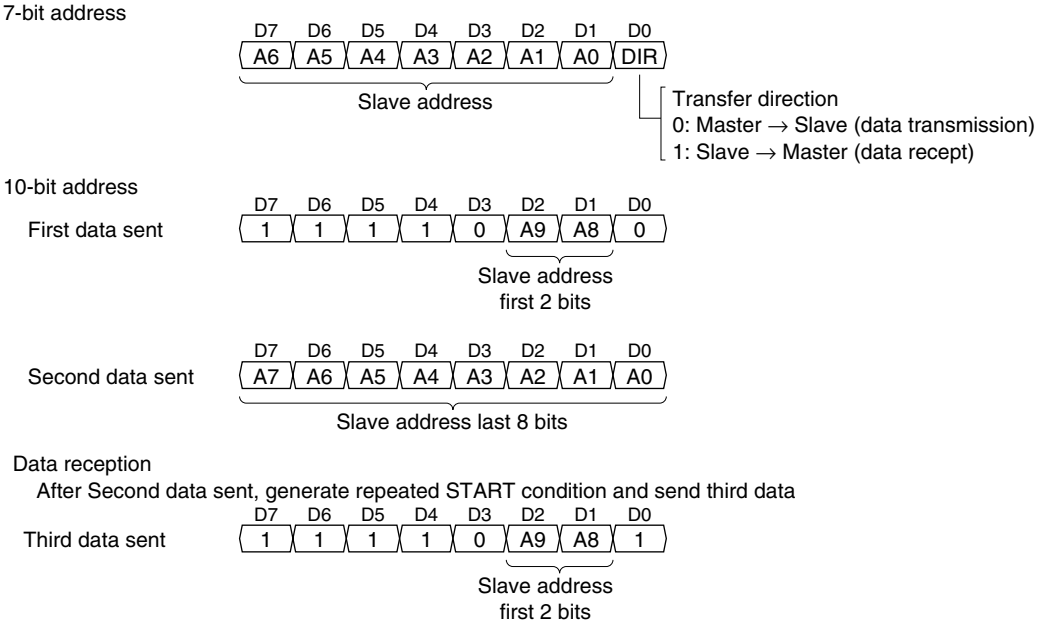


Figure 20.5.2: Slave address and transmission data specifying transfer direction

Transfer direction indicates the data transfer direction after the slave address. This is set to 0 when sending data from the master to the slave and to 1 when receiving data from the slave.
To send a slave address, set the transmission address to RTDT[7:0] (D[7:0]/I2C_DAT register). At the same time, set the TXE (D9/I2C_DAT register) transmitting the address to 1.

- **RTDT[7:0]**: Receive/Transmit Data Bits in the I²C Data (I2C_DAT) Register (D[7:0]/0x4344)
- **TXE**: Transmit Execution Bit in the I2C Data (I2C_DAT) Register (D9/0x4344)

After the slave address has been output, data can be sent and received as many times as required. Data must be sent or received according to the transfer direction set together with the slave address.

Data transmission control

The procedure for transmitting data is described below. Data transmission is performed by the same procedure as for slave address transmission.
To send byte data, set the transmission data to RTDT[7:0] (D[7:0]/I2C_DAT register). Set TXE (D9/I2C_DAT register) to 1 to transmit 1 byte.

When TXE is set to 1, the I²C master module begins data transmission in sync with the clock. If the previous data is currently being transmitted, data transmission starts after this has been completed.
The I²C master module first transfers the data written to the shift register, then starts outputting the clock from SCL0. Resetting TXE to 0 at this point generates an interrupt, enabling the subsequent transmission data and TXE to be reset.
The data bits in the shift register are shifted in sequence at the clock falling edge and output via the SDA0 pin with the MSB leading.
The I²C master module outputs 9 clocks with each data transmission. In the 9th clock cycle, an ACK or NAK is received from the slave device with the SDA0 signal as high impedance.
The slave device returns ACK(0) to the master if the data is received. If the data is not received, SDA is not pulled down, which the I²C master module interprets to mean an NAK(1) (transmission failed).

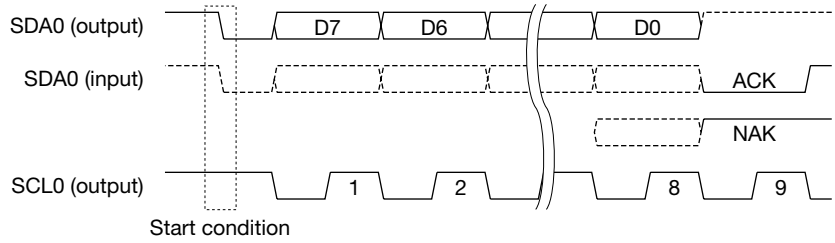


Figure 20.5.3: ACK and NAK

The I²C master module includes two status bits, TBUSY (D8/I2C_CTL register) and RTACK (D8/I2C_DAT register), for transmission control.

* **TBUSY**: Transmit Busy Flag in the I²C Control (I2C_CTL) Register (D8/0x4342)
* **RTACK**: Receive/Transmit ACK Bit in the I²C Data (I2C_DAT) Register (D8/0x4344)

The TBUSY flag indicates the data transmission status. This flag becomes 1 when transmission starts (including slave address transmission) and reverts to 0 once data transmission ends.
Inspect the flag to check whether the I²C master module is currently transmitting or at standby.
The RTACK bit indicates whether or not the slave device returned an ACK for the previous transmission. RTACK is 0 if an ACK was returned and 1 if ACK was not returned.

Data receipt control

The procedure for receiving data is described below. To receive data, the slave address must be sent with the transfer direction bit set to 1.
To receive data, set RXE (D10/I2C_DAT register) to 1 for receiving 1 byte.
TXE (D9/I2C_DAT register) is set to 1 when sending the slave address, but RXE can also be set to 1 at the same time. If both TXE and RXE are set to 1, TXE takes priority.

* **RXE**: Receive Execution Bit in the I²C Data (I2C_DAT) Register (D10/0x4344)

When the RXE bit is set to 1, allowing receiving to start, the I²C master module starts outputting the clock from the SCL0 pin with the SDA line at high impedance. The data is shifted into the shift register with the clock pulses, with the MSB leading.
RXE is reset to 0 when D7 is loaded.
The receive data is loaded to RTDT[7:0] once the 8-bit data has been received in the shift register. The I²C master module includes two status bits for receive control: RBRDY (D11/I2C_DAT register) and RBUSY (D9/I2C_CTL register).

* **RBRDY**: Receive Buffer Ready Bit in the I²C Data (I2C_DAT) Register (D11/0x4344)
* **RBUSY**: Receive Busy Flag in the I²C Control (I2C_CTL) Register (D9/0x4342)

The RBRDY flag indicates the receive data status. This flag becomes 1 when the data received in the shift register is loaded to RTDT[7:0] and reverts to 0 when the receive data is read out from RTDT[7:0]. Interrupts can also be generated once the flag value becomes 1.
The RBUSY flag indicates the receiving operation status. This flag is 1 when receiving starts and reverts to 0 when the data is received. It also reverts to 0 for the Wait state. Inspect the flag to determine whether the I²C master module is currently receiving or in standby.
To wait for reception using polling, follow the procedures given below using the RBUSY flag.
Interrupts to the CPU are prohibited because polling accurately determines the two state transitions 3 and 4.

1. Prohibits CPU interrupts using di instruction.
2. Writes 1 to RXE to prepare for receiving.
3. Waits for RBUSY to become 1 (reception start).
4. Waits for RBUSY to become 0 (reception end).
5. Reads out RTDT (received data).
6. Returns to CPU interrupt permitted state using ei instruction.

The I²C master module outputs 9 clocks with each data receipt. In the 9th clock cycle, an ACK or NAK is sent to the slave from the SDA0 pin. The bit state sent can be set in RTACK (D8/I2C_DAT register). To send ACK, set RTACK to 0. To send NAK, set RTACK to 1.

Data transfer end (Stop condition generation)

To end data transfers after all data has been transferred, the I²C master (this module) must generate a stop condition. This stop condition applies when the SCL line is maintained at High and the SDA line changes from Low to High.

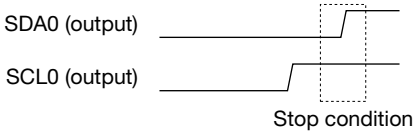


Figure 20.5.4: Stop condition

The stop condition is generated by setting STP (D1/I2C_CTL register) to 1.

∗ **STP**: Stop Control Bit in the I²C Control (I2C_CTL) Register (D1/0x4342)

When STP is set to 1, the I²C master module switches the SDA line from Low to High and generates a stop condition while maintaining the I²C bus SCL line at High. The I2C bus subsequently switches to free state.
When transmission or reception ends, TBUSY or RBUSY is cleared. Then, after a period longer than the 1/4 cycle of I²C clock, STP can set to 1. If I²C master communicate with slave device which has clock stretch function, STP can not be set to 1 until slave device finishes clock stretching. For this case, wait time is necessary before STP is set to 1. STP is reset to 0 when the stop condition is generated.

Continuing data transfer (Repeated start condition generation)

To make it possible to continue with a different data transfer after data transfer completion, the I²C master (this module) can generate a repeated start condition.

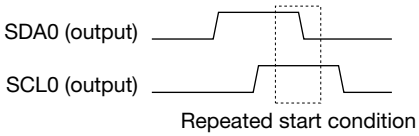


Figure 20.5.5: Repeated start condition

The repeated start condition is generated by setting STRT (D0/I2C_CTL register) to 1 when the I²C bus is busy.

∗ **STRT**: Start Control Bit in the I2C Control (I2C_CTL) Register (D0/0x4342)

STRT is automatically reset to 0 once the repeated start condition is generated. Slave address transmission is subsequently possible with the I²C bus remaining in the busy state.

Disabling data transfer

After STOP condition generation, write 0 to I2CMEN to disable data transfers. For this case, the STP may be polled to determine the end of STOP condition generation when it is cleared.
If I2CEN is set to 0 when I²C bus is busy, SCL0, SDA0 output level nor no information is guaranteed.

Timing chart

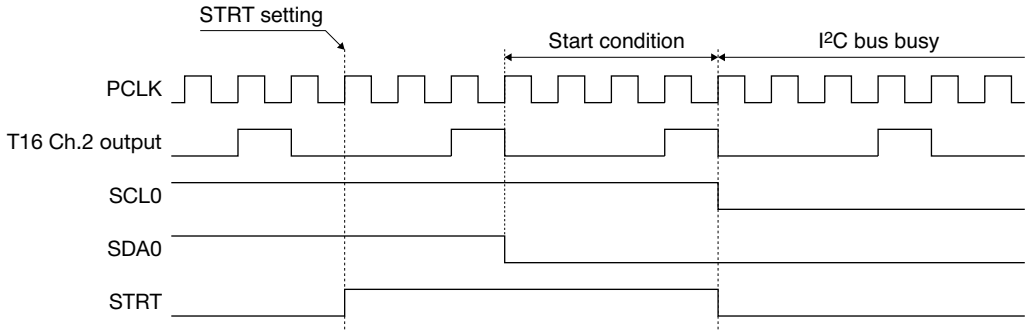


Figure 20.5.6: Start condition generation

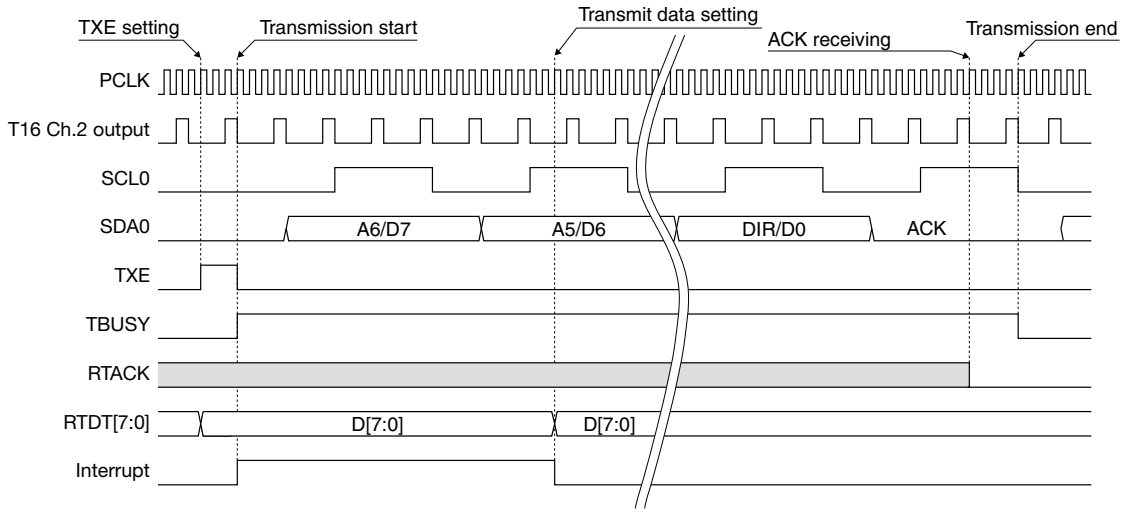


Figure 20.5.7: Slave address transmission/data transmission

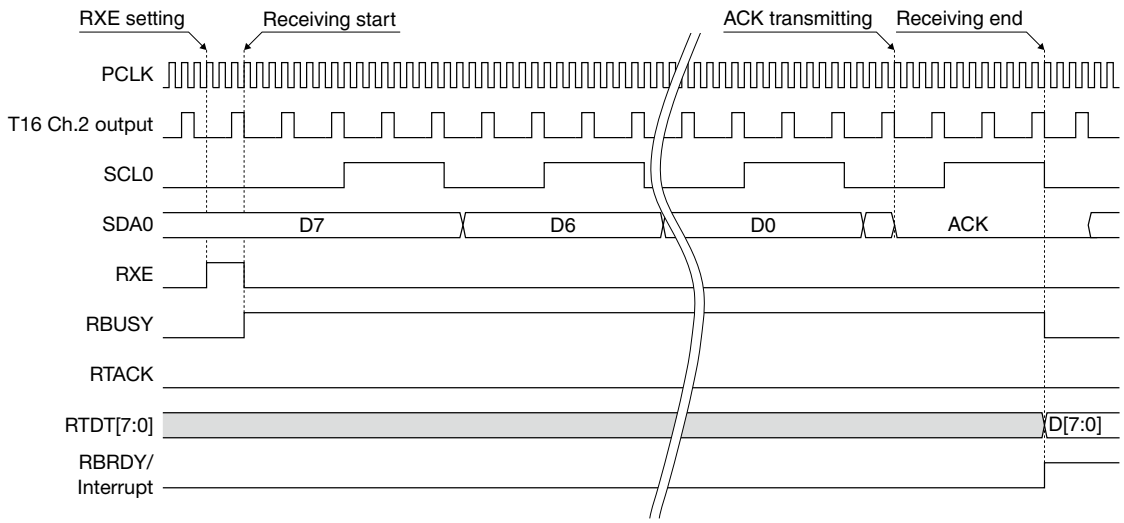


Figure 20.5.8: Data receiving

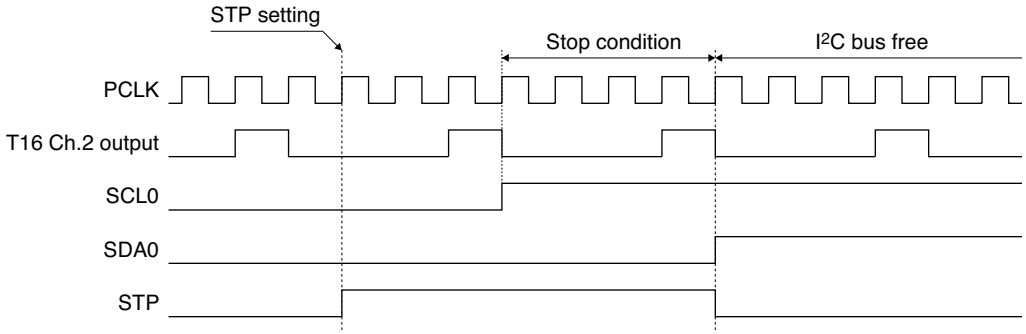


Figure 20.5.9: Stop condition generation

20.6 I²C Master Interrupts

The I²C master module includes a function for generating the following two different interrupt types.
- Transmit buffer empty interrupt
- Receive buffer full interrupt

The I²C master module outputs one interrupt signal shared by the two above interrupt factor types to the interrupt controller (ITC).

Transmit buffer empty interrupt

To use this interrupt, set TINTE (D0/I2C_ICTL register) to 1. If TINTE is set to 0 (default), interrupt requests for this factor will not be sent to the ITC.

∗ **TINTE**: Transmit Interrupt Enable Bit in the I²C Interrupt Control (I2C_ICTL) Register (D0/0x4346)

If transmit buffer empty interrupts are permitted (TINTE = 1), an interrupt request pulse is output to the ITC as soon as the transmit data set in RTDT[7:0] (D[7:0]/I2C_DAT register) is transferred to the shift register.

∗ **RTDT[7:0]**: Receive/Transmit Data Bits in the I²C Data (I2C_DAT) Register (D[7:0]/0x4344)

An interrupt occurs if other interrupt conditions are satisfied.
Transmit buffer empty interrupt occurs when the data was only sent.

• The clear method of transmit buffer empty flag
  Write the data to RTDT/I2CM_DAT.
  When TXE/I2CM_DAT is 0, the data doesn't send and the flag is only cleared.

Receive buffer full interrupt

To use this interrupt, set RINTE (D1/I2C_ICTL register) to 1. If RINTE is set to 0 (default), interrupt requests for this factor will not be sent to the ITC.

∗ **RINTE**: Receive Interrupt Enable Bit in the I²C Interrupt Control (I2C_ICTL) Register (D1/0x4346)

If receive buffer full interrupts are permitted (RINTE = 1), an interrupt request pulse is output to the ITC as soon as the data received in the shift register is loaded to RTDT[7:0].

An interrupt occurs if other interrupt conditions are met.
Receive buffer full interrupt occurs when the data was only received.

• The clear method of receive buffer full flag
  Read the data from RTDT/I2CM_DAT.

Note: When I2CM interrupt occurs, decide the transmit buffer empty interrupt or the receive buffer full interrupt by the program sequence of the I²C master. There're not registers to decide which interrupt occurred.

Interrupt vectors

The I²C master module interrupt vector numbers and vector addresses are as listed below.

Vector number: 19 (0x13)
Vector address: TTBR + 0x4c

Other interrupt settings

The ITC allows the priority of I²C master module interrupts to be set between level 0 (the default value) and level 7. To generate actual interrupts, the PSR (S1C17 core internal processor status register) IE (interrupt enable) bit must be set to 1.

For more information on interrupt processing, see "6 Interrupt Controller (ITC)."

20.7 Control Register Details

Table 20.7.1: I²CM register list

Address	Register name		Function
0x4340	I2C_EN	I²C Enable Register	I²C master module enable
0x4342	I2C_CTL	I²C Control Register	I²C master control and transfer status display
0x4344	I2C_DAT	I²C Data Register	Transfer data
0x4346	I2C_ICTL	I²C Interrupt Control Register	I²C master interrupt control

The I²C master module registers are described in detail below. These are 16-bit registers.

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

0x4340: I²C Enable Register (I2C_EN)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
I²C Enable Register (I2C_EN)	**0x4340** (16 bits)	D15–1	**–**	reserved	–	–	–	0 when being read.
		D0	**I2CEN**	I²C enable	1 Enable 0 Disable	0	R/W	

D[15:1] Reserved

D0 I2CEN: I²C Enable Bit

Permits or prohibits I²C master module operation.
1 (R/W): Permitted
0 (R/W): Prohibited (default)

Setting I2CEN to 1 starts the I²C master module operation, enabling data transfer.
Setting I2CEN to 0 stops the I²C master module operation.

0x4342: I²C Control Register (I2C_CTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
I²C Control Register (I2C_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

Indicates I²C master module receive operation status.

1 (R): Busy

0 (R): Idle (Default)

The RBUSY bit is set to 1 when I²C master module has started data reception, and the value is retained during the reception. When the receive process has been completed, the RBUSY bit is cleared to 0.

D8 TBUSY: Transmit Busy Flag

Indicates I²C master transmit operation status.

1 (R): Busy

0 (R): Idle (Default)

The RBUSY bit is set to 1 when I²C master module has started data transmissin, and the value is retained during the transmission. When the transmit process has been completed, the RBUSY bit is cleared to 0.

D[7:5] Reserved

D4 NSERM: Noise Remove On/Off Bit

Turns the noise filter function on or off.

1 (R/W): On

0 (R/W): Off (default)

The I²C master module incorporates a function for filtering noise from the SDA0 and SCL0 pin input signals. This function is enabled by setting NSERM to 1.

Note that using this function requires setting the I²C master clock (16-bit timer Ch.2 output clock) frequency to 1/6 or less of PCLK.

D[3:2] Reserved

D1 STP: Stop Control Bit

Generates the stop condition.

1 (R/W): Stop condition generated

0 (R/W): Disabled (default)

Setting the STP bit 1 makes the I²C master module generate the stop condition by switching the SDA line from Low to High while keeping the I²CM bus SCL line in High state. The I²C bus is in free status in the subsequent processes.

When transmission or reception ends, TBUSY or RBUSY is cleared. Then, after a period longer than the 1/4 cycle of I²C clock, STP can set to 1.

The generation of the stop condition automatically resets the STP bit to 0.

D0 STRT: Start Control Bit

Generates the start condition.
1 (R/W): Start condition generated
0 (R/W): Disabled (default)

With STRT set at 1, the I²C master module generates the start condition by changing the SDA line to Low while maintaining the I²C bus SCL line at High. The I²C bus subsequently becomes busy.
Set STRT to 1 when data transfer starts.

Registers should be set in the following sequence to generate start conditions:
1. Set the slave address in RTDT[7:0] (D[7:0]/I2C_DAT register). (First transmission data for 10-bit addresses, see Figure 20.5.2)
2. Set TXE (D9/I2C_DAT register) to 1.
3. Set STRT to 1.

STRT is automatically reset to 0 once the start condition is generated.

0x4344: I²C Data Register (I2C_DAT)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
I²C Data Register (I2C_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 Flag

Indicates the receive buffer status.

1 (R): Receive data ready

0 (R): Receive data empty (default)

The RBRDY flag is turned to 1 when data received by a shift register is loaded to RTDT[7:0] (D[7:0]), and returned to 0 when the received data is read from RTDT[7:0]. An interrupt can be generated once this flag is turned to 1.

Note: Use the RBUSY flag to wait for reception in the polling process. The RBRDY flag cannot be used for the polling standby. Refer to the description related to data reception control.

D10 RXE: Receive Execution Bit

Receives 1 byte of data.

1 (R/W): Data receipt start

0 (R/W): Disabled (default)

Setting RXE to 1 and TXE (D9) to 0 starts receiving for 1 byte of data. RXE can be set to 1 for subsequent receipt, even if the slave address is being sent or data is being received. RXE is reset to 0 as soon as D6 is loaded to the shift register.

D9 TXE: Transmit Execution Bit

Transmits 1 byte of data.

1 (R/W): Data transmission start

0 (R/W): Disabled (default)

Transmission is started by setting the transmission data to RTDT[7:0] (D[7:0]) and writing 1 to TXE. TXE can be set to 1 for subsequent transmission, even if the slave address or data is being sent. TXE is reset to 0 as soon as the data set in RTDT[7:0] is transferred to the shift register.

D8 RTACK: Receive/Transmit ACK Bit

When transmitting data

Indicates the response bit status.

1 (R/W): Error (NAK)

0 (R/W): ACK (default)

RTACK becomes 0 when ACK is returned from the slave after 1 byte of data is sent, indicating that the slave has received the data correctly. If RTACK is 1, the slave device is not operating or the data was not received correctly.

When receiving data

Sets the response bit sent to the slave.

1 (R/W): Error (NAK)

0 (R/W): ACK (default)

To return an ACK after data has been received, RTACK should be set to 0 before the I²C master module sends the response bit.

To return an NAK, set RTACK to 1.

D[7:0] RTDT[7:0]: Receive/Transmit Data Bits

When sending data

Set the transmission data. (Default: 0x0)

Data transmission is started by setting TXE (D9) to 1. If a slave address or data is currently being transmitted, transmission begins once the previous transmission is completed. Serial converted data is output from the SDA0 pin with MSB leading and bits set to 0 as Low level.

A transmit buffer empty interrupt factor is generated as soon as the data written to this register is transferred to the shift register, after which the subsequent transmission data can be written.

When receiving data

Read the receive data. (Default: 0x0)

Data receipt is started by setting RXE (D10) to 1. If a slave address is currently being transmitted or data is currently being received, the new receipt starts once the previous data has been transferred. The RBRDY flag (D11) is set and a receive buffer full interrupt factor generated as soon as receipt is complete and the shift register data is transferred to this register. Data can then be read until the subsequent data has been received. If the subsequent data is received before this register is read out, the contents are overwritten by the most recent received data.

Serial data input from the SDA0 pin with MSB leading is converted to parallel, with the High level bit set to 1 and the Low level bit set to 0, then loaded to this register.

0x4346: I²C Interrupt Control Register (I2C_ICTL)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
I²C Interrupt Control Register (I2C_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

Permits or prohibits receive buffer full I²C master module interrupts.

1 (R/W): Permitted

0 (R/W): Prohibited (default)

Setting RINTE to 1 permits the output of I²C master interrupt requests to the ITC due to a receive data buffer full. These interrupt requests are generated when the data received in the shift register is transferred to RTDT[7:0] (D[7:0]/I2C_DAT register) (when receipt is complete).

I²C master interrupts are not generated by receive data buffer full if RINTE is set to 0.

D0 TINTE: Transmit Interrupt Enable Bit

Permits or prohibits transmit buffer empty I²C master module interrupts.

1 (R/W): Permitted

0 (R/W): Prohibited (default)

Setting TINTE to 1 permits the output of I²C master module interrupt requests to the ITC due to a transmit buffer empty. These interrupt requests are generated when the data written to RTDT[7:0] (D[7:0]/I2C_DAT register) is transferred to the shift register.

I²C master interrupts are not generated by transmit buffer empty if TINTE is set to 0.

21 I²C Slave (I²CS)

21.1 Configuration of the I²C Slave Module

The S1C17003 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 21.1.1 shows the structure of the I²C slave module.

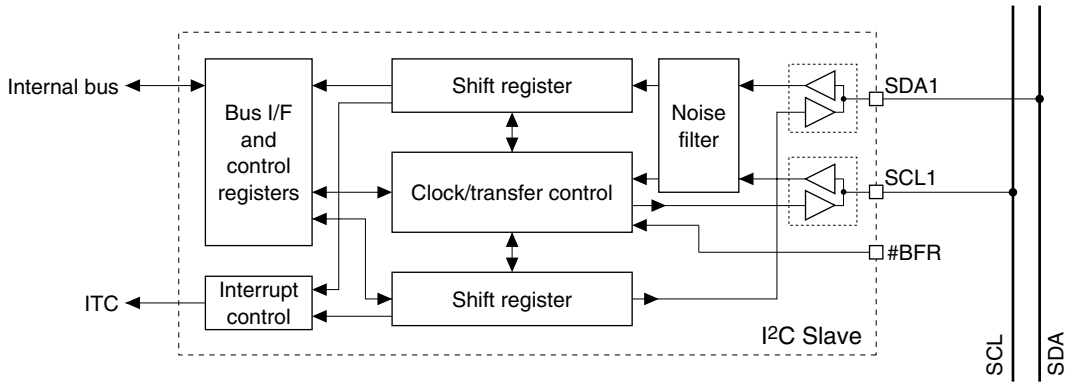


Figure 21.1.1 Structure of I²C Slave Module

Note: The I²C slave module does not support general call address and 10-bit address mode.

21.2 I²C Slave I/O Pins

Table 21.2.1 lists the I²C slave pins.

Table 21.2.1 List of I²C Slave Pins

Pin name	I/O	Size	Function
SDA1 (P34)	I/O	1	I²C slave data input/output pin This pin inputs serial data from the I²C bus and outputs serial data to the I²C bus.
SCL1 (P33)	I/O	1	I²C slave clock input/output pin This pin inputs the SCL line status and outputs low level to the I²C bus when clock stretch.
#BFR (P35)	I	1	I²C slave bus free request input pin A low pulse input to this pin requests the I²C slave to release the I²C bus. When the bus free request input has been enabled with software, a low pulse initializes the communication process of the I²C slave module and sets the SDA1 and SCL1 pins to high impedance state.

The I²C slave input/output pins (SDA1, SCL1, and #BFR) 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 I²C slave, the pin functions must be switched using the Port Function Select Register.
For details on switching pin function, "10.2 Input/Output Pin Function Selection (Port MUX)"

P34 → SDA1
- **P34MUX**: P34 Port Function Select Bit in the P3 Port Function Select (P3_PMUX) Register (D1-0/0x52a7)

P33 → SCL1
- **P33MUX**: P33 Port Function Select Bit in the P3 Port Function Select (P3_PMUX) Register (D7-6/0x52a6)

P35 → #BFR
- **P35MUX**: P35 Port Function Select Bit in the P3 Port Function Select (P3_PMUX) Register (D3-2/0x52a7)

21.3 I²C Slave Clock

The I²C slave module inputs via the SCL1 pin a clock that has been ouput from the external I²C master device, and use the clock to send/receive data.

The I²C slave module also uses the system clock (PCLK) for its operations. The PCLK frequency must be set eight-times or higher than the SCL1 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 “21.4.3 Optional Functions” for details.

21.4 Initializing the I²C Slave

21.4.1 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 SDA1 and SCL1 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 #BFR pin

The I²C slave module can accept bus free requests using the #BFR 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 is required. Two PCLK cycles or more pulse width is recommended) input to the #BFR pin sets BFREQ (D4/I2CS_STAT register) to 1. This initializes the I²C slave communication process and puts the SDA1 and SCL1 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.

21.4.2 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).

21.4.3 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 SCL1 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 SCL1 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: When I²C slave module is slave transceiver mode, the data setup time with clock stretching (= the period from outputting the MSB of SDATA[7:0] on I2CS_SDA pin to ending I2CS_SCL Low hold) depends on the PCLK frequency.

∗ **CLKSTR_EN**: Clock Stretch On/Off Bit in the I²C Slave Control (I2CS_CTL) Register (D3/0x4366)

Asynchronous address detection

The I²C 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 I²C slave address sent from the master in this status.
The asynchronous address detection function in this module is disabled by default. When using the asynchronous address detection function, set ASDET_EN (D1/I2CS_CTL register) to 1.

∗ **ASDET_EN**: Async. Address Detection On/Off Bit in the I²C 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 I²C slave module when the asynchronous address detection function has been enabled, 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 reset ASDET_EN to 0 in the interrupt handler routine. Data transfer will be able to resume normally after the master retries transmission. After the master generates a STOP condition to put the I²C bus into free status, the asynchronous address detection function can be enabled again to lower the operating speed.

Notes:
- When the asynchronous address detection function is enabled, the I²C 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 I²C slave module contains a function to remove noise from the SDA1 and SCL1 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 I²C Slave Control (I2CS_CTL) Register (D2/0x4366)

21.5 Data Transmit/Receive Control

Before starting data transfer, set up the conditions by the procedure below.
(1) Initialize the I²C slave module. See Section 21.4.
(2) Set up the interrupt conditions if the I²C slave interrupt is used. See Section 21.6.

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 (SDA1 = 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).

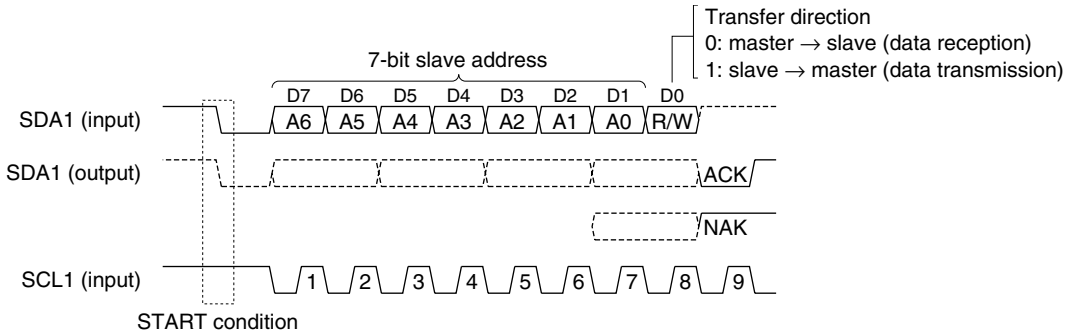


Figure 21.5.1 Receiving Slave Address and Data Direction Bit

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. STOP condition detection clears BUSY. STOP or Repeated START condition detection clears SELECTED. 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 (SCL1) 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 in 0 becomes invalid. Therefore, the 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 in 0 becomes invalid. Therefore, the 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 (SCL1) 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 SCL1 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 SDA1 pin in sync with the SCL1 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.

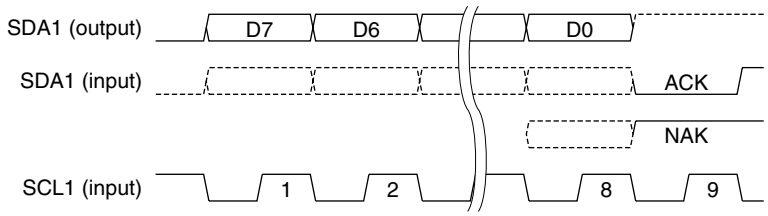


Figure 21.5.2 ACK and NAK

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 SDA1 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 SDA1 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 SDA1 line. An interrupt can be generated when DMS is set to 1, so an error handling can be performed in the interrupt handler routine. DMS is cleared by writing 1.

