EPSON
EXCEED YOUR VISION

CMOS 16-BIT SINGLE CHIP MICROCONTROLLER

S1C17001
Technical Manual

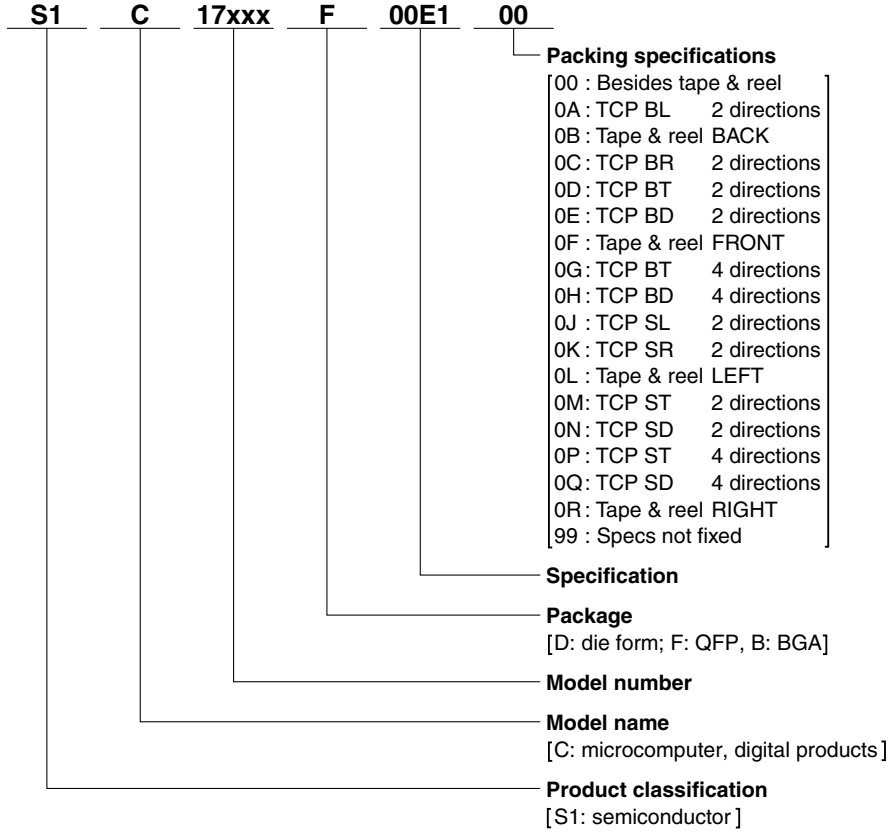
SEIKO EPSON CORPORATION

Rev. 1.3

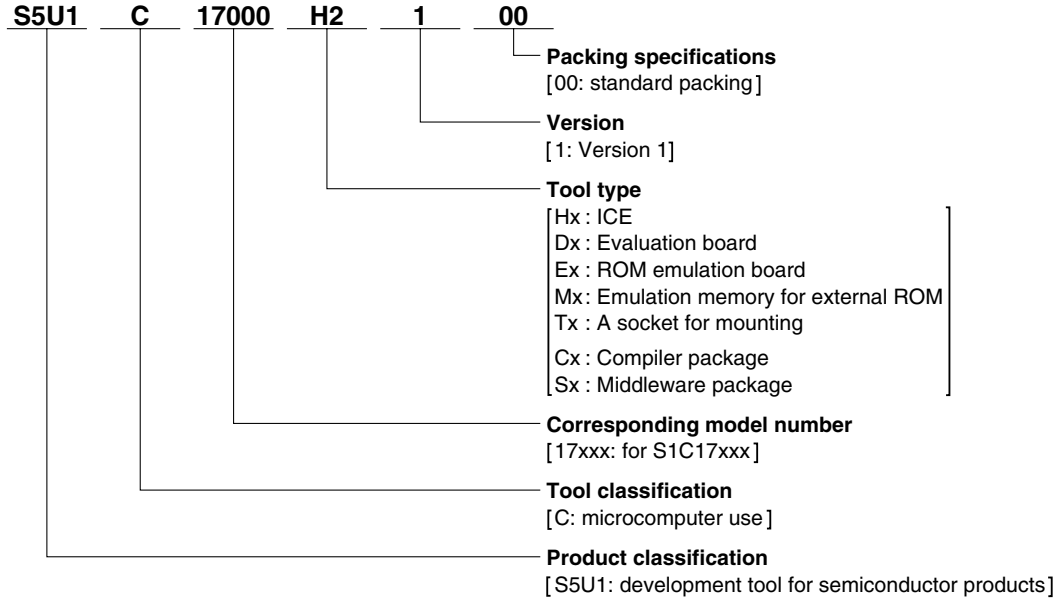
NOTICE

Configuration of product number

Devices



Development tools



– Contents –

1 Overview

The S1C7001 is a 16-bit MCU featuring high-speed low-power operations, compact dimensions, wide address space, and on-chip ICE. In addition to the S1C17 CPU core, it incorporates 32 Kbytes of ROM, 2 Kbytes of RAM, a serial interface supporting various sensors such as UART, SPI, and I²C high-bit-rate and IrDA1.0 compatibility, 8-bit timer, 16-bit timer, PWM & capture timer, clock timer, stopwatch timer, watchdog timer, and 28 general purpose input/output ports.
It allows 8.2 MHz high-speed operation at an operating voltage of just 1.8 V, and executes single commands using a single clock with 16-bit RISC processing.

1.1 Features

The main features of the S1C17001 are listed below.

●CPU	• Epson original 16-bit RISC CPU core S1C17
●Main (OSC3) oscillator circuit	• Crystal oscillator circuit, ceramic oscillator circuit, or external clock input 8.2 MHz (max)
●Sub (OSC1) oscillator circuit	• Crystal oscillator circuit or external clock input 32.786 kHz (typ)
●Internal ROM	• 32 Kbytes
●Internal RAM	• 2 Kbytes
●Input/output port	• Max. 28-bit general purpose input/output (shared with peripheral circuit input/output pins)
●Serial interface	• SPI (master/slave) 1ch. • UART (IrDA1.0 compatible) 1ch. • I²C (master) 1ch. • Remote controller (REMC) 1ch.
●Timer	• 8-bit timer (T8F) 1ch. • 16-bit timer (T16) 3ch. • PWM& capture timer (T16E) 1ch. • Clock timer (CT) 1ch. • Stopwatch timer (SWT) 1ch. • Watchdog timer (WDT) 1ch. • 8-bit OSC1 timer (T8OSC1) 1ch.
●Interrupt	• Reset • NMI • Hardware interrupt x14 (8 levels)
●Power supply voltage	• Core voltage (LV_{DD}) 1.65 V to 2.7 V • I/O voltage (HV_{DD}) 1.65 V to 3.6 V
●Operating temperature	• -40°C to 85°C
●Current consumption (typ.)	• 0.5 µA in SLEEP mode • 2.5 µA in HALT mode (32 kHz) • 10 µA when operating (32 kHz) 1800 µA when operating (8 MHz)
●Configuration as shipped	• WCSP-48pin package
●Mask ROM code development Flash memory	• S1C17704 (refer to Appendix E for details)

1-2 Block Diagram

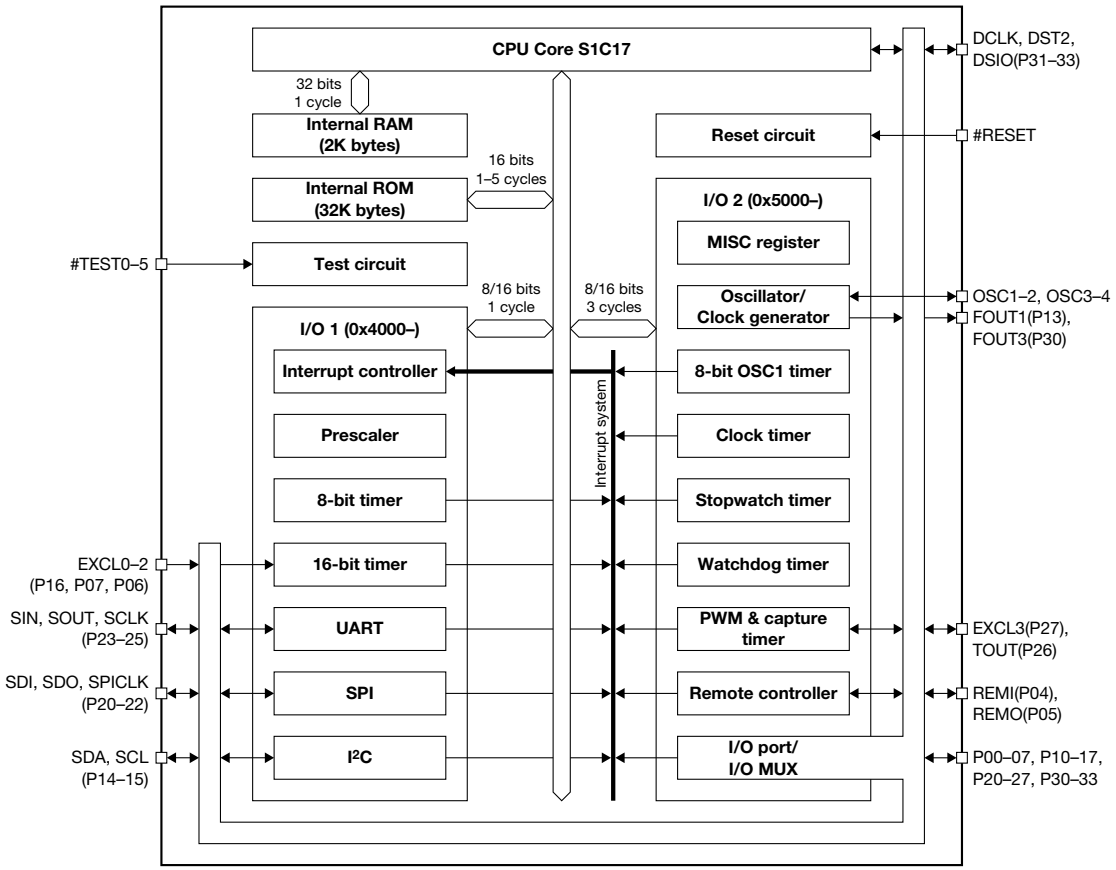
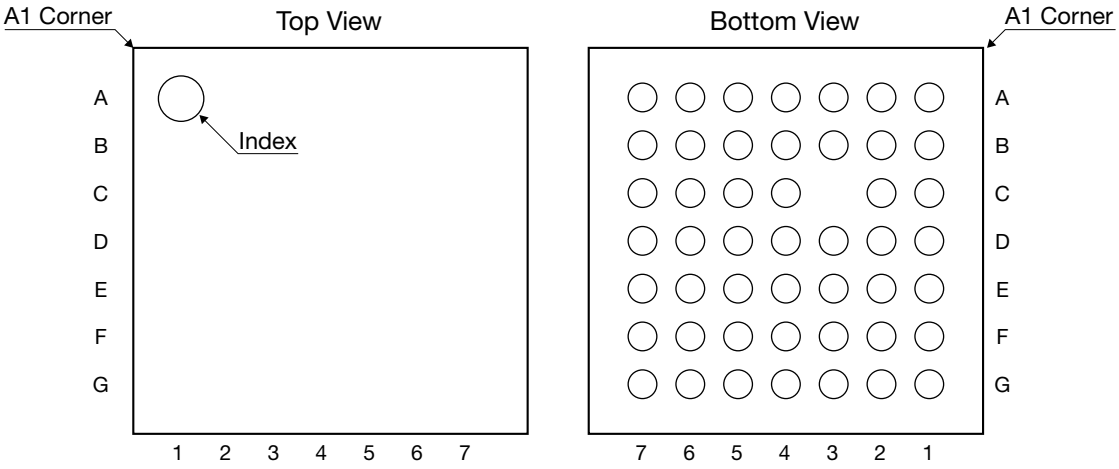


Figure 1.2.1: Block diagram

1.3 Pins

1.3.1 Pinout Diagram



Top View

	1	2	3	4	5	6	7
A	**#TEST2**	**P05** REMO	**VSS**	**P02**	**HVDD**	**P11**	**#TEST3**
B	**P07** EXCL1	**P06** EXCL2	**P04** REMI	**P03**	**P12**	**#RESET**	**VSS**
C	**P15** SCL	**P16** EXCL0		**P01**	**P10**	**P27** EXCL3	**P26** TOUT
D	**LVDD**	**VSS**	**P14** SDA	**P00**	**P13** FOUT1	**LVDD**	**P30** FOUT3
E	**DSIO** P33	**DST2** P32	**DCLK** P31	**P22** SPICLK	**P25** SCLK	**VSS**	**OSC1**
F	**TEST0**	**P17** #SPISS	**P20** SDI	**P23** SIN	**VSS**	**#TEST4**	**OSC2**
G	**#TEST1**	**P21** SDO	**HVDD**	**P24** SOUT	**OSC4**	**OSC3**	**#TEST5**