Note: If the I2CS module has sent back a NAK as the response to the address sent by the master when the conditions shown below are all met, the master must wait for 33 µs or more before it can send another slave address (except when the master sends the I2CS slave address again).
1. The transfer rate is set to 320 kbps or higher.
2. The asynchronous address detection function is enabled.
3. The I2CS module is placed into transfer standby state and OSC1 is used as the operating clock (PCLK).

∗ **DMS**: Output Data Mismatch Bit in the I²C Slave Status (I2CS_STAT) Register (D3/0x4368)

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 SDA1 pin in sync with the SCL1 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 (SCL1) 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 SCL1 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 (SDA1 = low) every time an 8-bit data has been received (by default setting). The response code can be changed to NAK (SDA1 = 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 (SCL1) after RXRDY has been set to 1 by receiving data just prior to one required for returning NAK.

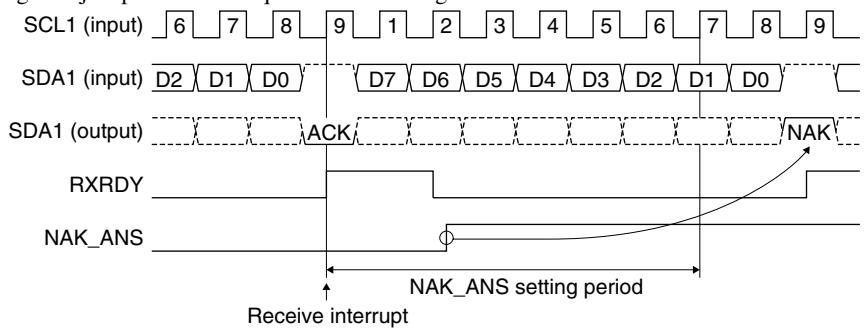


Figure 21.5.3 Setting NAK_ANS and NAK Response Timing

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 SDA line is pulled up from low to high with the SCL line held at high.

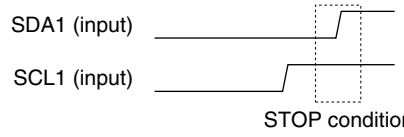


Table 21.5.4 STOP Condition

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 SDA1 and SCL1 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

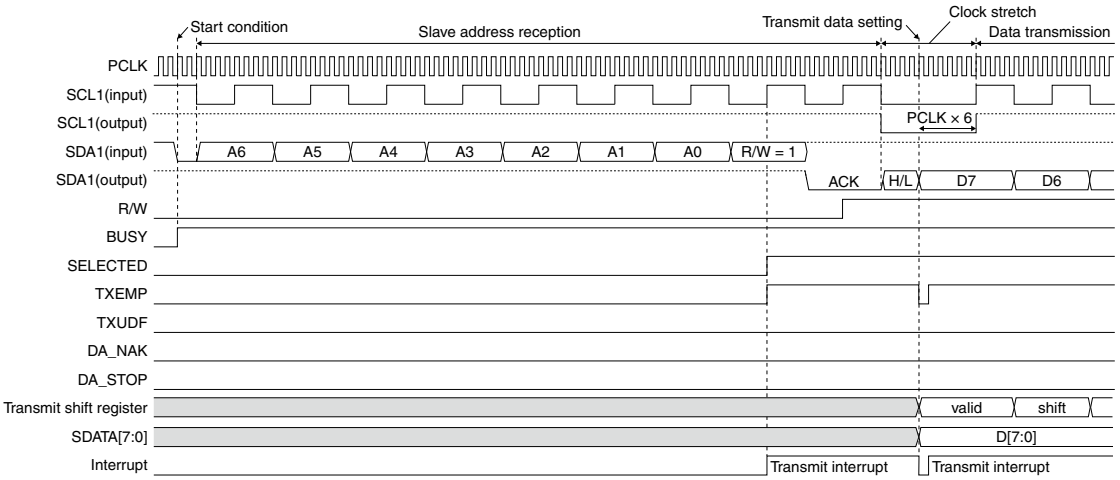


Figure 21.5.5 I²C Slave Timing Chart 1 (START condition → data transmission)

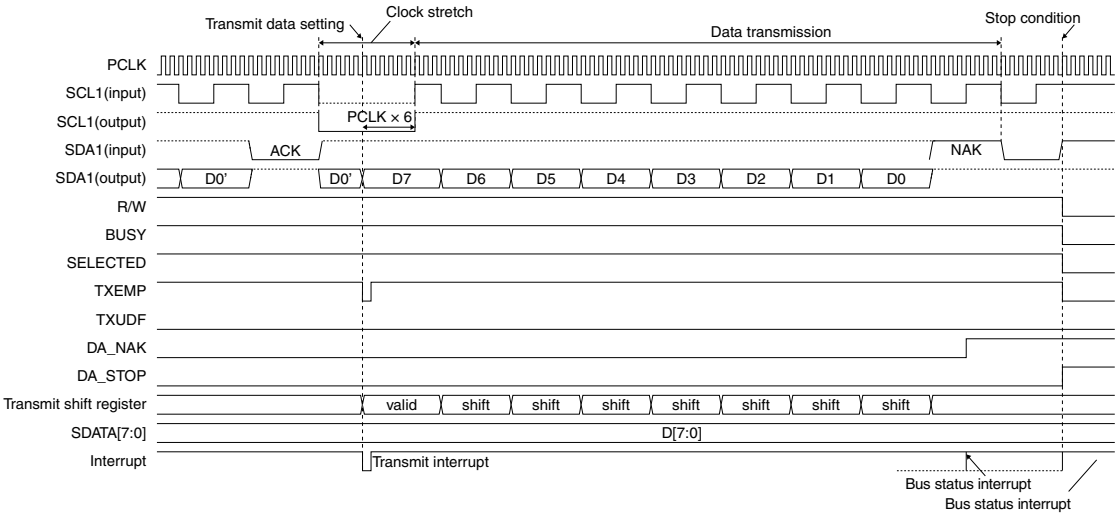


Figure 21.5.6 I²C Slave Timing Chart 2 (data transmission → STOP condition)

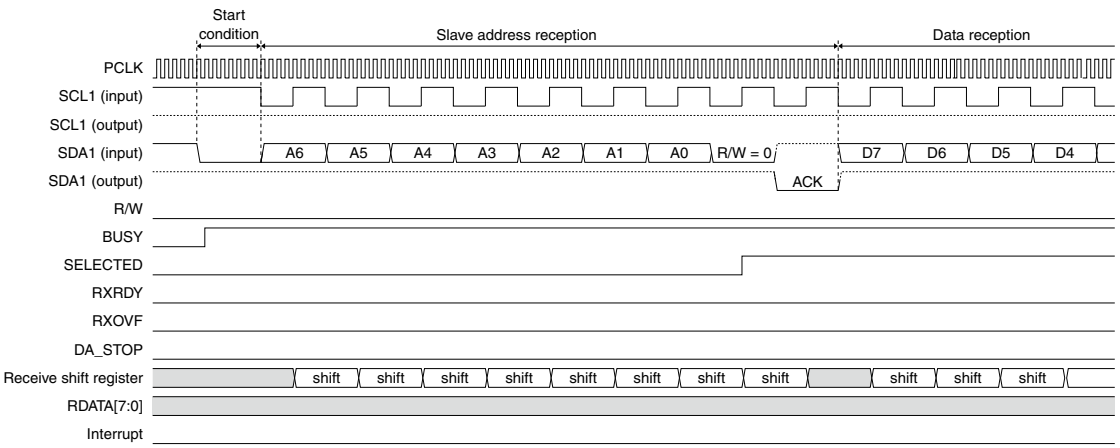


Figure 21.5.7 I²C Slave Timing Chart 3 (START condition → data reception)

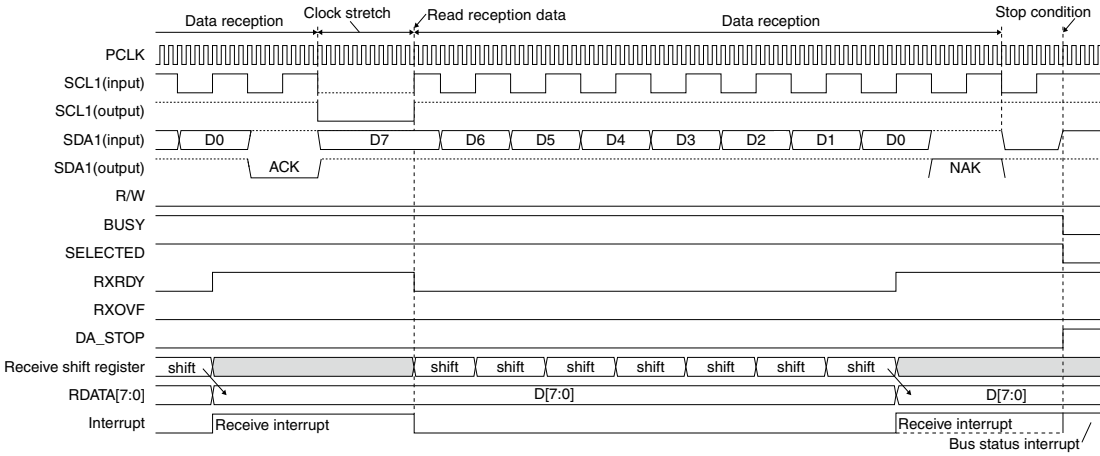


Figure21.5.8 I²C Slave Timing Chart 4 (data reception → STOP condition)

21.6 I²C Slave Interrupt

The I²C 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]**: I²C Slave Transmit Data Bits in the I²C Slave Transmit Data (I2CS_TRNS) Register (D[7:0]/0x4360)
∗ **TXEMP**: Transmit Data Empty Bit in the I²C 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 I²C 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]**: I²C Slave Receive Data Bits in the I²C Slave Receive Data (I2CS_RECV) Register (D[7:0]/0x4362)
∗ **RXRDY**: Receive Data Ready Bit in the I²C 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 I²C Slave Interrupt Control (I2CS_ICTL) Register (D1/0x436c)

Bus status interrupt

The I²C slave module provides the status bits listed below to represent the transmit/receive and I²C bus statuses (see Section 21.5 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 I²C 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 I²C 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 I²C Slave Status (I2CS_STAT) Register (D1/0x4368)
4. DMS: set to 1 when the SDA1 line status is different from transfer data
   * **DMS**: Output Data Mismatch Bit in the I²C Slave Status (I2CS_STAT) Register (D3/0x4368)

   DMA will also be set to 1 when another slave device issues ACK to this I²C slave address (when ASDET_EN (D1/I2CS_CTL register) = 0).

   Note: When the master device of the I²C bus, which has multiple slave devices connected including this IC, starts communication with another slave device, the I²C 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 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 I²C 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 21.6.1 shows the control bits for the I²C slave interrupt in the ITC.

Table 21.6.1 ITC Registers

Cause of interrupt	Interrupt level setup bits
Bus status/Transmit/receive	ILV13[2:0] (D[10:8]/ITC_ILV6)

ITC_ILV6 register (0x4312)

The ITC sends an interrupt request to the S1C17 Core.
The interrupt level setup bits set the interrupt level (0 to 7) of the I²C slave interrupt.
An interrupt request to the S1C17 Core is accepted only when all the conditions described below are met.

- The interrupt enable bit of I²C slave module (peripheral module) 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 "6. Interrupt Controller (ITC)."

Interrupt vector

The following shows the vector number and vector address for the I²C slave interrupt:

Table 21.6.2 I²C Slave Interrupt Vectors

Cause of interrupt	Vector number	Vector address
Bus status/Transmit/receive	17 (0x11)	TTBR + 0x44

21.7 Details of Control Registers

Table 21.7.1 List of I²C Slave Registers

Address	Register name		Function
0x4360	I2CS_TRNS	I²C Slave Transmit Data Write Register	I²C slave transmit data
0x4362	I2CS_RECV	I²C Slave Receive Data Read Register	I²C slave receive data
0x4364	I2CS_SADRS	I²C Slave Address Setup Register	Sets the I²C slave address.
0x4366	I2CS_CTL	I²C Slave Control Register	Controls the I²C slave module.
0x4368	I2CS_STAT	I²C Slave Status Register	Indicates the I²C slave bus status.
0x436a	I2CS_ASTAT	I²C Slave Access Status Register	Indicates the I²C slave access status.
0x436c	I2CS_ICTL	I²C Slave Interrupt Control Register	Controls the I²C slave interrupt.

The following describes each I²C 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 SDA1 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 (SCL1) 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 (SCL1) 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 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.

0x4362: I²C Slave Receive Data Register (I2CS_RECV)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
I²C Slave Receive Data Register (I2CS_RECV)	**0x4362** (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	**RDATA[7:0]**	I²C slave receive data	0–0xff	0x0	R	

D[15:8] Reserved

D[7:0] RDATA[7:0]: I²C Slave Receive Data Bits

The received data can be read from this register. (Default: 0x0)

The serial data input from the SDA1 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 I²C slave clock (SCL1) 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: I²C Slave Control Register (I2CS_CTL)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
I²C Slave Control Register (I2CS_CTL)	0x4366 (16 bits)	D15–9	–	reserved	–				–	–	0 when being read.
		D8	TBUF_CLR	I2CS_TRNS register clear	1	Clear state	0	Normal	0	R/W	
		D7	I2C_EN	I²C slave enable	1	Enable	0	Disable	0	R/W	
		D6	SOFTRESET	Software reset	1	Reset	0	Cancel	0	R/W	
		D5	NAK_ANS	NAK answer	1	NAK	0	ACK	0	R/W	
		D4	BFREQ_EN	Bus free request enable	1	Enable	0	Disable	0	R/W	
		D3	CLKSTR_EN	Clock stretch On/Off	1	On	0	Off	0	R/W	
		D2	NF_EN	Noise filter On/Off	1	On	0	Off	0	R/W	
		D1	ASDET_EN	Async.address detection On/Off	1	On	0	Off	0	R/W	
		D0	COM_MODE	I²C slave communication mode	1	Active	0	Standby	0	R/W	NAK responsee when standby

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: I²C Slave Enable Bit

Enables/disables operation of the I²C slave module.
1 (R/W): Enable
0 (R/W): Disable (default)

When I2C_EN is set to 1, the I²C slave module is activated and data transfer is enabled.
When I2C_EN is set to 0, the I²C slave module goes off.

D6 SOFTRESET: Software Reset Bit

Resets the I²C slave module.
1 (R/W): Reset
0 (R/W): Cancel reset state (default)

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 SDA1 and SCL1 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.

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 I²C slave module sends back an ACK (SDA1 = low) or a NAK (SDA1 = Hi-Z). Either ACK or NAK should be specified using NAK_ANS within 7 cycles of the I²C slave clock (SCL1) after RXRDY has been set to 1 by receiving the previous data.

D4 BFREQ_EN: Bus Free Request Enable Bit

Enables/disables I²C bus free requests by inputting a low pulse to the #BFR pin.
1 (R/W): Enable
0 (R/W): Disable (default)

To accept I²C 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 I²C slave communication process and puts the SDA1 and SCL1 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 #BFR pin are ignored and BFREQ is not set to 1.

D3 CLKSTR_EN: Clock Stretch On/Off Bit

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

After data and ACK are transmitted or received, the slave device may issue a wait request to the master device until it is ready to transmit/receive by pulling the SCL1 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 SCL1 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 I²C slave module contains a function to remove noise from the SDA1 and SCL1 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 I²C 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 I²C slave address sent from the master in this status. This function is enabled by setting ASDET_EN to 1. If the slave address sent from the master has matched with one that has been set in this I²C slave module when the asynchronous address detection function has been enabled, 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 reset ASDET_EN to 0 in the interrupt handler routine. Data transfer will be able to resume normally after the master retries transmission. After the master generates a STOP condition to put the I²C bus into free status, the asynchronous address detection function can be enabled again to lower the operating speed.

Notes:
- When the asynchronous address detection function is enabled, the I²C signals are input without passing through the noise filter. Therefore, the slave address may not be detected in a high-noise environment.
- When the asynchronous address detection function is enabled, data transfer cannot be performed even if the PCLK frequency is eight-times or higher than the transfer rate. Be sure to disable the asynchronous address detection function during normal operation.

D0 COM_MODE: I²C Slave Communication Mode Bit

Enables/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 I²C slave operation. 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).

0x4368: I²C Slave Status Register (I2CS_STAT)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
I²C Slave Status Register (I2CS_STAT)	0x4368 (16 bits)	D15–8	–	reserved	–				–	–	0 when being read.
		D7	BSTAT	Bus status transition	1	Changed	0	Unchanged	0	R	
		D6	–	reserved	–				–	–	0 when being read.
		D5	TXUDF RXOVF	Transmit data underflow Receive data overflow	1	Occurred	0	Not occurred	0	R/W	Reset by writing 1.
		D4	BFREQ	Bus free request	1	Occurred	0	Not occurred	0	R/W	
		D3	DMS	Output data mismatch	1	Error	0	Normal	0	R/W	
		D2	ASDET	Async. address detection status	1	Detected	0	Not detected	0	R/W	
		D1	DA_NAK	NAK receive status	1	NAK	0	ACK	0	R/W	
		D0	DA_STOP	STOP condition detect	1	Detected	0	Not detected	0	R/W	

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 I²C bus free request input status.

1 (R/W): Request has been issued

0 (R/W): Request has not been issued (default)

If BFREQ_EN (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 #BFR pin sets BFREQ to 1 and the bus free request is accepted. When a bus free request is accepted, the I²C slave module initializes the I²C communication process and puts the SDA1 and SCL1 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 #BFR pin are ignored and BFREQ is not set to 1.

D3 DMS: Output Data Mismatch Bit

Represents the results of comparison between output data and SDA1 line status.
1 (R/W): Error has been occurred
0 (R/W): Error has not been occurred (default)

The SDA1 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 SDA1 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 SDA1 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 I²C bus, which has multiple slave devices connected including this IC, starts communication with another slave device, the I²C 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 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 I²C 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 I²C 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 I²C slave address sent from the master in this status. ASDET is set to 1 if the slave address of the I²C slave module is detected when the asynchronous address detection function has been enabled by setting ASDET_EN (D1/I2CS_CTL register). The I²C slave module returns a NAK to the I²C master to request for resending the slave address. At the same time, an interrupt signal is output to the ITC if the interrupt is enabled with BSTAT_IEN (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 I²C slave module is selected as the slave device (SELECTED (D1/I2CS_ASTAT register) = 1), the I²C slave module sets DA_STOP to 1. At the same time, it initializes the I²C communication process.
When DA_STOP is set to 1, an interrupt signal is output to the ITC if the interrupt is enabled with BSTAT_IEN (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: I²C Slave Access Status Register (I2CS_ASTAT)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
I²C Slave Access Status Register (I2CS_ASTAT)	0x436a (16 bits)	D15–5	–	reserved	–				–	–	0 when being read.
		D4	RXRDY	Receive data ready	1	Ready	0	Not ready	0	R	
		D3	TXEMP	Transmit data empty	1	Empty	0	Not empty	0	R	
		D2	BUSY	I²C bus status	1	Busy	0	Free	0	R	
		D1	SELECTED	I²C slave select status	1	Selected	0	Not selected	0	R	
		D0	R/W	Read/write direction	1	Output	0	Input	0	R	

D[15:5] Reserved

D4 RXRDY: Receive Data Ready Bit

Indicates that the received data is ready to read.

1 (R): Received data ready

0 (R): No received data (default)

When the received data is loaded to the I2CS_RECV register, RXRDY is set to 1. At the same time, an interrupt signal is output to the ITC if the interrupt is enabled with RXRDY_IEN (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: I²C Bus Status Bit

Indicates the I²C bus status.

1 (R): Bus busy status

0 (R): Bus free status (default)

When the I²C slave module detects a START condition or detects that the SCL1 or SDA1 signal goes low, BUSY is set to 1 to indicate that the I²C bus enters busy status. The slave select status whether this module is selected as the slave device or not does not affect the BUSY status. After BUSY is set to 1, it is reset to 0 when a STOP condition is detected.

D1 SELECTED: I²C Slave Select Status Bit

Indicates that this module is selected as the I²C slave device.

1 (R): Selected

0 (R): Not selected (default)

When the slave address that is set in this module is received, SELECTED is set to 1 to indicate that this module is selected as the I²C slave device. After SELECTED is set to 1, it is reset to 0 when a STOP condition or a Repeated START condition is detected.

D0 R/W: Read/Write Direction Bit

Represents the transfer direction bit value.

1 (R): Output (master read operation)

0 (R): Input (master write operation) (default)

The transfer direction bit value that has been received with the slave address is set to R/W. Use R/W to select the transmit- or receive-handling.

0x436c: I²C Slave Interrupt Control Register (I2CS_ICTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
I²C Slave Interrupt Control Register (I2CS_ICTL)	0x436c (16 bits)	D15–3	–	reserved	–	–	–	0 when being read.
		D2	BSTAT_IEN	Bus status interrupt enable	1 Enable 0 Disable	0	R/W	
		D1	RXRDY_IEN	Receive interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	TXEMP_IEN	Transmit interrupt enable	1 Enable 0 Disable	0	R/W	

D[15:3] Reserved

D2 BSTAT_IEN: Bus Status Interrupt Enable Bit

Enables/disables the bus status interrupt.
1 (R/W): Enable
0 (R/W): Disable (default)

When BSTAT_IEN is set to 1, I²C 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 I²C slave receive interrupt.
1 (R/W): Enable
0 (R/W): Disable (default)

When RXRDY_IEN is set to 1, I²C 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 I²C slave transmit interrupt.
1 (R/W): Enable
0 (R/W): Disable (default)

When TXEMP_IEN is set to 1, I²C 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.

21.8 Precautions

- The I²C 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 I²C 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 I²C bus, which has multiple slave devices connected including this IC, starts communication with another slave device, the I²C 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 SDA1 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 I²C 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 I²C slave clock (SCL1) 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 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 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 I²C slave clock (SCL1) 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 I²C slave clock (SCL1) after RXRDY is set (a receive interrupt occurs)
  To return NAK, NAK_ANS should be set within this period.
- If the I2CS module has sent back a NAK as the response to the address sent by the master when the conditions shown below are all met, the master must wait for 33 μs or more before it can send another slave address (except when the master sends the I2CS slave address again).

  1. The transfer rate is set to 320 kbps or higher.
  2. The asynchronous address detection function is enabled.
  3. The I2CS module is placed into transfer standby state and OSC1 is used as the operating clock (PCLK).

22 Remote Controller (REMC)

22.1 REMC Configuration

The S1C17003 incorporates a remote controller (REMC) module for generating infrared remote control communication signals. The REMC module consists of a carrier generation circuit for generating a carrier signal using the prescaler output clock, an 8-bit down-counter for counting the transferred data length, a modulation circuit for generating transmission data of the specified carrier length, and an edge detection circuit for detecting input signal rising and falling edges.
The module is also capable of generating counter underflow interrupts indicating that the specified data length has been transmitted and input rising/falling edge detection interrupts for data receipt processing.
Figure 22.1.1 shows the REMC module configuration.

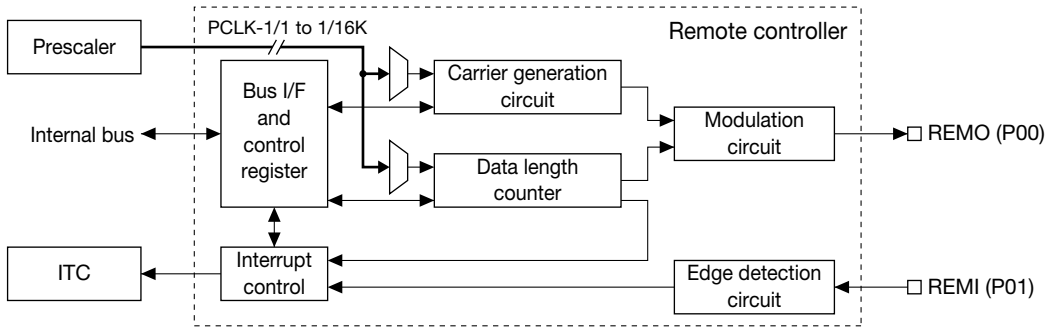


Figure 22.1.1: REMC module configuration

22.2 REMC Input/output Pin

Table 22.2.1 lists the REMC input/output pins.

Table 22.2.1: REMC input/output pin list

Pin name	I/O	Qty	Function
REMI (P01)	I	1	Remote control transmit data input pin Inputs receive data.
REMO (P00)	O	1	Remote control transmit data output pin Outputs modulated remote control transmit data.

The REMC module input/output pins (REMI, REMO) are shared with general purpose input/output port pins (P01, P00) and are initially set as general purpose input/output port pins. The function must be switched using the P0_PMUX register setting to use general purpose input/output port pins as REMC input/output pins. Switch the pins to REMC input/output by setting the following control bits to 1.

P01 → REMI

* **P01MUX**: P01 Port Function Select Bit in the P0 Port Function Select (P0_PMUX) Register (D2/0x52a0)

P00 → REMO

* **P00MUX**: P00 Port Function Select Bit in the P0 Port Function Select (P0_PMUX) Register (D0/0x52a0)

For detailed information on pin function switching, refer to "10.2 Input/output Pin Function Selection (Port MUX)."

22.3 Carrier Generation

The REMC module incorporates a carrier generation circuit that generates a carrier signal for transmission in accordance with the clock set by the software and carrier H and L section lengths.

The prescaler output clock is used for the carrier signal generation clock. The prescaler generates 15 different clocks, dividing the PCLK clock from 1/1 to 1/16K. One is selected by CGCLK[3:0] (D[15:12]/REMC_PSC register).

* **CGCLK[3:0]**: Carrier Generator Clock Select Bits in the REMC Prescaler Clock Select (REMC_PSC) Register (D[15:12]/0x5340)

Table 22.3.1: Carrier generation clock selection

CGCLK[3:0]	Prescaler output clock	CGCLK[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)

For more information on prescaler control, refer to "9 Prescaler (PSC)."

Note: The prescaler must run before the REMC module.

The carrier H and L section lengths are set by REMCH[5:0] (D[5:0]/REMC_CARH register) and REMCL[5:0] (D[13:8]/REMC_CARL register), respectively. These registers set a value corresponding to the number of clock cycles selected above + 1.

* **REMCH[5:0]**: H Carrier Length Setup Bits in the REMC H Carrier Length Setup (REMC_CARH) Register (D[5:0]/0x5342)
* **REMCL[5:0]**: L Carrier Length Setup Bits in the REMC L Carrier Length Setup (REMC_CARL) Register (D[13:8]/0x5342)

The carrier H and L section lengths can be calculated as follows:

$$\text{Carrier H section length} = \frac{\text{REMCH} + 1}{\text{clk\_in}} \text{ [s]}$$

$$\text{Carrier L section length} = \frac{\text{REMCL} + 1}{\text{clk\_in}} \text{ [s]}$$

REMCH: Carrier H section length register data value
REMCL: Carrier L section length register data value
clk_in: Prescaler output clock frequency

The carrier signal is generated from these settings as shown in Figure 22.3.1.

Example: CGCLK[3:0] = 0x2 (PCLK-1/4), REMCH[5:0] = 2, REMCL[5:0] = 1

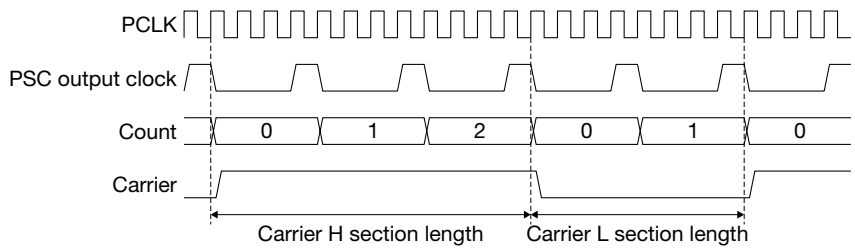


Figure 22.3.1: Carrier signal generation

22.4 Data Length Counter Clock Settings

The data length counter is an 8-bit counter for setting data lengths when transmitting data.
When a value corresponding to the data pulse width is written during data transmission, the data length counter begins counting down from that value, generating an underflow interrupt factor and halting when the counter reaches 0.
The subsequent transmit data is set using this interrupt.

This counter is also used for data receiving, enabling measurement of the receive data length. Interrupts can be generated at the input signal rising or falling edges when receiving data. The data pulse length can be obtained from the difference between data pulses by setting the data length counter to 0xff using the interrupt when the input changes and by reading out the count value when a subsequent interrupt occurs due to input changes.

This data length counter count clock also uses a prescaler output clock and can select one of 15 different types. The prescaler output clock is selected by the control bit LCCLK[3:0] (D[11:8]/REMC_CFG register) provided separately to the carrier generation clock.

* **LCCLK[3:0]**: Length Counter Clock Select Bits in the REMC Prescaler Clock Select (REMC_CFG) Register (D[11:8]/0x5340)

Table 22.4.1: Data length counter clock selection

LCCLK[3:0]	Prescaler output clock	LCCLK[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)

The data length counter can count up to 256. The count clock should be selected to ensure that the data length fits within this range.

22.5 Data Transfer Control

Make the following settings before starting data transfers.
(1) Set the carrier signal. (See Section 22.3.)
(2) Select the data length counter clock. (See Section 22.4.)
(3) Set the interrupt conditions. (See Section 22.6.)

Note: Make sure the REMC module is halted (when REMEN/REMC_CFG register = 0) before changing the above settings.

∗ **REMEN**: REMC Enable Bit in the REMC Configuration (REMC_CFG) Register (D0/0x5340)

Data transfer control

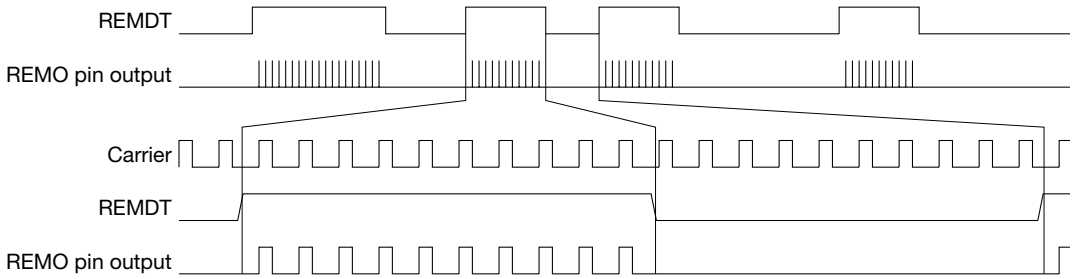


Figure 22.5.1: Data transmission

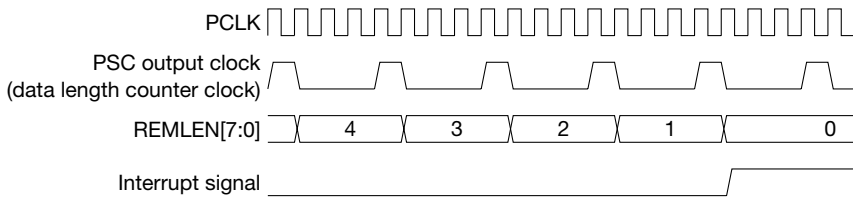


Figure 22.5.2: Underflow interrupt generation timing

(1) Data transmit mode setting
Set REMC to transmit mode by writing 0 to REMMD (D1/REMC_CFG register).

∗ **REMMD**: REMC Mode Select Bit in the REMC Configuration (REMC_CFG) Register (D1/0x5340)

(2) Permit data transmission
Permit REMC operation by setting REMEN (D0/REMC_CFG register) to 1. This initiates REMC transmission.

Set REMDT (D0/REMC_ST register) to 0 and REMLEN[7:0] (D[7:0]/REMC_LCNT register) to 0x0 before setting REMEN to 1 to prevent unnecessary data transmission.

(3) Transmission data settings
Set the data to be transmitted (High or Low) to REMDT (D0/REMC_ST register).

∗ **REMDT**: Transmit/Receive Data Bit in the REMC Status (REMC_ST) Register (D0/0x5344)

Setting REMDT to 1 outputs High; setting it to 0 outputs Low from the REMO pin after being modulated by the carrier signal.

(4) Data pulse length setting
Set the value corresponding to the data pulse length (High or Low section) at the start of transmission to REMLEN[7:0] (D[15:8]/REMC_LCNT register) to set to the data length counter.

∗ **REMLEN[7:0]**: Transmit/Receive Data Length Count Bits in the REMC Length Counter (REMC_LCNT) Register (D[15:8]/0x5344)

Given below are the values to which the data length counter is set:

Setting = Data pulse length (seconds) x prescaler output clock frequency (Hz)
The data length counter begins counting down from the value written using the prescaler output clock selected. An underflow interrupt factor occurs when the data length counter value reaches 0. If interrupts are permitted, an REMC interrupt request is output to the interrupt controller (ITC). The data length counter stops counting when it reaches 0.

(5) Interrupt processing

To transmit the subsequent data, set the subsequent data (step 3) and set the data pulse length (step 4) as part of the interrupt processing routine generated by the data length counter underflow.

(6) Data transmission end

To end data transmission, set REMEN to 0 after the final data transmission is complete (after underflow interrupt has occurred).

Data receipt control

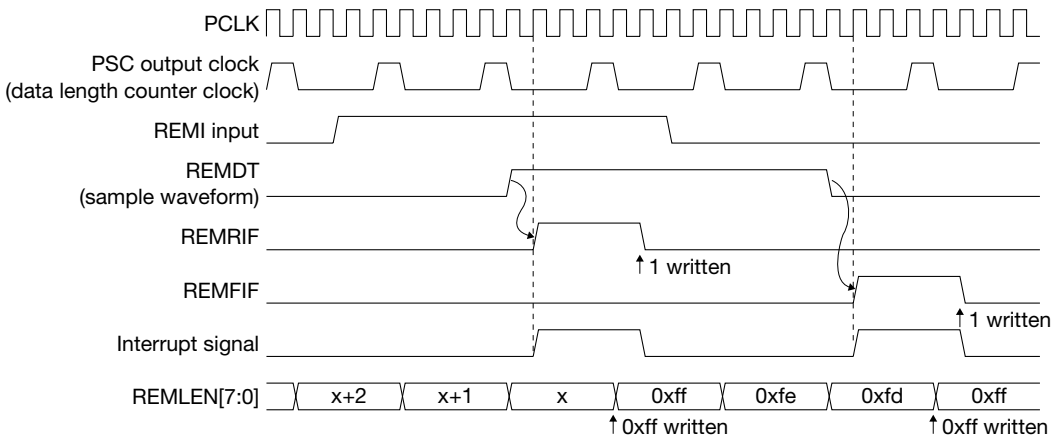


Figure 22.5.3: Data receipt

(1) Data receipt mode setting

Set REMC to receipt mode by writing 1 to REMMD (D1/REMC_CFG register).

(2) Permit data receipt

Permit REMC operation by setting REMEN (D0/REMC_CFG register) to 1. This initiates REMC transmission (input edge detection operation).

REMC detects input changes (signal rising or falling edges) by sampling the input signal from the REMI pin using the prescaler output clock selected for carrier generation. If a signal edge is detected, a rising or falling edge interrupt factor is generated. An REMC interrupt request is output to the ITC if interrupts are permitted. Rising edge and falling edge interrupts can be individually permitted or blocked.

Note that if the signal level after the input has changed is not detected for at least two continuous sampling clock cycles, the interrupt factor is interpreted as noise, and no rising or falling edge interrupt is generated.

(3) Interrupt processing

When a rising edge or falling edge interrupt occurs, 0xff is written to REMLEN[7:0] (D[15:8]/REMC_LCNT register) as part of the interrupt processing routine and set as the value of the data length counter.

The data length counter begins counting down using the selected prescaler output clock from the value written.

The data received can be read out from REMDT (D0/REMC_LCNT register).

The subsequent trailing or rising edge interrupt is generated once the data pulse ends, at which point the data length counter is read out. The data length can be calculated from the difference between 0xff and the value read. To receive the subsequent data, set the data length counter to 0xff once again, then wait for the subsequent interrupt.

If the data length counter becomes 0 after being set to 0xff without the occurrence of an edge interrupt, either data receiving is complete or a receive error has occurred. Data length counter underflow interrupts are generated even when receiving data and should be used for end/error processing.

(4) Data receipt end

To end data receipt, write 0 to REMEN after the final data has been received.

22.6 REMC Interrupts

The REMC module includes functions to generate the following three different interrupt types.
- Underflow interrupt
- Rising edge interrupt
- Falling edge interrupt

The REMC module outputs one interrupt signal shared by the three interrupt factors above to the interrupt controller (ITC). To identify the interrupt factor that occurred, inspect the interrupt flag within the REMC module.

Underflow interrupt

Generated when the data length counter has counted down to 0, this interrupt request sets the interrupt flag REMUIF (D0/REMC_IFLG register) inside the REMC to 1.
When data is being transmitted, the underflow interrupt indicates that the specified data length has been transmitted. When receiving data, the underflow interrupt indicates that data has been received or that a receive error has occurred.

∗ **REMUIF**: Underflow Interrupt Flag in the REMC Interrupt Flag (REMC_INT) Register (D0/0x5346)

To use this interrupt, set REMUIE (D0/REMC_IMSK register) to 1. If REMUIE is set to 0 (default), REMUIF will not be set to 1, and the interrupt request attributable to this factor will not be sent to the ITC.

∗ **REMUIE**: Underflow Interrupt Enable Bit in the REMC Interrupt Mask (REMC_IMSK) Register (D0/0x5346)

When REMUIF is set to 1, REMC outputs an interrupt request signal to the ITC. This interrupt request sets the REMC interrupt flag to 1 within the ITC, and generates an interrupt if the ITC and S1C17 core interrupt conditions are met.
REMUIF should be inspected as part of the REMC interrupt processing routine to determine whether the REMC interrupt is attributable to data length counter underflow.
The interrupt factor should be cleared as part of the interrupt processing routine by resetting both the ITC REMC interrupt flag and REMC module REMUIF (i.e., setting both to 1).

Rising edge interrupt

Generated when the REMI pin input signal changes from Low to High, this interrupt request sets the interrupt flag REMRIF (D1/REMC_IFLG register) to 1 within the REMC.
When data is being received, the data length counter can be operated between this interrupt and a falling edge interrupt to calculate the received data pulse width from that count value.

∗ **REMRIF**: Rising Edge Interrupt Flag in the REMC Interrupt Flag (REMC_INT) Register (D1/0x5346)

To use this interrupt, set REMRIE (D1/REMC_IMSK register) to 1. If REMRIE is set to 0 (default), REMRIF is not set to 1 and the interrupt request for this factor is not sent to the ITC.

∗ **REMRIE**: Rising Edge Interrupt Enable Bit in the REMC Interrupt Mask (REMC_INT) Register (D1/0x5346)

When REMRIF is set to 1, REMC outputs an interrupt request to the ITC. This interrupt request signal sets the REMC interrupt flag to 1 within the ITC, generating an interrupt if the ITC and S1C17 core interrupt conditions are met.
REMRIF should be inspected as part of the REMC interrupt processing routine to determine whether the REMC interrupt is attributable to input signal rising edge.
The interrupt factor should be cleared as part of the interrupt processing routine by resetting both the ITC REMC interrupt flag and REMC module REMRIF (i.e., setting both to 1).

Falling edge interrupt

Generated when the REMI pin input signal changes from High to Low, this interrupt request sets the interrupt flag REMRIF (D2/REMC_INT register) to 1 within the REMC.
When data is being received, the data length counter can be operated between this interrupt and a falling edge interrupt to calculate the received data pulse width from that count value.

* **REMFIF**: Falling Edge Interrupt Flag in the REMC Interrupt Flag (REMC_INT) Register (D2/0x5346)

To use this interrupt, set REMFIE (D2/REMC_IMSK register) to 1. If REMFIE is set to 0 (default), REMFIF is not set to 1 and the interrupt request for this factor is not sent to the ITC.

* **REMFIE**: Falling Edge Interrupt Enable Bit in the REMC Interrupt Mask (REMC_INT) Register (D2/0x5346)

When REMFIF is set to 1, REMC outputs an interrupt request to the ITC. This interrupt request signal sets the REMC interrupt flag to 1 within the ITC, generating an interrupt if the ITC and S1C17 core interrupt conditions are met.
REMFIF should be inspected as part of the REMC interrupt processing routine to determine whether the REMC interrupt is attributable to input signal falling edge.
The interrupt factor should be cleared as part of the interrupt processing routine by resetting both the ITC REMC interrupt flag and REMC module REMFIF (i.e., setting both to 1).

Interrupt vectors

The REMC interrupt vector numbers and vector addresses are as listed below.

Vector number: 20 (0x14)
Vector address: TTBR + 0x50

Other interrupt settings

The ITC allows the priority of REMC interrupts to be set between level 0 (the default value) and level 7. To generate actual interrupts, the PSR (S1C17 core internal processor status register) IE (interrupt enable) bit must be set to 1.
For more information on interrupt processing, see “6 Interrupt Controller (ITC).”

22.7 Control Register Details

Table 22.7.1: REMC register list

Address	Register name		Function
0x5340	REMC_CFG	REMC Configuration Register	Clock and transfer control
0x5342	REMC_CAR	REMC Carrier Length Setup Register	Carrier H/L section length setting
0x5344	REMC_LCNT	REMC Length Counter Register	Transfer bit and tranfer data length setting
0x5346	REMC_INT	REMC Interrupt Mask Register	Interrupt control

The REMC registers are described in detail below. These are 16-bit registers.

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

0x5340: REMC Configuration Register (REMC_CFG)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
REMC Configuration Register (REMC_CFG)	0x5340 (16 bits)	D15–12	CGCLK[3:0]	Carrier generator clock select (Prescaler output clock)	CGCLK[3:0] / LCCLK[3:0]: 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	
		D11–8	LCCLK[3:0]	Length counter clock select (Prescaler output clock)		0x0	R/W	
		D7–2	–	reserved	–	–	–	0 when being read.
		D1	REMMD	REMC mode select	1 Receive / 0 Transmit	0	R/W	
		D0	REMEN	REMC enable	1 Enable / 0 Disable	0	R/W	

D[15:12] CGCLK[3:0]: Carrier Generator Clock Select Bits

Select a carrier generation clock from the 15 prescaler output clocks.

Table 22.7.2: Carrier generation clock selection

CGCLK[3:0]	Prescaler output clock	CGCLK[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)

D[11:8] LCCLK[3:0]: Length Counter Clock Select Bits

Select a data length counter clock from the 15 prescaler output clocks.

Table 22.7.3: Carrier generation clock selection

LCCLK[3:0]	Prescaler output clock	LCCLK[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: The clock should be set only while the REMC module is stopped (REMEN(D0) = 0).

D[7:2] Reserved

D1 REMMD: REMC Mode Select Bit

Selects the transfer direction.

1 (R/W): Receive

0 (R/W): Transmit (default)

D0 REMEN: REMC Enable Bit

Permits or prohibit data transfer by the REMC module
1 (R/W): Permitted
0 (R/W): Prohibited (default)
Setting REMEN to 1 begins transmission or receiving in accordance with REMMD (D1) settings.
Setting REMEN to 0 halts REMC module operations.

0x5342: REMC Carrier Length Setup Register (REMC_CAR)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
REMC Carrier Length Setup Register (REMC_CAR)	0x5342 (16 bits)	D15–14	–	reserved	–	–	–	0 when being read.
		D13–8	REMCL[5:0]	L carrier length setup	0x0 to 0x3f	0x0	R/W	
		D7–6	–	reserved	–	–	–	0 when being read.
		D5–0	REMCH[5:0]	H carrier length setup	0x0 to 0x3f	0x0	R/W	

D[15:14] Reserved

D[13:8] REMCL[5:0]: L Carrier Length Setup Bits

Set the carrier signal L section length. (Default: 0x0)
Specify a value corresponding to the number of carrier generation clock cycles selected by CGCLK[3:0] (D[15:12]/REMC_CFG register) + 1.
Calculate carrier L section length as follows:

$$\text{Carrier L section length} = \frac{\text{REMCL} + 1}{\text{clk\_in}} \text{ [s]}$$

REMCL: REMCL[5:0] settings
clk_in: Prescaler output clock frequency

The H section length is specified by REMCH[5:0] (D[5:0]).
The carrier signal is generated from these settings as shown in Figure 22.7.1.

D[7:6] Reserved

D[5:0] REMCH[5:0]: H Carrier Length Setup Bits

Set the carrier signal H section length. (Default: 0x0)
Specify a value corresponding to the number of carrier generation clock cycles selected by CGCLK[3:0] (D[15:12]/REMC_CFG register) + 1.
Calculate carrier H section length as follows:

$$\text{Carrier H section length} = \frac{\text{REMCH} + 1}{\text{clk\_in}} \text{ [s]}$$

REMCH: REMCH[5:0] settings
clk_in: Prescaler output clock frequency

The L section length is specified by REMCL[5:0] (D[13:8]).
The carrier signal is generated from these settings as shown in Figure 22.7.1.

Example: CGCLK[3:0] = 0x2 (PCLK-1/4), REMCH[5:0] = 2, REMCL[5:0] = 1

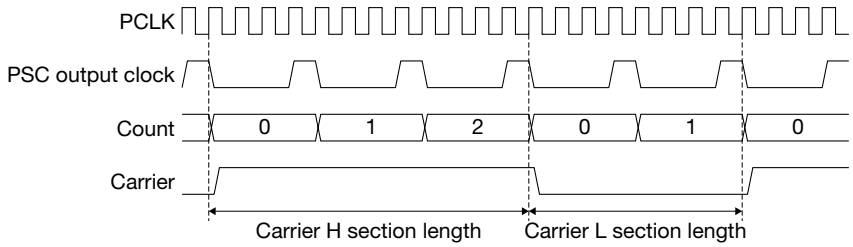


Figure 22.7.1: Carrier signal generation

0x5344: REMC Length Counter Register (REMC_LCNT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
REMC Length Counter Register (REMC_LCNT)	**0x5344** (16 bits)	D15–8	**REMLEN[7:0]**	Transmit/receive data length count (down counter)	0x0 to 0xff	0x0	R/W	
		D7–1	**–**	reserved	–	–	–	0 when being read.
		D0	**REMDT**	Transmit/receive data	1 1 (H) 0 0 (L)	0	R/W	

D[15:8] REMLEN[7:0]: Transmit/Receive Data Length Count Bits

Sets the data length counter value and begins counting. (Default: 0x0)
The counter stops when it reaches 0 and generates an underflow interrupt factor.

For data transmission

Sets the transmit data length for data transmission.
When a value corresponding to the data pulse width is written, the data length counter begins counting down from that value, generating an underflow interrupt and halting when the counter reaches 0.
The subsequent transmit data is set using this interrupt.

For data receiving

Interrupts can be generated at the input signal rising or falling edges when receiving data. The data pulse length can be obtained from the difference by setting the data length counter to 0xff using the interrupt when the input changes and reading out the count value when the next interrupt occurs due to an input change.

D[7:1] Reserved

D0 REMDT: Transmit/Receive Data Bit

Sets the transmit data for data transmission. Receive data can be read when receiving data.
1 (R/W): 1 (H)
0 (R/W): 0 (L) (default)

If REMEN (D0/REMC_CFG register) is set to 1, the REMDT setting is modulated by the carrier signal for data transmission and output from the REMO pin. For data receiving, this bit is set to the value corresponding to the signal level of the data pulse input.

0x5346: REMC Interrupt Control Register (REMC_INT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
REMC Interrupt Control Register (REMC_INT)	0x5346 (16 bits)	D15–11	–	reserved	–	–	–	0 when being read.
		D10	REMFIF	Falling edge interrupt flag	1 Cause of interrupt occurred / 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D9	REMRIF	Rising edge interrupt flag		0	R/W	
		D8	REMUIF	Underflow interrupt flag		0	R/W	
		D7–3	–	reserved	–	–	–	0 when being read.
		D2	REMFIE	Falling edge interrupt enable	1 Enable / 0 Disable	0	R/W	
		D1	REMRIE	Rising edge interrupt enable	1 Enable / 0 Disable	0	R/W	
		D0	REMUIE	Underflow interrupt enable	1 Enable / 0 Disable	0	R/W	

This register indicates the occurrence status of interrupt factors arising from data length counter underflow, input signal rising edge, or input signal falling edge. When an REMC interrupt occurs, the interrupt flag in this register should be inspected to identify the interrupt factor.
Setting the corresponding interrupt enable bit to 1 sets the interrupt flag to 1 when a data length counter underflow, input signal rising edge, or input signal falling edge occurs. The REMC outputs an interrupt request signal to the ITC at the same time, which sets the REMC interrupt flag to 1 within the ITC and generates an interrupt if the ITC and S1C17 core interrupt conditions are met.

Note:
- **To prevent interrupt recurrences, the REMC module interrupt flag must be reset within the interrupt processing routine following an REMC interrupt.**
- **To prevent generating unnecessary interrupts, reset the interrupt flag before permitting interrupts by the interrupt enable bit.**

D[15:11] Reserved

D10 REMFIF: Falling Edge Interrupt Flag

Interrupt flag indicating the falling edge interrupt occurrence status.
1(R): Interrupt factor present
0(R): No interrupt factor (default)
1(W): Reset flag
0(W): Disabled

Setting REMFIE (D2/REMC_IMSK register) to 1 sets SIF1 to 1 at the input signal falling edge.

D9 REMRIF: Rising Edge Interrupt Flag

Interrupt flag indicating the rising edge interrupt factor occurrence status.
1(R): Interrupt factor present
0(R): No interrupt factor (default)
1(W): Reset flag
0(W): Disabled

Setting REMRIF (D1/REMC_IMSK register) to 1 sets REMRIE to 1 at the input signal falling edge.

D8 REMUIF: Underflow Interrupt Flag

Interrupt flag indicating the underflow interrupt factor occurrence status.
1(R): Interrupt factor present
0(R): No interrupt factor (default)
1(W): Reset flag
0(W): Disabled

Setting REMUIE (D1/REMC_IMSK register) to 1 sets REMUIF to 1 when a data length counter underflow occurs.

D[7:3] Reserved

D2 REMFIE: Falling Edge Interrupt Enable Bit

Permits or blocks input signal falling edge interrupts.
1 (R/W): Interrupt permitted
0 (R/W): Interrupt prohibited (default)

D1 REMRIE: Rising Edge Interrupt Enable Bit

Permits or blocks input signal rising edge interrupts.

1 (R/W): Interrupt permitted

0 (R/W): Interrupt prohibited (default)

D0 REMUIE: Underflow Interrupt Enable Bit

Permits or blocks data length counter underflow interrupts.

1 (R/W): Interrupt permitted

0 (R/W): Interrupt prohibited (default)

22.8 Precautions

- The prescaler must run before operating the REMC module.
- To prevent interrupt recurrences, the REMC module interrupt flag must be reset within the interrupt processing routine following an REMC interrupt.
- To prevent unwanted interrupts, reset the interrupt flag before permitting interrupts with the interrupt enable bit.

23 A/D Converter (ADC10SA)

23.1 Outline of A/D Converter

S1C17003 has built-in A/D converter with the following characteristics.

- Conversion method: Successive approximation type
- Resolution: 10 bit
- Input channel: 4 channels
- A/D Conversion clock: 2MHz (Max.), 16kHz (Min.)
- Conversion time: 9 clock (sampling time)
  + 11 clock (conversion time) = 20 clock
  Min. 10 μsec (when 2 MHz input block selected)
  Max. 1,250 μsec (when 16 kHz input block selected)
- Analog input voltage range: V_{SS}–AV_{DD}(=V_{DD})
- Built-in Sampling & hold circuit
- Converter mode (4 types):
  1 time conversion of single channel
  1 time conversion of Multi channels
  Continuous conversion of single channel (end with software control)
  Continuous conversion of multi channels (end with software control)
- Conversion trigger (3 types):
  Software trigger
  External terminal (#ADTRG) trigger
  16bit timer Ch.0 underflow trigger
- Conversion result 10bit can be read by filling to the upper side /lower side.
- Interruption
  Conversion completion interruption
  Conversion result overwrite error interruption

Figure 23.1.1 shows structure of A/D converter.

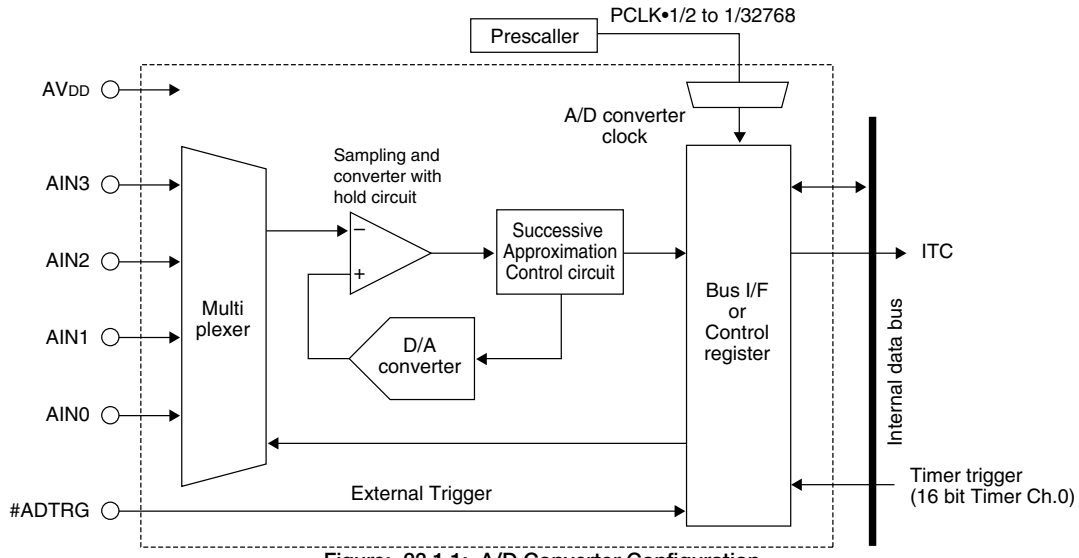


Figure: 23.1.1: A/D Converter Configuration

23.2 ADC Terminal

Figure 23.2.1 shows input/output terminal list of A/D converter.

Table 23.2.1: Input/output terminal of A/D converter

Terminal	I/O	Number	Function
#ADTRG (P03)	I	1	A/D converter external trigger terminal
AIN3 (P17)	I	1	A/D converter Ch.3 analog input terminal
AIN2 (P20)	I	1	A/D converter Ch.2 analog input terminal
AIN1 (P21)	I	1	A/D converter Ch.1 analog input terminal
AIN0 (P22)	I	1	A/D converter Ch.0 analog input terminal
AV_{DD}	-	1	Analog voltage Set as AV_{DD}=V_{DD}. Set as AV_{DD}=V_{DD}, even when A/D converter is not used.

P03 → #ADTRG
- **P03MUX**: P03 Port Function Select Bit in the P0 Port Function Select (P0_PMUX) Register (D4/0x52a1)

P17 → AIN3
- **P17MUX**: P1 Port Function Select Bit in the P1 Port Function Select (P1_PMUX) Register (D2/0x52a3)

P20 → AIN2
- **P20MUX**: P2 Port Function Select Bit in the P1 Port Function Select (P2_PMUX) Register (D2/0x52a4)

P21 → AIN1
- **P21MUX**: P2 Port Function Select Bit in the P1 Port Function Select (P2_PMUX) Register (D2/0x52a4)

P22 → AIN0
- **P22MUX**: P2 Port Function Select Bit in the P1 Port Function Select (P2_PMUX) Register (D2/0x52a4)

Refer to "10.2 Input/Output Pin Function Selection (Port MUX)" for the details of terminal function and switching of function.

Notes : 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.

23.3 A/D Converter Settings

To use the A/D converter, the following settings are required in advance.

1. Setting for analog input pins ... See section 23.2
2. Setting for A/D conversion clock
3. Selection of the start/end channels for analog conversion process
4. Setting of A/D conversion mode
5. Selection of the trigger type
6. Setting of sampling time
7. Setting of conversion result storage mode
8. Setting for interrupts... See section 23.6

Note: Be sure to disable the A/D converter (set ADEN(DO/ADC10_CTL register)=0) before configuring those settings. Changing settings in enabled state can cause a malfunction.

∗ **ADEN**: A/D Enable Bit in the A/D Control/Status (ADC10_CTL) Register (DO/0x5384)

Setting for A/D conversion clock

To use the A/D converter, the peripheral clock (PCLK) supplied from the clock generator (CLG) and a division clock supplied from the Prescaler (PSC) must be turned on.
For details, refer to the "8.3 Peripheral Module Clock (PCLK) Control", and "9.1 Prescaler Configuration."
The A/D converter can select the Prescaler-supplied division clock from 15 types shown in the table 23.3.1.Use ADDF[3:0] (D[3:0]/ADC10_DIV register) for the selection.

∗ **ADDF[3:0]**: A/D Converter Clock Divided Frequency Selection Bits in the ADCIO Divided Frequency (ADC10_DIV) Register(D[3:0]/Ox5386)

Table 23.3.1: Selection of A/D conversion clock

ADDF3:0	A/D clock
0xf	reserved
0xe	PCLK•1/32768
0xd	PCLK•1/16384
0xc	PCLK•1/8192
0xb	PCLK•1/4096
0xa	PCLK•1/2048
0x9	PCLK•1/1024
0x8	PCLK•1/521
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)

Note: • For information about restriction of input clock frequencies, refer to "26.5 A/D Converter Characteristics."
• Do not start A/D conversion while clock output from the Prescaler to the AD converter is turned off, or turn off clock output from the Prescaler while A/D conversion is in process. It can cause a malfunction.

Selection of the start/end channels for analog conversion process

The channels used for the A/D conversion should be selected from pins (channels) configured for analog input. This setting enables single converting operation to process the serial A/D conversion over multiple channels. Use ADCS[2:0] (D[10:8]/ADC10_TRG register) and ADCE[2:0] (D[13:11]/ADC10_TRG register) to specify the start and end channel respectively for conversion process.

* **ADCS[2:0]**: A/D Converter Start Channel Selection Bits in the ADC10 Trigger/Channel Select (ADC10_TRG) Register (D[10:8]/0x5382)
* **ADCE[2:0]**: A/D Converter End Channel Selection Bits in the ADC10 Trigger/Channel Select (ADC10_TRG) Register (D[13:11]/0x5382)

Table 23.3.2: Relation between ADCS/ADCE and input channels.

ADCS[2:0]/ADCE[2:0]	Select channel
0x7	Dummy ch[7:4] 0 (Vss level)
0x6	
0x5	
0x4	
0x3	AIN3
0x2	AIN2
0x1	AIN1
0x0	AIN0

(Default: 0x0)

Example: A/D conversion process of single operation

ADCS[2:0] = 0, ADCE[2:0] = 0: Convert only AIN0.

ADCS[2:0] = 0, ADCE[2:0] = 3: Convert serially in the order of AIN0→AIN1→AIN2→AIN3

ADCS[2:0] = 2, ADCE[2:0] = 1: Convert serially in the order of AIN2→AIN3→(Dummy ch[7:4])→AIN0→AIN1

Setting of A/D conversion mode

Single conversion or serial conversion can be selected for the A/D converter by using ADMS (D5/ADC10_TRG register).

* **ADMS**: A/D Conversion Mode Selection Bit in the ADC10 Trigger/Channel Select (ADC10_TRG) Register (D6/0x5382)

1. Single conversion mode (ADMS=0)

This mode performs a single A/D conversion of all inputs to channels in the range specified by ADCS[2:0] (D[10:8]/ADC10_TRG register) and ADCE[2:0] (D[13:11]/ADC10_TRG register), and then stops.

2. Serial conversion mode (ADMS=1)

This mode keeps performing A/D conversion of channels in the range specified by ADCS[2:0] or ADCE[2:0] until software stops the process.

The mode is set to single conversion after the initial reset.

Selection of the trigger type

Select the type of trigger starting A/D conversion from 3 types shown in the table 23.3.2 and specify it by ADTS[1:0] (D[5:4]/ADC10_TRG register).

* **ADTS[1:0]:** A/D Conversion Trigger Selection Bits in the ADC10 Trigger/Channel Select (ADC10_TRG) Register (D[5:4]/0x5382)

Table 23.3.3: Selection of the trigger type

ADTS[1:0]	Trigger source
0x3	External trigger (#ADTRG pin)
0x2	reserved
0x1	16-bit programmable timer Ch.0
0x0	Software trigger

(Default: 0x0)

1. External trigger (#ADTRG)

This type uses an input signal via the #ADTRG pin as a trigger.

To use this trigger type, the #ADTRG pin must be configured using the Port Function Select Register. This type starts A/D conversion by detecting negative edge of #ADTRG signal.

2. 16-bit timer (T16) Ch.0

This type uses an underflow signal of 16-bit timer (T16) Ch.0 as a trigger. The type is effective when periodic A/D conversion is required because the cycle of the signal can be configured programmably by the timer. For settings for the timer, refer to "11 16-bit Timer (T16)."

3. Software trigger

This type uses the software's writing 1 to ADCTL (D1/AD_CTL register) as a trigger to start A/D conversion.

* **ADCTL:** A/D Conversion Control/Status Bit in the ADC10 Control/Status (ADC10_CTL) Register (D1/0x5382)

Setting of sampling time

This A/D converter provides ADST[2:0] (D[2:0]/ADC10_TRG register) enabling the input sampling time of analog signals to be configured to 8 steps (2 to 9 of the conversion clock).

The register must be set to default (ADST[2:0]=111).

∗ **ADST[2:0]:** Sampling Clock Count Bits in the ADC10 Control/Status (AD_CTL) Register (D[2:0]/0x5382)

Setting of conversion result storage mode

After completing A/D conversion, this 10-bit A/D converter stores the 10-bit conversion result in the A/D conversion result storage register ADD[15:0] (D[15:0]/ADC10_ADD register).

∗ **ADD[15:0]:** A/D Converted Data Bits in the ADC10 Conversion Result (ADC10_ADD) Register (D[15:0]/0x5380)

The conversion result storage mode can configure STMD (D[7]/ADC10_TRG register), and select either high-order or low-order to store 10-bit A/D conversion result in ADD[15=0].

∗ **STMD:** Converted Data Store Mode Bits in the ADC10 Trigger/Channel Select (ADC10_TRG) Register (D[7]/0x5382)

STMD=0: ADD[15:10]=0, ADD[9]= conversion result [MSB], ADD[0]= conversion result [LSB]
STMD=1: ADD[15]=[MSB], ADD[6]= conversion result [LSB], ADD[5:0]=0

23.4 A/D Conversion Control and Operations

The following shows the process of A/D conversion operation

1. Activating A/D converter circuit
2. Starting A/D conversion
3. Reading A/D conversion result
4. Completing A/D conversion

Activating A/D converter circuit

After configuring settings shown in the previous section, write 1 to ADEN (DO/ADC10_CTL register) to enable the A/D converter. This allows the A/D converter to permit a trigger to start A/D conversion. To reconfigure or disable the A/D converter, set the ADEN bit to 0.

* **ADEN**: A/D Enable Bit in the ADC10 Control/Status (ADC10_CTL) Register (D0/0x5384)

Starting A/D conversion

The A/D converter starts A/D conversion if a trigger is input when the ADEN bit is set to 1. When software trigger is selected, A/D conversion starts by writing 1 to ADCTL (Dl/ADC10_CTL register).

* **ADCTL**: A/D Conversion Control Bit in the ADC10 Control/Status (ADC10_CTL) Register (D1/0x5384)

Triggers other than selected by ADTS[1:0](D[5:4]/ADC10_TRG register) are not permitted.

* **ADTS[1:0]**: A/D Conversion Trigger Selection Bits in the ADC10 Trigger/Channel Select (ADC10_TRG) Register (D[5:4]/0x5382)

Once a trigger is input, the A/D converter processes the sampling of analog input signals from the conversion starting channel selected by ADCS[2:0] (D[10:8]/ADC10_TRG register) to perform A/D conversion.

* **ADCS[2:0]**: A/D Converter Start Channel Selection Bits in the ADC10 Trigger/Channel Select (ADC10_TRG) Register (D[10:8]/0x5382)

The ADCTL bit used for software trigger turns to 1 even by the trigger of other type, enabling itself to be used as the status bit for A/D conversion.
ADICH[2:0] (D[2=0]/ADC10_CTL register) can read the channel in conversion process.

* **ADICH[2:0]**: Internal Conversion Channel Status Bits in the ADC10 Control/Status (ADC10_CTL) Register (D[14:12]/0x5384)

Reading A/D conversion result

After completing A/D conversion, the A/D converter stores conversion result in 10-bit data register ADD[15:0] (D[15:0]/ADC10_ADD register), and set the conversion complete flag ADCF (D8/ADC10_CTL register). If ADCS[2:0] (D[10:8]/ADC10_TRG register) and ADCE[2:0] (D[13:11]/ADC10_TRG register) specify multiple channels, the A/D converter continues A/D conversion for subsequent channels.

* **ADD[15:0]**: A/D Converted Data Bits in the ADC10 Conversion Result (ADC10_ADD) Register (D[15:0]/0x5380)
* **ADCF**: Conversion-Complete Flag Bit in the ADC10 Control/Status (ADC10_CTL) Register (D8/0x5384
* **ADCE[2:0]**: End Channel Selection Bits in the ADC10 Trigger/Channel Select (ADC10_TRG) Register (D[13:11]/0x5382)

A/D conversion result is stored in ADD[15:0] each time when conversion for a channel is completed. The conversion complete interrupt can be generated concurrently with the storing. The interrupt is usually used to read converted data. If you do not use the conversion complete interrupt, check that the conversion complete factor ADCF (D8/ADD[15:0] register) is set to 1, and then read conversion result from ADD ADD[15:0]. By reading the ADD [15:0] value, the conversion complete interrupt and the ADCF flag are automatically set to 0.

When the serial conversion mode has been selected, conversion result must be read from ADD[15:0] before the next conversion is completed. If you cannot read the conversion result before ADD[15:0] is updated while the conversion complete flag ADCF (D8/ADC10_CTL register) is set to 1, the overwrite error flag ADOWE (D9/ADC10_CTL register) is set to 1, so that you can check that the conversion result has been overwritten. You can also generate the conversion data overwrite interrupt concurrently with overwriting. After reading the conversion result from ADD[15:0], read also the ADOWE flag, or check that the conversion data overwrite interrupt has not occurred so that the read data is valid.
Once the ADOWE flag has been set, it is not reset until software write 1 to the flag. If the ADOWE flag has been reset, the conversion data overwrite interrupt can be stopped to occur.
Note that setting ADOWE flag to 1 also sets the ADCF flag. Therefore, read converted data to reset ADCF to 0.

∗ **ADOWE**: Overwrite Error Flag Bit in the ADC10 Control/Status (ADC10_CTL) Register (D9/0x5384)

Note: Occurrence of an overwrite error does not stop serial conversion process.