Figure 1.3.1.1: Pinout diagram (WCSP-48pin)

1.3.2 Package (QFP12-48pin)

QFP12-48pin package

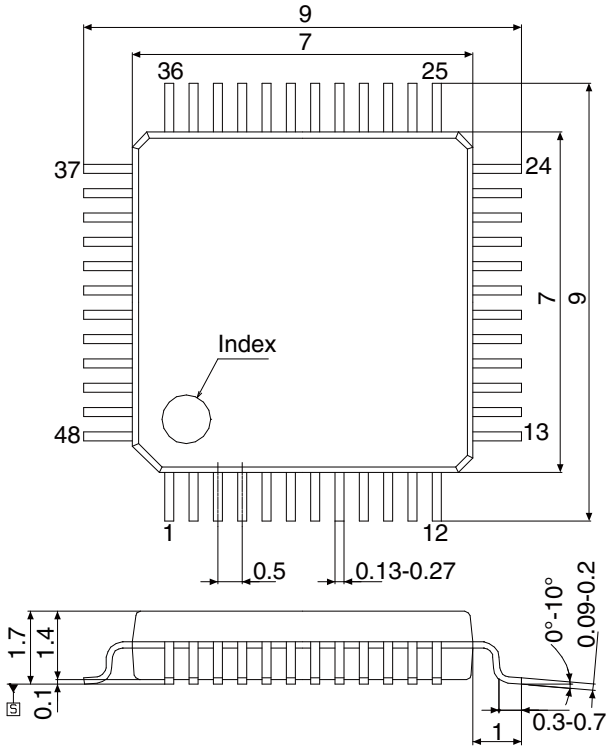
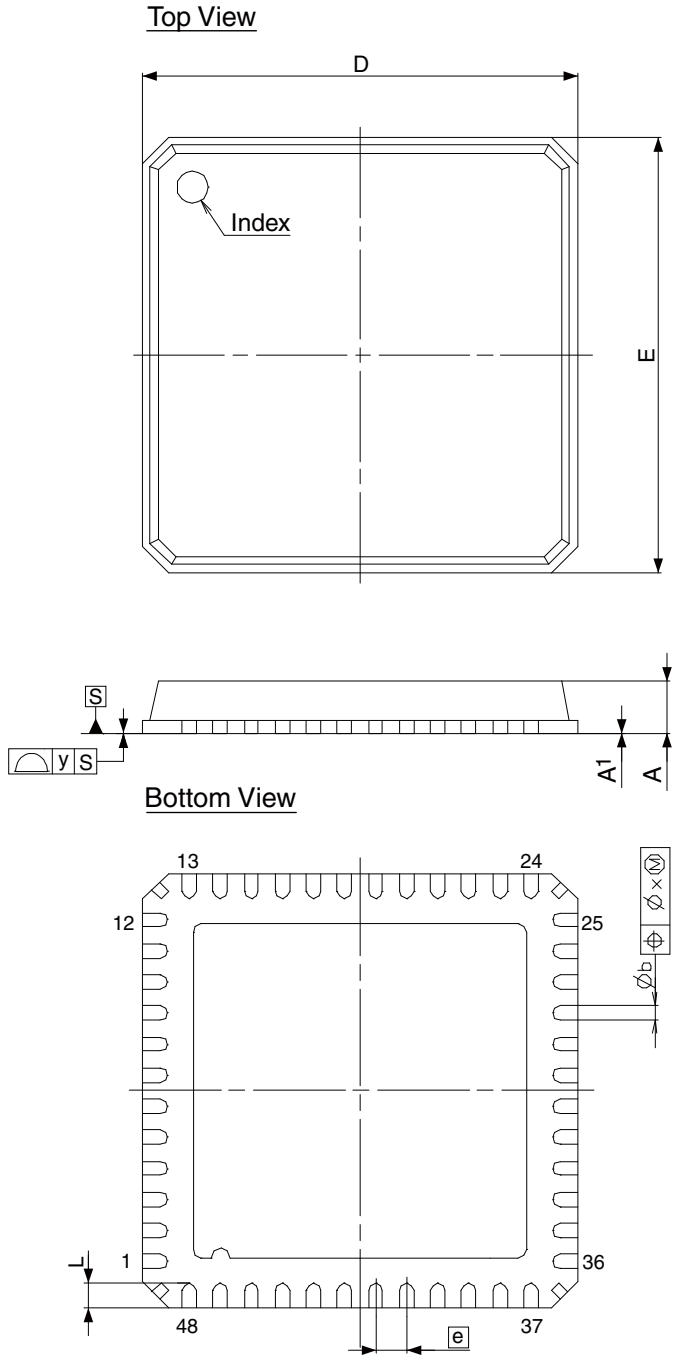


Figure 1.3.2.1 QFP12-48pin package scale

QFN7-48pin package



Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	-	7	-
E	-	7	-
A	-	-	1
A_1	0	-	-
b	0.17	-	0.3
e	-	0.5	-
L	0.3	-	0.5
X	-	-	0.1
y	-	-	0.08

Figure 1.3.2.2 QFN7-48pin package scale

1.3.3 Pin Descriptions

Table 1.3.3.1: Pin descriptions

Pin/ball No.			Name	I/O	Default status	Function
QFN7 -48PIN	QFP12 -48PIN	WCSP				
37	37	A1	**#TEST2**	I	I(Pull-Up)	Test pin (fixed at High during normal operations)
38	38	A2	**P05**/REMO	I/O	I(Pull-Up)	Input/output port pin (with interrupt)*/ Remote output pin
39	39	B3	**P04**/REMI	I/O	I(Pull-Up)	Input/output port pin (with interrupt)*/ Remote input pin
40	40	A3	**VSS**	-	-	Power supply pin (GND)
41	41	B4	**P03**	I/O	I(Pull-Up)	Input/output port pin (with interrupt)
42	42	A4	**P02**	I/O	I(Pull-Up)	Input/output port pin (with interrupt)
43	43	C4	**P01**	I/O	I(Pull-Up)	Input/output port pin (with interrupt)
44	44	D4	**P00**	I/O	I(Pull-Up)	Input/output port pin (with interrupt)
45	45	A5	**HVDD**	-	-	Power supply pin (HVDD+)
46	46	B5	**P12**	I/O	I(Pull-Up)	Input/output port pin (with interrupt)
47	47	A6	**P11**	I/O	I(Pull-Up)	Input/output port pin (with interrupt)
48	48	C5	**P10**	I/O	I(Pull-Up)	Input/output port pin (with interrupt)
1	1	A7	**#TEST3**	I	I(Pull-Up)	Test pin (fixed at High during normal operations)
2	2	B6	**#RESET**	I	I(Pull-Up)	Initial set input pin
3	3	B7	**VSS**	-	-	Power supply pin (GND)
4	4	C6	**P27**/EXCL3	I/O	I(Pull-Up)	Input/output port pin*/ T16E external clock input pin
5	5	C7	**P26**/TOUT	I/O	I(Pull-Up)	Input/output port pin*/ T16E PWM signal output pin
6	6	D5	**P13**/FOUT1	I/O	I(Pull-Up)	Input/output port pin (with interrupt)*/ OSC1 clock output pin
7	7	D7	**P30**/FOUT3	I/O	I(Pull-Up)	Input/output port pin*/ OSC3 division clock output pin
8	8	D6	**LVDD**	-	-	Power supply pin (LVDD+)
9	9	E6	**VSS**	-	-	Power supply pin (GND)
10	10	E7	**OSC1**	I	I	OSC1 oscillator input pin (permits external clock input)
11	11	F7	**OSC2**	O	O	OSC1 oscillator output pin
12	12	F6	**#TEST4**	I	I(Pull-Up)	Test pin (fixed at High during normal operations)
13	13	G7	**#TEST5**	I	I(Pull-Up)	Test pin (fixed at High during normal operations)
14	14	G6	**OSC3**	I	I	OSC3 oscillator input pin (permits external clock input)
15	15	G5	**OSC4**	O	O	OSC3 oscillator output pin
16	16	F5	**VSS**	-	-	Power supply pin (GND)
17	17	E5	**P25**/SCLK	I/O	I(Pull-Up)	Input/output port pin*/ UART clock input pin
18	18	G4	**P24**/SOUT	I/O	I(Pull-Up)	Input/output port pin*/ UART data output pin
19	19	F4	**P23**/SIN	I/O	I(Pull-Up)	Input/output port pin*/ UART data input pin
20	20	G3	**HVDD**	-	-	Power supply pin (HVDD+)
21	21	E4	**P22**/SPICLK	I/O	I(Pull-Up)	Input/output port pin*/ SPI clock input/output pin
22	22	G2	**P21**/SDO	I/O	I(Pull-Up)	Input/output port pin*/ SPI data output pin
23	23	F3	**P20**/SDI	I/O	I(Pull-Up)	Input/output port pin*/ SPI data input pin
24	24	F2	**P17**/#SPISS	I/O	I(Pull-Up)	Input/output port pin (with interrupt)*/ SPI slave select input pin
25	25	G1	**#TEST1**	I	I(Pull-Up)	Test pin (fixed at High during normal operations)
26	26	F1	**TEST0**	I	I(Pull-Down)	Test pin (fixed at Low during normal operations)
27	27	E3	**DCLK**/P31	I/O	O(H)	On-chip debugger clock output pin* / input/output port pin
28	28	E2	**DST2**/P32	I/O	O(L)	On-chip debugger status output pin* / input/output port pin
29	29	E1	**DSIO**/P33	I/O	I(Pull-Up)	On-chip debugger data input/output pin*/ input/output port pin
30	30	D2	**VSS**	-	-	Power supply pin (GND)
31	31	D1	**LVDD**	-	-	Power supply pin (LVDD+)
32	32	D3	**P14**/SDA	I/O	I(Pull-Up)	Input/output port pin (with interrupt)*/ I2C data input/output pin
33	33	C1	**P15**/SCL	I/O	I(Pull-Up)	Input/output port pin (with interrupt)*/ I2C clock output pin
34	34	C2	**P16**/EXCL0	I/O	I(Pull-Up)	Input/output port pin (with interrupt)*/ T16 Ch.0 external clock input pin
35	35	B1	**P07**/EXCL1	I/O	I(Pull-Up)	Input/output port pin (with interrupt)*/ T16 Ch.1 external clock input pin
36	36	B2	**P06**/EXCL2	I/O	I(Pull-Up)	Input/output port pin (with interrupt)*/ T16 Ch.2 external clock input pin

Note: Pins appearing in bold and functions indicated by "*" are default settings.

2 CPU

The S1C17001 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 command 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

Command set

- Code length — Fixed 16-bit length
- Number of commands — 111 basic commands (184 in total)
- Execution cycle — Main commands executed in one cycle
- Immediate expansion commands — Expansion of immediate to 24 bits
- Compact, high-speed command 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 command 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 command)
- SLEEP (slp command)

2.2 CPU Registers

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

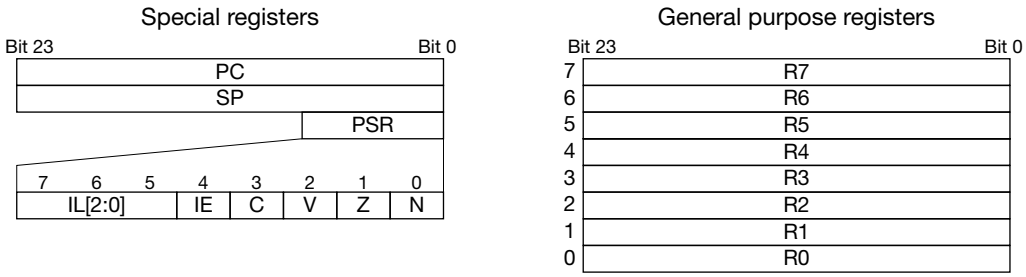


Figure 2.2.1: Registers

2.3 Command Set

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

Table 2.3.1: S1C17 core command 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 command
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

*1: Command 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.
*2: Coprocessor commands are reserved, since the S1C17001 does not include a coprocessor.

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 S1C17001 vector table starts from address 0x8000. The vector table base address can be read from the TTBR (vector table base register) at address 0xffff80.
Table 2.4.1 shows the S1C17001 vector table.