Completing A/D conversion

- **For single conversion mode (ADMS=0)**
  Single conversion mode stops the conversion process once completing a cycle from the start channel specified by ADCS[2:0] (D[10:8]/ADC10_TRG register) to the end channel ADCE[2:0] (D[13:11]/ADC10_TRG register). After the completion, ADCTL (D1/ADC10_CTL register) is returned to 0.

  ∗ **ADMS**: Conversion Mode Selection bit in the ADC10 Trigger/Channel Select (ADC10_TRG) Register (D6/0x5382)

- **For serial conversion mode (ADMS=1)**
  Serial conversion mode maintains the A/D conversion cycles from the start channel to the end channel continuously. Hardware in this mode does not stop the cycles. Use software to set ADCTL (D1/ADC10_CTL register) to 1 to terminate the process forcibly. You cannot get data under A/D conversion process when forcible termination occurs.

Figure 23.4.1 shows A/D conversion operation.

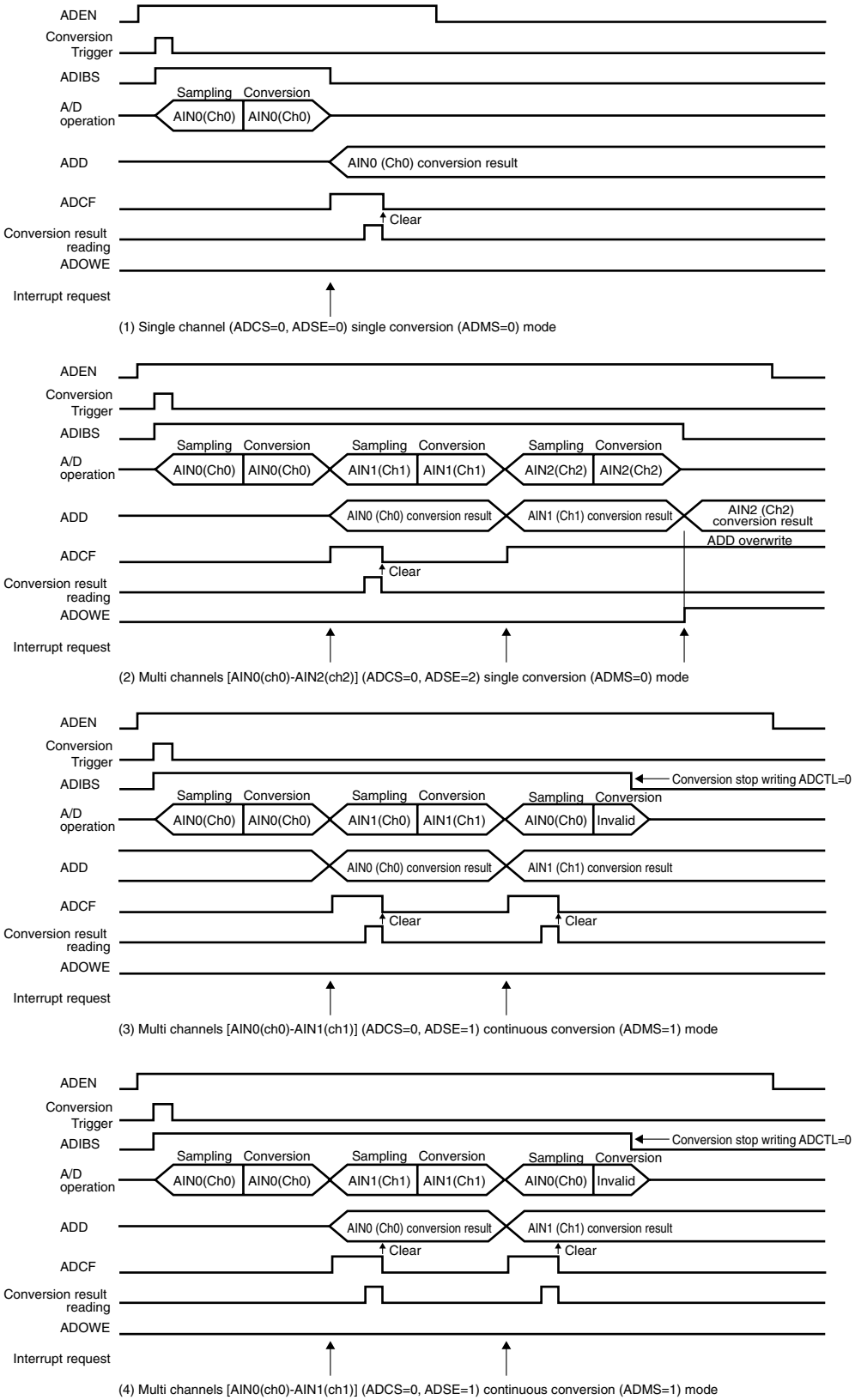


Figure 23.4.1: Operation of A/D conversion

23.5 A/D Converter interrupt

The A/D converter provides function of generating the following 2 types of interrupts.

- Conversion complete interrupt
- Conversion data overwrite interrupt

The A/D converter outputs an interrupt signal shared by those 2 types of interrupt factors to the interrupt controller (ITC). To determine the generated interrupt factor, read a relevant interrupt factor register.

Conversion complete interrupt

Once completing A/D conversion for a channel, and if ADCIE (D4/ADC10_CTL register) is set to 1 (default:0), the A/D converter outputs the conversion complete interrupt signal to the controller (ITC) to request an interrupt.

∗ **ADCIE**: Conversion-Complete Interrupt Enable Bit in the ADC10 Control/Status (ADC10_CTL) Register (D4/0x5384)

By reading ADD[15:0] (D[15:0]/ADC10_ADD register), the conversion complete interrupt factor is automatically cleared, and ADCF (D8/ADC10_CTL register) is also reset from 1 to 0. To disable generation of the conversion complete interrupt, set the ADCIE bit to 0.

Conversion data overwrite interrupt

When the ADD[15:0] register has not been read before it is overwritten by the subsequent A/D conversion result, and if ADOIE (D5/ADC10_CTL register) is set to 1 (default:0), the A/D converter outputs the conversion data overwrite interrupt signal to the controller (ITC) to request an interrupt.

∗ **ADOIE**: Overwrite Interrupt Enable Bit in the ADC10 Control/Status (ADC10_CTL) Register (D5/0x5384)

By writing 1 to ADOWE (D9/ADC10_CTL register), the conversion data overwrite interrupt factor is reset to 0.

∗ **ADOWE**: Overwrite Error Flag Bit in the ADC10 Control/Status (ADC10_CTL) Register (D9/0x5384)

To disable generation of the conversion data overwrite interrupt, set the ADOIE bit to 0.

ITC register for A/D converter interrupts

Table 23.5.1 shows the ICT control register corresponding to the A/D converter interrupt factors.

Table 23.5.1: ITC register

Interrupt factor	interrupt level setting bit
Conversion complete/Conversion data overwrite	ILV18[2:0] (D[2:0]/ITC_LV9)

ITC_LV9 register (0x4318)

Specify the A/D converter interrupt level (0 to 7) for the interrupt level bit. If the same interrupt level is specified, out-of-range interrupts have higher priority while the conversion complete interrupt has lower. The S1C17 core permits an interrupt when all of the following conditions are met:

- The interrupt enable bit of the A/D converter module is set to 1.
- The IE (interrupt enable) bit of the processor status register inside the S1C17 core (PSR) is set to 1.
- A/D converter interrupts are set to higher level than the value set in Interrupt Level (IL) of PSR.
- NMI or other interrupt factor with higher priority has not occurred.

For details on the interrupt control register and its operation when an interrupt occurs, refer to "6. Interrupt Controller (ITC)."

Interrupt vector

The following shows the vector number and address of A/D converter interrupts.

Table 23.5.2: A/D converter interrupt vector

Interrupt factor	Vector No.	Vector address
Conversion complete/Conversion data overwrite	22 (0x16)	TTBR + 0x58

23.6 Controlling Register Details

Table 23.6.1: ADC10SA register list

Address	Register name		Function
0x5380	ADC10_ADD	ADC10 Conversion Rersult Register	AD conversion result
0x5382	ADC10_TRG	ADC10 Trigger/Channel Select Register	Setting of conversion trigger/conversion channel
0x5384	ADC10_CTL	ADC10 Control/Status Register	Conversion control, conversion status
0x5386	ADC10_DIV	ADC10 Divided Frequency Register	A/D conversion clock division setting

Each register of ADC10SA module is explained below. These are 16 bites registers.

Note: Write 0 in "Reserved" bit while writing the data to the register. Do not write 1.

0x5380: ADC10 Conversion Result Register (ADC10_ADD)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
A/D Conversion Result Register (ADC10_ADD)	**0x5380** (16 bits)	D15–0	**ADD[15:0]**	A/D converted data @STMD=0 ADD[15:10]=60, ADD9=MSB, ADD0=LSB @STMD=1 ADD15=MSB, ADD6=LSB, ADD[5.:0]=60	0–1023	0	R	

D[15:0] ADD[15:0]: A/D Converted Data Bits

A/D conversion result is stored.
Storage methods can be changed by settings of STMD register.

STMD=0 ADD[15:10]=0, ADD[9]=MSB, ADD[0]=LSB
STMD=1 ADD[15]=MSB, ADD[6]=LSB, ADD[5:0]=0

This register is read only so writing is not possible.
Data is 0 at the time of initial setting.

0x5382: ADC10 Trigger/Channel Selection Register (ADC10_TRG)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
A/D trigger/ Channel Select (ADC10_TRG)	0x5382 (16 bits)	D15–14	–	reserved	–	–	–	0 when being read.
		D13–11	**ADCE[2:0]**	End channel selection	0x0–0x7	0	R/W	
		D10–8	**ADCS[2:0]**	Start channel selection	0x0–0x7	0	R/W	
		D7	**STMD**	Covnerted data store mode	1 {AD[9:0], 6'b0} / 0 {6'b0, AD[9:0]}	0	R/W	
		D6	**ADMS**	Conversion mode selection	1 continuous / 0 single	0	R/W	
		D5–4	**ADTS[1:0]**	Conversion trigger selection	ADST[1:0] trigger: 0x3 #ADTRG pin; 0x2 reserved; 0x1 16bit timer; 0x0 software	0	R/W	
		D3	–	reserved	–	–	–	0 when being read.
		D2–0	**ADST[2:0]**	Sampling clock count	ADST[2:0] count clock: 0x7 9clocks; 0x6 8clocks; 0x5 7clocks; 0x4 6clocks; 0x3 5clocks; 0x2 4clocks; 0x1 3clocks; 0x0 2clocks	0x7	R/W	Must set 0x7

D[15:14] Reserved

D[13:11] ADCE[2:0]: End Channel Selection Bits

Set the conversion end channel within (0–7) channel numbers.
Analog input from the channel set by ADCS register up to the channel set by register can be converted continuously in 1 A/D conversion.
When A/D is to be converted only for 1 channel, set the same channel numbers in ADCS register and ADCE register.
ADCE is set to 0 (AIN0) at the time of initial reset.

D[10:8] ADCS[2:0]: Start Channel Selection Bit

Set the conversion start channel with channel numbers (0–7).
Analog input from the channel set by this register up to the channel set by ADCE register can be converted continuously in 1 A/D conversion.
When A/D is converted for only 1 channel, set the same channel number in ADCS register and ADCE register.
ADCS is set to 0 (AIN0) at the time of initial reset.

D7 STMD: Converted Data Store Mode Bit

Select the method to store conversion result to ADD register.
For the details, refer to ADD register.
STMD is set to 0 (ADD [15:10]=6'b0, ADD[9]=MSB, ADD[0]=LSB) at the time of initial reset.

D6 ADMS: Conversion Mode Selection Bit

Select the A/D conversion mode.

1 (R/W) : Continuous conversion mode
0 (R/W) : Single conversion mode

A/D converter is set to continuous mode by writing 1 to ADMS. A/D conversion in the range of channel selected by ADCS and ADCE can be performed continuously till software stops it.
When ADMS is 0, it operates in single conversion mode and A/D conversion for all inputs in the range of channel selected by ADCS and ADCE register is performed once and stopped.
ADMS is set to 0 (single conversion mode) at the time of initial reset.

D[5:4] ADTS [1:0]: Conversion Trigger Selection Bits

Select the trigger method by which A/D conversion is started.

Table 23.6.2: Trigger selection

ADTS1	ADTS0	Trigger
1	1	External trigger (# ADTRG)
1	0	Reserved
0	1	16 bits programmable timer
0	0	Software

When external trigger is used, select the # ADTRG from the port MUX (For the details, refer to I/O port section and port MUX section).
When 16 bits programmable timer ch0 is used, since the underflow signal becomes trigger, set the period and other settings by programmable timer.
ADTS is set to 0 (software trigger) at the time of initial reset.

D3 Reserved

D[2:0] ADST [2:0]: Sampling clock Count Bits

Sets the sampling time of analog input.

Table 23.6.3: Trigger selection

ADST2	ADST1	ADST0	Sampling time
1	1	1	9 clocks
1	1	0	8 clocks
1	0	1	7 clocks
1	0	0	6 clocks
0	1	1	5 clocks
0	1	0	4 clocks
0	0	1	3 clocks
0	0	0	2 clocks

Clock number is an input clock number of A/D converter.
ADST is set to 111 (9 clocks) at the time of initial reset.
ADST must be set to 111 (9 clocks). Do not change the register value.

0x5384: ADC10 Control/Status Register (ADC10_CTL)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
A/D Control/ Status Register (ADC10_CTL)	0x5384 (16 bits)	D15	–	reserved	–				–	–	0 when being read.
		D14–12	ADICH	Internal conversion channel status	0x0–0x7				0	R	
		D11	–	reserved	–				–	–	0 when being read.
		D10	ADIBS	Internal busy status	1	busy	0	idle	0	R	
		D9	ADOWE	Overwrite error flag	1	Error	0	Normal	0	R/W	Reset by writing 1
		D8	ADCR	Conversion-complete flag	1	Completed	0	Not conmpleted	0	R	Reset when ADADD is read.
		D7–6	–	reserved					–	–	0 when being read.
		D5	ADOIE	Overwrite interrupt enable	1	Enable	0	Disable	0	R/W	
		D4	ADCIE	Conversion-complete interrupt enable	1	Enable	0	Disable	0	R/W	
		D3–2	–	reserved					–	–	0 when being read.
		D1	ADCTL	conversion control	1	Start/Run	0	Stop	0	R/W	Stop by writing 0
		D0	ADEN	A/D enable	1	Enable	0	Disable	0	R/W	

D15 Reserved

D[14:12] ADICH [2:0]: Internal Conversion Channel Status Bits

Shows the channel numbers (0–7) during A/D conversion (4–7 are Dummy ch).
When multi channels make A/D conversion, channels currently under conversion can be confirmed by reading this bit.
ADICH is set to 0 (AIN0) at the time of initial reset.

D11 Reserved

D10 ADIBS: Internal Busy status Bits

Shows the status of A/D converter.

1 (R/W) : during conversion
0 (R/W) : Conversion complete

1 is output during A/D conversion and 0 is output after A/D conversion completion.

D9 ADOWE: Overwrite Error Flag Bit

This is an interruption flag indicating the status of conversion data overwrite cause

1 (R) : With Interruption cause
0 (R) : Without interruption cause (default)
1 (W) : Reset the flag
0 (W) : Disable

When multi channels make A/D conversion, ADOWE set to 1 if conversion result of next channel is written (overwritten) to conversion data register before resetting the conversion end flag ADCF set by conversion of previous channel by reading conversion data. At that time, if the ADOIE (D5/ ADC10_CTL register) is set to 1, overwrite interruption request signal related for ITC is output. If interruption conditions of ITC and S1C17 core are valid, interruption will be occurred.
ADOWE is reset by writing 1.

Note: • After generating overwrite interruption, it is necessary to reset the ADOWE in interruption process routine to prevent the regeneration of same interruption.
• Before permitting overwrite interruption by ADOIE, reset the ADOWE to prevent the generation of unnecessary interruption.

D8 ADCF: Conversion Complete Flag Bit

It is an interruption flag indicating condition for generating conversion completion cause.

1 (R) : With interruption cause
0 (R) : Without interruption cause (default)
1 (W) : Disable
1 (W) : Disable

After completion of A/D conversion, if the conversion data is stored to ADD(D[15:0]/ADC10_ADD register) it is set to 1. At that time, if the ADCIE(D4/ADC10_CTL register) is set to 1, conversion completion interruption request signal for ITC is output. If interruption conditions of ITC and S1C17 core are valid, interruption will be occurred.
It is reset to 0 by reading ADD.
When multiple channels make A/D conversion, if next A/D conversion is finished in the status where ADCF is 1 (before reading conversion data), data register overwrites to the new conversion result and overwrite error is generated. Therefore, it is necessary to reset the ADCF by reading the conversion data before completing the next A/D conversion.

D[7:6] Reserved

D5 ADOIE: Overwrite Interrupt Enable Bit

Permits or prohibits the generation of overwrite interruption of A/D conversion result for CPU.

1 (R/W) : Interruption permitted
0 (R/W) : Interruption prohibited

In the interruption enable bit that controls the overwrite interruption of A/D conversion result, when ADOIE is set to 1, interruption is permitted and when it sets to 0, interruption is prohibited. ADOIE is set to 0 (Interruption prohibition) at the time of initial reset.

D4 ADCIE: Conversion-complete Interrupt Enable Bit

Permits or prohibits the generation of A/D conversion complete interrupt for CPU.

1 (R/W) : Interruption permitted
0 (R/W) : Interruption prohibited

In the interruption enable bit that controls the A/D conversion complete interruption, when ADCIE is set to 1, interruption is permitted and when it sets to 0, interruption is prohibited. ADCIE is set to 0 (Interruption prohibition) at the time of initial reset.

D[3:2] Reserved

D1 ADCTL: Conversion Control Bit

Controls the A/D conversion.

1 (R/W) : Software trigger
0 (R/W) : A/D conversion stop

If the A/D conversion is started by software trigger, 1 is written to ADCTL. In case of other trigger methods, ADCTL is set to 1 by hardware.
ADCTL retained to 1 during A/D conversion.
At the time of single conversion mode, if the A/D conversion of specified channels is stopped, ADCTL returns to 0 and A/D conversion circuit is stopped. When A/D conversion of continuous mode is stopped, write 0 to ADCTL.
When ADEN is 0, a trigger is not accepted.
ADCTL is set to 0 (A/D conversion stop) at the time of initial reset.

D0 ADEN: A/D Enable Bit

Set the A/D converter to enable (conversion possible status).

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

A/D converter is enabled by writing 1 to ADEN and it is a condition where A/D conversion (trigger can be received) can be started. When ADEN is 0, A/D converter is set to default status and trigger is not received.
Furthermore, when the A/D converter of mode and start/complete channels is to be set, it is set after setting ADEN to 0 in order to avoid the error operation.
ADEN is set to 0 (disable) at the time of initial reset.

0x5386: ADC10 Divided Frequency Register (ADC10_DIV)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
A/D Divided	**0x5386**	D15–4		reserved	0–1023		0	R	0 when being read.
Frequency Register (ADC_DIV)	(16 bits)	D3–0	**ADDF[3:0]**	A/D converter clock divided frequency select	ADDF[3:0]	clock	0	R/W	
					0xf	Reserved			
					0xe	PCLK.1/32768			
					0xd	PCLK.1/16384			
					0xc	PCLK.1/8192			
					0xb	PCLK.1/4096			
					0xa	PCLK.1/2048			
					0x9	PCLK.1/1024			
					0x8	PCLK.1/512			
					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			

D[15:4] Reserved

D[3:0] ADCTL: A/D Converter Clock Divided Frequency Select Bits
A/D conversion clock can be selected from 16 types mentioned above.

Note: • Prescaler should be operated is the precondition for the operation of A/D converter. For the details, refer to CLG chapter, PCLK control section, PSC chapter, prescaler structure section.
• For information about restriction of input clock frequencies, refer to "26.5 A/D Converter Characteristics."
• When the clock output from prescaler to A/D converter is OFF, never start the A/D conversion nor set the clock output of prescaler during A/D conversion operation to OFF. Otherwise it may cause an error.

23.7 Notes

- When A/D converter like mode or start/complete channel is to be set, A/D converter should be set to disable status (ADEN (D0/ADC10_CTL register) is 0). It may cause an error if enable status is changed.
- For information about restriction of A/D conversion clock frequencies, refer to “26.5 A/D Converter Characteristics.”
- If the clock output from prescaler to A/D converter is OFF, never start the A/D conversion nor set the clock output of prescaler during A/D conversion operation to OFF. Otherwise it may cause an error.
- ADCF (D8/ADC_CTL register) and ADOWE (D9/ADOWE) will not be fixed after initial reset.
  Reset the program to prevent the generation of unnecessary interruption.
- After interruption, reset the PSR or interruption cause flag before executing instruction to prevent regeneration of interruption due to same reason.
- If the external triggers are used as A/D conversion triggers, ensure to maintain the length more than 2 cycles of S1C17 core operation clock for Low period of input to # ADTRG terminal.

24 On-chip Debugger (DBG)

24.1 Resource Requirements and Debugging Tool

Debugging work area

Debugging requires a 64-byte debugging work area. In the S1C17003, RAM addresses 0x0007c0 to 0x0007ff are assigned as the debugging work area. When using the debugging function, avoid using this area for any other user applications.

The start address for this debugging work area can be read from the DBRAM register (0xffff90).

Debugging tool

Debugging involves connecting an ICD (In-Circuit Debugger) such as S5U1C17001H (ICD Mini) to the S1C17003 debug pin and inputting the debug instruction from the PC debugger.

The following tools are required:

- S1C17 Family In-Circuit Debugger (e.g., S5U1C17001H)
- S1C17 Family C compiler package (S5U1C17001C)

Debug pins

The following debug pins are used to connect an ICD (e.g., S5U1C17001H).

Table 24.1.1: Debug pin list

Pin name	I/O	Qty	Function
DCLK (P43)	O	1	On-chip debugger clock output pin Outputs a clock to the ICD.
DSIO (P41)	I/O	1	On-chip debugger data input/output pin Used for inputting/outputting debugging data and inputting break signals.
DST2 (P42)	O	1	On-chip debugger status signal output pin Outputs the processor status during debugging.

Shared with general purpose input/output port pins (P43, P42, P41), the on-chip debugger input/output pins (DCLK, DST2, DSIO) are initially set for use as debugger pins. If the debugging function is not used, these pins can be switched via the P4_PMUX register to enable use as general purpose input/output port pins. Set the control bits shown below to 1 to switch the pins to general purpose input/output port use.

DCLK → P43

∗ **P43MUX**: P43 Port Function Select Bit in the P4 Port Function Select (P4_PMUX) Register (D6/0x52a8)

DST2 → P42

∗ **P42MUX**: P42 Port Function Select Bit in the P4 Port Function Select (P4_PMUX) Register (D4/0x52a8)

DSIO → P41

∗ **P41MUX**: P41 Port Function Select Bit in the P4 Port Function Select (P4_PMUX) Register (D2/0x52a8)

For more information on pin function and switching, refer to "10.2 Input/Output Pin Function Selection (Port MUX)."

24.2 Debug Break Operation Status

The S1C17 core switches to debug mode when the `brk` instruction is executed or a debug interrupt is generated by a break signal (Low) input to the DSIO pin. This state persists until the `retd` instruction is executed.
During this time, hardware interrupts and NMIs are disabled.
The default setting halts peripheral circuit operations. This setting can be modified even when debugging is underway.

Peripheral circuits that operate using the prescaler output clock

- 8-bit timer
- 16-bit timer
- PWM timer
- Remote controller
- P port
- UART
- SPI
- I^2C (master/slave)
- ADC

With the default settings, the prescaler will stop in debug mode, also stopping the peripheral circuits above that use the prescaler output clock. The prescaler includes PRUND (D1/PSC_CTL register) to specify prescaler operations during debug mode. When PRUND is set to 1, the prescaler operates even in debug mode, allowing the peripheral circuits above to operate as well. When PRUND is 0 (default), the prescaler and the peripheral circuits above will stop when the S1C17 core switches to debug mode.

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

Peripheral circuits that operate using the OSC1 clock

- Clock timer
- Watchdog timer
- Stopwatch timer
- 8-bit OSC1 timer

The MISC register includes O1DBG (D0/MISC_OSC1 register) to specify the operation of the above OSC1 peripheral circuits during debug mode. When O1DBG is set to 1, the OSC1 peripheral circuits operate even in debug mode. When O1DBG is 0 (default), the OSC1 peripheral circuits will stop when the S1C17 core switches to debug mode.

∗ **O1DBG**: OSC1 Peripheral Control (in Debug Mode) Bit in the OSC1 Peripheral Control (MISC_OSC1) Register (D0/0x5324)

24.3 Additional Debugging Function

The S1C17003 expands the following on-chip debugging functions of the S1C17 core.

Branching destination in debug mode

When a debug interrupt is generated, the S1C17 core enters debug mode and branches to the debug processing routine. In this process, the S1C17 core is designed to branch to address 0xfffc00. In addition to this branching destination, the S1C17003 also allows designation of address 0x0 (beginning address of internal RAM) as the branching destination when debug mode is activated. The branching destination address is selected using DBADR (D8/MISC_IRAMSZ register). When the DBADR is set to "0" (default), the branching destination is set to 0xfffc00. When it is set to "1," the branching destination is set to 0x0.

- **DBADR**: Debug Base Address Select Bit in the IRAM Size Select (MISC_IRAMSZ) Register (D8/0x5326)

Adding instruction breaks

The S1C17 core supports two instruction breaks (hardware PC breaks). The S1C17003 increased this number to five, adding the control bits and registers given below.

- **IBE2**: Instruction Break #2 Enable Bit in the Debug Control (DCR) Register (D5/0xffffa0)
- **IBE3**: Instruction Break #3 Enable Bit in the Debug Control (DCR) Register (D6/0xffffa0)
- **IBE4**: Instruction Break #4 Enable Bit in the Debug Control (DCR) Register (D7/0xffffa0)
- **IBAR2[23:0]**: Instruction Break Address #2 Bits in the Instruction Break Address (IBAR2) Register 2 (D[23:0]/0xffffb8)
- **IBAR3[23:0]**: Instruction Break Address #3 Bits in the Instruction Break Address (IBAR3) Register 3 (D[23:0]/0xffffbc)
- **IBAR4[23:0]**: Instruction Break Address #4 Bits in the Instruction Break Address (IBAR4) Register 4 (D[23:0]/0xffffd0)

To use five hardware PC breaks (including four user breaks, and one reserved), the S1C17 Software Integrated Development Environment GNU17 (ver. 1.2.1 or later) must be installed.

24.4 Control Register Details

Table 24.4.1: Debug register list

Address	Register name		Function
0x5322	MISC_OSC1	OSC1 Peripheral Control Register	OSC1 operation peripheral function setting for debugging
0x5326	MISC_IRAMSZ	IRAM Size Select Register	IRAM size selection
0xffff90	DBRAM	Debug RAM Base Register	Debug RAM base address display
0xffffa0	DCR	Debug Control Register	Debug control
0xffffb8	IBAR2	Instruction Break Address Register 2	Instruction break address #2 setting
0xffffbc	IBAR3	Instruction Break Address Register 3	Instruction break address #3 setting
0xffffd0	IBAR4	Instruction Break Address Register 4	Instruction break address #4 setting

The debug registers are described in detail below. These are 8-bit registers.

Note:
- **When data is written to the registers, the "Reserved" bits must always be written as 0 and not 1.**
- **For debug registers not described here, refer to the *S1C17 Core Manual*.**

0x5322: OSC1 Peripheral Control Register (MISC_OSC1)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
OSC1 Peripheral Control Register (MISC_OSC1)	**0x5322** (16 bits)	D15–1	**–**	reserved	–				–	–	0 when being read.
		D0	**O1DBG**	OSC1 peripheral control in debug mode	1	Run	0	Stop	0	R/W	

D[7:1] Reserved

D0 O1DBG: OSC1 Peripheral Control in Debug Mode Bit

Sets OSC1 peripheral circuit operation in debug mode.

1 (R/W): Operate

0 (R/W): Stop (default)

OSC1 peripheral circuit refers to the following peripheral circuits that operate using the OSC1 clock.

- Clock timer
- Watchdog timer
- Stopwatch timer
- 8-bit OSC1 timer

0x5326: IRAM Size Select Register (MISC_IRAMSZ)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
IRAM Size Select Register (MISC_IRAMSZ)	0x5326 (16 bits)	D15–9	–	reserved	–	–	–	0 when being read.
		D8	DBADR	Debug base address select	1 0x0 0 0xfffc00	0	R/W	
		D7–2	–	reserved	–	–	–	0 when being read.
		D1–0	IRAMSZ[1:0]	IRAM size select	IRAMSZ[1:0] / Read cycle 0x3 2KB 0x2 4KB 0x1 8KB 0x0 12KB	0x0	R/W	

D[15:9] Reserved

D8 DBADR: Debug Base Address Select Bit

Selects the address to branch to in the event of a debug interrupt.
1(R/W): 0x0
0(R/W): 0xfffc00 (default)

D[7:2] Reserved

D[1:0] IRAMSZ[1:0]: IRAM Size Select Bits

Selects the size of the internal RAM to be used.

Table 24.4.2 Selecting the size of internal RAM

IRAMSZ[1:0]	Internal RAM size
0x3	2KB
0x2	4KB
0x1	8KB
0x0	12KB

(Default: 0x2)

Note: The IRAM Size Select Register is write-protected. To rewrite this register, the write-protection must be overridden by writing 0x96 to the MISC Protect Register (0x5324). Normally, the MISC Protect Register (0x5324) should be set to a value other than 0x96, except when rewriting the IRAM Size Select Register. Unnecessary rewriting of the IRAM Size Select Register may result in system malfunctions.

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	0x0fc0	0x0fc0	R	

D[31:24] Not used (Fixed at 0)

D[23:0] DBRAM[23:0]: Debug RAM Base Address Bits

Read-only register containing the initial address of the debugging work area (64 bytes).

0xffffa0: Debug Control Register (DCR)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Debug Control Register (DCR)	0xffffa0 (8 bits)	D7	IBE4	Instruction break #4 enable	1	Enable	0	Disable	0	R/W	
		D6	IBE3	Instruction break #3 enable	1	Enable	0	Disable	0	R/W	
		D5	IBE2	Instruction break #2 enable	1	Enable	0	Disable	0	R/W	
		D4	DR	Debug request flag	1	Occurred	0	Not occurred	0	R/W	Reset by writing 1.
		D3	IBE1	Instruction break #1 enable	1	Enable	0	Disable	0	R/W	
		D2	IBE0	Instruction break #0 enable	1	Enable	0	Disable	0	R/W	
		D1	SE	Single step enable	1	Enable	0	Disable	0	R/W	
		D0	DM	Debug mode	1	Debug mode	0	User mode	0	R	

D7 IBE4: Instruction Break #4 Enable Bit

Permits or prohibits instruction break #4.
1(R/W): Permit
0(R/W): Prohibit (default)

If this bit is set to 1, the instruction fetch address and the value set in the Instruction Break Address Register 4 (0xffffd0) are compared. If they match, an instruction break is generated. If this bit is set to 0, no comparison is performed.

D6 IBE3: Instruction Break #3 Enable Bit

Permits or prohibits instruction break #3.
1(R/W): Permit
0(R/W): Prohibit (default)

If this bit is set to 1, the instruction fetch address and the value set in the Instruction Break Address Register 3 (0xffffbc) are compared. If they match, an instruction break is generated. If this bit is set to 0, no comparison is performed.

D5 IBE2: Instruction Break #2 Enable Bit

Permits or prohibits instruction break #2.
1(R/W): Permit
0(R/W): Prohibit (default)

If this bit is set to 1, the instruction fetch address and the value set in the Instruction Break Address Register 2 (0xffffb8) are compared. If they match, an instruction break is generated. If this bit is set to 0, no comparison is performed.

D4 DR: Debug Request Flag

Indicates the presence or absence of an external debug request.
1(R): Request generated
0(R): None (default)
1(W): Resets flag
0(W): Invalid

This flag is cleared (reset to 0) when 1 is written. It must be cleared before the debug processing routine is terminated by the retd instruction.

D3 IBE1: Instruction Break #1 Enable Bit

Permits or prohibits instruction break #1.
1(R/W): Permit
0(R/W): Prohibit (default)

If this bit is set to 1, the instruction fetch address and the value set in the Instruction Break Address Register 1 (0xffffb4) are compared. If they match, an instruction break is generated. If this bit is set to 0, no comparison is performed.

D2 **IBE0: Instruction Break #0 Enable Bit**

Permits or prohibits instruction break #0.
1(R/W): Permit
0(R/W): Prohibit (default)

If this bit is set to 1, the instruction fetch address and the value set in the Instruction Break Address Register 0 (0xffffb0) are compared. If they match, an instruction break is generated. If this bit is set to 0, no comparison is performed.

D1 **SE: Single Step Enable Bit**

Permits or prohibits single-step operations.
1(R/W): Permit
0(R/W): Prohibit (default)

D0 **DM: Debug Mode Bit**

Indicates the processor operating mode (debug mode or user mode).
1(R): Debug mode
0(R): User mode (default)

0xffffb8: Instruction Break Address Register 2 (IBAR2)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Instruction Break Address Register 2 (IBAR2)	**0xffffb8** (32 bits)	D31–24	**–**	reserved	–	–	–	0 when being read.
		D23–0	**IBAR2[23:0]**	Instruction break address #2 IBAR223 = MSB IBAR20 = LSB	0x0 to 0xffffff	0x0	R/W	

D[31:24] Reserved

D[23:0] IBAR2[23:0]: Instruction Break Address #2 Bits

Sets instruction break address #2. (default: 0x000000)

0xffffbc: Instruction Break Address Register 3 (IBAR3)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Instruction Break Address Register 3 (IBAR3)	**0xffffbc** (32 bits)	D31–24	–	reserved	–	–	–	0 when being read.
		D23–0	**IBAR3[23:0]**	Instruction break address #3 IBAR323 = MSB IBAR30 = LSB	0x0 to 0xffffff	0x0	R/W	

D[31:24] Reserved

D[23:0] IBAR3[23:0]: Instruction Break Address #3 Bits

Sets instruction break address #3. (default: 0x000000)

0xffffd0: Instruction Break Address Register 4 (IBAR4)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Instruction Break Address Register 4 (IBAR4)	**0xffffd0** (32 bits)	D31–24	**–**	reserved	–	–	–	0 when being read.
		D23–0	**IBAR4[23:0]**	Instruction break address #4 IBAR423 = MSB IBAR40 = LSB	0x0 to 0xffffff	0x0	R/W	

D[31:24] Reserved

D[23:0] IBAR4[23:0]: Instruction Break Address #4 Bits

Sets instruction break address #4. (default: 0x000000)

25 Multiplier/Divider

25.1 Overview

The S1C17003 incorporates a coprocessor that provides signed/unsigned 16 x 16 bit multiplication functions, 16 ÷ 16 bit division functions, and signed 16 x 16 bit + 32-bit Product-sum calculation (MAC, Multiplyord Accumulator) functions enabling overflow detection.
Use of these functions is discussed below.

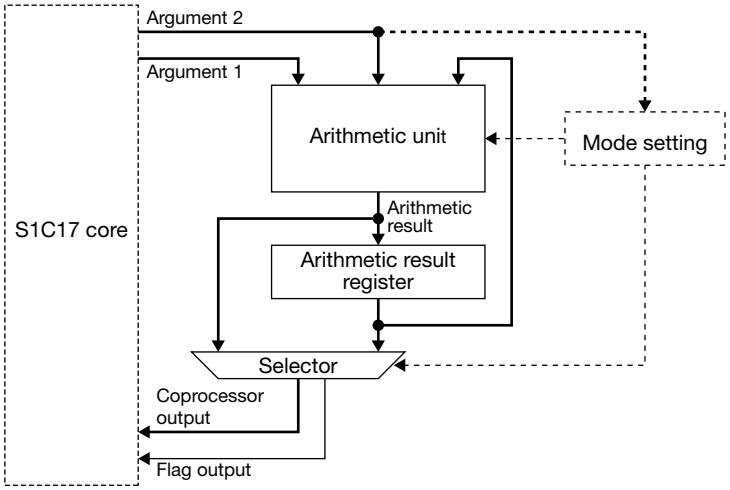


Figure 25.1.1: Multiplier/divider block diagram

Table 25.1.1: Arithmetic cycles

Operation	Cycles
Multiplication	1 cycle
Product-sum calculation	1 cycle
Division	17 to 20 cycles

25.2 Operating Mode and Output Mode

The multiplier/divider operates in accordance with the operating mode specified by the application program. The multiplier/divider supports six different operations, as shown in Table 25.2.1.
The multiplication, division, and MAC arithmetic results are 32-bit data. This means the S1C17 core cannot read out results in a single access cycle. The output mode is provided to specify whether the first 16 bits or last 16 bits of the multiplier/divider arithmetic results are read out.
Specify the operating and output modes by writing 7-bit data to the multiplier/divider internal mode setting register. Use the "`ld.cw`" instruction for 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)

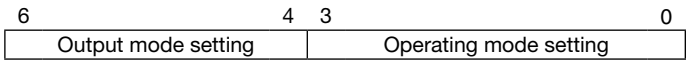


Figure 25.2.1: Mode setting registers

Table 25.2.1: Mode setting

Setting (D[6:4])	Output mode	Setting (D[3:0])	Operating mode
0x0	Last 16-bit output mode Last 16 bits of the arithmetic results are read out as coprocessor output.	0x0	Initialization mode 0 Clears the arithmetic results register to 0x0.
0x1	First 16-bit output mode First 16 bits of the arithmetic results are read out as coprocessor output.	0x1	Initialization mode 1 Loads the 16-bit arithmetic augend into the last 16 bits of the arithmetic results register.
0x2 to 0x7	Reserved	0x2	Initialization mode 2 Loads the 32-bit arithmetic augend into the arithmetic results register.
		0x3	Arithmetic results reading mode Outputs the arithmetic results register data without performing calculations.
		0x4	Unsigned multiplication mode Performs unsigned multiplication.
		0x5	Signed multiplication mode Performs signed multiplication.
		0x6	Reserved
		0x7	Signed MAC mode Performs signed MAC multiplication.
		0x8	Unsigned division mode Performs unsigned division.
		0x9	Signed division mode Performs signed division.
		0xa to 0xf	Reserved

25.3 Multiplication

The multiplication function executes "A (32 bits) = B (16 bits) x C (16 bits)."
To perform multiplication, set the operating mode to 0x4 (unsigned multiplication) or 0x5 (signed multiplication). Next, transfer the 16-bit multiplicand (B) and 16-bit multiplier (C) to the multiplier/divider using the "`ld.ca`" instruction. Half of the arithmetic result (16 bits, A [15:0] or A[31:16], depending on output mode) is returned to the CPU register, together with the flag status.
The remaining half of the arithmetic result is read out by setting the multiplier/divider to arithmetic result reading mode.

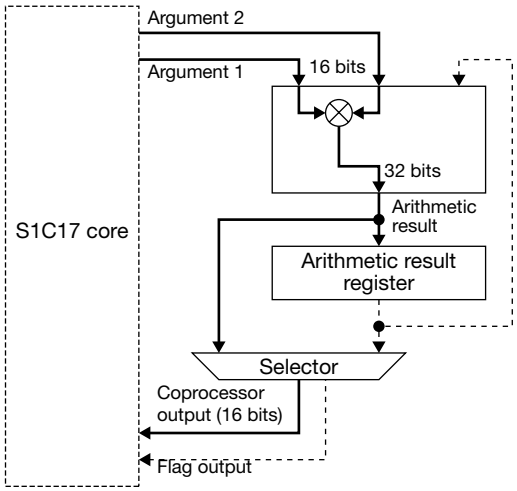


Figure 25.3.1: Multiplier mode data paths

Table 25.3.1: Multiplier mode operations

Mode setting	Instruction	Operation	Flag	Remarks
0x04 or 0x05	`ld.ca %rd,%rs`	res[31:0] ← %rd × %rs %rd ← res[15:0]	psr (CVZN) ← 0b0000	The arithmetic result register retains arithmetic results until the results are overwritten by another 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: Arithmetic result register

Examples:

```
ld.cw %r0,0x4   ; Mode setting (unsigned multiplication mode & last 16 bit output mode)
ld.ca %r0,%r1   ; Executes "res = %r0 x %r1" and loads the last 16 bits of the result to %r0 register.
ld.cw %r0,0x13  ; Mode setting (arithmetic result reading mode & first 16 bit output mode)
ld.ca %r1,%r0   ; Loads the first 16 bits of the result to %r1 register.
```

25.4 Division

The division function executes "A (16 bits) = B (16 bits) ÷ C (16 bits), D (16 bits) = Remainder."
To perform a division, set the operating mode to 0x8 (unsigned division) or 0x9 (signed division). Next, transfer the 16-bit dividend (B) and 16-bit divisor (C) to the multiplier/divider using the "ld.ca" instruction. The quotient will be placed in the lower 16 bits of the arithmetic result register, while the remainder is placed in the upper 16 bits. When the calculation is completed, the 16 bits corresponding to the quotient or remainder as specified in the output mode and the flag status are returned to the CPU register. The other 16 bits of the arithmetic result can be read out by setting the multiplier/divider to arithmetic result reading mode.

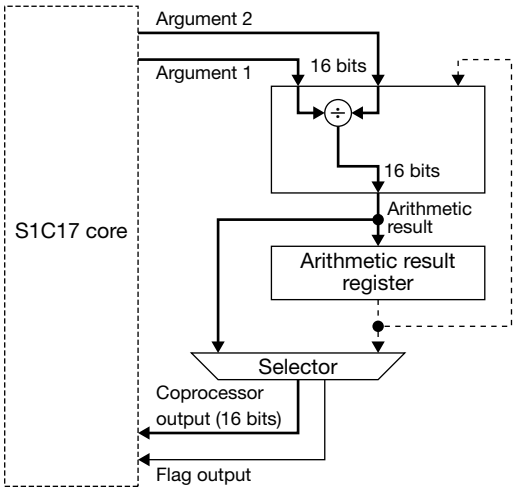


Figure 25.4.1 Division mode data path

Table 25.4.1 Division mode operations

Mode setting	Instruction	Operation	Flag	Remarks
0x08 or 0x09	`ld.ca %rd,%rs`	res[31:0] ← %rd ÷ %rs %rd ← res[15:0] (quotient)	psr (CVZN) ← 0b0000	The arithmetic result register retains the calculated result until it is overwritten by the result of another arithmetic operation.
	(`ext` *imm9*) `ld.ca %rd,`*imm7*	res[31:0] ← %rd ÷ *imm7/16* %rd ← res[15:0] (quotient)		
0x18 or 0x19	`ld.ca %rd,%rs`	res[31:0] ← %rd ÷ %rs %rd ← res[31:16] (remainder)		
	(`ext` *imm9*) `ld.ca %rd,`*imm7*	res[31:0] ← %rd ÷ *imm7/16* %rd ← res[31:16] (remainder)		

res: Arithmetic result register

Example:

```
ld.cw %r0,0x8   ; Mode setting (unsigned division mode & lower 16-bit output mode)
ld.ca %r0,%r1   ; Executes "res = %r0 + %r1" and loads the lower 16 bits (quotient) of the result to the
                  %r0 register.
ld.cw %r0,0x13  ; Mode setting (arithmetic result reading mode & upper 16-bit output mode)
ld.ca %r1,%r0   ; Loads the upper 16 bits (remainder) of the result to the %r1 register.
```

25.5 Product-sum Operation

The Product-sum operation function executes "A (32 bits) = B (16 bits) x C (16 bits) + A (32 bits)."
The initial value (A) must be set to the arithmetic result register before performing Product-sum operations.
To clear the arithmetic result register (A = 0), set the operating mode to 0x0. There is no need to send 0x0 to the multiplier/divider using separate instructions.
To load 16-bit or 32-bit values to the arithmetic result register, set the operating mode to 0x1 (16 bits) or 0x2 (32 bits). Next, transfer the initial value to the multiplier/divider using the "`ld.cf`" instruction.

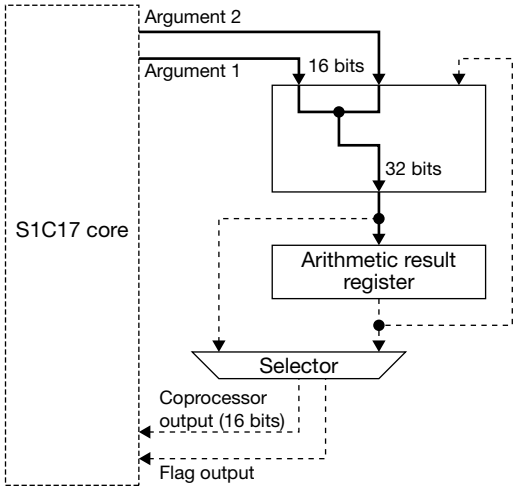


Figure 25.5.1: Initialization mode data paths

Table 25.5.1: Arithmetic result register initialization

Mode setting	Instruction	Operation	Remarks
0x0	–	res[31:0] ← 0x0	Initializes using operating mode settings only (no data transfer).
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: Arithmetic result register

To perform MAC operations, set the operating mode to 0x7 (signed MAC). Next, transfer the 16-bit multiplicand (B) and 16-bit multiplier (C) to the multiplier/divider using the "`ld.ca`" instruction. Half of the arithmetic result (16 bits, A [15:0] or A[31:16], depending on output mode) is returned to the CPU register together with the flag status. The remaining half of the arithmetic result is read out by setting the multiplier/divider to arithmetic result reading mode.
The PSR overflow flag (V) is set to 1 by the arithmetic results. Other flags are cleared to 0.
Transfer the required number of multiplicands and multipliers to continue MAC operations without switching to arithmetic result reading mode. In this case, there is no need to set to MAC mode each time data is sent.

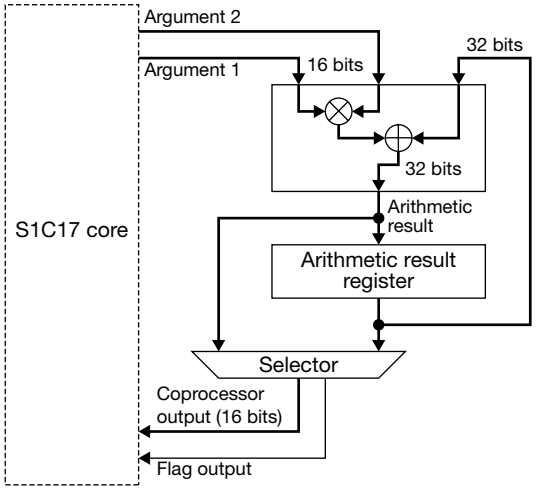


Figure 25.5.2: MAC mode data paths

Table 25.5.2: MAC mode operations

Mode setting	Instruction	Operation	Flag	Remarks
0x07	`ld.ca %rd,%rs`	res[31:0] ← %rd × %rs + res[31:0] %rd ← res[15:0]	If overflow occurs psr (CVZN) ← 0b0100 Other cases psr (CVZN) ← 0b0000	The arithmetic result register retains arithmetic results until the results are overwritten by another 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: Arithmetic result register

Examples:

```
ld.cw %r0,0x7  ;  Mode setting (signed MAC mode & last 16 bit output mode)
ld.ca %r0,%r1  ;  Executes "res = %r0 x %r1 + res" and loads the last 16 bits of the result to %r0 register.
ld.cw %r0,0x13 ;  Mode setting (arithmetic result reading mode & first 16 bit output mode)
ld.ca %r1,%r0  ;  Loads first 16 bits of the result to %r1 register.
```

Overflow flag (V) setting conditions

If the multiplication result sign, arithmetic result register sign, and arithmetic result sign satisfy the following conditions in MAC operations, an overflow occurs, and the overflow flag (V) is set to 1.

Table 25.5.3: Overflow flag (V) setting conditions

Mode setting	Multiplication result sign	Arithmetic result register sign	Arithmetic result sign
0x07	0 (Positive)	0 (Positive)	1 (Negative)
0x07	1 (Negative)	1 (Negative)	0 (Positive)

An overflow occurs if positive values are summed giving a negative result in MAC operations or if negative values are summed giving a positive result. The result is retained in the coprocessor until the overflow flag (V) is cleared.

Overflow flag (V) clear conditions

The overflow flag (V) set is cleared if the "`ld.ca`" instruction is executed for MAC operation without causing an overflow or if the "`ld.ca`" or "`ld.cf`" instruction is executed in other than arithmetic result reading mode.

25.6 Arithmetic Results Reading

Since the "ld.ca" instruction cannot load 32-bit arithmetic results to the CPU register, multiplication and Product -sum operation return half of the arithmetic result (16 bits, A[15:0] or A[31:16], depending on output mode) together with the flag status to the CPU register. The remaining half of the arithmetic result is read by setting the multiplier to arithmetic result reading mode. The arithmetic result register retains arithmetic results until the results are overwritten by another operation.

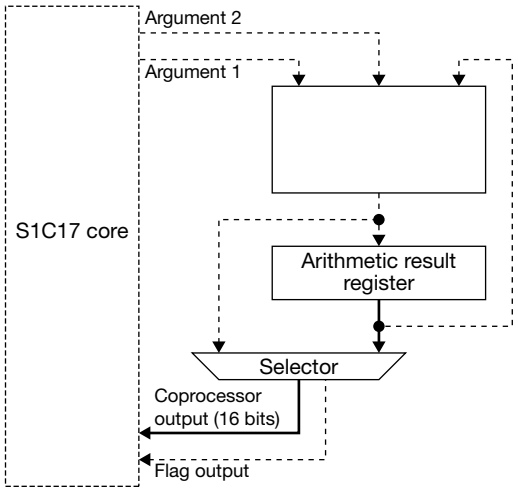


Figure 25.6.1: Arithmetic result reading mode data paths

Table 25.6.1: Arithmetic result reading mode operations

Mode setting	Instruction	Operation	Flag	Remarks
0x03	ld.ca %rd,%rs	%rd ← res[15:0]	psr (CVZN) ← 0b0000	This operating mode does not affect the arithmetic 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: Arithmetic result register

26 Electrical Characteristics

26.1 Absolute Maximum Ratings

Item	Code	Condition	Rating	Units
Core power supply voltage	LV_{DD}		-0.3 to 2.5	V
I/O power supply voltage	HV_{DD}		-0.3 to 4.0	V
Analog power supply voltage	AV_{DD}		-0.3 to 4.0	V
Input voltage	HV_I		-0.3 to HV_{DD} + 0.5	V
	LV_I		-0.3 to LV_{DD} + 0.5	
Analog input voltage	AV_I		-0.3 to AV_{DD} + 0.3	V
Output voltage	V_O		-0.3 to HV_{DD} + 0.5	V
High-level output current	I_{OH}	1 pin	-10	mA
		Total for all pins	-40	mA
Low-level output current	I_{OL}	1 pin	-10	mA
		Total for all pins	-40	mA
Storage temperature	Tstg		-65 to 150	°C
Soldering temperature/time	Tsol		260°C, 10 s (leads)	–

26.2 Recommended Operating Conditions

Item	Code	Condition	Min.	Typ.	Max.	Units
Core power supply voltage	LV_{DD}		1.65	–	1.95	V
I/O power supply voltage	HV_{DD}		1.65	–	3.6	V
Analog power supply voltage*1	AV_{DD}	P17, P[22:20]=When analog signal input	2.7	–	3.6	V
		P17, P[22:20]=When digital signal input	1.65	–	3.6	V
Input voltage	HV_I		V_{SS}	–	HV_{DD}	V
	LV_I		V_{SS}	–	LV_{DD}	V
Analog input voltage	AV_I		V_{SS}	–	AV_{DD}	V
Operating frequency	fosc3	Crystal/ceramic oscillation	5	–	20	MHz
	fosc1	Crystal oscillation	–	32.768	–	kHz
Operating temperature	Ta		-40	–	85	°C
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 P17, P[22:20] 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.

26.3 Current Consumption

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

Item	Code	Condition	Min.	Typ.	Max.	Units
SLEEP current consumption	I_{SLP}	OSC1: Off, OSC3: Off	–	1	–	µA
HALT current consumption	I_{HALT1}	OSC1: 32kHz, OSC3: Off	–	3.3	–	µA
	I_{HALT2}	OSC1: 32kHz, OSC3: 1MHz	–	180	–	µA
	I_{HALT3}	OSC1: 32kHz, OSC3: 4MHz	–	300	–	µA
	I_{HALT4}	OSC1: 32kHz, OSC3: 8MHz	–	500	–	µA
	I_{HALT5}	OSC1: 32kHz, OSC3: 20MHz	–	1.3	–	mA
Execution current *1 consumption	I_{EXE1}	OSC1: 32kHz, OSC3: Off	–	8.0	–	µA
	I_{EXE2}	OSC1: 32kHz, OSC3: 1MHz	–	350	–	µA
	I_{EXE3}	OSC1: 32kHz, OSC3: 4MHz	–	840	–	µA
	I_{EXE4}	OSC1: 32kHz, OSC3: 8MHz	–	1.6	–	mA
	I_{EXE5}	OSC1: 32kHz, OSC3: 20MHz	–	4.0	–	mA
ADC operating *2 current consumption	I_{ADC}	ADC enable	–	260	–	µA

*1: Execution current consumption is the value for continuous operations while fetching the test program (ALU instruction 60.5%, branch instruction 17%, memory read 12%, memory write 10.5%) from flash memory.
*2: Current consumption at AV_{DD}

26.4 Input/Output Terminal Characteristics

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

Item	Code	Condition	Min.	Typ.	Max.	Units
Input leakage current	I_{LI}	HV_{DD}=Max, V_I=Max.	-5	–	5	µA
High level output current	I_{OH}	V_{OH} = HV_{DD}-0.4V HV_{DD} = Min.	-1	–	–	mA
Low level output current	I_{OL}	V_{OL} = 0.4V HV_{DD} = Min.	1	–	–	mA
Positive trigger input voltage	V_{T1+}	HV_{DD}=Max.	0.6	–	1.4	V
Negative trigger input voltage	V_{T1-}	HV_{DD}=Min.	0.3	–	1.1	V
Hysteresis voltage	ΔV	HV_{DD}=Min.	0.02	–	–	V
Input pull-up resistance	R_{IN1}	Pxx, V_I=0V	48	120	300	kΩ
	R_{IN2}	#RESET, V_I=0V	96	240	600	kΩ
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 stated: LV_{DD} = 1.8V±0.15V, HV_{DD} = 2.7 to 3.6V, V_{SS} = 0V, Ta = -40 to 85°C

Item	Code	Condition	Min.	Typ.	Max.	Units
Input leakage current	I_{LI}	HV_{DD}=Max, V_I=Max.	-5	–	5	V
High level output current	I_{OH}	V_{OH}=HV_{DD}-0.4V, HV_{DD}=Min.	- 1.7	–	–	mA
Low level output current	I_{OL}	V_{OL}=0.4V, HV_{DD}=Min.	1.7	–	–	mA
Positive trigger input voltage	V_{T1+}	HV_{DD}=Max.	1.2	–	2.7	V
Negative trigger input voltage	V_{T1-}	HV_{DD}=Min.	0.5	–	1.8	V
Hysteresis voltage	ΔV	HV_{DD}=Min.	0.2	–	–	V
Input pull-up resistance	R_{IN1}	Pxx, V_I=0V	25	50	144	kΩ
	R_{IN2}	#RESET, V_I=0V	50	100	288	kΩ
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

26.5 A/D Converter Characteristics

Unless otherwise stated: 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	Code	Condition	Min.	Typ.	Max.	Units
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.

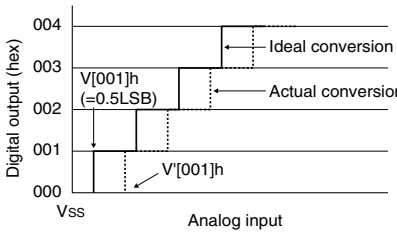
A/D conversion error

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

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

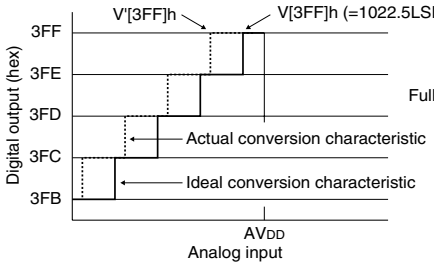
$$1LSB' = \frac{V'[3FF]h - V'[001]h}{2^{10} - 2}$$

■ **Zero scale error**



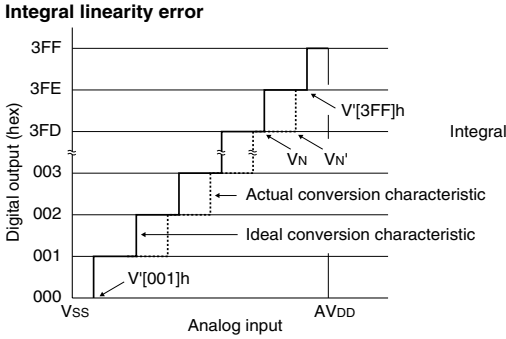
$$\text{Zero scale error } E_{ZS} = \frac{(V'[001]h - 0.5LSB') - (V[001]h - 0.5LSB)}{1LSB} \text{ [LSB]}$$

■ **Full scale error**



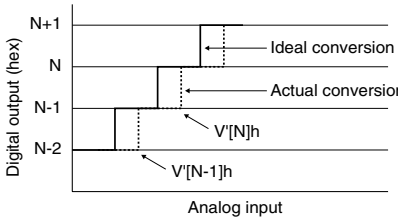
$$\text{Full scale error } E_{FS} = \frac{(V'[3FF]h + 0.5LSB') - (V[3FF]h + 0.5LSB)}{1LSB} \text{ [LSB]}$$

■ **Integral linearity error**



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

■ **Differential linearity error**



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

26.6 SPI Characteristics

Master mode

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

Item	Code	Min.	Typ.	Max.	Units
SPICLK cycle time	t_{SPCK}	200	–	–	ns
SDI setup time	t_{SDS}	60	–	–	ns
SDI hold time	t_{SDH}	10	–	–	ns
SDO output delay time	t_{SDO}	–	–	20	ns

Slave mode

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

Item	Code	Min.	Typ.	Max.	Units
SPICLK cycle time	t_{SPCK}	200	–	–	ns
SDI setup time	t_{SDS}	10	–	–	ns
SDI hold time	t_{SDH}	10	–	–	ns
SDO output delay time	t_{SDO}	–	–	60	ns

26.7 I²C Characteristics

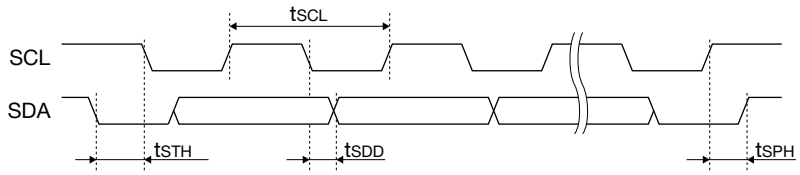


Figure 26.7.1: I²C Timing

Unless otherwise stated: V_{DD} = 1.8 to 3.6V, V_{SS} = 0V, Ta = -25 to 70°C

Item	Code	Min.	Typ.	Max.	Units
SCL cycle time	t_{SCL}	2500			ns
Start condition hold time	t_{STH}	1/f_{SYS}			ns
Data output delay time	t_{SDD}	1/f_{SYS}			ns
Stop condition hold time	t_{SPH}	1/f_{SYS}			ns

∗ f_{SYS}: System operation clock frequency

26.8 Oscillation Circuit 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 oscillator

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

Item	Code	Condition	Min.	Typ.	Max.	Units
Oscillation start time	t_{STA1}	When the recommended parts shown in Section "27. Basic External Wiring Diagram" are used			1	s

OSC3 crystal oscillator

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

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

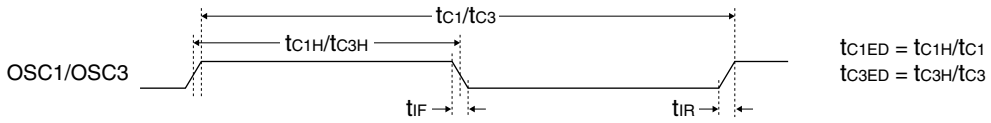
Item	Code	Condition	Min.	Typ.	Max.	Units
Oscillation start time	t_{STA3}	When the recommended parts shown in Section "27. Basic External Wiring Diagram" are used			10	ms

OSC3 ceramic oscillator

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

Item	Code	Condition	Min.	Typ.	Max.	Units
Oscillation start time	t_{STA3}	When the recommended parts shown in Section "27. Basic External Wiring Diagram" are used			1	ms

26.9 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

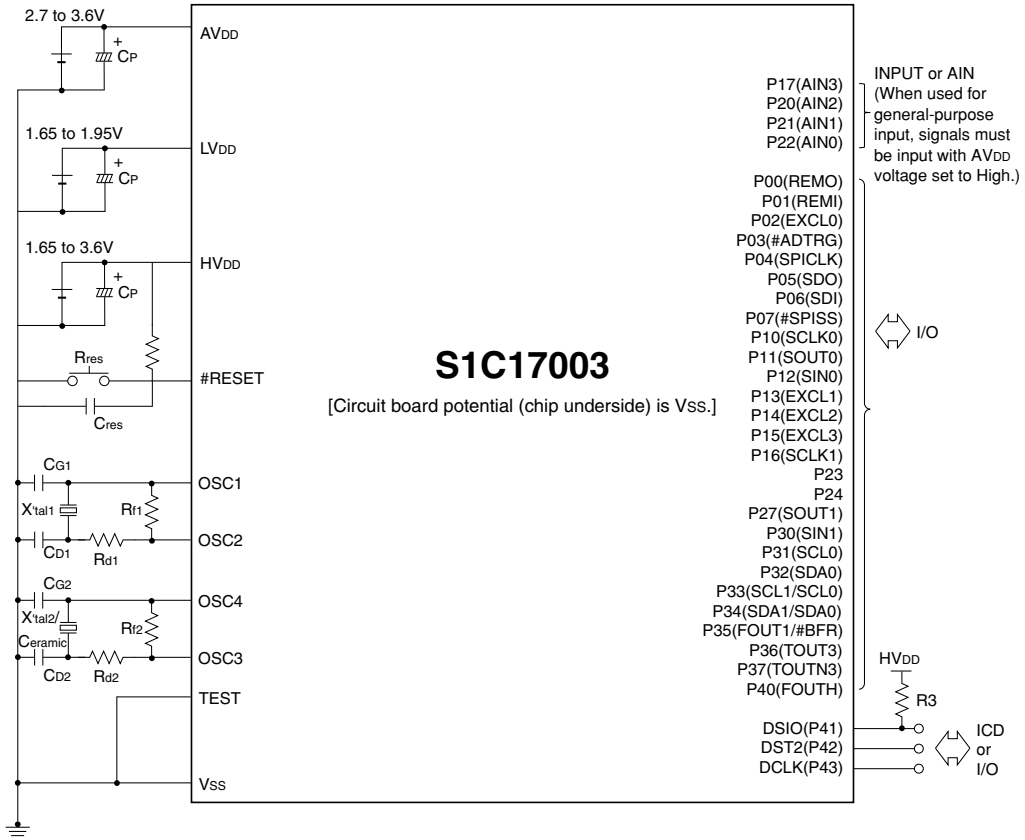
Item	Symbol	Min.	Typ.	Max.	Unit
OSC1 external clock cycle time	t_{C1}		30.51		µs
OSC1 external clock input duty	t_{C1ED}	45		55	%
OSC1 external clock input rise time	t_{IF}			5	ns
OSC1 external clock input fall time	t_{IR}			5	ns

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}			5	ns
OSC3 external clock input fall time	t_{IR}			5	ns

27 Basic External Connection 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.

28 Package

28.1 TQFP12-64 pin package

(Units: mm)

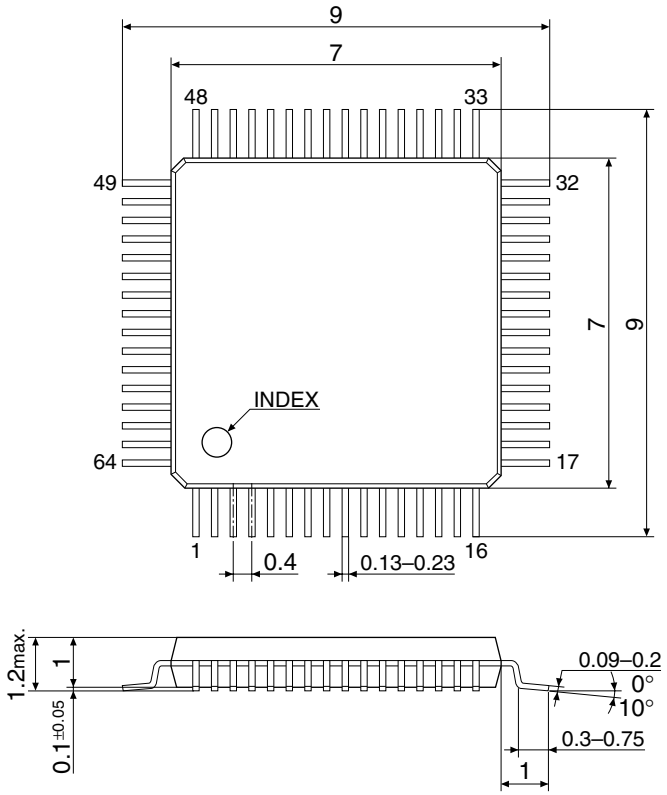
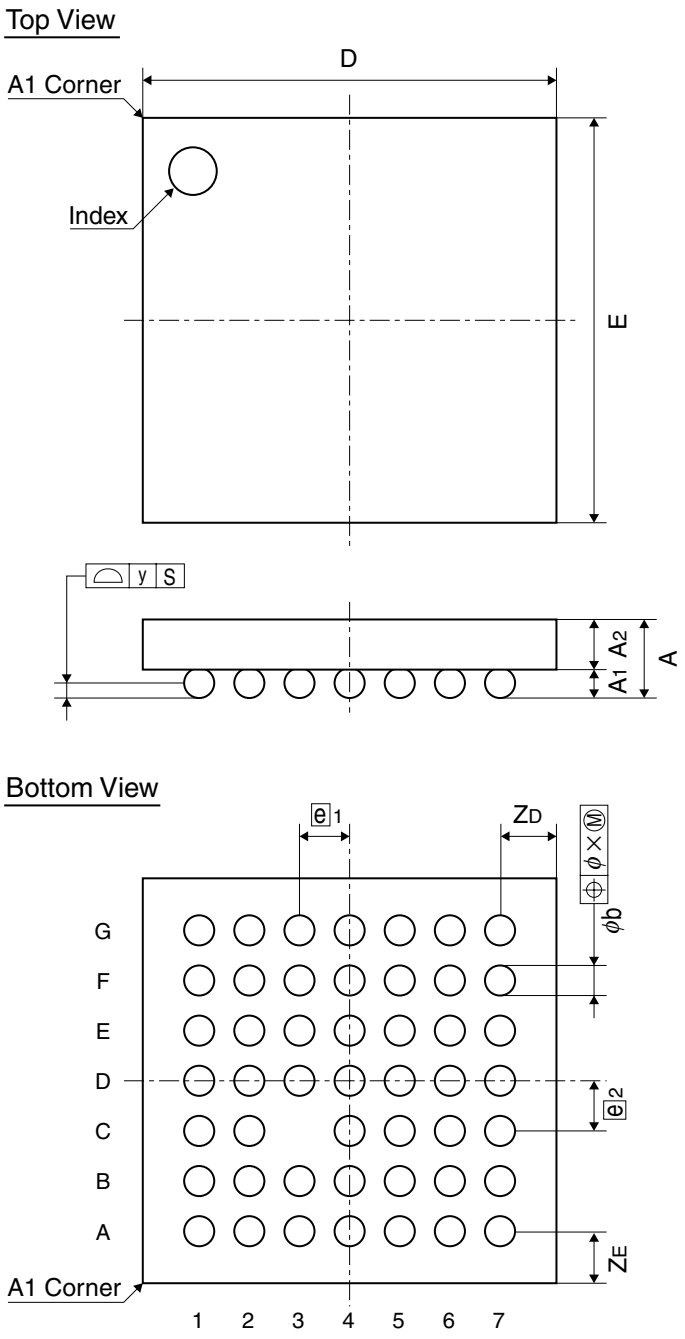


Figure 28.1.1 TQFP12-64pin package scale

28.2 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	–
e_1	–	0.40	–
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 28.2.1 WCSP-48 package scale

28.3 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)

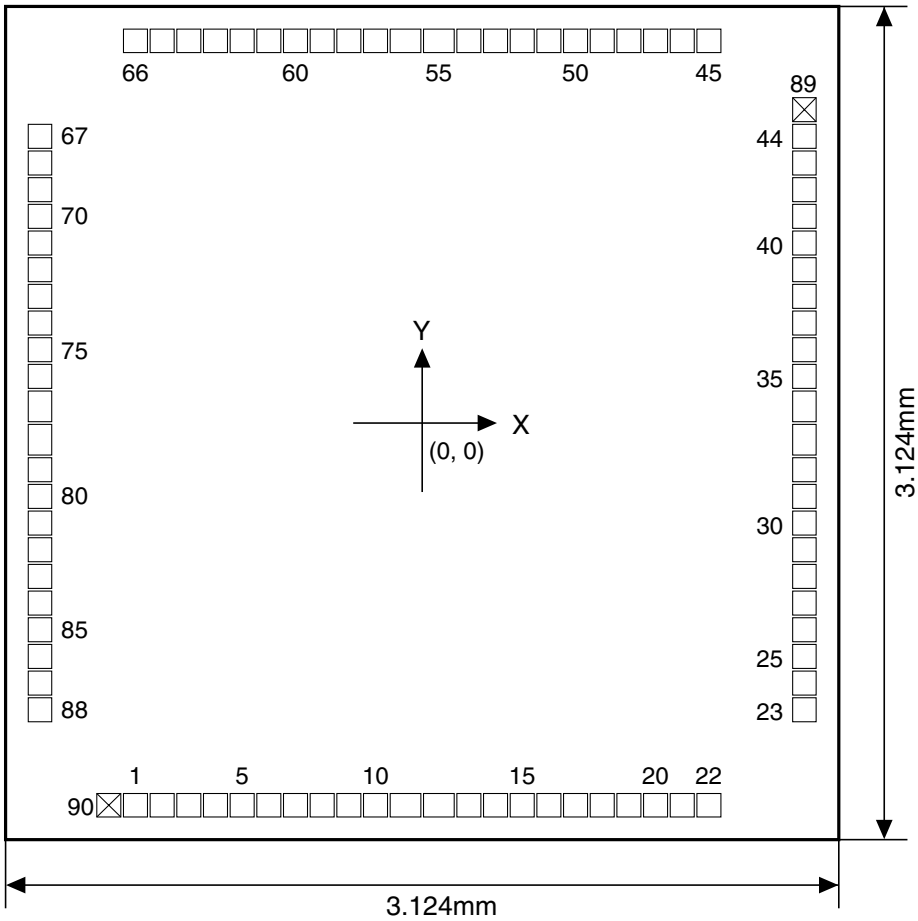
   Thermal resistance = 90–100°C/W

   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.