Table 2.4.1: Vector table

Vector No./ Software interrupt No.	Vector address	Hardware interrupt name	Hardware interrupt factor	Priority
0 (0x00)	0x8000	Reset	• Low input to #RESET pin • Watchdog timer overflow *2	1
1 (0x01)	0x8004	Irregular address interrupt	Memory access command	2
–	(0xfffc00)	Debug interrupt	brk command etc.	3
2 (0x02)	0x8008	NMI	Watchdog timer overflow *2	4
3 (0x03)	0x800c	reserved	–	–
4 (0x04)	0x8010	P0 port interrupt	P00 to P07 port input	High *1
5 (0x05)	0x8014	P1 port interrupt	P10 to P17 port input	↑
6 (0x06)	0x8018	Stopwatch timer interrupt	• Timer 100 Hz signal • Timer 10 Hz signal • Timer 1 Hz signal	
7 (0x07)	0x801c	Clock timer interrupt	• Timer 32 Hz signal • Timer 8 Hz signal • Timer 2 Hz signal • Timer 1 Hz signal	
8 (0x08)	0x8020	8-bit OSC1 timer interrupt	Compare match	
9 (0x09)	0x8024	reserved	–	
10 (0x0a)	0x8028			
11 (0x0b)	0x802c	PWM & capture timer interrupt	• Compare A • Compare B	
12 (0x0c)	0x8030	8-bit timer interrupt	Timer underflow	
13 (0x0d)	0x8034	16-bit timer Ch.0 interrupt	Timer underflow	
14 (0x0e)	0x8038	16-bit timer Ch.1 interrupt	Timer underflow	
15 (0x0f)	0x803c	16-bit timer Ch.2 interrupt	Timer underflow	
16 (0x10)	0x8040	UART interrupt	• Transmit buffer empty • Receive buffer full • Receive error	
17 (0x11)	0x8044	Remote controller interrupt	• Data length counter underflow • Input rising edge detection • Input falling edge detection	
18 (0x12)	0x8048	SPI interrupt	• Transmit buffer empty • Receive buffer full	
19 (0x13)	0x804c	I²C interrupt	• Transmit buffer empty • Receive buffer full	
20 (0x14) : 31 (0x1f)	0x8050 : 0x807c	reserved	–	↓ Low *1

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

0xffff80: Vector Table Base Register (TTBR)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Vector Table Base Register (TTBR)	**0xffff80** (32 bits)	D31–24	**–**	Unused (fixed at 0)	0x0	0x0	R	
		D23–0	**TTBR[23:0]**	Vector table base address	0x8000	0x8000	R	

2.5 Processor Information

The S1C17001 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 S1C17001 memory map.

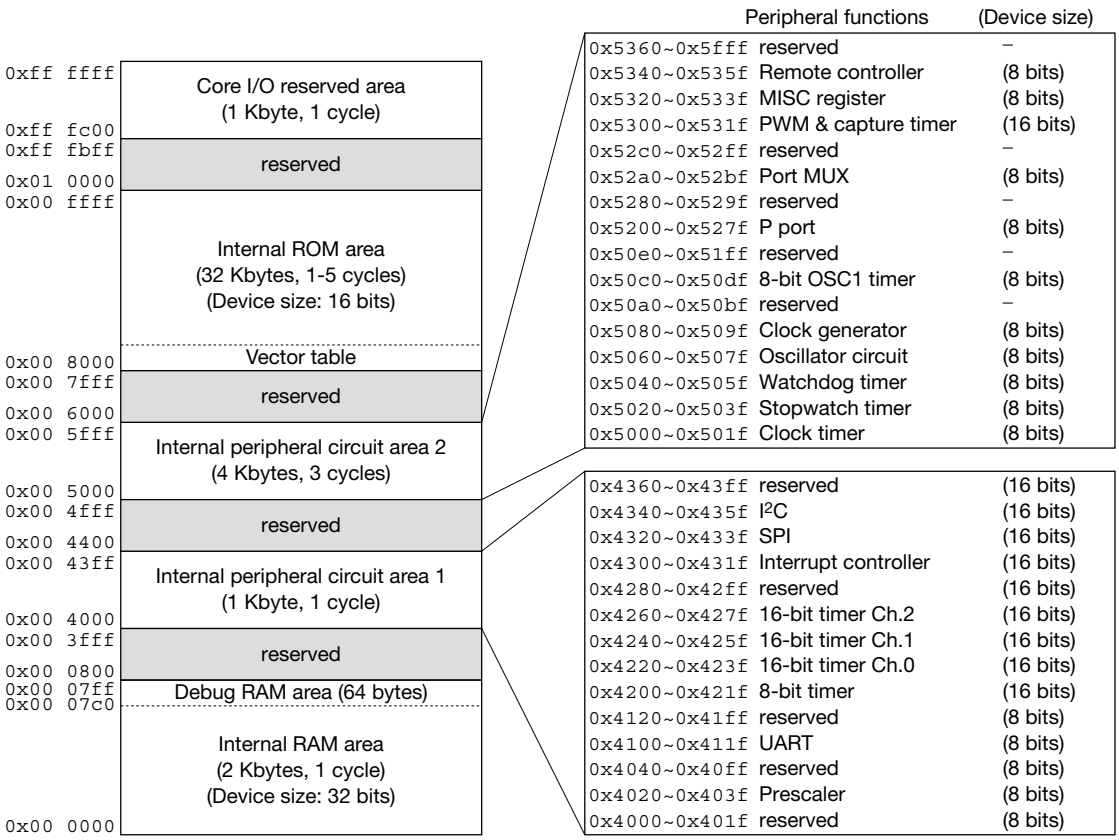


Figure 3.1: S1C17001 memory map

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 command (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.

Bus cycle calculation example

Number of bus cycles when accessing internal peripheral circuit area 2 (8-bit device, 3 cycles) from CPU using 16-bit read/write command:
3 cycles x 2 bus accesses = 6 CCLK cycles

3.1.1 Access Size Restrictions

When programming, note that the modules listed below are subject to access size restrictions.

SPI, I²C

The SPI and I²C registers can be accessed only with 16-bit read/write commands.

All other modules can be accessed using 8-bit, 16-bit, and 32-bit commands. 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 Command Execution Cycle Restrictions

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

- If a command is executed for an internal ROM area while accessing internal ROM and internal peripheral circuit area 2 (0x5000 onward) data
- If a command is executed for an internal RAM area while accessing internal RAM area data

3.2 Internal ROM Area

3.2.1 Internal ROM

The 32 Kbyte area from address 0x8000 to 0xffff is ROM. This area can be used for writing application programs and data. Address 0x8000 is defined as the vector table base address, and the vector table must be placed at the start of this area (refer to "2.4 Vector Table"). ROM reads take 1 to 5 cycles.

3.2.2 ROM Read Access Cycle Settings

Set the IROM area read access cycles using FLCYC[2:0] (D[2:0]/MISC_FL register) to retain compatibility with S1C17701. Normally, set FLCYC[2:0] to 0x4.

0x5320: ROM Control Register (MISC_FL)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
ROM Control Register (MISC_FL)	**0x5320** (8 bits)	D7–3	–	reserved	–		–	–	0 when being read.
		D2–0	**FLCYC[2:0]**	ROM read access cycle	FLCYC[2:0]	Read cycle	0x3	R/W	
					0x7–0x5	reserved			
					0x4	1 cycle			
					0x3	5 cycles			
					0x2	4 cycles			
					0x1	3 cycles			
					0x0	2 cycles			

3.3 Internal RAM Area

3.3.1 Internal RAM

The 2 Kbyte area from address 0x0 to 0x7ff is RAM. This RAM can be accessed in one cycle. In addition to storing variables, it can also be used to copy command codes and execute them rapidly in RAM.

Note: The last 64 bytes of the internal RAM (0x7c0 to 0x7ff) 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.

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 (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 three cycles.

- 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 timer (T8OSC1, 8-bit device)
- Input/output port & port MUX (P, 8-bit device)
- PWM & capture timer (T16E, 16-bit device)
- MISC register (MISC, 8-bit device)
- Remote controller (REMC, 8-bit device)

3.4.3 I/O Map

The I/O map for the internal peripheral circuit area is shown below. For more information on control registers, refer to the I/O register list in the Appendix or the corresponding peripheral circuit explanations.

Note: Addresses indicated as "Reserved" or blank unused peripheral circuit areas should not be accessed by application programs.

Table 3.4.3.1: I/O map (internal peripheral circuit area 1)

Peripheral circuit	Address	Register name		Function
Prescaler (8-bit device)	0x4020	PSC_CTL	Prescaler Control Register	Prescaler start/stop control
	0x4021 to 0x403f	–	–	Reserved
UART (with IrDA) (8-bit device)	0x4100	UART_ST	UART Status Register	Transfer, buffer, and error status display
	0x4101	UART_TXD	UART Transmit Data Register	Transmission data
	0x4102	UART_RXD	UART Receive Data Register	Received data
	0x4103	UART_MOD	UART Mode Register	Transfer data format setting
	0x4104	UART_CTL	UART Control Register	Data transfer control
	0x4105	UART_EXP	UART Expansion Register	IrDA mode setting
	0x4106 to 0x411f	–	–	Reserved
8-bit timer (with F mode) (16-bit device)	0x4200	T8F_CLK	8-bit Timer Input Clock Select Register	Prescaler output clock selection
	0x4202	T8F_TR	8-bit Timer Reload Data Register	Reload data setting
	0x4204	T8F_TC	8-bit Timer Counter Data Register	Counter data
	0x4206	T8F_CTL	8-bit Timer Control Register	Timer mode setting and timer RUN/STOP
	0x4208 to 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 to 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 to 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 to 0x427f	–	–	Reserved
Interrupt controller (16-bit device)	0x4300	ITC_IFLG	Interrupt Flag Register	Interrupt occurrence status display/reset
	0x4302	ITC_EN	Interrupt Enable Register	Maskable interrupt permission/prohibition
	0x4304	ITC_CTL	ITC Control Register	ITC operation permission/prohibition
	0x4306	ITC_ELV0	External Interrupt Level Setup Register 0	P0/P1 port interrupt level and trigger mode setting
	0x4308	ITC_ELV1	External Interrupt Level Setup Register 1	Stopwatch timer and clock timer interrupt level and trigger mode setting
	0x430a	ITC_ELV2	External Interrupt Level Setup Register 2	8-bit OSC1 timer interrupt level and trigger mode setting
	0x430c	ITC_ELV3	External Interrupt Level Setup Register 3	PWM & capture timer interrupt level and trigger mode setting
	0x430e	ITC_ILV0	Internal Interrupt Level Setup Register 0	8-bit timer and 16-bit timer Ch.0 interrupt level setting
	0x4310	ITC_ILV1	Internal Interrupt Level Setup Register 1	16-bit timer Ch.1 and 16-bit timer Ch.2 interrupt level setting
	0x4312	ITC_ILV2	Internal Interrupt Level Setup Register 2	UART and remote controller interrupt level setting
	0x4314	ITC_ILV3	Internal Interrupt Level Setup Register 3	SPI and I²C interrupt level setting
	0x4316 to 0x431f	–	–	Reserved
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	Received data
	0x4326	SPI_CTL	SPI Control Register	SPI mode and data transfer permission setting
	0x4328 to 0x433f	–	–	Reserved
I²C (16-bit device)	0x4340	I2C_EN	I²C Enable Register	I²C module enable
	0x4342	I2C_CTL	I²C Control Register	I²C control and transfer status display
	0x4344	I2C_DAT	I²C Data Register	Transfer data
	0x4346	I2C_ICTL	I²C Interrupt Control Register	I²C interrupt control
	0x4348 to 0x435f	–	–	Reserved

Table 3.4.3.2: I/O map (internal peripheral circuit area 2)

Peripheral circuit	Address	Register name		Function
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 to 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 to 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 to 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 to 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 to 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 to 0x50df	–	–	Reserved
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_IO	P0 Port I/O Direction Control Register	P0 port input/output direction selection
	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 to 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_IO	P1 Port I/O Direction Control Register	P1 port input/output direction selection
	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 to 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_IO	P2 Port I/O Direction Control Register	P2 port input/output direction selection
	0x5223	P2_PU	P2 Port Pull-up Control Register	P2 port pull-up control
	0x5224 to 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_IO	P3 Port I/O Direction Control Register	P3 port input/output direction selection
	0x5233	P3_PU	P3 Port Pull-up Control Register	P3 port pull-up control
	0x5234 to 0x527f	–	–	Reserved
	0x52a0	P0_PMUX	P0 Port Function Select Register	P0 port function selection
	0x52a1	P1_PMUX	P1 Port Function Select Register	P1 port function selection
	0x52a2	P2_PMUX	P2 Port Function Select Register	P2 port function selection
	0x52a3	P3_PMUX	P3 Port Function Select Register	P3 port function selection
	0x52a4 to 0x52bf	–	–	Reserved

Peripheral circuit	Address	Register name		Function
PWM & capture timer (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 mask setting
	0x530c	T16E_IFLG	PWM Timer Interrupt Flag Register	Interrupt occurrence status display/reset
	0x530e to 0x531f	–	–	Reserved
MISC register (8-bit device)	0x5320	MISC_FL	ROM Control Register	ROM access condition setting
	0x5321	–	–	Reserved
	0x5322	MISC_OSC1	OSC1 Peripheral Control Register	OSC1 operation peripheral function setting for debugging
	0x5323 to 0x533f	–	–	Reserved
Remote controller (8-bit device)	0x5340	REMC_CFG	REMC Configuration Register	Transfer selection and permission
	0x5341	REMC_PSC	REMC Prescaler Clock Select Register	Prescaler output clock selection
	0x5342	REMC_CARH	REMC H Carrier Length Setup Register	Carrier H section length setting
	0x5343	REMC_CARL	REMC L Carrier Length Setup Register	Carrier L section length setting
	0x5344	REMC_ST	REMC Status Register	Transfer bit
	0x5345	REMC_LCNT	REMC Length Counter Register	Transfer data length setting
	0x5346	REMC_IMSK	REMC Interrupt Mask Register	Interrupt mask setting
	0x5347	REMC_IFLG	REMC Interrupt Flag Register	Interrupt occurrence status display/reset
	0x5348 to 0x535f	–	–	Reserved

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	0xffff80	TTBR	Vector Table Base Register	Vector table base address display
	0xffff84	IDIR	Processor ID Register	Processor ID display
	0xffff90	DBRAM	Debug RAM Base Register	Debugging RAM base address display

For more information on TTBR, refer to “2.4 Vector Table”; and for more information on IDIR, refer to “2.5 Processor Information.”
For more information on DBRAM, refer to “22 On-chip Debugging (DBG).”