28.4 Pad Layout

28.4.1 Diagram of Pad Layout



Chip thickness: 400 µm

Appendix A: I/O Register List

Peripheral circuit	Address	Register name		Function
Prescaler (8-bit device)	0x4020	PSC_CTL	Prescaler Control Register	Prescaler start/stop control
	0x4021~0x403f	–	–	Reserved
UART (with IrDA) Ch.0 (8-bit device)	0x4100	UART_ST0	UART Ch.0 Status Register	Transfer, buffer, error status display
	0x4101	UART_TXD0	UART Ch.0 Transmit Data Register	Transmission data
	0x4102	UART_RXD0	UART Ch.0 Receive Data Register	Receiving data
	0x4103	UART_MOD0	UART Ch.0 Mode Register	Transfer data format setting
	0x4104	UART_CTL0	UART Ch.0 Control Register	Data transfer control
	0x4105	UART_EXP0	UART Ch.0 Expansion Register	IrDA mode setting
	0x4106~0x411f	–	–	Reserved
UART (with IrDA) Ch.1 (8-bit device)	0x4120	UART_ST1	UART Ch.1 Status Register	Transfer, buffer, error status display
	0x4121	UART_TXD1	UART Ch.1 Transmit Data Register	Transmission data
	0x4122	UART_RXD1	UART Ch.1 Receive Data Register	Receiving data
	0x4123	UART_MOD1	UART Ch.1 Mode Register	Transfer data format setting
	0x4124	UART_CTL1	UART Ch.1 Control Register	Data transfer control
	0x4125	UART_EXP1	UART Ch.1 Expansion Register	IrDA mode setting
	0x4126~0x413f	–	–	Reserved
8-bit timer (with F mode) Ch.0 (16-bit device)	0x4200	T8F_CLK0	8-bit Timer Ch.0 Input Clock Select Register	Prescaler output clock selection
	0x4202	T8F_TR0	8-bit Timer Ch.0 Reload Data Register	Reload data setting
	0x4204	T8F_TC0	8-bit Timer Ch.0 Counter Data Register	Counter data
	0x4206	T8F_CTL0	8-bit Timer Ch.0 Control Register	Timer mode setting and timer RUN/STOP
	0x4208	T8F_INT0	8-bit Timer Ch.0 Interrupt Control Register	Interrupt control
	0x420a~0x421f	–	–	Reserved
16-bit timer Ch.0 (16-bit device)	0x4220	T16_CLK0	16-bit Timer Ch.0 Input Clock Select Register	Prescaler output clock selection
	0x4222	T16_TR0	16-bit Timer Ch.0 Reload Data Register	Reload data setting
	0x4224	T16_TC0	16-bit Timer Ch.0 Counter Data Register	Counter data
	0x4226	T16_CTL0	16-bit Timer Ch.0 Control Register	Timer mode setting and timer RUN/STOP
	0x4228	T16_INT0	16-bit Timer Ch.0 Interrupt Control Register	Interrupt control
	0x422a~0x423f	–	–	Reserved
16-bit timer Ch.1 (16-bit device)	0x4240	T16_CLK1	16-bit Timer Ch.1 Input Clock Select Register	Prescaler output clock selection
	0x4242	T16_TR1	16-bit Timer Ch.1 Reload Data Register	Reload data setting
	0x4244	T16_TC1	16-bit Timer Ch.1 Counter Data Register	Counter data
	0x4246	T16_CTL1	16-bit Timer Ch.1 Control Register	Timer mode setting and timer RUN/STOP
	0x4248	T16_INT1	16-bit Timer Ch.1 Interrupt Control Register	Interrupt control
	0x424a~0x425f	–	–	Reserved
16-bit timer Ch.2 (16-bit device)	0x4260	T16_CLK2	16-bit Timer Ch.2 Input Clock Select Register	Prescaler output clock selection
	0x4262	T16_TR2	16-bit Timer Ch.2 Reload Data Register	Reload data setting
	0x4264	T16_TC2	16-bit Timer Ch.2 Counter Data Register	Counter data
	0x4266	T16_CTL2	16-bit Timer Ch.2 Control Register	Timer mode setting and timer RUN/STOP
	0x4268	T16_INT2	16-bit Timer Ch.2 Interrupt Control Register	Interrupt control
	0x426a~0x427f	–	–	Reserved
8-bit timer (with F mode) Ch.1 (16-bit device)	0x4280	T8F_CLK1	8-bit Timer Ch.1 Input Clock Select Register	Prescaler output clock selection
	0x4282	T8F_TR1	8-bit Timer Ch.1 Reload Data Register	Reload data setting
	0x4284	T8F_TC1	8-bit Timer Ch.1 Counter Data Register	Counter data
	0x4286	T8F_CTL1	8-bit Timer Ch.1 Control Register	Timer mode setting and timer RUN/STOP
	0x4288	T8F_INT1	8-bit Timer Ch.1 Interrupt Control Register	Interrupt control
	0x428a~0x429f	–	–	Reserved
Interrupt controller (16-bit device)	0x4300~0x4304	–	–	Reserved
	0x4306	ITC_LV0	Interrupt Level Setup Register 0	P0/P1 interrupt level setting
	0x4308	ITC_LV1	Interrupt Level Setup Register 1	SWT/CT interrupt level setting
	0x430a	ITC_LV2	Interrupt Level Setup Register 2	T8OSC1 interrupt level setting
	0x430c	ITC_LV3	Interrupt Level Setup Register 3	T16E Ch.0 interrupt level setting
	0x430e	ITC_LV4	Interrupt Level Setup Register 4	T8F Ch.0/Ch.1, T16 Ch.0 interrupt level setting
	0x4310	ITC_LV5	Interrupt Level Setup Register 5	T16 Ch.1/Ch.2 interrupt level setting
	0x4312	ITC_LV6	Interrupt Level Setup Register 6	UART Ch.0, I^2C/UART Ch.1 slave interrupt level setting
	0x4314	ITC_LV7	Interrupt Level Setup Register 7	SPI/I^2C master interrupt level setting
	0x4316	ITC_LV8	Interrupt Level Setup Register 8	REMC interrupt level setting
	0x4318	ITC_LV9	Interrupt Level Setup Register 9	ADC10SA interrupt setting
	0x431a~0x431f	–	–	Reserved

Peripheral circuit	Address	Register name		Function
SPI (16-bit device)	0x4320	SPI_ST	SPI Status Register	Transfer and buffer status display
	0x4322	SPI_TXD	SPI Transmit Data Register	Transmission data
	0x4324	SPI_RXD	SPI Receive Data Register	Receiving data
	0x4326	SPI_CTL	SPI Control Register	SPI mode and data transfer permission setting
	0x4328~0x433f	–	–	Reserved
I^2C (master) (16-bit device)	0x4340	I2C_EN	I^2C Enable Register	I^2C module enable
	0x4342	I2C_CTL	I^2C Control Register	I^2C control and transfer status display
	0x4344	I2C_DAT	I^2C Data Register	Transfer data
	0x4346	I2C_ICTL	I^2C Interrupt Control Register	I^2C interrupt control
	0x4348~0x435f	–	–	Reserved
I^2C (salave) (16-bit device)	0x4360	I2CS_TRNS	I^2C Slave Transfer Data Write Register	Transmission data
	0x4362	I2CS_RECV	I^2C Slave Receive Data Read Register	Receiving data
	0x4364	I2CS_SADRS	I^2C Slave Address Set Register	Slave address data
	0x4366	I2CS_CTL	I^2C Slave Control Register	I^2C slave control
	0x4368	I2CS_STAT	I^2C Slave Status Register	I^2C slave status display
	0x436a	I2CS_ASTAT	I^2C Slave Access Status Register	I^2C slave transfer status display
	0x436c	I2CS_ICTL	I^2C Slave Interrupt Control Register	I^2C slave interrupt control
	0x4370~0x437f	–	–	Reserved
Clock timer (8-bit device)	0x5000	CT_CTL	Clock Timer Control Register	Timer reset and RUN/STOP control
	0x5001	CT_CNT	Clock Timer Counter Register	Counter data
	0x5002	CT_IMSK	Clock Timer Interrupt Mask Register	Interrupt mask setting
	0x5003	CT_IFLG	Clock Timer Interrupt Flag Register	Interrupt occurrence status display/reset
	0x5004~0x501f	–	–	Reserved
Stopwatch timer (8-bit device)	0x5020	SWT_CTL	Stopwatch Timer Control Register	Timer reset and RUN/STOP control
	0x5021	SWT_BCNT	Stopwatch Timer BCD Counter Register	BCD counter data
	0x5022	SWT_IMSK	Stopwatch Timer Interrupt Mask Register	Interrupt mask setting
	0x5023	SWT_IFLG	Stopwatch Timer Interrupt Flag Register	Interrupt occurrence status display/reset
	0x5024~0x503f	–	–	Reserved
Watchdog timer (8-bit device)	0x5040	WDT_CTL	Watchdog Timer Control Register	Timer reset and RUN/STOP control
	0x5041	WDT_ST	Watchdog Timer Status Register	Timer mode setting and NMI status display
	0x5042~0x505f	–	–	Reserved
Oscillator circuit (8-bit device)	0x5060	OSC_SRC	Clock Source Select Register	Clock source selection
	0x5061	OSC_CTL	Oscillation Control Register	Oscillation control
	0x5062	OSC_NFEN	Noise Filter Enable Register	Noise filter ON/OFF
	0x5063	–	–	Reserved
	0x5064	OSC_FOUT	FOUT Control Register	Clock external output control
	0x5065	OSC_T8OSC1	T8OSC1 Clock Control Register	8-bit OSC1 timer clock setting
	0x5066~0x507f	–	–	Reserved
Clock generator (8-bit device)	0x5080	CLG_PCLK	PCLK Control Register	PCLK feed control
	0x5081	CLG_CCLK	CCLK Control Register	CCLK division ratio setting
	0x5082~0x509f	–	–	Reserved
8-bit OSC1 timer (8-bit device)	0x50c0	T8OSC1_CTL	8-bit OSC1 Timer Control Register	Timer mode setting and timer RUN/STOP
	0x50c1	T8OSC1_CNT	8-bit OSC1 Timer Counter Data Register	Counter data
	0x50c2	T8OSC1_CMP	8-bit OSC1 Timer Compare Data Register	Compare data setting
	0x50c3	T8OSC1_IMSK	8-bit OSC1 Timer Interrupt Mask Register	Interrupt mask setting
	0x50c4	T8OSC1_IFLG	8-bit OSC1 Timer Interrupt Flag Register	Interrupt occurrence status display/reset
	0x50c5	T8OSC1_DUTY	8-bit OSC1 Timer PWM Data Register	PWM output data setting
	0x50c6~0x50df	–	–	Reserved

Peripheral circuit	Address	Register name		Function
P port & port MUX (8-bit device)	0x5200	P0_IN	P0 Port Input Data Register	P0 port input data
	0x5201	P0_OUT	P0 Port Output Data Register	P0 port output data
	0x5202	P0_OEN	P0 Port Output Enable Register	P0 port output enable
	0x5203	P0_PU	P0 Port Pull-up Control Register	P0 port pull-up control
	0x5204	–	–	Reserved
	0x5205	P0_IMSK	P0 Port Interrupt Mask Register	P0 port interrupt mask setting
	0x5206	P0_EDGE	P0 Port Interrupt Edge Select Register	P0 port interrupt edge selection
	0x5207	P0_IFLG	P0 Port Interrupt Flag Register	P0 port interrupt occurrence status display/reset
	0x5208	P0_CHAT	P0 Port Chattering Filter Control Register	P0 port chattering filter control
	0x5209	P0_KRST	P0 Port Key-Entry Reset Configuration Register	P0 port key entry reset setting
	0x520a	P0_IEN	P0 Port Input Enable Register	P0 port input enable
	0x520b~0x520f	–	–	Reserved
	0x5210	P1_IN	P1 Port Input Data Register	P1 port input data
	0x5211	P1_OUT	P1 Port Output Data Register	P1 port output data
	0x5212	P1_OEN	P1 Port Output Enable Register	P1 port output enable
	0x5213	P1_PU	P1 Port Pull-up Control Register	P1 port pull-up control
	0x5214	–	–	Reserved
	0x5215	P1_IMSK	P1 Port Interrupt Mask Register	P1 port interrupt mask setting
	0x5216	P1_EDGE	P1 Port Interrupt Edge Select Register	P1 port interrupt edge selection
	0x5217	P1_IFLG	P1 Port Interrupt Flag Register	P1 port interrupt occurrence status display/reset
	0x5218	P1_CHAT	P1 Port Chattering Filter Control Register	P1 port chattering filter control
	0x5219	–	–	Reserved
	0x521a	P1_IEN	P1 Port Input Enable Register	P1 port input enable
	0x521b~0x521f	–	–	Reserved
	0x5220	P2_IN	P2 Port Input Data Register	P2 port input data
	0x5221	P2_OUT	P2 Port Output Data Register	P2 port output data
	0x5222	P2_OEN	P2 Port Output Enable Register	P2 port output enable
	0x5223	P2_PU	P2 Port Pull-up Control Register	P2 port pull-up control
	0x5224~0x5229	–	–	Reserved
	0x522a	P2_IEN	P2 Port Input Enable Register	P2 port input enable
	0x522b~0x522f	–	–	Reserved
	0x5230	P3_IN	P3 Port Input Data Register	P3 port input data
	0x5231	P3_OUT	P3 Port Output Data Register	P3 port output data
	0x5232	P3_OEN	P3 Port Output Enable Register	P3 port output enable
	0x5233	P3_PU	P3 Port Pull-up Control Register	P3 port pull-up control
	0x5234~0x5239	–	–	Reserved
	0x523a	P3_IEN	P3 Port Input Enable Register	P3 port input enable
	0x523b~0x523f	–	–	Reserved
	0x5240	P4_IN	P4 Port Input Data Register	P4 port input data
	0x5241	P4_OUT	P4 Port Output Data Register	P4 port output data
	0x5242	P4_OEN	P4 Port Output Enable Register	P4 port output enable
	0x5243	P4_PU	P4 Port Pull-up Control Register	P4 port pull-up control
	0x5244~0x5249	–	–	Reserved
	0x524a	P4_IEN	P4 Port Input Enable Register	P4 port input enable
	0x524b~0x527f	–	–	Reserved
	0x52a0~0x52a1	P0_PMUX	P0 Port Function Select Register	P0 port function selection
	0x52a2~0x52a3	P1_PMUX	P1 Port Function Select Register	P1 port function selection
	0x52a4~0x52a5	P2_PMUX	P2 Port Function Select Register	P2 port function selection
	0x52a6~0x52a7	P3_PMUX	P3 Port Function Select Register	P3 port function selection
	0x52a8	P4_PMUX	P4 Port Function Select Register	P4 port function selection
	0x52a9~0x52bf	–	–	Reserved
PWM timer Ch.0 (16-bit device)	0x5300	T16E_CA	PWM Timer Compare Data A Register	Compare data A setting
	0x5302	T16E_CB	PWM Timer Compare Data B Register	Compare data B setting
	0x5304	T16E_TC	PWM Timer Counter Data Register	Counter data
	0x5306	T16E_CTL	PWM Timer Control Register	Timer mode setting and timer RUN/STOP
	0x5308	T16E_CLK	PWM Timer Input Clock Select Register	Prescaler output clock selection
	0x530a	T16E_IMSK	PWM Timer Interrupt MASK Register	Interrupt factor mask selection
	0x530c	T16E_IFLG	PWM Timer Interrupt Flag Register	Interrupt factor checking
	0x530e~0x531f	–	–	Reserved
MISC register (16-bit device)	0x5320	MISC_FL	FLASHC/SRAMC Control Register	FLASHC/SRAMC access condition setting
	0x5322	MISC_OSC1	OSC1 Peripheral Control Register	OSC1 operation peripheral function setting for debugging
	0x5324	MISC_PROT	MISC Protect Register	MISC register write protection
	0x5326	MISC_IRAMSZ	IRAM Size Select Register	IRAM size selection
	0x5328	MISC_TTBRL	Vactor Table Address Low Register	Vector table address setting
	0x532a	MISC_TTBRH	Vector Table Address High Register	Vector table address setting
	0x532c	MISC_PSR	PSR Register	PSR status reading
	0x532e~0x533f	–	–	Reserved

Peripheral circuit	Address	Register name		Function
Remote controller (16-bit device)	0x5340	REMC_CFG	REMC Configuration Register	Clock transfer control
	0x5342	REMC_CAR	REMC Carrier Length Setup Register	Carrier H/L section length setting
	0x5344	REMC_LCNT	REMC Length Counter Register	Transfer bit and transfer data length setting
	0x5346	REMC_INT	REMC Interrupt Control Register	Interrupt control
	0x5348~0x535f	–	–	Reserved
A/D converter (16-bit device)	0x5380	ADC10_ADD	ADC10 Conversion Result Register	A/D conversion result
	0x5382	ADC10_TRG	ADC10 Trigger/Channel Select Register	Conversion Trigger/channel setting
	0x5384	ADC10_CTL	ADC10 Control/Status Register	Conversion control/status
	0x5386	ADC10_DIV	ADC10 divided frequency Register	A/D conversion clock divided frequency setting
	0x5388~0x539f	–	–	Reserved
S1C17 core I/O	0xffff84	IDIR	Processor ID Register	Processor ID display
	0xffff90	DBRAM	Debug RAM Base Register	Debugging RAM base address display
	0xffffa0	DCR	Debug Control Register	Debug control
	0xffffb8	IBAR2	Instruction Break Address Register 2	Instruction break address #2 setting
	0xffffbc	IBAR3	Instruction Break Address Register 3	Instruction break address #3 setting
	0xffffd0	IBAR4	Instruction Break Address Register 4	Instruction break address #4 setting

Note: Addresses marked as “Reserved” or unused peripheral circuit areas not marked in the table must not be accessed by application programs.

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	

0x4100–0x4124 UART (with IrDA)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
UART Ch.0 Status Register (UART_ST0)	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 Ch.0 Transmit Data Register (UART_TXD0)	0x4101 (8 bits)	D7–0	TXD[7:0]	Transmit data TXD7(6) = MSB TXD0 = LSB	0x0 to 0xff (0x7f)	0x0	R/W	
UART Ch.0 Receive Data Register (UART_RXD0)	0x4102 (8 bits)	D7–0	RXD[7:0]	Receive data in the receive data buffer RXD7(6) = MSB RXD0 = LSB	0x0 to 0xff (0x7f)	0x0	R	Older data in the buffer is read out first.
UART Ch.0 Mode Register (UART_MOD0)	0x4103 (8 bits)	D7–5	–	reserved	–	–	–	0 when being read.
		D4	CHLN	Character length	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 Ch.0 Control Register (UART_CTL0)	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 Ch.0 Expansion Register (UART_EXP0)	0x4105 (8 bits)	D7	–	reserved	–	–	–	0 when being read.
		D6–4	IRCLK[2:0]	IrDA receive detection clock select	IRCLK[2:0] 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	0x0	R/W	
		D3–1	–	reserved	–	–	–	0 when being read.
		D0	IRMD	IrDA mode select	1 On 0 Off	0	R/W	
UART Ch.1 Status Register (UART_ST1)	0x4120 (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 Ch.1 Transmit Data Register (UART_TXD1)	0x4121 (8 bits)	D7–0	TXD[7:0]	Transmit data TXD7(6) = MSB TXD0 = LSB	0x0 to 0xff (0x7f)	0x0	R/W	
UART Ch.1 Receive Data Register (UART_RXD1)	0x4122 (8 bits)	D7–0	RXD[7:0]	Receive data in the receive data buffer RXD7(6) = MSB RXD0 = LSB	0x0 to 0xff (0x7f)	0x0	R	Older data in the buffer is read out first.
UART Ch.1 Mode Register (UART_MOD1)	0x4123 (8 bits)	D7–5	–	reserved	–	–	–	0 when being read.
		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 Ch.1 Control Register (UART_CTL1)	0x4124 (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	

0x4125 UART (with IrDA)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
UART Ch.1 Expansion Register (UART_EXP1)	**0x4125** (8 bits)	D7	–	reserved	–	–	–	0 when being read.
		D6–4	**IRCLK[2:0]**	IrDA receive detection clock select	IRCLK[2:0] / 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	0x0	R/W	
		D3–1	–	reserved	–	–	–	0 when being read.
		D0	**IRMD**	IrDA mode select	1 On / 0 Off	0	R/W	

0x4200–0x4208 8-bit Timer (with Fine Mode) Ch.0

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
8-bit Timer Ch.0 Input Clock Select Register (T8F_CLK0)	**0x4200** (16 bits)	D15–4	**–**	reserved	–	–	–	0 when being read.
		D3–0	**DF[3:0]**	8-bit 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	
8-bit Timer Ch.0 Reload Data Register (T8F_TR0)	**0x4202** (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	
8-bit Timer Ch.0 Counter Data Register (T8F_TC0)	**0x4204** (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	
8-bit Timer Ch.0 Control Register (T8F_CTL0)	**0x4206** (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	
8-bit Timer Ch.0 Interrupt Control Register (T8F_INT0)	**0x4208** (16 bits)	D15–9	**–**	reserved	–	–	–	0 when being read.
		D8	**T8IE**	8-bit timer interrupt enable	1 Enable / 0 Disable	0	R/W	
		D7–1	**–**	reserved	–	–	–	0 when being read.
		D0	**T8IF**	8-bit timer interrupt flag	1 Cause of interrupt occurred / 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.

0x4220–0x4244 16-bit Timer

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
16-bit Timer Ch.0 Input Clock Select Register (T16_CLK0)	**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 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	
16-bit Timer Ch.0 Reload Data Register (T16_TR0)	**0x4222** (16 bits)	D15–0	**TR[15:0]**	16-bit timer reload data TR15 = MSB TR0 = LSB	0x0 to 0xffff	0x0	R/W	
16-bit Timer Ch.0 Counter Data Register (T16_TC0)	**0x4224** (16 bits)	D15–0	**TC[15:0]**	16-bit timer counter data TC15 = MSB TC0 = LSB	0x0 to 0xffff	0xffff	R	
16-bit Timer Ch.0 Control Register (T16_CTL0)	**0x4226** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10	**CKACTV**	External clock active level select	1 High / 0 Low	1	R/W	
		D9–8	**CKSL[1:0]**	Input clock and pulse width measurement mode select	CKSL[1:0] / Mode 0x3 reserved 0x2 Pulse width 0x1 External clock 0x0 Internal clock	0x0	R/W	
		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	
16-bit Timer Ch.0 Interrupt Control Register (T16_INT0)	**0x4228** (16 bits)	D15–9	**–**	reserved	–	–	–	0 when being read.
		D8	**T16IE**	16-bit timer interrupt enable	1 Enable / 0 Disable	0	R/W	
		D7–1	**–**	reserved	–	–	–	0 when being read.
		D0	**T16IF**	16-bit timer interrupt flag	1 Cause of interrupt occurred / 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
16-bit Timer Ch.1 Input Clock Select Register (T16_CLK1)	**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 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	
16-bit Timer Ch.1 Reload Data Register (T16_TR1)	**0x4242** (16 bits)	D15–0	**TR[15:0]**	16-bit timer reload data TR15 = MSB TR0 = LSB	0x0 to 0xffff	0x0	R/W	
16-bit Timer Ch.1 Counter Data Register (T16_TC1)	**0x4244** (16 bits)	D15–0	**TC[15:0]**	16-bit timer counter data TC15 = MSB TC0 = LSB	0x0 to 0xffff	0xffff	R	

0x4246–0x4268 16-bit Timer

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
16-bit Timer Ch.1 Control Register (T16_CTL1)	0x4246 (16 bits)	D15–11	–	reserved	–	–	–	0 when being read.
		D10	CKACTV	External clock active level select	1 High 0 Low	1	R/W	
		D9–8	CKSL[1:0]	Input clock and pulse width measurement mode select	CKSL[1:0] Mode: 0x3 reserved; 0x2 Pulse width; 0x1 External clock; 0x0 Internal clock	0x0	R/W	
		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	
16-bit Timer Ch.1 Interrupt Control Register (T16_INT1)	0x4248 (16 bits)	D15–9	–	reserved	–	–	–	0 when being read.
		D8	T16IE	16-bit timer interrupt enable	1 Enable 0 Disable	0	R/W	
		D7–1	–	reserved	–	–	–	0 when being read.
		D0	T16IF	16-bit timer interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
16-bit Timer Ch.2 Input Clock Select Register (T16_CLK2)	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: 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	
16-bit Timer Ch.2 Reload Data Register (T16_TR2)	0x4262 (16 bits)	D15–0	TR[15:0]	16-bit timer reload data TR15 = MSB TR0 = LSB	0x0 to 0xffff	0x0	R/W	
16-bit Timer Ch.2 Counter Data Register (T16_TC2)	0x4264 (16 bits)	D15–0	TC[15:0]	16-bit timer counter data TC15 = MSB TC0 = LSB	0x0 to 0xffff	0xffff	R	
16-bit Timer Ch.2 Control Register (T16_CTL2)	0x4266 (16 bits)	D15–11	–	reserved	–	–	–	0 when being read.
		D10	CKACTV	External clock active level select	1 High 0 Low	1	R/W	
		D9–8	CKSL[1:0]	Input clock and pulse width measurement mode select	CKSL[1:0] Mode: 0x3 reserved; 0x2 Pulse width; 0x1 External clock; 0x0 Internal clock	0x0	R/W	
		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	
16-bit Timer Ch.2 Interrupt Control Register (T16_INT2)	0x4268 (16 bits)	D15–9	–	reserved	–	–	–	0 when being read.
		D8	T16IE	16-bit timer interrupt enable	1 Enable 0 Disable	0	R/W	
		D7–1	–	reserved	–	–	–	0 when being read.
		D0	T16IF	16-bit timer interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.

0x4280–0x4288 8-bit Timer (with Fine Mode) Ch.1

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
8-bit Timer Ch.1 Input Clock Select Register (T8F_CLK1)	**0x4280** (16 bits)	D15–4	–	reserved	–	–	–	0 when being read.
		D3–0	**DF[3:0]**	8-bit 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	
8-bit Timer Ch.1 Reload Data Register (T8F_TR1)	**0x4282** (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	
8-bit Timer Ch.1 Counter Data Register (T8F_TC1)	**0x4284** (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	
8-bit Timer Ch.1 Control Register (T8F_CTL1)	**0x4286** (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	
8-bit Timer Ch.1 Interrupt Control Register (T8F_INT1)	**0x4288** (16 bits)	D15–9	–	reserved	–	–	–	0 when being read.
		D8	**T8IE**	8-bit timer interrupt enable	1 Enable 0 Disable	0	R/W	
		D7–1	–	reserved	–	–	–	0 when being read.
		D0	**T8IF**	8-bit timer interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.