This page intentionally left blank.

4 Power Supply Voltage

The S1C17001 operation power supply voltages are given below.

Core voltage (LV_{DD}): 1.65 V to 2.7 V
I/O voltage (HV_{DD}): 1.65 V to 3.6 V

Supply voltages within the respective ranges to LV_{DD} and HV_{DD} pins with the V_{SS} pin as GND.
The S1C17001 has two LV_{DD} pins, two HV_{DD} pins, and five V_{SS} pins. All must be connected to the + power supply and GND. None should be left open.

This page intentionally left blank.

5 Initial Reset

5.1 Initial Reset Factors

Shown below are the three different initial reset factors for initializing S1C17001 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.

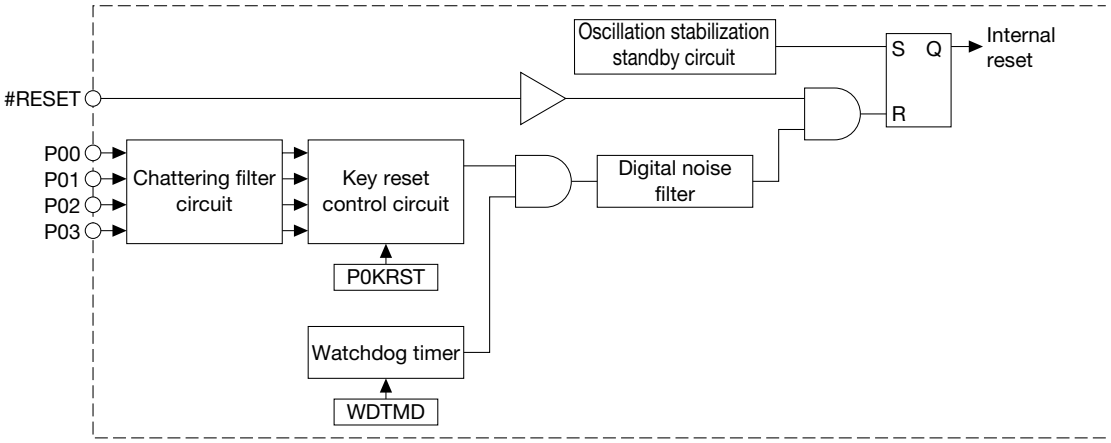


Figure 5.1.1: Initial reset circuit configuration

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 S1C17001 reliably, the #RESET pin must be maintained at Low level for at least the specified duration after the power supply voltage rises. (Refer to "24.5 AC 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 S1C17001 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 (1024/fosc3 seconds*) 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 OSC3 clock after the reset is canceled.

*fosc3: 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 command execution.

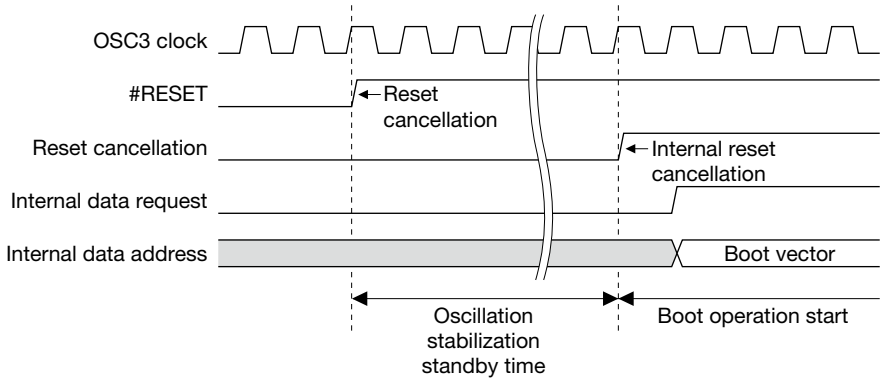


Figure 5.2.1: Sequence of operations after initial reset cancellation

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 and display memory should be initialized via software, since they are 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 S1C17001 features the following 14 different types of hardware interrupts:

1. P00 to P07 input interrupt (8 types)
2. P10 to P17 input interrupt (8 types)
3. Stopwatch timer interrupt (3 types)
4. Clock timer interrupt (4 types)
5. 8-bit OSC1 timer interrupt (1 type)
6. PWM & capture timer interrupt (2 types)
7. 8-bit timer interrupt (1 type)
8. 16-bit timer Ch.0 interrupt (1 type)
9. 16-bit timer Ch.1 interrupt (1 type)
10. 16-bit timer Ch.2 interrupt (1 type)
11. UART interrupt (3 types)
12. Remote controller interrupt (3 types)
13. SPI interrupt (2 types)
14. I²C interrupt (2 types)

The various interrupt circuits include interrupt flags to indicate an interrupt request from a neighboring module and interrupt enable bits to permit/prohibit interrupts. The interrupt level (priority) for determining the processing order when multiple interrupts occur simultaneously can be set separately for each interrupt circuit.
Each interrupt circuit includes the number of interrupt factors indicated in parentheses above. The respective peripheral module register controls the specific interrupt factor used to generate the interrupt request to the ITC. For detailed information on interrupt factors and interrupt factor control, refer to the discussion of the peripheral module.
Figure 6.1.1 illustrates the interrupt system configuration.

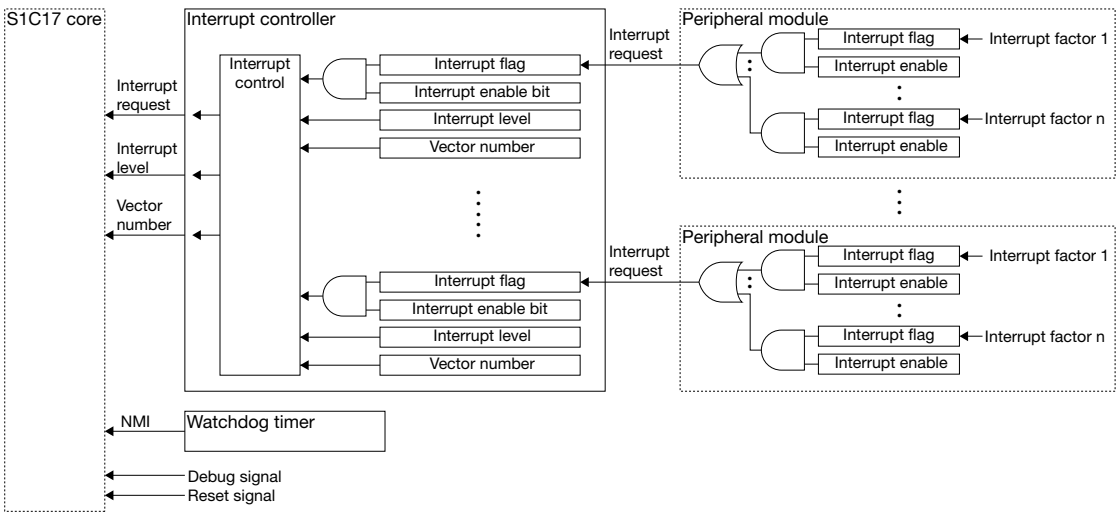


Figure 6.1.1: Interrupt system

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 S1C17001 vector table starts from address 0x8000. The vector table base address can be read from the TTBR register (0xffff80).
Table 6.2.1 shows the S1C17001 vector table.

Table 6.2.1: Vector table

Vector No./ Software interrupt No.	Vector address	Hardware interrupt name	Hardware interrupt factor	Priority
0 (0x00)	0x8000	Reset	• Low input to #RESET pin • Watchdog timer overflow *2	1
1 (0x01)	0x8004	Irregular address interrupt	Memory access command	2
–	(0xfffc00)	Debug interrupt	brk command etc.	3
2 (0x02)	0x8008	NMI	Watchdog timer overflow *2	4
3 (0x03)	0x800c	reserved	–	–
4 (0x04)	0x8010	P0 port interrupt	P00 to P07 port input	High *1
5 (0x05)	0x8014	P1 port interrupt	P10 to P17 port input	↑
6 (0x06)	0x8018	Stopwatch timer interrupt	• Timer 100 Hz signal • Timer 10 Hz signal • Timer 1 Hz signal	
7 (0x07)	0x801c	Clock timer interrupt	• Timer 32 Hz signal • Timer 8 Hz signal • Timer 2 Hz signal • Timer 1 Hz signal	
8 (0x08)	0x8020	8-bit OSC1 timer interrupt	Compare match	
9 (0x09)	0x8024	reserved	–	
10 (0x0a)	0x8028			
11 (0x0b)	0x802c	PWM & capture timer interrupt	• Compare A • Compare B	
12 (0x0c)	0x8030	8-bit timer interrupt	Timer underflow	
13 (0x0d)	0x8034	16-bit timer Ch.0 interrupt	Timer underflow	
14 (0x0e)	0x8038	16-bit timer Ch.1 interrupt	Timer underflow	
15 (0x0f)	0x803c	16-bit timer Ch.2 interrupt	Timer underflow	
16 (0x10)	0x8040	UART interrupt	• Transmit buffer empty • Receive buffer full • Receive error	
17 (0x11)	0x8044	Remote controller interrupt	• Data length counter underflow • Input rising edge detection • Input falling edge detection	
18 (0x12)	0x8048	SPI interrupt	• Transmit buffer empty • Receive buffer full	
19 (0x13)	0x804c	I²C interrupt	• Transmit buffer empty • Receive buffer full	
20 (0x14) : 31 (0x1f)	0x8050 : 0x807c	reserved	–	↓ Low *1

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

Vector numbers 4 to 19 are assigned maskable interrupts supported by the S1C17001.

6.3 Maskable Interrupt Control

6.3.1 ITC Enable

To use ITC, set ITEN (D0/ITC_CTL register) to 1.

∗ **ITEN: ITC Enable Bit in the ITC Control (ITC_CTL) Register (D0/0x4304)**

If ITEN is 0, maskable interrupts will not occur, regardless of the other register settings.

6.3.2 Interrupt Request from Peripheral Module and Interrupt Flag

If an interrupt factor for a permitted interrupt occurs in a peripheral module, that module sends an interrupt request signal to the ITC. This interrupt request signal causes the corresponding interrupt flag inside the ITC to be set to 1. The interrupt flag is maintained at 1 until it is reset to 0, indicating that an interrupt request was received from the peripheral module. The interrupt flag status can be read from the ITC_IFLG register (0x4300).
Table 6.3.2.1 shows the correspondence between interrupt factors and interrupt flags.

Table 6.3.2.1: Hardware interrupt factors and interrupt flags

Vector No.	Hardware interrupt request	Interrupt flag
4	P0 port interrupt: P00 to P07 port input	EIFT0 (D0/ITC_IFLG register)
5	P1 port interrupt: P10 to P17 port input	EIFT1 (D1/ITC_IFLG register)
6	Stopwatch timer interrupt: 100 Hz/10 Hz/1 Hz signal	EIFT2 (D2/ITC_IFLG register)
7	Clock timer interrupt: 32 Hz/8 Hz/2 Hz/1 Hz signal	EIFT3 (D3/ITC_IFLG register)
8	8-bit OSC1 timer interrupt: Compare match	EIFT4 (D4/ITC_IFLG register)
11	PWM & capture timer interrupt: Compare A/Compare B match	EIFT7 (D7/ITC_IFLG register)
12	8-bit timer interrupt: Timer underflow	IIFT0 (D8/ITC_IFLG register)
13	16-bit timer Ch.0 interrupt: Timer underflow	IIFT1 (D9/ITC_IFLG register)
14	16-bit timer Ch.1 interrupt: Timer underflow	IIFT2 (D10/ITC_IFLG register)
15	16-bit timer Ch.2 interrupt: Timer underflow	IIFT3 (D11/ITC_IFLG register)
16	UART interrupt: Transmit buffer empty/Receive buffer full/Receive error	IIFT4 (D12/ITC_IFLG register)
17	Remote controller interrupt: Data length counter underflow/Input rise-up/Input drop-off	IIFT5 (D13/ITC_IFLG register)
18	SPI interrupt: Transmit buffer empty/Receive buffer full	IIFT6 (D14/ITC_IFLG register)
19	I²C interrupt: Transmit buffer empty/Receive buffer full	IIFT7 (D15/ITC_IFLG register)

The ITC generates interrupts to the S1C17 core using the interrupt flags.
If an interrupt flag is set to 1 with the interrupt permitted (refer to next section for details), the ITC sends the interrupt request, interrupt level, and vector number signals to the S1C17 core.

An interrupt flag set to 1 is reset by writing 1. The interrupt flag should be reset to 0 during the interrupt processing routine. If the interrupt flag is not reset by the interrupt processing routine, the same interrupt will recur after the interrupt processing routine has ended. (The interrupt is prohibited during interrupt processing and returned to the permitted state on execution of the reti command after interrupt processing.)
Note however that the interrupt flags (EIFT0 to EIFT7) for interrupts set in the level trigger (refer to Section 6.3.5) are not reset by writing 1. These interrupt flags are reset when the interrupt source sets the interrupt signal to inactive.

Refer to the interrupt source module section for detailed information on the conditions under which interrupt factors arise and individual module interrupt settings are made.

6.3.3 Interrupt Permission/Prohibition

Sending an interrupt request to the S1C17 core requires first permitting the individual interrupts using the interrupt enable bit inside the ITC_EN register (0x4302) corresponding to the interrupt flag. Setting the interrupt enable bit to 1 permits interrupts, while setting it to 0 (default) prohibits interrupts. The interrupt enable bit does not affect the interrupt flag. Interrupt flags for interrupt requests generated by a peripheral module will be set regardless of the interrupt enable bit setting.
Table 6.3.3.1 shows the correspondence between interrupt enable bits and interrupt flags.

Table 6.3.3.1: Interrupt enable bit list

Vector No.	Hardware interrupt	Interrupt flag	Interrupt enable bit
4	P0 port interrupt	EIFT0 (D0/ITC_IFLG register)	EIEN0 (D0/ITC_EN register)
5	P1 port interrupt	EIFT1 (D1/ITC_IFLG register)	EIEN1 (D1/ITC_EN register)
6	Stopwatch timer interrupt	EIFT2 (D2/ITC_IFLG register)	EIEN2 (D2/ITC_EN register)
7	Clock timer interrupt	EIFT3 (D3/ITC_IFLG register)	EIEN3 (D3/ITC_EN register)
8	8-bit OSC1 timer interrupt	EIFT4 (D4/ITC_IFLG register)	EIEN4 (D4/ITC_EN register)
11	PWM & capture timer interrupt	EIFT7 (D7/ITC_IFLG register)	EIEN7 (D7/ITC_EN register)
12	8-bit timer interrupt	IIFT0 (D8/ITC_IFLG register)	IIEN0 (D8/ITC_EN register)
13	16-bit timer Ch.0 interrupt	IIFT1 (D9/ITC_IFLG register)	IIEN1 (D9/ITC_EN register)
14	16-bit timer Ch.1 interrupt	IIFT2 (D10/ITC_IFLG register)	IIEN2 (D10/ITC_EN register)
15	16-bit timer Ch.2 interrupt	IIFT3 (D11/ITC_IFLG register)	IIEN3 (D11/ITC_EN register)
16	UART interrupt	IIFT4 (D12/ITC_IFLG register)	IIEN4 (D12/ITC_EN register)
17	Remote controller interrupt	IIFT5 (D13/ITC_IFLG register)	IIEN5 (D13/ITC_EN register)
18	SPI interrupt	IIFT6 (D14/ITC_IFLG register)	IIEN6 (D14/ITC_EN register)
19	I²C interrupt	IIFT7 (D15/ITC_IFLG register)	IIEN7 (D15/ITC_EN register)

Note: • To prevent generating unnecessary interrupts, always set the interrupt flags before permitting interrupts by writing 1 to the interrupt enable bit.
- To generate an actual interrupt, the IE bit in the S1C17 core Processor Status Register (PSR) must be set to 1, in addition to the interrupt enable bit. The S1C17 core will not accept maskable interrupt requests if the IE bit is set to 0. In this case, interrupt requests from the ITC will be retained and accepted after the IE bit is set to 1.

6.3.4 Processing for Multiple Interrupts

The ITC ITC_ELV*x* and ITC_ILV*x* registers (0x4306 to 0x4314) set the interrupt levels (0 to 7) for the various interrupt factors.

Table 6.3.4.1: Interrupt level setting bits

Vector No.	Hardware interrupt	Interrupt level setting bit	Register address
4	P0 port interrupt	EILV0[2:0] (D[2:0]/ITC_ELV0 register)	0x4306
5	P1 port interrupt	EILV1[2:0] (D[10:8]/ITC_ELV0 register)	0x4306
6	Stopwatch timer interrupt	EILV2[2:0] (D[2:0]/ITC_ELV1 register)	0x4308
7	Clock timer interrupt	EILV3[2:0] (D[10:8]/ITC_ELV1 register)	0x4308
8	8-bit OSC1 timer interrupt	EILV4[2:0] (D[2:0]/ITC_ELV2 register)	0x430a
11	PWM & capture timer interrupt	EILV7[2:0] (D[10:8]/ITC_ELV3 register)	0x430c
12	8-bit timer interrupt	IILV0[2:0] (D[2:0]/ITC_ILV0 register)	0x430e
13	16-bit timer Ch.0 interrupt	IILV1[2:0] (D[10:8]/ITC_ILV0 register)	0x430e
14	16-bit timer Ch.1 interrupt	IILV2[2:0] (D[2:0]/ITC_ILV1 register)	0x4310
15	16-bit timer Ch.2 interrupt	IILV3[2:0] (D[10:8]/ITC_ILV1 register)	0x4310
16	UART interrupt	IILV4[2:0] (D[2:0]/ITC_ILV2 register)	0x4312
17	Remote controller interrupt	IILV5[2:0] (D[10:8]/ITC_ILV2 register)	0x4312
18	SPI interrupt	IILV6[2:0] (D[2:0]/ITC_ILV3 register)	0x4314
19	I²C interrupt	IILV7[2:0] (D[10:8]/ITC_ILV3 register)	0x4314

The interrupt level can range from 0 to 7.
The interrupt level set is issued to the S1C17 core at the same time as an interrupt request from the ITC. This interrupt level is used in the S1C17 core to prohibit subsequent interrupts with the same or lower levels (refer to Section 6.3.6).
Initial resets reset all interrupt levels to 0. The S1C17 core rejects interrupt requests if the interrupt level is 0.

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_ELV*x* and ITC_ILV*x* registers 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.

6.3.5 Interrupt Trigger Modes

The ITC includes two trigger modes – pulse trigger mode and level trigger mode – which enable acceptance of interrupt requests setting the EIFT flag as pulse or level signals.
Trigger mode can be selected using the EITG*x* bits within the ITC_ELV*x* register (0x4306 to 0x4308). Setting the EITG*x* bits to 1 selects the level trigger mode; setting them to 0 (default) selects pulse trigger mode.

Note: All EITG*x* bits should be set to 1 (level trigger mode) for the S1C17001.

Table 6.3.5.1: Trigger mode selector bits

Hardware interrupt	Trigger mode selector bit	Register address
P0 port interrupt	EITG0 (D4/ITC_ELV0 register)	0x4306
P1 port interrupt	EITG1 (D12/ITC_ELV0 register)	0x4306
Stopwatch timer interrupt	EITG2 (D4/ITC_ELV1 register)	0x4308
Clock timer interrupt	EITG3 (D12/ITC_ELV1 register)	0x4308
8-bit OSC1 timer interrupt	EITG4 (D4/ITC_ELV2 register)	0x430a
PWM & capture timer interrupt	EITG7 (D12/ITC_ELV3 register)	0x430c

The module setting the IIFT flag outputs a pulse signal only as the interrupt request to the ITC. No trigger mode selector bit is provided.

Pulse trigger mode

In pulse trigger mode, the ITC samples the interrupt signal using the system clock rising edge. If a pulse High period is detected, the ITC sets the interrupt flag (IIFT*x*) to 1 and stops sampling that interrupt signal. The ITC resumes sampling of the interrupt signal after the application program resets the interrupt flag (IIFT*x*) to 0 (via interrupt processing routine).

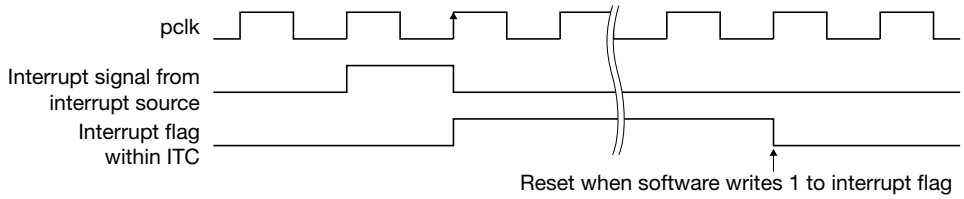


Figure 6.3.5.1: Pulse trigger mode

Note: The S1C17001 interrupts listed below are in pulse trigger mode. If an interrupt occurs, reset the interrupt flag IIFT*x* (to 1) within the interrupt processing routine.

- **8-bit timer interrupt**
- **16-bit timer Ch.0 interrupt**
- **16-bit timer Ch.1 interrupt**
- **16-bit timer Ch.2 interrupt**
- **UART interrupt**
- **Remote controller interrupt**
- **SPI interrupt**
- **I²C interrupt**

Level trigger mode

In level trigger mode, the ITC samples the interrupt signal continuously using the system clock rising edge. The interrupt flag (EIFT*x*) is set to 1 if High level is detected and is reset to 0 if Low level is subsequently detected. Since interrupt flags (EIFT*x*) cannot be reset by writing 1 in this mode, the interrupt signal is held at High until the interrupt source module is accepted by the S1C17 core, and the interrupt signal must subsequently be cleared.

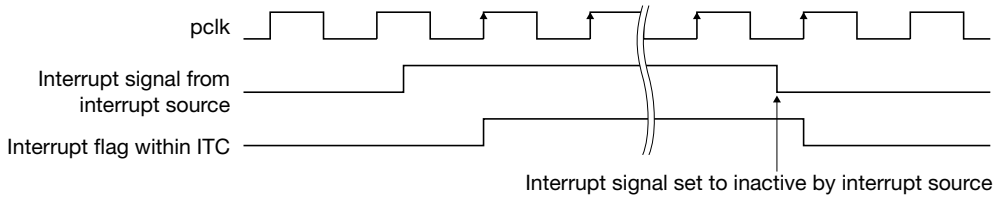


Figure 6.3.5.2: Level trigger mode

Note: The S1C17001 interrupts listed below are in level trigger mode. The interrupt flag within peripheral modules must be reset (to 1) within the interrupt processing routine rather than EIFT*x*

- P0 port interrupt
- P1 port interrupt
- Stopwatch timer interrupt
- Clock timer interrupt
- 8-bit OSC1 timer interrupt
- PWM & capture timer interrupt

For more information on interrupt flags for resetting, refer to the peripheral module description.

6.3.6 S1C17 Core Interrupt Processing

Maskable interrupts for the S1C17 core occur when all of the following conditions are met:
- ITEN (D0/ITC_CTL register) has been set to 1.

  * **ITEN: ITC Enable Bit in the ITC Control (ITC_CTL) Register (D0/0x4304)**

- The corresponding interrupt enable bit has been set to 1 for the interrupt factor.
- 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.

When an interrupt factor occurs, the corresponding interrupt flag is set to 1. This state is maintained until reset by the program or hardware (for interrupts set in level trigger mode). The interrupt factor is not cleared even if the conditions listed above remain unmet when the interrupt factor occurs. An interrupt occurs when the above conditions are met.
If multiple maskable interrupt factors occur simultaneously, the interrupt factor with the highest level 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 command 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 command returns the PSR to the state before the interrupt. The program resumes processing following the command being executed at the time the interrupt occurred via the next branch.