0x4306–0x4318 Interrupt Controller

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Interrupt Level Setup Register 0 (ITC_LV0)	**0x4306** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**ILV1[2:0]**	P1 interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**ILV0[2:0]**	P0 interrupt level	0 to 7	0x0	R/W	
Interrupt Level Setup Register 1 (ITC_LV1)	**0x4308** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10-8	**ILV3[2:0]**	CT interrupt level	0 to 7	0x0	R/W	
		D7-3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**ILV2[2:0]**	SWT interrupt level	0 to 7	0x0	R/W	
Interrupt Level Setup Register 2 (ITC_LV2)	**0x430a** (16 bits)	D15–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**ILV4[2:0]**	T8OSC1 interrupt level	0 to 7	0x0	R/W	
Interrupt Level Setup Register 3 (ITC_LV3)	**0x430c** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**ILV7[2:0]**	T16E Ch.0 interrupt level	0 to 7	0x0	R/W	
		D7–0	**–**	reserved	–	–	–	0 when being read.
Interrupt Level Setup Register 4 (ITC_LV4)	**0x430e** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**ILV9[2:0]**	T16 Ch.0 interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**ILV8[2:0]**	T8F Ch.0/Ch.1 interrupt level	0 to 7	0x0	R/W	
Interrupt Level Setup Register 5 (ITC_LV5)	**0x4310** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**ILV11[2:0]**	T16 Ch.2 interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**ILV10[2:0]**	T16 Ch.1 interrupt level	0 to 7	0x0	R/W	
Interrupt Level Setup Register 6 (ITC_LV6)	**0x4312** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**ILV13[2:0]**	UART Ch.1/I^2C slave interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**ILV12[2:0]**	UART Ch.0 interrupt level	0 to 7	0x0	R/W	
Interrupt Level Setup Register 7 (ITC_LV7)	**0x4314** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**ILV15[2:0]**	I^2C Master interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**ILV14[2:0]**	SPI interrupt level	0 to 7	0x0	R/W	
Interrupt Level Setup Register 8 (ITC_LV8)	**0x4316** (16 bits)	D15–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**ILV16[2:0]**	REMC interrupt level	0 to 7	0x0	R/W	
Interrupt Level Setup Register 9 (ITC_LV9)	**0x4318** (16 bits)	D15–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**ILV18[2:0]**	ADC10SA interrupt level	0 to 7	0x0	R/W	

0x4320–0x4326 SPI

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SPI Status Register (SPI_ST)	**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 Transmit Data Register (SPI_TXD)	**0x4322** (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	**SPTDB[7:0]**	SPI transmit data buffer SPTDB7 = MSB SPTDB0 = LSB	0x0 to 0xff	0x0	R/W	
SPI Receive Data Register (SPI_RXD)	**0x4324** (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	**SPRDB[7:0]**	SPI receive data buffer SPRDB7 = MSB SPRDB0 = LSB	0x0 to 0xff	0x0	R	
SPI Control Register (SPI_CTL)	**0x4326** (16 bits)	D15–10	–	reserved	–	–	–	0 when being read.
		D9	**MCLK**	SPI clock source select	1 T16 Ch.1 0 PCLK•1/4	0	R/W	
		D8	**MLSB**	LSB/MSB first mode select	1 LSB 0 MSB	0	R/W	
		D7–6	–	reserved	–	–	–	0 when being read.
		D5	**SPRIE**	Receive data buffer full int. enable	1 Enable 0 Disable	0	R/W	
		D4	**SPTIE**	Transmit data buffer empty int. enable	1 Enable 0 Disable	0	R/W	
		D3	**CPHA**	Clock phase select	1 Data out 0 Data in	0	R/W	These bits must be
		D2	**CPOL**	Clock polarity select	1 Active L 0 Active H	0	R/W	set before setting
		D1	**MSSL**	Master/slave mode select	1 Master 0 Slave	0	R/W	SPEN to 1.
		D0	**SPEN**	SPI enable	1 Enable 0 Disable	0	R/W	

0x4340–0x4346 I²C Master

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
I²C Enable Register (I2C_EN)	**0x4340** (16 bits)	D15–1	**–**	reserved	–	–	–	0 when being read.
		D0	**I2CEN**	I²C enable	1 Enable 0 Disable	0	R/W	
I²C Control Register (I2C_CTL)	**0x4342** (16 bits)	D15–10	**–**	reserved	–	–	–	0 when being read.
		D9	**RBUSY**	Receive busy flag	1 Busy 0 Idle	0	R	
		D8	**TBUSY**	Transmit busy flag	1 Busy 0 Idle	0	R	
		D7–5	**–**	reserved	–	–	–	0 when being read.
		D4	**NSERM**	Noise remove on/off	1 On 0 Off	0	R/W	
		D3–2	**–**	reserved	–	–	–	0 when being read.
		D1	**STP**	Stop control	1 Stop 0 Ignored	0	R/W	
		D0	**STRT**	Start control	1 Start 0 Ignored	0	R/W	
I²C Data Register (I2C_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	
I²C Interrupt Control Register (I2C_ICTL)	**0x4346** (16 bits)	D15–2	**–**	reserved	–	–	–	0 when being read.
		D1	**RINTE**	Receive interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	**TINTE**	Transmit interrupt enable	1 Enable 0 Disable	0	R/W	

0x4360–0x436c I²C Slave

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
I²C Slave Transmit Data Register (I2CS_TRNS)	0x4360 (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	SDATA[7:0]	I²C slave transmit data	0–0xff	0x0	R/W	
I²C Slave Receive Data Register (I2CS_RECV)	0x4362 (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	RDATA[7:0]	I²C slave receive data	0–0xff	0x0	R	
I²C Slave Address Setup Register (I2CS_SADRS)	0x4364 (16 bits)	D15–7	–	reserved	–	–	–	0 when being read.
		D6–0	SADRS[6:0]	I²C slave address	0–0x7f	0x0	R/W	
I²C Slave Control Register (I2CS_CTL)	0x4366 (16 bits)	D15–9	–	reserved	–	–	–	0 when being read.
		D8	TBUF_CLR	I2CS_TRNS register clear	1 Clear state 0 Normal	0	R/W	
		D7	I2C_EN	I²C slave enable	1 Enable 0 Disable	0	R/W	
		D6	SOFTRESET	Software reset	1 Reset 0 Cancel	0	R/W	
		D5	NAK_ANS	NAK answer	1 NAK 0 ACK	0	R/W	
		D4	BFREQ_EN	Bus free request enable	1 Enable 0 Disable	0	R/W	
		D3	CLKSTR_EN	Clock stretch On/Off	1 On 0 Off	0	R/W	
		D2	NF_EN	Noise filter On/Off	1 On 0 Off	0	R/W	
		D1	ASDET_EN	Async.address detection On/Off	1 On 0 Off	0	R/W	
		D0	COM_MODE	I²C slave communication mode	1 Active 0 Standby	0	R/W	NAK response when standby
I²C Slave Status Register (I2CS_STAT)	0x4368 (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7	BSTAT	Bus status transition	1 Changed 0 Unchanged	0	R	
		D6	–	reserved	–	–	–	0 when being read.
		D5	TXUDF	Transmit data underflow	1 Occurred 0 Not occurred	0	R/W	Reset by writing 1.
			RXOVF	Receive data overflow				
		D4	BFREQ	Bus free request	1 Occurred 0 Not occurred	0	R/W	
		D3	DMS	Output data mismatch	1 Error 0 Normal	0	R/W	
		D2	ASDET	Async. address detection status	1 Detected 0 Not detected	0	R/W	
		D1	DA_NAK	NAK receive status	1 NAK 0 ACK	0	R/W	
		D0	DA_STOP	STOP condition detect	1 Detected 0 Not detected	0	R/W	
I²C Slave Access Status Register (I2CS_ASTAT)	0x436a (16 bits)	D15–5	–	reserved	–	–	–	0 when being read.
		D4	RXRDY	Receive data ready	1 Ready 0 Not ready	0	R	
		D3	TXEMP	Transmit data empty	1 Empty 0 Not empty	0	R	
		D2	BUSY	I²C bus status	1 Busy 0 Free	0	R	
		D1	SELECTED	I²C slave select status	1 Selected 0 Not selected	0	R	
		D0	R/W	Read/write direction	1 Output 0 Input	0	R	
I²C Slave Interrupt Control Register (I2CS_ICTL)	0x436c (16 bits)	D15–3	–	reserved	–	–	–	0 when being read.
		D2	BSTAT_IEN	Bus status interrupt enable	1 Enable 0 Disable	0	R/W	
		D1	RXRDY_IEN	Receive interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	TXEMP_IEN	Transmit interrupt enable	1 Enable 0 Disable	0	R/W	

0x5000–0x5003 Clock Timer

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Clock Timer Control Register (CT_CTL)	0x5000 (8 bits)	D7–5	–	reserved	–	–	–	0 when being read.
		D4	CTRST	Clock timer reset	1 Reset 0 Ignored	0	W	
		D3–1	–	reserved	–	–	–	
		D0	CTRUN	Clock timer run/stop control	1 Run 0 Stop	0	R/W	
Clock Timer Counter Register (CT_CNT)	0x5001 (8 bits)	D7–0	CTCNT[7:0]	Clock timer counter value	0x0 to 0xff	0	R	
Clock Timer Interrupt Mask Register (CT_IMSK)	0x5002 (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.
		D3	CTIE32	32 Hz interrupt enable	1 Enable 0 Disable	0	R/W	
		D2	CTIE8	8 Hz interrupt enable	1 Enable 0 Disable	0	R/W	
		D1	CTIE2	2 Hz interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	CTIE1	1 Hz interrupt enable	1 Enable 0 Disable	0	R/W	
Clock Timer Interrupt Flag Register (CT_IFLG)	0x5003 (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.
		D3	CTIF32	32 Hz interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D2	CTIF8	8 Hz interrupt flag		0	R/W	
		D1	CTIF2	2 Hz interrupt flag		0	R/W	
		D0	CTIF1	1 Hz interrupt flag		0	R/W	

0x5020–0x5023 Stopwatch Timer

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Stopwatch Timer Control Register (SWT_CTL)	**0x5020** (8 bits)	D7–5	–	reserved	–	–	–	0 when being read.
		D4	**SWTRST**	Stopwatch timer reset	1 Reset 0 Ignored	0	W	
		D3–1	–	reserved	–	–	–	
		D0	**SWTRUN**	Stopwatch timer run/stop control	1 Run 0 Stop	0	R/W	
Stopwatch Timer BCD Counter Register (SWT_BCNT)	**0x5021** (8 bits)	D7–4	**BCD10[3:0]**	1/10 sec. BCD counter value	0 to 9	0	R	
		D3–0	**BCD100[3:0]**	1/100 sec. BCD counter value	0 to 9	0	R	
Stopwatch Timer Interrupt Mask Register (SWT_IMSK)	**0x5022** (8 bits)	D7–3	–	reserved	–	–	–	0 when being read.
		D2	**SIE1**	1 Hz interrupt enable	1 Enable 0 Disable	0	R/W	
		D1	**SIE10**	10 Hz interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	**SIE100**	100 Hz interrupt enable	1 Enable 0 Disable	0	R/W	
Stopwatch Timer Interrupt Flag Register (SWT_IFLG)	**0x5023** (8 bits)	D7–3	–	reserved	–	–	–	0 when being read.
		D2	**SIF1**	1 Hz interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D1	**SIF10**	10 Hz interrupt flag		0	R/W	
		D0	**SIF100**	100 Hz interrupt flag		0	R/W	

0x5040–0x5041 Watchdog Timer

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Watchdog Timer Control Register (WDT_CTL)	0x5040 (8 bits)	D7–5	–	reserved	–	–	–	0 when being read.
		D4	WDTRST	Watchdog timer reset	1 Reset 0 Ignored	0	W	
		D3–0	WDTRUN[3:0]	Watchdog timer run/stop control	Other than 1010 Run 1010 Stop	1010	R/W	
Watchdog Timer Status Register (WDT_ST)	0x5041 (8 bits)	D7–2	–	reserved	–	–	–	0 when being read.
		D1	WDTMD	NMI/Reset mode select	1 Reset 0 NMI	0	R/W	
		D0	WDTST	NMI status	1 NMI occurred 0 Not occurred	0	R	

0x5060–0x5065 Oscillator

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Clock Source Select Register (OSC_SRC)	**0x5060** (8 bits)	D7–2	**–**	reserved	–	–	–	0 when being read.
		D1	**–**	reserved	–	1	R	1 when being read.
		D0	**CLKSRC**	System clock source select	1 OSC1 0 HSCLK	0	R/W	
Oscillation Control Register (OSC_CTL)	**0x5061** (8 bits)	D7–6	**–**	reserved	–	0	–	0 when being read.
		D5–4	**OSC3WT[1:0]**	OSC3 wait cycle select	OSC3WT[1:0] / Wait cycle 0x3 / 128 cycles 0x2 / 256 cycles 0x1 / 512 cycles 0x0 / 1024 cycles	0x0	R/W	
		D3–2	**–**	reserved	–	–	–	0 when being read.
		D1	**OSC1EN**	OSC1 enable	1 Enable 0 Disable	0	R/W	
		D0	**OSC3EN**	OSC3 enable	1 Enable 0 Disable	1	R/W	
Noise Filter Enable Register (OSC_NFEN)	**0x5062** (8 bits)	D7–2	**–**	reserved	–	–	–	0 when being read.
		D1	**RSTFE**	Reset noise filter enable	1 Enable 0 Disable	1	R/W	
		D0	**NMIFE**	NMI noise filter enable	1 Enable 0 Disable	0	R/W	
FOUT Control Register (OSC_FOUT)	**0x5064** (8 bits)	D7–4	**–**	reserved	–	–	–	0 when being read.
		D3–2	**FOUTHD [1:0]**	FOUTH clock division ratio select	FOUTHD[1:0] / Division ratio 0x3 / reserved 0x2 / HSCLK•1/4 0x1 / HSCLK•1/2 0x0 / HSCLK•1/1	0x0	R/W	Note: FOUTHD must be operated while FOUT1E and FOUT1E are disabled.
		D1	**FOUTHE**	FOUTH output enable	1 Enable 0 Disable	0	R/W	
		D0	**FOUT1E**	FOUT1 output enable	1 Enable 0 Disable	0	R/W	
T8OSC1 Clock Control Register (OSC_T8OSC1)	**0x5065** (8 bits)	D7–4	**–**	reserved	–	–	–	0 when being read.
		D3–1	**T8O1CK[2:0]**	T8OSC1 clock division ratio select	T8O1CK[2:0] / Division ratio 0x7–0x6 / reserved 0x5 / OSC1•1/32 0x4 / OSC1•1/16 0x3 / OSC1•1/8 0x2 / OSC1•1/4 0x1 / OSC1•1/2 0x0 / OSC1•1/1	0x0	R/W	Note: T8O1CK must be operated while T8O1CE is disabled.
		D0	**T8O1CE**	T8OSC1 clock output enable	1 Enable 0 Disable	0	R/W	

0x5080–0x5081 Clock Generator

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
PCLK Control Register (CLG_PCLK)	**0x5080** (8 bits)	D7–2	–	reserved	–		–	–	0 when being read.
		D1–0	**PCKEN[1:0]**	PCLK enable	PCKEN[1:0]	PCLK supply	0x3	R/W	
					0x3	Enable			
					0x2	Not allowed			
					0x1	Not allowed			
					0x0	Disable			
CCLK Control Register (CLG_CCLK)	**0x5081** (8 bits)	D7–2	–	reserved	–		–	–	0 when being read.
		D1–0	**CCLKGR[1:0]**	CCLK clock gear ratio select	CCLKGR[1:0]	Gear ratio	0x0	R/W	
					0x3	1/8			
					0x2	1/4			
					0x1	1/2			
					0x0	1/1			

0x50c0–0x50c5 8-bit OSC1 Timer

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
8-bit OSC1 Timer Control Register (T8OSC1_CTL)	**0x50c0** (8 bits)	D7–5	–	reserved	–	–	–	0 when being read.
		D4	**T8ORST**	Timer reset	1 Reset 0 Ignored	0	W	
		D3–2	–	reserved	–	–	–	
		D1	**T8ORMD**	Count mode select	1 One shot 0 Repeat	0	R/W	
		D0	**T8ORUN**	Timer run/stop control	1 Run 0 Stop	0	R/W	
8-bit OSC1 Timer Counter Data Register (T8OSC1_CNT)	**0x50c1** (8 bits)	D7–0	**T8OCNT[7:0]**	Timer counter data T8OCNT7 = MSB T8OCNT0 = LSB	0x0 to 0xff	0x0	R	
8-bit OSC1 Timer Compare Data Register (T8OSC1_CMP)	**0x50c2** (8 bits)	D7–0	**T8OCMP[7:0]**	Compare data T8OCMP7 = MSB T8OCMP0 = LSB	0x0 to 0xff	0x0	R/W	
8-bit OSC1 Timer Interrupt Mask Register (T8OSC1_IMSK)	**0x50c3** (8 bits)	D7–1	–	reserved	–	–	–	0 when being read.
		D0	**T8OIE**	8-bit OSC1 timer interrupt enable	1 Enable 0 Disable	0	R/W	
8-bit OSC1 Timer Interrupt Flag Register (T8OSC1_IFLG)	**0x50c4** (8 bits)	D7–1	–	reserved	–	–	–	0 when being read.
		D0	**T8OIF**	8-bit OSC1 timer interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
8-bit OSC1 Timer PWM Duty Data Register (T8OSC1_DUTY)	**0x50c5** (8 bits)	D7–0	**T8ODTY[7:0]**	PWM output duty data T8ODTY7 = MSB T8ODTY0 = LSB	0x0 to 0xff	0x0	R/W	

0x5200–0x5213 P Port & Port MUX

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
P0 Port Input Data Register (P0_IN)	**0x5200** (8 bits)	D7–0	**P0IN[7:0]**	P0[7:0] port input data	1: 1 (H) 0: 0 (L)	×	R	
P0 Port Output Data Register (P0_OUT)	**0x5201** (8 bits)	D7–0	**P0OUT[7:0]**	P0[7:0] port output data	1: 1 (H) 0: 0 (L)	0	R/W	
P0 Port Output Enable Register (P0_IO)	**0x5202** (8 bits)	D7–0	**P0OEN[7:0]**	P0[7:0] port output enable	1: Enable 0: Disable	0	R/W	
P0 Port Pull-up Control Register (P0_PU)	**0x5203** (8 bits)	D7–0	**P0PU[7:0]**	P0[7:0] port pull-up enable	1: Enable 0: Disable	1 (0xff)	R/W	
P0 Port Interrupt Mask Register (P0_IMSK)	**0x5205** (8 bits)	D7–0	**P0IE[7:0]**	P0[7:0] port interrupt enable	1: Enable 0: Disable	0	R/W	
P0 Port Interrupt Edge Select Register (P0_EDGE)	**0x5206** (8 bits)	D7–0	**P0EDGE[7:0]**	P0[7:0] port interrupt edge select	1: Falling edge 0: Rising edge	0	R/W	
P0 Port Interrupt Flag Register (P0_IFLG)	**0x5207** (8 bits)	D7–0	**P0IF[7:0]**	P0[7:0] port interrupt flag	1: Cause of interrupt occurred 0: Cause of interrupt not occurred	0	R/W	Reset by writing 1.
P0 Port Chattering Filter Control Register (P0_CHAT)	**0x5208** (8 bits)	D7	**–**	reserved	–	–	–	0 when being read.
		D6–4	**P0CF2[2:0]**	P0[7:4] chattering filter time	P0CF2[2:0] / Filter time 0x7: 16384/f_{PCLK} 0x6: 8192/f_{PCLK} 0x5: 4096/f_{PCLK} 0x4: 2048/f_{PCLK} 0x3: 1024/f_{PCLK} 0x2: 512/f_{PCLK} 0x1: 256/f_{PCLK} 0x0: None	0x0	R/W	
		D3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**P0CF1[2:0]**	P0[3:0] chattering filter time	P0CF1[2:0] / Filter time 0x7: 16384/f_{PCLK} 0x6: 8192/f_{PCLK} 0x5: 4096/f_{PCLK} 0x4: 2048/f_{PCLK} 0x3: 1024/f_{PCLK} 0x2: 512/f_{PCLK} 0x1: 256/f_{PCLK} 0x0: None	0x0	R/W	
P0 Port Key-Entry Reset Configuration Register (P0_KRST)	**0x5209** (8 bits)	D7–2	**–**	reserved	–	–	–	0 when being read.
		D1–0	**P0KRST[1:0]**	P0 port key-entry reset configuration	P0KRST[1:0] / Configuration 0x3: P0[3:0] = 0 0x2: P0[2:0] = 0 0x1: P0[1:0] = 0 0x0: Disable	0x0	R/W	
P0 Port Input Enable Register (P0_IEN)	**0x520a** (8 bits)	D7–0	**P0IEN[7:0]**	P0[7:0] port input enable	1: Enable 0: Disable	0xff	R/W	
P1 Port Input Data Register (P1_IN)	**0x5210** (8 bits)	D7–0	**P1IN[7:0]**	P1[7:0] port input data	1: 1 (H) 0: 0 (L)	×	R	
P1 Port Output Data Register (P1_OUT)	**0x5211** (8 bits)	D7–0	**P1OUT[7:0]**	P1[7:0] port output data	1: 1 (H) 0: 0 (L)	0	R/W	
P1 Port Output Enable Register (P1_IO)	**0x5212** (8 bits)	D7–0	**P1OEN[7:0]**	P1[7:0] port output enable	1: Enable 0: Disable	0	R/W	
P1 Port Pull-up Control Register (P1_PU)	**0x5213** (8 bits)	D7–0	**P1PU[7:0]**	P1[7:0] port pull-up enable	1: Enable 0: Disable	1 (0xff)	R/W	

0x5215–0x523a P Port & Port MUX

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
P1 Port Interrupt Mask Register (P1_IMSK)	**0x5215** (8 bits)	D7–0	**P1IE[7:0]**	P1[7:0] port interrupt enable	1 Enable 0 Disable	0	R/W	
P1 Port Interrupt Edge Select Register (P1_EDGE)	**0x5216** (8 bits)	D7–0	**P1EDGE[7:0]**	P1[7:0] port interrupt edge select	1 Falling edge 0 Rising edge	0	R/W	
P1 Port Interrupt Flag Register (P1_IFLG)	**0x5217** (8 bits)	D7–0	**P1IF[7:0]**	P1[7:0] port interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
P1 Port Chattering Filter Control Register (P1_CHAT)	**0x5218** (8 bits)	D7	**–**	reserved	–	–	–	0 when being read.
		D6–4	**P1CF2[2:0]**	P1[7:4] chattering filter time	P0CF2[2:0] Filter time 0x7 16384/f_{PCLK} 0x6 8192/f_{PCLK} 0x5 4096/f_{PCLK} 0x4 2048/f_{PCLK} 0x3 1024/f_{PCLK} 0x2 512/f_{PCLK} 0x1 256/f_{PCLK} 0x0 None	0 0x0	R/W R/W	
		D3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**P1CF1[2:0]**	P1[3:0] chattering filter time	P0CF1[2:0] Filter time 0x7 16384/f_{PCLK} 0x6 8192/f_{PCLK} 0x5 4096/f_{PCLK} 0x4 2048/f_{PCLK} 0x3 1024/f_{PCLK} 0x2 512/f_{PCLK} 0x1 256/f_{PCLK} 0x0 None	0x0	R/W	
P1 Port Input Enable Register (P1_IEN)	**0x521a** (8 bits)	D7–0	**P1IEN[7:0]**	P1[7:0] port input enable	1 Enable 0 Disable	0xff	R/W	
P2 Port Input Data Register (P2_IN)	**0x5220** (8 bits)	D7–0	**P2IN[7:0]**	P2[7:0] port input data	1 1 (H) 0 0 (L)	×	R	P25, P26 are 0, when being read.
P2 Port Output Data Register (P2_OUT)	**0x5221** (8 bits)	D7–0	**P2OUT[7:0]**	P2[7:0] port output data	1 1 (H) 0 0 (L)	0	R/W	
P2 Port Output Enable Register (P2_IO)	**0x5222** (8 bits)	D7–0	**P2OEN[7:0]**	P2[7:0] port output enable	1 Enable 0 Disable	0	R/W	
P2 Port Pull-up Control Register (P2_PU)	**0x5223** (8 bits)	D7–0	**P2PU[7:0]**	P2[7:0] port pull-up enable	1 Enable 0 Disable	1 (0xff)	R/W	
P2 Port Input Enable Register (P2_IEN)	**0x522a** (8 bits)	D7–0	**P2IEN[7:0]**	P2[7:0] port input enable	1 Enable 0 Disable	0xff	R/W	
P3 Port Input Data Register (P3_IN)	**0x5230** (8 bits)	D7–0	**P3IN[7:0]**	P3[7:0] port input data	1 1 (H) 0 0 (L)	×	R	
P3 Port Output Data Register (P3_OUT)	**0x5231** (8 bits)	D7–0	**P3OUT[7:0]**	P3[7:0] port output data	1 1 (H) 0 0 (L)	0	R/W	
P3 Port Output Enable Register (P3_IO)	**0x5232** (8 bits)	D7–0	**P3OEN[7:0]**	P3[7:0] port output enable	1 Enable 0 Disable	0	R/W	
P3 Port Pull-up Control Register (P3_PU)	**0x5233** (8 bits)	D7–0	**P3PU[7:0]**	P3[7:0] port pull-up enable	1 Enable 0 Disable	1 (0xff)	R/W	
P3 Port Input Enable Register (P3_IEN)	**0x523a** (8 bits)	D7–0	**P3IEN[7:0]**	P3[7:0] port input enable	1 Enable 0 Disable	0xff	R/W	

0x5240–0x52a4 P Port & Port MUX

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
P4 Port Input Data Register (P4_IN)	0x5240 (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.
		D3–0	P4IN[7:0]	P4[3:0] port input data	1 1 (H) 0 0 (L)	×	R	
P4 Port Output Data Register (P4_OUT)	0x5241 (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.
		D3–0	P4OUT[3:0]	P4[3:0] port output data	1 1 (H) 0 0 (L)	0	R/W	
P4 Port Output Enable Register (P4_IO)	0x5242 (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.
		D3–0	P4OEN[3:0]	P4[3:0] port output enable	1 Enable 0 Disable	0	R/W	
P4 Port Pull-up Control Register (P4_PU)	0x5243 (8 bits)	D7–4	–	reserved	–	–	–	1 when being read.
		D3–0	P4PU[3:0]	P4[3:0] port pull-up enable	1 Enable 0 Disable	1 (0xff)	R/W	
P4 Port Input Enable Register (P4_IEN)	0x524a (8 bits)	D7–4	–	reserved	–	–	–	1 when being read.
		D3–0	P4IEN[3:0]	P4[3:0] port input enable	1 Enable 0 Disable	0xff	R/W	
P0 Port Function Select Register (P0_PMUX)	0x52a0 (8 bits)	D7	–	reserved	–	–	–	0 when being read.
		D6	P03MUX	P03 port function select	1 #ADTRG 0 P03	0	R/W	
		D5	–	reserved	–	–	–	0 when being read.
		D4	P02MUX	P02 port function select	1 reserved 0 P02/EXCL0	0	R/W	
		D3	–	reserved	–	–	–	0 when being read.
		D2	P01MUX	P01 port function select	1 REMI 0 P01	0	R/W	
		D1	–	reserved	–	–	–	0 when being read.
		D0	P00MUX	P00 port function select	1 REMO 0 P00	0	R/W	
P0 Port Function Select Register (P0_PMUX)	0x52a1 (8 bits)	D7	–	reserved	–	–	–	0 when being read.
		D6	P07MUX	P07 port function select	1 #SPISS 0 P07	0	R/W	
		D5	–	reserved	–	–	–	0 when being read.
		D4	P06MUX	P06 port function select	1 SDI 0 P06	0	R/W	
		D3	–	reserved	–	–	–	0 when being read.
		D2	P05MUX	P05 port function select	1 SDO 0 P05	0	R/W	
		D1	–	reserved	–	–	–	0 when being read.
		D0	P04MUX	P04 port function select	1 SPICLK 0 P04	0	R/W	
P1 Port Function Select Register (P1_PMUX)	0x52a2 (8 bits)	D7–6	P13MUX [1:0]	P13 port function select	P13MUX[1:0] Port 0x3 Reserved 0x2 Reserved 0x1 Reserved 0x0 P13/EXCL1	0	R/W	
		D5	–	reserved	–	–	–	0 when being read.
		D4	P12MUX	P12 port function select	1 SIN 0 P12	0	R/W	
		D3	–	reserved	–	–	–	0 when being read.
		D2	P11MUX	P11 port function select	1 SOUT 0 P11	0	R/W	
		D1	–	reserved	–	–	–	0 when being read.
		D0	P10MUX	P10 port function select	1 SCLK 0 P10	0	R/W	
P1 Port Function Select Register (P1_PMUX)	0x52a3 (8 bits)	D7	–	reserved	–	–	–	0 when being read.
		D6	P17MUX	P17 port function select	1 AIN3 0 P17	0	R/W	
		D5–4	P16MUX [1:0]	P16 port function select	P16MUX[1:0] Port 0x3 Reserved 0x2 Reserved 0x1 SCLK1 0x0 P16/EXCL4	0	R/W	
		D3–2	P15MUX [1:0]	P15 port function select	P15MUX[1:0] Port 0x3 Reserved 0x2 Reserved 0x1 Reserved 0x0 P15/EXCL3	0	R/W	
		D1–0	P14MUX [1:0]	P14 port function select	P14MUX[1:0] Port 0x3 Reserved 0x2 Reserved 0x1 Reserved 0x0 P14/EXCL2	0	R/W	
P2 Port Function Select Register (P2_PMUX)	0x52a4 (8 bits)	D7	–	reserved	–	–	–	0 when being read.
		D6	P23MUX	P23 port function select	1 Reserved 0 P23	0	R/W	
		D5	–	reserved	–	–	–	0 when being read.
		D4	P22MUX	P22 port function select	1 AIN0 0 P22	0	R/W	
		D3	–	reserved	–	–	–	0 when being read.
		D2	P21MUX	P21 port function select	1 AIN1 0 P21	0	R/W	
		D1	–	reserved	–	–	–	0 when being read.
		D0	P20MUX	P20 port function select	1 AIN2 0 P20	0	R/W	

0x52a5–0x52a8 P Port & Port MUX

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
P2 Port Function Select Register (P2_PMUX)	**0x52a5** (8 bits)	D7–6	**P27MUX [1:0]**	P27 port function select	P27MUX[1:0] Port 0x3 Reserved 0x2 Reserved 0x1 SOUT1 0x0 P27	0	R/W	
		D5–1	**–**	reserved	–	–	–	0 when being read.
		D0	**P24MUX**	P24 port function select	1 Reserved 0 P24	0	R/W	
P3 Port Function Select Register (P3_PMUX)	**0x52a6** (8 bits)	D7–6	**P33MUX [1:0]**	P33 port function select	P33MUX[1:0] Port 0x3 Reserved 0x2 SCL0 0x1 SCL1 0x0 P33	0	R/W	
		D5–4	**P32MUX [1:0]**	P32 port function select	P32MUX[1:0] Port 0x3 Reserved 0x2 Reserved 0x1 SDA0 0x0 P32	0	R/W	
		D3–2	**P31MUX [1:0]**	P31 port function select	P31MUX[1:0] Port 0x3 Reserved 0x2 Reserved 0x1 SCL0 0x0 P31	0	R/W	
		D1–0	**P30MUX [1:0]**	P30 port function select	P30MUX[1:0] Port 0x3 Reserved 0x2 Reserved 0x1 SIN1 0x0 P30	0	R/W	
P3 Port Function Select Register (P3_PMUX)	**0x52a7** (8 bits)	D7–6	**P37MUX [1:0]**	P37 port function select	P37MUX[1:0] Port 0x3 TOUT4 0x2 Reserved 0x1 TOUTN3 0x0 P37	0	R/W	
		D5–4	**P36MUX [1:0]**	P36 port function select	P36MUX[1:0] Port 0x3 Reserved 0x2 Reserved 0x1 TOUT3 0x0 P36	0	R/W	
		D3–2	**P35MUX [1:0]**	P35 port function select	P35MUX[1:0] Port 0x3 Reserved 0x2 #BFR 0x1 FOUT1 0x0 P35	0	R/W	
		D1–0	**P34MUX [1:0]**	P34 port function select	P34MUX[1:0] Port 0x3 Reserved 0x2 SDA0 0x1 SDA1 0x0 P34	0	R/W	
P4 Port Function Select Register (P4_PMUX)	**0x52a8** (8 bits)	D7	**–**	reserved	–	–	–	0 when being read.
		D6	**P43MUX**	P43 port function select	1 P43 0 DCLK	0	R/W	
		D5	**–**	reserved	–	–	–	0 when being read.
		D4	**P42MUX**	P42 port function select	1 P42 0 DST2	0	R/W	
		D3	**–**	reserved	–	–	–	0 when being read.
		D2	**P41MUX**	P41 port function select	1 P41 0 DSI0	0	R/W	
		D1	**–**	reserved	–	–	–	0 when being read.
		D0	**P40MUX**	P40 port function select	1 FOUTH 0 P40	0	R/W	

0x5300–0x530c PWM & Capture Timer

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
PWM Timer Compare Data A Register (T16E_CA)	0x5300 (16 bits)	D15–0	T16ECA[15:0]	Compare data A T16ECA15 = MSB T16ECA0 = LSB	0x0 to 0xffff	0x0	R/W	
PWM Timer Compare Data B Register (T16E_CB)	0x5302 (16 bits)	D15–0	T16ECB[15:0]	Compare data B T16ECB15 = MSB T16ECB0 = LSB	0x0 to 0xffff	0x0	R/W	
PWM Timer Counter Data Register (T16E_TC)	0x5304 (16 bits)	D15–0	T16ETC[15:0]	Counter data T16ETC15 = MSB T16ETC0 = LSB	0x0 to 0xffff	0x0	R/W	
PWM Timer Control Register (T16E_CTL)	0x5306 (16 bits)	D15–9	–	reserved	–	–	–	0 when being read.
		D8	INITOL	Internal output level	1 High, 0 Low	0	R/W	
		D7	–	reserved	–	–	–	
		D6	SELFM	Fine mode select	1 Fine mode, 0 Normal mode	0	R/W	
		D5	CBUFEN	Comparison buffer enable	1 Enable, 0 Disable	0	R/W	
		D4	INVOUT	Inverse output	1 Invert, 0 Normal	0	R/W	
		D3	CLKSEL	Input clock select	1 External, 0 Internal	0	R/W	
		D2	OUTEN	Clock output enable	1 Enable, 0 Disable	0	R/W	
		D1	T16ERST	Timer reset	1 Reset, 0 Ignored	0	W	0 when being read.
		D0	T16ERUN	Timer run/stop control	1 Run, 0 Stop	0	R/W	
PWM Timer Input Clock Select Register (T16E_CLK)	0x5308 (16 bits)	D15–4	–	reserved	–	–	–	0 when being read.
		D3–0	T16EDF[3:0]	Timer input clock select (Prescaler output 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	
PWM Timer Interrupt Mask Register (T16E_IMSK)	0x530a (16 bits)	D15–2	–	reserved	–	–	–	0 when being read.
		D1	CBIE	Compare B interrupt enable	1 Enable, 0 Disable	0	R/W	
		D0	CAIE	Compare A interrupt enable	1 Enable, 0 Disable	0	R/W	
PWM Timer Interrupt Flag Register (T16E_IFLG)	0x530c (16 bits)	D15–2	–	reserved	–	–	–	0 when being read.
		D1	CBIF	Compare B interrupt flag	1 Cause of interrupt occurred, 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D0	CAIF	Compare A interrupt flag		0	R/W	