6.4 NMI

The S1C17001 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* commands. The vector table vector number (0 to 31) is specified by the operand immediate *imm5*. With the intl command, *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 by Interrupt Factors

HALT and SLEEP modes are canceled by interrupt factors, and the CPU starts up.
The interrupt factors capable of starting the CPU and specific program execution details after CPU startup (whether to branch into an interrupt processing routine) depend on the clock states in HALT and SLEEP modes.
For more information, refer to “B.1 Clock Control Power Saving” in Appendix B.

6.7 Control Register Details

Table 6.7.1 ITC register list

Address	Register name		Function
0x4300	ITC_IFLG	Interrupt Flag Register	Indicates and resets interrupt occurrence status.
0x4302	ITC_EN	Interrupt Enable Register	Permits/blocks maskable interrupts.
0x4304	ITC_CTL	ITC Control Register	Permits/blocks ITC operation.
0x4306	ITC_ELV0	External Interrupt Level Setup Register 0	Sets P0 and P1 port interrupt level and trigger mode.
0x4308	ITC_ELV1	External Interrupt Level Setup Register 1	Sets stopwatch timer, clock timer interrupt level and trigger mode.
0x430a	ITC_ELV2	External Interrupt Level Setup Register 2	Sets 8-bit OSC1 timer interrupt level and trigger mode.
0x430c	ITC_ELV3	External Interrupt Level Setup Register 3	Sets PWM & capture timer interrupt level and trigger mode.
0x430e	ITC_ILV0	Internal Interrupt Level Setup Register 0	Sets 8-bit timer and 16-bit timer Ch.0 interrupt level.
0x4310	ITC_ILV1	Internal Interrupt Level Setup Register 1	Sets 16-bit timer Ch.1 and 16-bit timer Ch.2 interrupt level.
0x4312	ITC_ILV2	Internal Interrupt Level Setup Register 2	Sets UART and remote controller interrupt level.
0x4314	ITC_ILV3	Internal Interrupt Level Setup Register 3	Sets SPI and I^2C interrupt levels.

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.

0x4300: Interrupt Flag Register (ITC_IFLG)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Interrupt Flag Register (ITC_IFLG)	**0x4300** (16 bits)	D15	**IIFT7**	I²C interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D14	**IIFT6**	SPI interrupt flag					0	R/W	
		D13	**IIFT5**	Remote controller interrupt flag					0	R/W	
		D12	**IIFT4**	UART interrupt flag					0	R/W	
		D11	**IIFT3**	16-bit timer Ch.2 interrupt flag					0	R/W	
		D10	**IIFT2**	16-bit timer Ch.1 interrupt flag					0	R/W	
		D9	**IIFT1**	16-bit timer Ch.0 interrupt flag					0	R/W	
		D8	**IIFT0**	8-bit timer interrupt flag					0	R/W	
		D7	**EIFT7**	PWM&capture timer interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1 in pulse trigger mode. Cannot be reset by software in level trigger mode.
		D6	**EIFT6**	reserved					0	R/W	
		D5	**EIFT5**	reserved					0	R/W	
		D4	**EIFT4**	8-bit OSC1 timer interrupt flag					0	R/W	
		D3	**EIFT3**	Clock timer interrupt flag					0	R/W	
		D2	**EIFT2**	Stopwatch timer interrupt flag					0	R/W	
		D1	**EIFT1**	P1 port interrupt flag					0	R/W	
		D0	**EIFT0**	P0 port interrupt flag					0	R/W	

D[15:8] IIFT[7:0]: Interrupt Flags (for Pulse Trigger)

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

The interrupt flags are reset to 1 if an interrupt factor occurs in the peripheral modules.
An interrupt is generated to the S1C17 core provided the following conditions are met:

1. The corresponding interrupt enable bit is set to 1.
2. No other interrupt having requests higher priority levels have occurred.
3. The PSR IE bit was set to 1 (interrupt permitted).
4. The corresponding interrupt level setting bit has been set to a higher level than the S1C17 core interrupt level (IL).

The interrupt flags are set to 1 when an interrupt factor occurs regardless of the interrupt enable bit or interrupt level setting bit states.
The interrupt flags must be reset and the PSR must be reset (by setting the IE bit to 1 or with the `reti` command) to accept the next interrupt after interrupt occurrence.
An interrupt factor flag set to 1 is reset by writing 1.

Table 6.7.2: Hardware interrupt factors and interrupt flags

Interrupt flag	Hardware interrupt factor
IIFT0 (D8)	8-bit timer interrupt: Timer underflow
IIFT1 (D9)	16-bit timer Ch.0 interrupt: Timer underflow
IIFT2 (D10)	16-bit timer Ch.1 interrupt: Timer underflow
IIFT3 (D11)	16-bit timer Ch.2 interrupt: Timer underflow
IIFT4 (D12)	UART interrupt: Transmit buffer empty/Receive buffer full/Receive error
IIFT5 (D13)	Remote controller interrupt: Data length counter underflow/Input rise-up/Input drop-off
IIFT6 (D14)	SPI interrupt: Transmit buffer empty/Receive buffer full
IIFT7 (D15)	I²C interrupt: Transmit buffer empty/Receive buffer full

D[7:0] EIFT[7:0]: Interrupt Flags (for Level Trigger)

These are interrupt flags indicating the interrupt factor occurrence status.

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

Refer to the description for IIFT[7:0].
Note that these interrupts must be set to level trigger mode in the ITC_ELV*x* registers (0x4306 to 0x430c). To reset the interrupt flags, rather than writing 1 to the bit, set the interrupt flag to 1 within the peripheral module.

Table 6.7.3: Hardware interrupt factors and interrupt flags

Interrupt flag	Hardware interrupt factor
EIFT0 (D0)	P0 port interrupt: P00 to P07 port input
EIFT1 (D1)	P1 port interrupt: P10 to P17 port input
EIFT2 (D2)	Stopwatch timer interrupt: 100 Hz/10 Hz/1 Hz signal
EIFT3 (D3)	Clock timer interrupt: 32 Hz/8 Hz/2 Hz/1 Hz signal
EIFT4 (D4)	8-bit OSC1 timer interrupt: Compare match
EIFT7 (D7)	PWM & capture timer interrupt: Compare A/Compare B match

Note: The interrupt flags are not reset even if maskable interrupt requests are accepted by the S1C17 core and branched to interrupt processing routines. Note that returning from an interrupt processing routine using the `reti` command without resetting the interrupt flags using the program will generate the same interrupt. Interrupt flags set to level trigger must be reset by the control register within the peripheral module.

0x4302: Interrupt Enable Register (ITC_EN)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Interrupt Enable Register (ITC_EN)	**0x4302** (16 bits)	D15	**IIEN7**	I²C interrupt enable	1	Enable	0	Disable	0	R/W	
		D14	**IIEN6**	SPI interrupt enable					0	R/W	
		D13	**IIEN5**	Remote controller interrupt enable					0	R/W	
		D12	**IIEN4**	UART interrupt enable					0	R/W	
		D11	**IIEN3**	16-bit timer Ch.2 interrupt enable					0	R/W	
		D10	**IIEN2**	16-bit timer Ch.1 interrupt enable					0	R/W	
		D9	**IIEN1**	16-bit timer Ch.0 interrupt enable					0	R/W	
		D8	**IIEN0**	8-bit timer interrupt enable					0	R/W	
		D7	**EIEN7**	PWM&capture timer interrupt enable					0	R/W	
		D6	**EIEN6**	reserved					0	R/W	
		D5	**EIEN5**	reserved					0	R/W	
		D4	**EIEN4**	8-bit OSC1 timer interrupt enable					0	R/W	
		D3	**EIEN3**	Clock timer interrupt enable					0	R/W	
		D2	**EIEN2**	Stopwatch timer interrupt enable					0	R/W	
		D1	**EIEN1**	P1 port interrupt enable					0	R/W	
		D0	**EIEN0**	P0 port interrupt enable					0	R/W	

D[15:0] IIEN[7:0], EIEN[7:0]: Interrupt Enable Bits

These bits permit or prohibit interrupt events.
1 (R/W): Interrupt permitted
0 (R/W): Interrupt prohibited (default)

Setting the interrupt enable bit to 1 permits interrupts. Setting it to 0 prohibits interrupts.
Even if interrupt prohibition has been set, the corresponding interrupt can still be used to cancel Stand-by mode.

Table 6.7.4: Hardware interrupt factors and interrupt enable bits

Interrupt enable bit	Hardware interrupt factor
EIEN0 (D0)	P0 port interrupt: P00 to P07 port input
EIEN1 (D1)	P1 port interrupt: P10 to P17 port input
EIEN2 (D2)	Stopwatch timer interrupt: 100 Hz/10 Hz/1 Hz signal
EIEN3 (D3)	Clock timer interrupt: 32 Hz/8 Hz/2 Hz/1 Hz signal
EIEN4 (D4)	8-bit OSC1 timer interrupt: Compare match
EIEN7 (D7)	PWM & capture timer interrupt: Compare A/Compare B match
IIEN0 (D8)	8-bit timer interrupt: Timer underflow
IIEN1 (D9)	16-bit timer Ch.0 interrupt: Timer underflow
IIEN2 (D10)	16-bit timer Ch.1 interrupt: Timer underflow
IIEN3 (D11)	16-bit timer Ch.2 interrupt: Timer underflow
IIEN4 (D12)	UART interrupt: Transmit buffer empty/Receive buffer full/Receive error
IIEN5 (D13)	Remote controller interrupt: Data length counter underflow/Input rise-up/Input drop-off
IIEN6 (D14)	SPI interrupt: Transmit buffer empty/Receive buffer full
IIEN7 (D15)	I²C interrupt: Transmit buffer empty/Receive buffer full

0x4304: ITC Control Register (ITC_CTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
ITC Control Register (ITC_CTL)	**0x4304** (16 bits)	D15–1	**–**	reserved	–	–	–	0 when being read.
		D0	**ITEN**	ITC enable	1 Enable 0 Disable	0	R/W	

D[15:1] Reserved

D0 ITEN: ITC Enable Bit

Permits interrupt control using the ITC.
1 (R/W): Permitted
0 (R/W): Prohibited (default)

Set to 1 before using the ITC.

0x4306: External Interrupt Level Setup Register 0 (ITC_ELV0)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
External Interrupt Level Setup Register 0 (ITC_ELV0)	**0x4306** (16 bits)	D15–13	**–**	reserved	–	–	–	0 when being read.
		D12	**EITG1**	P1 interrupt trigger mode	1 Level 0 Pulse	0	R/W	Be sure to set to 1.
		D11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**EILV1[2:0]**	P1 interrupt level	0 to 7	0x0	R/W	
		D7–5	**–**	reserved	–	–	–	0 when being read.
		D4	**EITG0**	P0 interrupt trigger mode	1 Level 0 Pulse	0	R/W	Be sure to set to 1.
		D3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**EILV0[2:0]**	P0 interrupt level	0 to 7	0x0	R/W	

D[15:13] Reserved

D12 EITG1: P1 Port Interrupt Trigger Mode Select Bit

Selects P1 port interrupt trigger mode. This should be set to 1 for the S1C17001.
1 (R/W): Level trigger mode
0 (R/W): Pulse trigger mode (default)

In pulse trigger mode, the ITC samples interrupt signals using system clock rising edges. When the pulse High period is detected, the ITC sets the interrupt flag (EIFT*x*) to 1 and stops sampling that interrupt signal. The ITC resumes interrupt signal sampling after the application program (interrupt processing routine) resets the interrupt flag (EIFT*x*) to 0.
In level trigger mode, the ITC samples interrupt signals using system clock rising edges. When High level is detected, the interrupt flag (EIFT*x*) is set to 1 and is subsequently reset to 0 when Low level is detected. Interrupt flags (EIFT*x*) cannot be reset by writing 1 in this mode. The interrupt signal must be maintained at High until the interrupt source module is accepted by the S1C17 core, and the interrupt signal must then be cleared.

D11 Reserved

D[10:8] EILV1[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_ELV*x* and ITC_ILV*x* registers (0x4306 to 0x4314) 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[7:5] Reserved

D4 EITG0: P0 Port Interrupt Trigger Mode Select Bit

Selects P0 port interrupt trigger mode. This should be set to 1 for the S1C17001.
1 (R/W): Level trigger mode
0 (R/W): Pulse trigger mode (default)

Refer to the EITG1 (D12) description.

D3 Reserved

D[2:0] EILV0[2:0]: P0 Port Interrupt Level Bits

Set the P0 port interrupt level (0 to 7). (Default: 0)
Refer to the EILV1[2:0] (D[10:8]) description.