0x5320–0x532c MISC Registers

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
ROM Control Register (MISC_FL)	**0x5320** (16 bits)	D15–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**FLCYC[2:0]**	FLASHC read access cycle	FLCYC[2:0] / Read cycle 0x7–0x5 / reserved 0x4 / 1 cycle 0x3 / 5 cycles 0x2 / 4 cycles 0x1 / 3 cycles 0x0 / 2 cycles	0x3	R/W	
OSC1 Peripheral Control Register (MISC_OSC1)	**0x5322** (16 bits)	D15–1	**–**	reserved	–	–	–	0 when being read.
		D0	**O1DBG**	OSC1 peripheral control in debug mode	1 Run / 0 Stop	0	R/W	
MISC Protect Register (MISC_PROT)	**0x5324** (16 bits)	D15–0	**PROT[15:0]**	MISC register write protect	Writing 0x96 removes the write protection of the MISC registers (0x5326–0x532a). Writing another value set the write protection.	0x0	R/W	
IRAM Size Select Register (MISC_IRAMSZ)	**0x5326** (16 bits)	D15–9	**–**	reserved	–	–	–	0 when being read.
		D8	**DBADR**	Debug base address select	1 0x0 / 0 0xfffc00	0	R/W	
		D7–2	**–**	reserved	–	–	–	0 when being read.
		D1–0	**IRAMSZ[1:0]**	IRAM size select	IRAMSZ[1:0] / Read cycle 0x3 / reserved 0x2 / reserved 0x1 / reserved 0x0 / reserved	0x2	R/W	
Vector Table Address Low Register (MISC_TTBRL)	**0x5328** (16 bits)	D15–8	**TTBR[15:8]**	Vector table base address A[15:8]	0x0–0xff	0x80	R/W	
		D7–0	**TTBR[7:0]**	Vector table base address A[7:0] (fixed at 0)	0x0	0x0	R	
Vector Table Address High Register (MISC_TTBRH)	**0x532a** (16 bits)	D15–8	**–**	reserved	–	–	–	0 when being read.
		D7–0	**TTBR[23:16]**	Vector table base address A[23:16]	0x0–0xff	0x0	R/W	
PSR Register (MISC_PSR)	**0x532c** (16 bits)	D15–8	**–**	reserved	–	–	–	0 when being read.
		D7–5	**PSRIL[2:0]**	PSR interrupt level (IL) bits	0x0 to 0x7	0x0	R	
		D4	**PSRIE**	PSR interrupt enable (IE) bit	1 1 (enable) / 0 0 (disable)	0	R	
		D3	**PSRC**	PSR carry (C) flag	1 1 (set) / 0 0 (cleared)	0	R	
		D2	**PSRV**	PSR overflow (V) flag	1 1 (set) / 0 0 (cleared)	0	R	
		D1	**PSRZ**	PSR zero (Z) flag	1 1 (set) / 0 0 (cleared)	0	R	
		D0	**PSRN**	PSR negative (N) flag	1 1 (set) / 0 0 (cleared)	0	R	

0x5340–0x5346 Remote Controller

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
REMC Configuration Register (REMC_CFG)	**0x5340** (16 bits)	D15–12	**CGCLK[3:0]**	Carrier generator clock select (Prescaler output clock)	CGCLK[3:0] / LCCLK[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	
		D11–8	**LCCLK[3:0]**	Length counter clock select (Prescaler output clock)		0x0	R/W	
		D7–2	**–**	reserved	–	–	–	0 when being read.
		D1	**REMMD**	REMC mode select	1 Receive 0 Transmit	0	R/W	
		D0	**REMEN**	REMC enable	1 Enable 0 Disable	0	R/W	
REMC Carrier Length Setup Register (REMC_CAR)	**0x5342** (16 bits)	D15–14	**–**	reserved	–	–	–	0 when being read.
		D13–8	**REMCL[5:0]**	L carrier length setup	0x0 to 0x3f	0x0	R/W	
		D7–6	**–**	reserved	–	–	–	0 when being read.
		D5–0	**REMCH[5:0]**	H carrier length setup	0x0 to 0x3f	0x0	R/W	
REMC Length Counter Register (REMC_LCNT)	**0x5344** (16 bits)	D15–8	**REMLEN[7:0]**	Transmit/receive data length count (down counter)	0x0 to 0xff	0x0	R/W	
		D7–1	**–**	reserved	–	–	–	0 when being read.
		D0	**REMDT**	Transmit/receive data	1 1 (H) 0 0 (L)	0	R/W	
REMC Interrupt Control Register (REMC_INT)	**0x5346** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10	**REMFIF**	Falling edge interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D9	**REMRIF**	Rising edge interrupt flag		0	R/W	
		D8	**REMUIF**	Underflow interrupt flag		0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2	**REMFIE**	Falling edge interrupt enable	1 Enable 0 Disable	0	R/W	
		D1	**REMRIE**	Rising edge interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	**REMUIE**	Underflow interrupt enable	1 Enable 0 Disable	0	R/W	

0x5380–0x5386 ADC10SA

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
A/D Conversion Result Register (ADC10_ADD)	0x5380 (16 bits)	D15–0	ADD[15:0]	A/D converted data @STMD=0 ADD[15:10]=6'b0,ADD9=MSB, ADD0=LSB @STMD=1 ADD15=MSB,ADD6=LSB, ADD[5:0]=6'b0	0-1023	0	R	
A/D Trigger/ Channel Select (ADC10_TRG)	0x5382 (16 bits)	D15–14	–	reserved	–	–	–	0 when being read.
		D13–11	ADCE[2:0]	End channel selection	0x0-0x7	0	R/W	
		D10–8	ADCS[2:0]	Start channel selection	0x0-0x7	0	R/W	
		D7	STMD	Converted data store mode	1 {AD[9:0], 6'b0} 0 { 6'b0, AD[9:0]}	0	R/W	
		D6	ADMS	Conversion mode selection	1 continuous 0 Single	0	R/W	
		D5–4	ADTS	Conversion trigger selection	ADTS[2:0] trigger 0x3 #ADTRG pin 0x2 reserved 0x1 16bit timer 0x0 software	0	R/W	
		D3	–	reserved	–	–	–	0 when being read.
		D2–0	ADST[2:0]	Sampling clock count	ADST[2:0] count clock 0x7 9clocks 0x6 8clocks 0x5 7clocks 0x4 6clocks 0x3 5clocks 0x2 4clocks 0x1 3clocks 0x0 2clocks	0x7	R/W	Must set 0x7
A/D Control/ Status Register (ADC10_CTL)	0x5384 (16 bits)	D15	–	reserved	–	–	–	0 when being read.
		D14–12	ADICH	Internal conversion channel status	0x0-0x7	0	R	
		D11	–	reserved	–	–	–	0 when being read.
		D10	ADIBS	Internal busy status	1 busy 0 idle	0	R	
		D9	ADOWE	Overwrite error flag	1 Error 0 Normal	0	R/W	Reset by writing 1
		D8	ADCF	Conversion-complete flag	1 Completed 0 Not completed	0	R	Reset when ADC10_ADD is read.
		D7–6	–	reserved	–	–	–	0 when being read.
		D5	ADOIE	Overwrite interrupt enable	1 Enable 0 Disable	0	R/W	
		D4	ADCIE	Conversion-complete interrupt enable	1 Enable 0 Disable	0	R/W	
		D3–2	–	reserved	–	–	–	0 when being read.
		D1	ADCTL	conversion control	1 Start/Run 0 Stop	0	R/W	Stop by writing 0
		D0	ADEN	A/D enable	1 Enable 0 Disable	0	R/W	
A/D divided frequency Register (ADC_DIV)	0x5386 (16 bits)	D15–4	–	reserved	–	–	–	0 when being read.
		D3–0	ADDF[3:0]	A/D converter clock divided frequency select	ADDF[3:0] clock 0xf Reserved 0xe PCLK・1/32768 0xd PCLK・1/16384 0xc PCLK・1/8192 0xb PCLK・1/4096 0xa PCLK・1/2048 0x9 PCLK・1/1024 0x8 PCLK・1/512 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	0	R/W	

0xffff84–0xffffd0 S1C17 Core I/O

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Processor ID Register (IDIR)	**0xffff84** (8 bits)	D7–0	**IDIR[7:0]**	Processor ID 0x10: S1C17 Core	0x10	0x10	R	
Debug RAM Base Register (DBRAM)	**0xffff90** (32 bits)	D31–24	**–**	Unused (fixed at 0)	0x0	0x0	R	
		D23–0	**DBRAM[23:0]**	Debug RAM base address	0x0fc0	0x0fc0	R	
Debug Control Register (DCR)	**0xffffa0** (8 bits)	D7	**IBE4**	Instruction break #4 enable	1 Enable 0 Disable	0	R/W	
		D6	**IBE3**	Instruction break #3 enable	1 Enable 0 Disable	0	R/W	
		D5	**IBE2**	Instruction break #2 enable	1 Enable 0 Disable	0	R/W	
		D4	**DR**	Debug request flag	1 Occurred 0 Not occurred	0	R/W	Reset by writing 1.
		D3	**IBE1**	Instruction break #1 enable	1 Enable 0 Disable	0	R/W	
		D2	**IBE0**	Instruction break #0 enable	1 Enable 0 Disable	0	R/W	
		D1	**SE**	Single step enable	1 Enable 0 Disable	0	R/W	
		D0	**DM**	Debug mode	1 Debug mode 0 User mode	0	R	
Instruction Break Address Register 2 (IBAR2)	**0xffffb8** (32 bits)	D31–24	**–**	reserved	–	–	–	0 when being read.
		D23–0	**IBAR2[23:0]**	Instruction break address #2 IBAR223 = MSB IBAR20 = LSB	0x0 to 0xffffff	0x0	R/W	
Instruction Break Address Register 3 (IBAR3)	**0xffffbc** (32 bits)	D31–24	**–**	reserved	–	–	–	0 when being read.
		D23–0	**IBAR3[23:0]**	Instruction break address #3 IBAR323 = MSB IBAR30 = LSB	0x0 to 0xffffff	0x0	R/W	
Instruction Break Address Register 4 (IBAR4)	**0xffffd0** (32 bits)	D31–24	**–**	reserved	–	–	–	0 when being read.
		D23–0	**IBAR4[23:0]**	Instruction break address #4 IBAR423 = MSB IBAR40 = LSB	0x0 to 0xffffff	0x0	R/W	

Appendix B: Power Saving

Current consumption will vary dramatically, depending on CPU operating mode, operation clock frequency, and the peripheral circuits being operated. Listed below are the control methods for saving power.

B.1 Clock Control Power Saving

Figure B.1.1 illustrates the S1C17003 clock system.

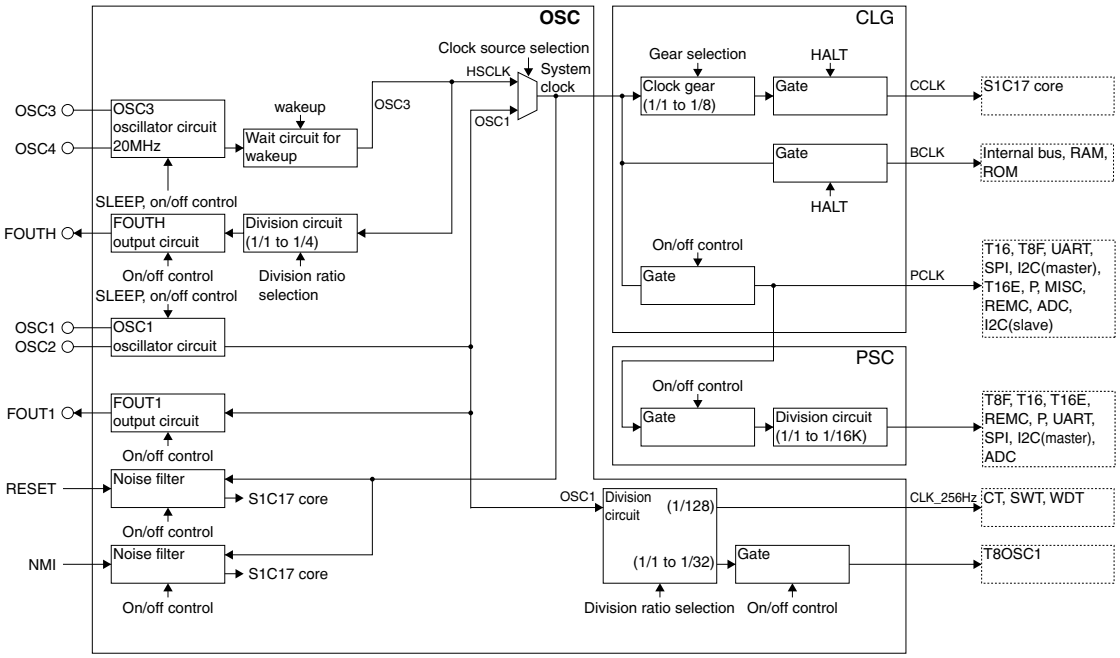


Figure B.1.1 Clock system

This section describes clock systems that can be controlled via software and power-saving control details. For more information on control registers and control methods, refer to the respective module sections.

System SLEEP (All clocks stopped)

- Execute `slp` instruction
  Execute the `slp` instruction when the entire system can be stopped. The CPU switches to SLEEP mode and the system clocks stop. This also stops all peripheral circuits using clocks. Starting up the CPU from SLEEP mode is therefore limited to startup using ports (described later).

System clocks

- Clock source selection (OSC module)
  Select between OSC3 and OSC1 for the system clock source. Reduce current consumption by selecting the OSC1 clock when low-speed processing is possible.
- OSC3 oscillation circuit stop (OSC module)
  You can reduce current consumption by using OSC1 as the system clock and stopping the OSC3 oscillation circuit.

CPU clock (CCLK)

- Execute the `halt` instruction
  Execute the `halt` instruction when program execution by the CPU is not required—for example, when only the display is required or for interrupt standby. The CPU switches to HALT mode and suspends operations, but the peripheral circuits maintain the status in place at the time of the `halt` instruction, enabling use of peripheral circuits for timers and interrupts. You can reduce power consumption even further by suspending unnecessary peripheral circuits before executing the halt instruction. The CPU is started from HALT mode using the port or interrupts from the peripheral circuit operating in HALT mode.
- Low-speed clock gear selection (CLG module)
  The CLG module can reduce CPU clock speeds to between 1/1 and 1/8 of the system clock via the clock gear settings. Reduce current consumption by operating the CPU at the minimum speed required for applications.

Peripheral clock (PCLK)

- PCLK stop (CLG module)
  Stop the PCLK clock feed from the CLG to peripheral circuits if none of the following peripheral circuits is required.

  Peripheral circuits operating with PCLK
  - Prescaler (PWM & capture timer, remote controller, P port)
  - UART
  - 8-bit timer
  - 16-bit timer Ch.0 to Ch.2
  - SPI
  - I^2C (master/slave)
  - P port and port MUX (control register, chattering filter)
  - PWM & capture timer
  - MISC register
  - Remote controller
  - A/D converter

The peripheral modules listed below are operated by clocks other than PCLK, except for control register access.
This means PCLK is not required after the control register has been set and operation started.

- Clock timer
- Stopwatch timer
- Watchdog timer
- 8-bit OSC1 timer

Table B.1.1 shows a list of methods for clock control and starting/stopping the CPU.

Table B.1.1: Clock control list

Current consumption	OSC1	OSC3	CPU (CCLK)	PCLK peripheral	OSC1 peripheral	CPU stop method	CPU startup method
↑ Low	Stop	Stop	Stop	Stop	Stop	Execute **slp** instruction	1
	Oscillation (system CLK)	Stop	Stop	Stop	Operation	Execute **halt** instruction	1, 2
	Oscillation (system CLK)	Stop	Stop	Operation	Operation	Execute **halt** instruction	1, 2, 3
	Oscillation (system CLK)	Stop	Operation(1/1)	Operation	Operation		
	Oscillation	Oscillation (system CLK)	Stop	Operation	Operation	Execute **halt** instruction	1, 2, 3
	Oscillation	Oscillation (system CLK)	Operation (Low gear)	Operation	Operation		
High ↓	Oscillation	Oscillation (system CLK)	Operation(1/1)	Operation	Operation		

HALT and SLEEP mode cancellation methods (CPU startup method)

1. Startup by port
   Started up by input/output port interrupt and debug interrupt (ICD forced break).
2. Startup by OSC1 peripheral circuit
   Started up by clock timer, stopwatch timer, watchdog timer, or 8-bit OSC1 timer interrupts.
3. Startup by PCLK peripheral circuit
   Started up by PCLK peripheral circuit interrupt.

Appendix C: Mounting Precautions

This section describes various precautions for circuit board design and IC mounting.

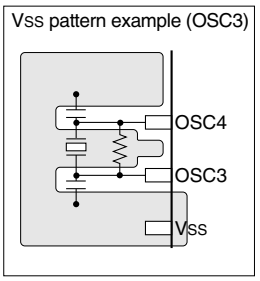
Oscillator circuit

- Oscillation characteristics depend on factors such as components used (oscillator, Rf, CG, CD) and circuit board patterns. In particular, with ceramic or crystal oscillators, select the appropriate external resistors (Rf,) and capacitors (CG, CD) only after fully evaluating components actually mounted on the circuit board.
- Oscillator clock disturbances caused by noise may cause malfunctions. To prevent such disturbances, consider the following points. The latest devices, in particular, are manufactured by microscopic processes, making them especially susceptible to noise.
  Areas in which noise countermeasures are especially important include the OSC2 pin and related circuit components and wiring. OSC1 pin handling is equally important. The noise precautions required for the OSC1 and OSC2 pins are described below.
  We also recommend applying similar noise countermeasures to high-speed oscillator circuits, such as the OSC3 and OSC4 pins and wiring.

(1) Components such as oscillators, resistors, and capacitors connected to the OSC1 (OSC3) and OSC2 (OSC4) pins should have the shortest connections possible.

(2) Wherever possible, avoid locating digital signal lines within 3 mm of the OSC1 (OSC3) and OSC2 (OSC4) pins or related circuit components and wiring. Rapidly-switching signals, in particular, should be kept at a distance from these components. Since the spacing between layers of multi-layer printed circuit boards is a mere 0.1 mm to 0.2 mm, the above precautions also apply when positioning digital signal lines on other layers. Never place digital signal lines alongside such components or wiring, even if more than 3 mm distance or located on other layers. Avoid crossing wires.

(3) Use VSS to shield OSC1 (OSC3) and OSC2 (OSC4) pins and related wiring (including wiring for adjacent circuit board layers).
Layers wired should be adequately shielded as shown to the right.
Fully ground adjacent layers, where possible. At minimum, shield the area at least 5 mm around the above pins and wiring.
Even after implementing these precautions, avoid configuring digital signal lines in parallel, as described in (2) above. Avoid crossing even on discrete layers, except for lines carrying signals with low switching frequencies.



(4) After implementing these precautions, check the output clock waveform by running the actual application program within the product.
Use an oscilloscope to check outputs from the FOUT1 and FOUTH pins.
You can check the quality of the OSC3 output waveform via the FOUTH output. Confirm that the frequency is as designed, is free of noise, and has minimal jitter.
You can check the quality of the OSC1 waveform via the FOUT1 output. In particular, enlarge the areas before and after the clock rising and falling edges and take special care to confirm that the regions approximately 100 ns to either side are free of clock or spiking noise.

Failure to observe precautions (1) to (3) adequately may lead to jitter in the OSC3 output and noise in the OSC1 output. Jitter in the OSC3 output will reduce operating frequencies, while noise in the OSC1 output will destabilize timers operated by the OSC1 clock as well as CPU core operations when the system clock switches to OSC1.

Reset circuit

- The reset signal input to the #RESET pin when power is turned on will vary, depending on various factors, such as power supply start-up time, components used, and circuit board patterns. Constants such as capacitance and resistance should be determined through thorough testing with real-world products. Account for resistance fluctuations when setting the #RESET pin pull-up resistance for constants settings.
- Components such as capacitors and resistors connected to the #RESET pin should have the shortest connections possible to prevent noise-induced resets.

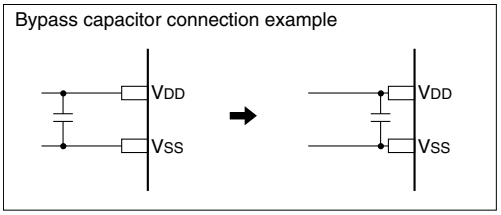
Power supply circuit

Sudden power supply fluctuations due to noise will cause malfunctions. Consider the following issues.

(1) Connections from the power supply to the VDD and VSS pins should be implemented via the shortest, thickest patterns possible.

(2) If a bypass capacitor is connected between VDD and VSS, connections between the VDD and VSS pins should be as short as possible.

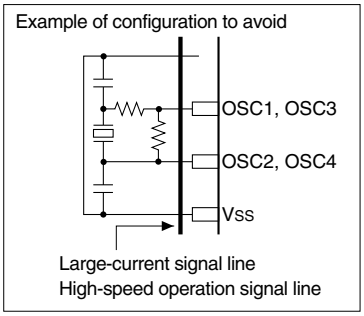
Bypass capacitor connection example



Signal line location

- To prevent electromagnetically-induced noise arising from mutual induction large-current signal lines should not be positioned close to circuits susceptible to noise, such as oscillators.
- Locating signal lines in parallel over significant distances or crossing signal lines operating at high speed will cause malfunctions due to noise generated by mutual interference.
  Specifically, avoid positioning crossing signal lines operating at high speed close to circuits susceptible to noise, such as oscillators.

Example of configuration to avoid



Noise-induced malfunctions

Check the following three points if you suspect the presence of noise-induced IC malfunctions.

(1) DSIO pin
Low-level noise to this pin will cause a switch to Debug mode. The switch to Debug mode can be confirmed by the clock output from DCLK and a High signal from the DST2 pin.
For the product version, we recommend connecting the DSIO pin directly to HV_{DD} or pulling up the DISO pin using a resistor not exceeding 10 kΩ.
The IC includes an internal pull-up resistor. The resistor has a relatively high impedance of 50 kΩ to 100 kΩ and is not noise-resistant.

(2) #RESET pin
Low-level noise to this pin will reset the IC. Depending on the input waveform, the reset may not proceed correctly.
This is more likely to occur if, due to circuit design choices, the impedance is high when the reset input is High.

(3) V_{DD} and V_{SS} power supply
The IC will malfunction the instant noise falling below the rated voltage is input.
Incorporate countermeasures on the circuit board, including close patterns for circuit board power supply circuits, noise-filtering decoupling capacitors, and surge/noise prevention components on the power supply line.

Perform the inspections described above using an oscilloscope capable of observing waveforms of at least 200 MHz. It may not be possible to observe high-speed noise events with a low-speed oscilloscope.

If you detect potential noise-induced malfunctions while observing the waveform with an oscilloscope, recheck with a low-impedance (less than 1 kΩ) resistor connecting the relevant pin to GND or to the power supply. Malfunctions at that pin are likely if changes are visible, such as the malfunction disappearing, becoming less frequent, or the phenomena changing.

The DSIO and #RESET input circuits described above detect input signal edges and are susceptible to malfunctions induced by spike noise. This makes these digital signal pins the most susceptible to noise.
To reduce potential noise, keep the following two points in mind when designing circuit boards:

(A) It is important to use low impedance resistors when driving the signals, as described above. Avoid connecting impedance exceeding 1 kΩ (ideally, 0 Ω) to the power supply or GND. The signal lines connected should be no longer than approximately 5 cm.

(B) Signals switching from 1 to 0 or 0 to 1 may generate noise if signal lines run parallel to other digital lines on the circuit board.
The highest risk of noise occurs in configurations in which a line is sandwiched between multiple signal lines that vary in synchrony. You can minimize noise effects by reducing the length of parallel sections (limit to a few cm) or by increasing the separation (to at least 2 mm).

Handling of light (for bare chip mounting)

The characteristics of semiconductor components can vary when exposed to light. ICs may malfunction or nonvolatile memory data may be corrupted if ICs are exposed to light.
Consider the following precautions for circuit boards and products in which this IC is mounted to prevent IC malfunctions attributable to light exposure.

(1) Design and mount the product so that the IC is shielded from light during use.

(2) Shield the IC from light during inspection processes.

(3) Shield the IC on the upper, underside, and side faces of the IC chip.

(4) Mount the IC chip within one week of opening the package. If the IC chip must be stored before mounting, take measures to ensure light shielding.

(5) Adequate evaluations are required to assess nonvolatile memory data retention characteristics before product delivery if the product is subjected to heat stress exceeding regular reflow conditions during mounting processes.

Miscellaneous

This product series is manufactured using 0.18 μm microscopic processes.
Although it is designed to ensure basic IC reliability meeting EIAJ and MIL standards, consider the following points when mounting the product.
All oscillator circuit input/output pins use direct connections to internal 0.18 μm transistors. In addition to physical damage during mounting, minor variations over time may result in electrical damage arising from disturbances in the form of voltages exceeding the absolute maximum rating (2.5 V). The following factors can give rise to these variations:

(1) electromagnetically-induced noise from industrial power supplies used in mounting reflow, reworking after mounting, and individual characteristic evaluation (testing) processes;

(2) electromagnetically-induced noise from a solder iron when soldering.

In particular, during soldering, take care to ensure that the soldering iron GND (tip potential) has the same potential as the IC GND.

Appendix D: Developing S1C17003 Mask ROM Code

(1) Use the S1C17602 Flash microcomputer to develop mask ROM code for the S1C17003.

(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 S1C17003 when moto2ff is executed.
- Data start address = 0x8000
- Data block size = 512 × 16 bits

(4) Take the differences listed in the table below into consideration and perform operation check using the S1C17602.

Circuit/function		S1C17602	S1C17003
Memory	Boot address	0x8000	←
	Flash	64KB	—
	Mask ROM	—	64KB
	IRAM	4KB	←
	Display RAM	40B	—
Clock	Maximum operating frequency	8.2MHz	20MHz
	IOSC oscillator circuit	Internal oscillator	—
	OSC3 oscillator circuit	Crystal/Ceramic/External input	←
	OSC1 oscillator circuit	Crystal/External input	←
Co-processor	Product-sum operation	16bit×16bit+32bit(1cycle)	←
	Divider	16bit÷16bit(20cycle)	←
Peripheral circuit	I/O port	36 (With Hi-z mode and input Schmitt switching)	30 (Without Hi-z mode, and with fixed Input Schmitt)
	Input port	—	4 (Interface level AV_{DD})
	SPI(master / slave)	1ch	←
	I^2C(master / slave)	1ch	←
	UART(with IrDA1.0)	2ch	←
	Infrared rays remote controller (REMC)	1ch	←
	8-bit timer with the fine mode (T8F)	2ch	←
	16-bit timer (T16)	3ch	←
	PWM timer & Capture timer (T16E)	1ch	←
	Clock timer(CT)	1ch	←
	Stopwatch timer (SWT)	1ch	←
	8-bit OSC1 timer (T8OSC1)	1ch	←
	Watchdog timer (WDT)	1ch	←
	LCD driver	36×8 / 40×4	—
	A/D converter	10bit×8ch (±1.5LSB)	10bit×4ch (±3LSB)
	R/F converter	2ch	—
	Power supply voltage detection (SVD) circuit	1.8 to 3.2V detect	—
Power supply	Power supply voltage	V_{DD}=1.8V to 3.6V (in normal operation) V_{DD}=2.7V to 3.6V (while writing to flash ROM) (Regulator is built-in)	HV_{DD}(I/O)=1.65V to 3.6V LV_{DD}(core)=1.65V to 1.95V AV_{DD}(Analog)=2.7V to 3.6V
Temperature	Operating temperatures	-20 to 70°C	-40 to 85°C
Configuration	Package	TQFP14-100 VFBGA7H-144 Bare chip 100μm pitch	TQFP12-64 WCSP 48pin Bare chip 100μm pitch

Revision History

Code No.	Page	Contents
411635101	All	New anactment
411635102	1-3	Figure 1.2.1 modified.
	1-6	Opening of Pad and Chip thickness added
	2-5	Description modified. Branch ipa.d→jpa.d
	3-4	0x5326 Register table and Table 3.3.1.1 modified. Descriptions added. • Please do not change the setting of IRAMSZ[2:0]/MISC_IRAMSZ Register from a default value.
	6-15	Descriptions modified. Set the UART Ch.1 or I2C (slave) interrupt level (0 to 7). (Default: 0)
	7-1, 8-1, AP-31	Delete ITC from the block that connect to PCLK
	7-2	Descriptions added. For information about input clock waveforms, refer to “26 Electrical Characteristics.”
	7-6, 7-7	Figure 7.7.2 and Figure 7.7.3 modified.
	8-3, 8-5, 8-7, AP-32	Description deleted. • Interrupt controller
	18-6	Descriptions modified. (2)RDRY = 1, RD2B = 0...of the receive data buffer must be read out before an overrun error occurs.
	18-7	Descriptions modified. After a data transfer is completed (both transmission and reception), data transfers are blocked by writing 0 to the RXEN bit.
	18-8	Descriptions modified. However, if the receive data buffer is not emptied...by the time this data has been received, the third data received in the shift register will not be sent to the buffer and generate an overrun error.
	18-14	Descriptions modified. FER is reset by writing 1. PER is reset by writing 1. OER is reset by writing 1.
	18-19	Descriptions modified. Preventing transfers by writing 0 to RXEN also clears transmit data buffer.
	18-21	Descriptions modified. RBFI bit in the UART_CTLx register Preventing transfer by setting RXEN to 0 clears (initializes) transfer data buffers. Before writing 0 to RXEN, confirm the absence of data in the buffers awaiting transmission.
	19-3	Descriptions modified. The Master mode SPI uses the internal clock output by the 16-bit timer Ch.1 as the SPI clock. Figure 19.3.1 modified. Figure 19.3.2 deleted. Description deleted. Since the internal circuit operates in sync with the PCLK clock, the input clock is used to synchronize the differentiated PCLK clock. Descriptions modified. Note: The duty ratio of the clock input via the SPICLK pin must be 50%.
	19-4, 19-5	Descriptions added. Note: When the SPI module is used in master...second and following bytes during continuous transfer. Figure 19.4.2 added.
	19-6, 19-7	Figure 19.5.1 and Figure 19.5.2 deleted. Figure 19.5.1 added.
	19-7	Descriptions modified. After a data transfer is completed ...guaranteed if SPEN is set to 0 while data is being sent or received.
	19-8	Descriptions modified. If SPTBE is 0,→If SPTBE is 1,
	19-11	Descriptions added. Note: Make sure that SPEN is set to 1 before...SPI_TXD register to start data transmission/reception.
	20-3	Descriptions added. If the I^2C master module communicates with a slave device which has clock stretching, Transfer rates are limited up to 50 kbits/s in the Standard-mode, up to 200 kbits in the Fast-mode.
	20-6, 20-7, 20-15	Descriptions modified. NACK→NAK
	20-6	Figure 20.5.2 modified.
	20-7	Descriptions modified. The data is shifted into the shift register with the clock pulses, ...RXE is reset to 0 when D7 is loaded.

	20-8	Descriptions modified. When transmission or reception ends, TBUSY or RBUSY is cleared. Then, after a ....clock cycle set. Descriptions modified. Disabling data transfer ...the I2C bus is in busy status, the SCL0 and SDA0 output levels and transfer data at that point cannot be guaranteed.
	20-9	Figure 20.5.6, Figure 20.5.7 and Figure 20.5.8 modified.
	20-10	Figure 20.5.9 modified. Descriptions modified. Transmit buffer empty interrupt occurs when the data was only sent....NOTE: When I2CM interrupt occurs, decide the transmit buffer empty interrupt or the receive buffer full interrupt by the program sequence of the I2C master. There're not registers to decide which interrupt occurred.
	20-13	Descriptions added. Setting the STP bit 1 makes the I²C master module...the 1/4 cycle of I2C clock, STP can set to 1.
	21-1	Figure 21.1.1 modified.
	21-2	Descriptions modified. I²C slave clock input/output pin...SCL line status and outputs low level to the I²C bus when clock stretch.
	21-4	Descriptions modified. (one system clock (PCLK) cycle is required. Two PCLK cycles or more pulse width is recommended) Descriptions added. Note: When I²C slave module is slave transceiver mode,...depends on the PCLK frequency.
	21-6	Descriptions modified. STOP condition detection clears BUSY. STOP or Repeated START condition detection clears SELECTED.
	21-7	Descriptions added. When the asynchronous address detection function is used, the data written before ASDET_EN is reset in 0 becomes invalid. Therefore, the transmission data must be written, after TXEMP has been set to 1.
	21-8	Descriptions added. Note: If the I2CS module has sent back a NAK as the response ...The I2CS module is placed into transfer standby state and OSC1 is used as the operating clock (PCLK).
	21-10, 21-11	Figure 21.5.5, Figure 21.5.6, Figure 21.5.7 and Figure 21.5.8 modified.
	21-12	Descriptions modified. 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
	21-22	Descriptions modified. Indicates that a stop condition or a repeated start condition is detected...At the same time, it initializes the I2C communication process.
	21-23	Descriptions modified. After SELECTED is set to 1, it is reset to 0 when a STOP condition or a Repeated START condition is detected.
	23-5	Descriptions modified. Low level→negative edge
	23-16	Descriptions modified. When ADEN is 0, a trigger is not accepted.
	23-18	Description deleted. Moreover, set back the input to #ADTRG terminal to High with less than 20 cycles of set input clock so that it will not be detected as next A/D conversion trigger.
	26-6	26.9 External Clock Input Characteristics added.

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.
Unit 715-723, 7/F Trade Square, 681 Cheung Sha Wan Road,
Kowloon, Hong Kong.
Phone: +852-2585-4600 Fax: +852-2827-4346

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: 411635102
First Issue January 2009
Revised April 2011Ⓗ