0x4308: External Interrupt Level Setup Register 1 (ITC_ELV1)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
External Interrupt Level Setup Register 1 (ITC_ELV1)	**0x4308** (16 bits)	D15–13	**–**	reserved	–	–	–	0 when being read.
		D12	**EITG3**	CT interrupt trigger mode	1 Level 0 Pulse	0	R/W	Be sure to set to 1.
		D11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**EILV3[2:0]**	CT interrupt level	0 to 7	0x0	R/W	
		D7–5	**–**	reserved	–	–	–	0 when being read.
		D4	**EITG2**	SWT interrupt trigger mode	1 Level 0 Pulse	0	R/W	Be sure to set to 1.
		D3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**EILV2[2:0]**	SWT interrupt level	0 to 7	0x0	R/W	

D[15:13] Reserved

D12 EITG3: Clock Timer Interrupt Trigger Mode Select Bit

Selects clock timer interrupt trigger mode. This should be set to 1 for the S1C17001.
1 (R/W): Level trigger mode
0 (R/W): Pulse trigger mode (default)

Refer to the ITC_ELV0 register (0x4306) EITG1 (D12) description.

D11 Reserved

D[10:8] EILV3[2:0]: Clock Timer Interrupt Level Bits

Set the clock timer interrupt level (0 to 7). (Default: 0)
Refer to the ITC_ELV0 register (0x4306) EILV1[2:0] (D[10:8]) description.

D[7:5] Reserved

D4 EITG2: Stopwatch Timer Interrupt Trigger Mode Select Bit

Selects stopwatch timer interrupt trigger mode. This should be set to 1 for the S1C17001.
1 (R/W): Level trigger mode
0 (R/W): Pulse trigger mode (default)

Refer to the ITC_ELV0 register (0x4306) EITG1 (D12) description.

D3 Reserved

D[2:0] EILV2[2:0]: Stopwatch Timer Interrupt Level Bits

Set the stopwatch timer interrupt level (0 to 7). (Default: 0)
Refer to the ITC_ELV0 register (0x4306) EILV1[2:0] (D[10:8]) description.

0x430a: External Interrupt Level Setup Register 2 (ITC_ELV2)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
External Interrupt Level Setup Register 2 (ITC_ELV2)	**0x430a** (16 bits)	D15–5	**–**	reserved	–	–	–	0 when being read.
		D4	**EITG4**	T8OSC1 interrupt trigger mode	1 Level 0 Pulse	0	R/W	Be sure to set to 1.
		D3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**EILV4[2:0]**	T8OSC1 interrupt level	0 to 7	0x0	R/W	

D[15:5] Reserved

D4 EITG4: 8-bit OSC1 Timer Interrupt Trigger Mode Select Bit

Selects 8-bit OSC1 timer interrupt trigger mode. This should be set to 1 for the S1C17001.
1 (R/W): Level trigger mode
0 (R/W): Pulse trigger mode (default)

Refer to the ITC_ELV0 register (0x4306) EITG1 (D12) description.

D3 Reserved

D[2:0] EILV4[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_ELV0 register (0x4306) EILV1[2:0] (D[10:8]) description.

0x430c: External Interrupt Level Setup Register 3 (ITC_ELV3)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
External Interrupt Level Setup Register 3 (ITC_ELV3)	**0x430c** (16 bits)	D15–13	**–**	reserved	–	–	–	0 when being read.
		D12	**EITG7**	T16E interrupt trigger mode	1 Level 0 Pulse	0	R/W	Be sure to set to 1.
		D11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**EILV7[2:0]**	T16E interrupt level	0 to 7	0x0	R/W	
		D7–0	**–**	reserved	–	–	–	0 when being read.

D[15:13] Reserved

D12 EITG7: PWM & Capture Timer Interrupt Trigger Mode Select Bit

Selects PWM & capture timer interrupt trigger mode. This should be set to 1 for the S1C17001.

1 (R/W): Level trigger mode

0 (R/W): Pulse trigger mode (default)

Refer to the ITC_ELV0 register (0x4306) EITG1 (D12) description.

D11 Reserved

D[10:8] EILV7[2:0]: PWM & Capture Timer Interrupt Level Bits

Set the PWM & capture timer interrupt level (0 to 7). (Default: 0)

Refer to the ITC_ELV0 register (0x4306) EILV1[2:0] (D[10:8]) description.

D[7:0] Reserved

0x430e: Internal Interrupt Level Setup Register 0 (ITC_ILV0)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Internal Interrupt Level Setup Register 0 (ITC_ILV0)	**0x430e** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**IILV1[2:0]**	T16 Ch.0 interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**IILV0[2:0]**	T8 interrupt level	0 to 7	0x0	R/W	

D[15:11] Reserved

D[10:8] IILV1[2:0]: 16-bit Timer Ch.0 Interrupt Level Bits

Set the 16-bit timer Ch.0 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 permi tted by the interrupt enable bit, the ITC sends the interrupt request with the highest level set by the ITC_ELV*x* and ITC_ILV*x* registers (0x4306 to 0x4314) 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] IILV0[2:0]: 8-bit Timer Interrupt Level Bits

Set the 8-bit timer interrupt level (0 to 7). (Default: 0)
Refer to the IILV1[2:0] (D[10:8]) description.

0x4310: Internal Interrupt Level Setup Register 1 (ITC_ILV1)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Internal Interrupt Level Setup Register 1 (ITC_ILV1)	**0x4310** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**IILV3[2:0]**	T16 Ch.2 interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**IILV2[2:0]**	T16 Ch.1 interrupt level	0 to 7	0x0	R/W	

D[15:11] Reserved

D[10:8] IILV3[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_ILV0 register (0x430e) IILV1[2:0] (D[10:8]) description.

D[7:3] Reserved

D[2:0] IILV2[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_ILV0 register (0x430e) IILV1[2:0] (D[10:8]) description.

0x4312: Internal Interrupt Level Setup Register 2 (ITC_ILV2)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Internal Interrupt Level Setup Register 2 (ITC_ILV2)	**0x4312** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**IILV5[2:0]**	REMC interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**IILV4[2:0]**	UART interrupt level	0 to 7	0x0	R/W	

D[15:11] Reserved

D[10:8] IILV5[2:0]: Remote Controller Interrupt Level Bits

Set the remote controller interrupt level (0 to 7). (Default: 0)
Refer to the ITC_ILV0 register (0x430e) IILV1[2:0] (D[10:8]) description.

D[7:3] Reserved

D[2:0] IILV4[2:0]: UART Interrupt Level Bits

Set the UART interrupt level (0 to 7). (Default: 0)
Refer to the ITC_ILV0 register (0x430e) IILV1[2:0] (D[10:8]) description.

0x4314: Internal Interrupt Level Setup Register 3 (ITC_ILV3)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Internal Interrupt Level Setup Register 3 (ITC_ILV3)	**0x4314** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**IILV7[2:0]**	I²C interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**IILV6[2:0]**	SPI interrupt level	0 to 7	0x0	R/W	

D[15:11] Reserved

D[10:8] IILV7[2:0]: I²C Interrupt Level Bits

Set the I²C interrupt level (0 to 7). (Default: 0)
Refer to the ITC_ILV0 register (0x430e) IILV1[2:0] (D[10:8]) description.

D[7:3] Reserved

D[2:0] IILV6[2:0]: SPI Interrupt Level Bits

Set the SPI interrupt level (0 to 7). (Default: 0)
Refer to the ITC_ILV0 register (0x430e) IILV1[2:0] (D[10:8]) description.

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` command.
- The S1C17001 interrupts listed below are in level trigger mode.
  - P0 port interrupt
  - P1 port interrupt
  - Stopwatch timer interrupt
  - Clock timer interrupt
  - 8-bit OSC1 timer interrupt
  - PWM & capture timer interrupt

  Make sure all EITG*x* bits within the ITC_ELV*x* registers (0x4306 to 0x430c) have been set to 1 (level trigger mode).
  The interrupt flag within peripheral modules must be reset (to 1) within the interrupt processing routine rather than EIFT*x*. For more information on interrupt flags for resetting, refer to the peripheral module description.

This page intentionally left blank.

7 Oscillator Circuit (OSC)

7.1 OSC Module Configuration

The S1C17001 incorporates two internal oscillator circuits (OSC3 and OSC1). The OSC3 oscillator circuit generates the main clock (max. 8.2 MHz) for high-speed operation of the S1C17 core and peripheral circuits. The OSC1 oscillator circuit generates a sub-clock (typ. 32.768 kHz) 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.
Figure 7.1.1 illustrates the clock system and OSC module configuration.

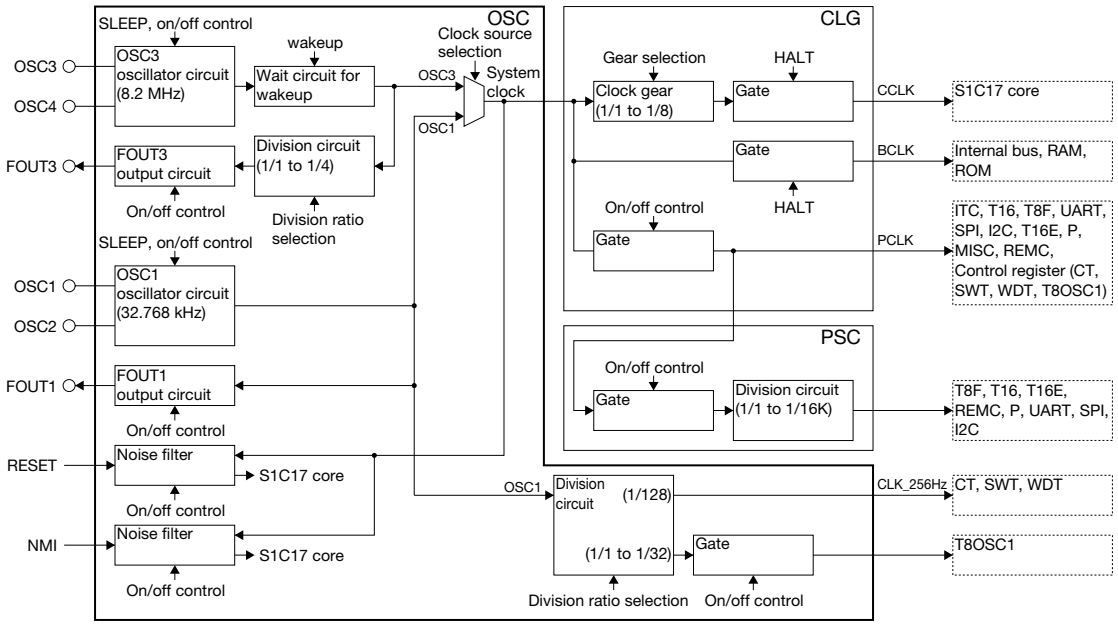
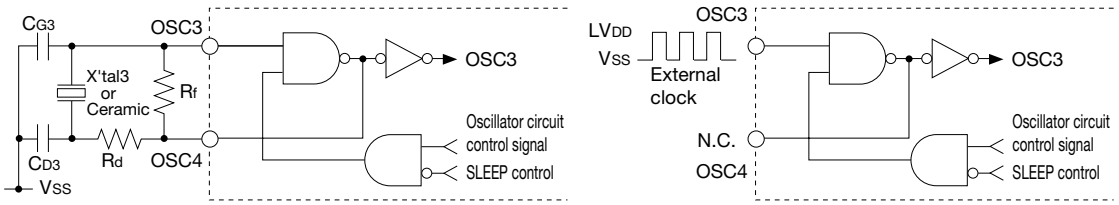


Figure 7.1.1: OSC module configuration

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

The OSC3 oscillator circuit generates the main clock (max. 8.2 MHz) for high-speed operation of the S1C17 core and peripheral circuits. The oscillator circuit can be either crystal- or ceramic-based. It also supports external clock input.



(1) Crystal/ceramic oscillator circuit (2) External clock input

Figure 7.2.1 illustrates the OSC3 oscillator circuit configuration.

When used as a crystal or ceramic 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. Two capacitors (C_{G3} and C_{D3}) should also be connected between the OSC3/OSC4 pins and V_{SS}. A drain resistor (R_d) should be connected between the OSC4 pin and C_{D3}, if required.
When used with external clock input, the OSC4 pin should be left free, and a clock with a duty ratio of 50% at LV_{DD} level should be input to the OSC3 pin.

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 initial resetting, OSC3EN is set to 1 and the OSC3 oscillator circuit is on. Since the OSC3 clock is used as the system clock, the S1C17 core begins operating using the OSC3 clock.

Stabilization wait time at start of OSC3 oscillation

The OSC3 oscillator circuit incorporates an oscillation stabilization wait timer to prevent malfunctions due to unstable clock operations at the start of OSC3 oscillation—for example, when power is first turned on, on awaking from SLEEP, or when the OSC3 oscillation circuit is turned on via software. The OSC3 clock is not fed to the system until the time set for this timer has elapsed.

Four different oscillation stabilization wait times can be selected using the OSC3WT[1:0] (D[5:4]/OSC_CTL register)

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

Table 7.2.2: 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 "24 Electrical Characteristics."

7.3 OSC1 Oscillator Circuit

The OSC1 oscillator circuit generates a (typ.) 32.768 kHz sub-clock.
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.
The oscillator circuit is crystal-based. The oscillator circuit also allows use of an external clock input.
Figure 7.3.1 illustrates the OSC1 oscillator circuit configuration.

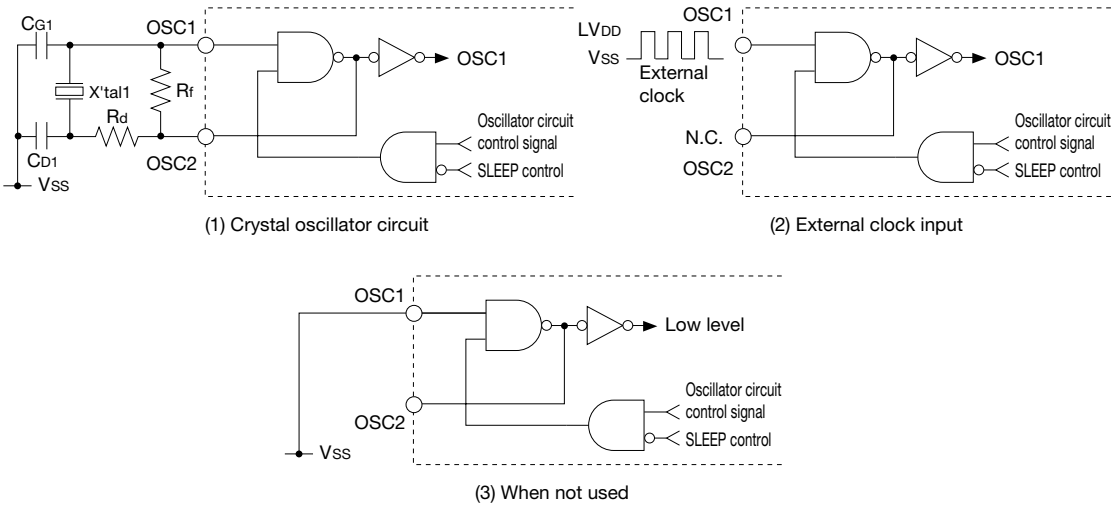


Figure 7.3.1: OSC1 oscillator circuit

When this is used as a crystal oscillator circuit, connect a crystal oscillator X'tal1 (typ. 32.768 kHz) and feedback resistor (R_f) between the OSC1 and OSC2 pins. Connect two capacitors (C_{G1} and C_{D1}) between the OSC1/OSC2 pins and V_{SS}. A drain resistor (R_d) should be connected between the OSC2 pin and C_{D1}, if required.
When used with external clock (max. 100 kHz) input, the OSC2 pin should be left free, and a clock with a duty ratio of 50% at LV_{DD} level should be input to the OSC1 pin.
If the OSC1 oscillator circuit is not used, connect the OSC1 pin to V_{SS} while leaving the OSC2 pin open.

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)

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.

7.4 System Clock Switching

The software can be used to select the OSC3 or OSC1 clocks as the system clock. If possible, you can reduce power consumption by stopping OSC3 oscillation after switching the system clock to OSC1.
The procedure is given below.

OSC3 to OSC1

1. If OSC1 oscillation is stopped, start oscillation by setting OSC1EN (D1/OSC_CTL register) to 1.
   * **OSC1EN**: OSC1 Enable Bit in the Oscillation Control (OSC_CTL) Register (D1/0x5061)
2. Set CLKSRC (D0/OSC_SRC register) to 1 and switch the system clock from OSC3 to OSC1.
   * **CLKSRC**: System Clock Source Select Bit in the Clock Source Select (OSC_SRC) Register (D0/0x5060)
3. If operation is not required for peripheral modules using OSC3 as an oscillation source, set OSC3EN (D0/OSC_CTL register) to 0 to stop OSC3 oscillation.
   * **OSC3EN**: OSC3 Enable Bit in the Oscillation Control (OSC_CTL) Register (D0/0x5061)

Note: • Switching the system clock from OSC3 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).
• OSC3 oscillation cannot be stopped before switching the system clock to OSC1.

OSC1 to OSC3

1. Set the OSC3WT[1:0] (D[5:4]/OSC_CTL register) to an oscillation stabilization wait time (see Table 7.2.2) at least as long as OSC3 oscillation start time. (Not necessary if already set.)
   * **OSC3WT[1:0]**: OSC3 Wait Cycle Select Bits in the Oscillation Control (OSC_CTL) Register (D[5:4]/0x5061)
2. If the OSC3 oscillation is stopped, set OSC3EN (D0/OSC_CTL register) to 1 to start oscillation. After starting OSC3 oscillation, the OSC3 clock is not fed until the time set in OSC3WT[1:0] (D[5:4]/OSC_CTL register) has elapsed.
3. Set CLKSRC (D0/OSC_SRC register) to 0 to switch the system clock from OSC1 to OSC3.
4. If operation is not required for peripheral modules using OSC1 as an oscillation source, set OSC1EN (D1/OSC_CTL register) to 0 to stop OSC1 oscillation.

Note: • Steps 1 and 2 are not required if the OSC3 oscillation circuit is already operating.
• 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 “24 Electrical Characteristics.”
• OSC1 oscillation cannot be stopped before switching the system clock to OSC3.

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)."

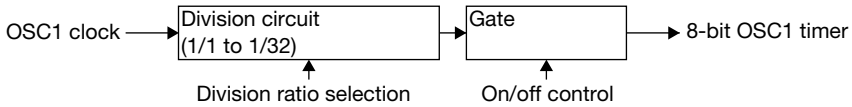


Figure 7.5.1: 8-bit OSC1 timer clock control circuit

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)

7.6 Clock External Output (FOUT3, FOUT1)

The OSC3 division clock (FOUT3) and OSC1 clock (FOUT1) can be output to devices outside the chip.

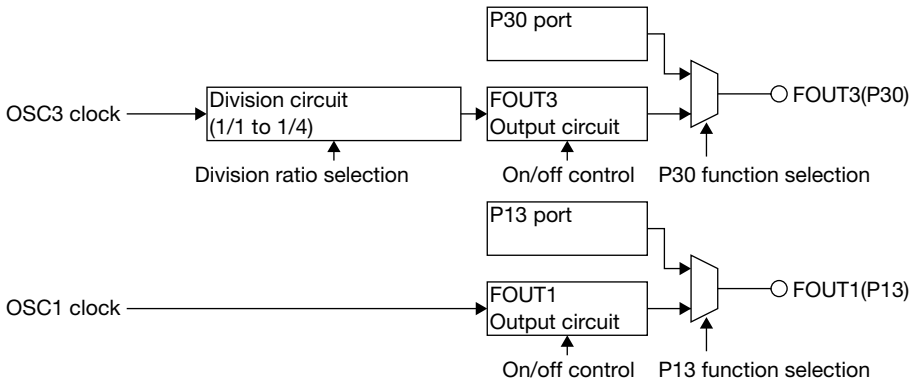


Figure 7.6.1: Clock output circuit

FOUT3 output

FOUT3 is the OSC3 division clock.

Output pin setting

The FOUT3 output pin is combined with the P30 port. This functions as the P30 port pin by default, so the pin function should be changed by writing 1 to P30MUX (D0/P3_PMUX register) if use is required for FOUT3 output.

∗ P30MUX: P30 Port Function Select Bit in the P3 Port Function Select (P3_PMUX) Register (D0/0x52a3)

FOUT3 clock frequency selection

Three different clock output frequencies can be selected. Select the division ratio for the OSC3 clock using FOUT3D[1:0] (D[3:2]/OSC_FOUT register).

∗ **FOUT3D[1:0]**: FOUT3 Clock Division Ratio Select Bits in the FOUT Control (OSC_FOUT) Register (D[3:2]/0x5064)

Table 7.6.1: FOUT3 clock division ratio selection

FOUT3D[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 FOUT3E (D1/OSC_FOUT register). Setting FOUT3E to 1 outputs the FOUT3 clock from the FOUT3 pin. Setting it to 0 halts output.

∗ **FOUT3E**: FOUT3 Output Enable Bit in the FOUT Control (OSC_FOUT) Register (D1/0x5064)

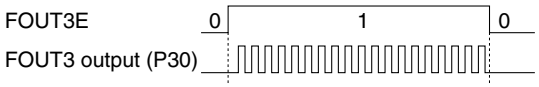


Figure 7.6.2: FOUT3 output

Note: Since the FOUT3 signal is asynchronized with FOUT3E writing, switching output on or off will generate certain hazards.

FOUT1 output

FOUT1 is the OSC1 clock.

Output pin setting

The FOUT1 output pin is combined with the P13 port. This functions as the P13 port pin by default, so the pin function should be changed by writing 1 to P13MUX (D3/P1_PMUX register) if use is required for FOUT1 output.

* **P13MUX**: P13 Port Function Select Bit in the P1 Port Function Select (P1_PMUX) Register (D3/0x52a1)

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)

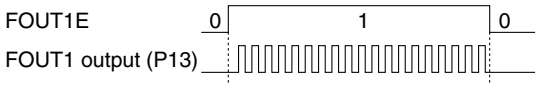


Figure 7.6.3: FOUT1 output

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)

The noise filters operate using the system clock (OSC3 or OSC1 clock) divided to 1/8. When activated, they filter out noise with pulses not exceeding two clock cycles.
This means the pulse width must be at least 16 cycles of the system clock to input as a valid signal.

Note: • All noise filters should normally be enabled.
• The S1C17001 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
0x5064	OSC_FOUT	FOUT Control Register	Clock external output control
0x5065	OSC_T8OSC1	T8OSC1 Clock Control Register	8-bit OSC1 timer clock setting

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)	D7–1	**–**	reserved	–	–	–	0 when being read.
		D0	**CLKSRC**	System clock source select	1 OSC1 0 OSC3	0	R/W	

D[7:1] Reserved

D0 CLKSRC: System Clock Source Select Bit

Selects the system clock source.
1 (R/W): OSC1
0 (R/W): OSC3 (default)

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

Note: If the system clock is switched from OSC3 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).

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	1	R/W	
		D0	OSC3EN	OSC3 enable	1 Enable 0 Disable	1	R/W	

D[7:6] Reserved

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 "24 Electrical Characteristics."

D1 OSC1EN: OSC1 Enable Bit

Permits or prohibits OSC1 oscillator circuit operation.
1 (R/W): Permitted (on) (default)
0 (R/W): Prohibited (off)

Note: • 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	1	R/W	

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. Pulses having widths of less than 16 cycles are filtered out as noise. 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) (default)
0 (R/W): Disabled (bypass)

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 S1C17001 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	FOUT3D[1:0]	FOUT3 clock division ratio select	FOUT3D[1:0] / Division ratio: 0x3 reserved; 0x2 OSC3-1/4; 0x1 OSC3-1/2; 0x0 OSC3-1/1	0x0	R/W	
		D1	FOUT3E	FOUT3 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] FOUT3D[1:0]: FOUT3 Clock Division Ratio Select Bits

Select the OSC3 clock division ratio to set the FOUT3 clock frequency.

Table 7.8.3: FOUT3 clock division ratio selection

FOUT3D[1:0]	Division ratio
0x3	Reserved
0x2	OSC3-1/4
0x1	OSC3-1/2
0x0	OSC3-1/1

(Default: 0x0)

D1 FOUT3E: FOUT3 Output Enable Bit

Permits or prohibits FOUT3 clock (OSC3 division clock) external output.
1 (R/W): Permitted (on)
0 (R/W): Prohibited (off) (default)

Setting FOUT3E to 1 outputs the FOUT3 clock from the FOUT3 pin. Setting it to 0 stops the output. The FOUT3 output pin is combined with the P30 port. This functions as the P30 port pin by default, so the pin function should be changed by writing 1 to P30MUX (D0/P3_PMUX register) if use is required for FOUT3 output.

* **P30MUX**: P30 Port Function Select Bit in the P3 Port Function Select (P3_PMUX) Register (D0/0x52a3)

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. The FOUT1 output pin is combined with the P13 port. This functions as the P13 port pin by default, so the pin function should be changed by writing 1 to P13MUX (D3/P1_PMUX register) if use is required for FOUT1 output.

* **P13MUX**: P13 Port Function Select Bit in the P1 Port Function Select (P1_PMUX) Register (D3/0x52a1)

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.

7.9 Precautions

- The OSC3 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 "24 Electrical Characteristics."
- Switching the system clock from OSC3 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 FOUT3/FOUT1 signal is asynchronized with FOUT3E/FOUT1E writing, switching output on or off will generate certain hazards.

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.

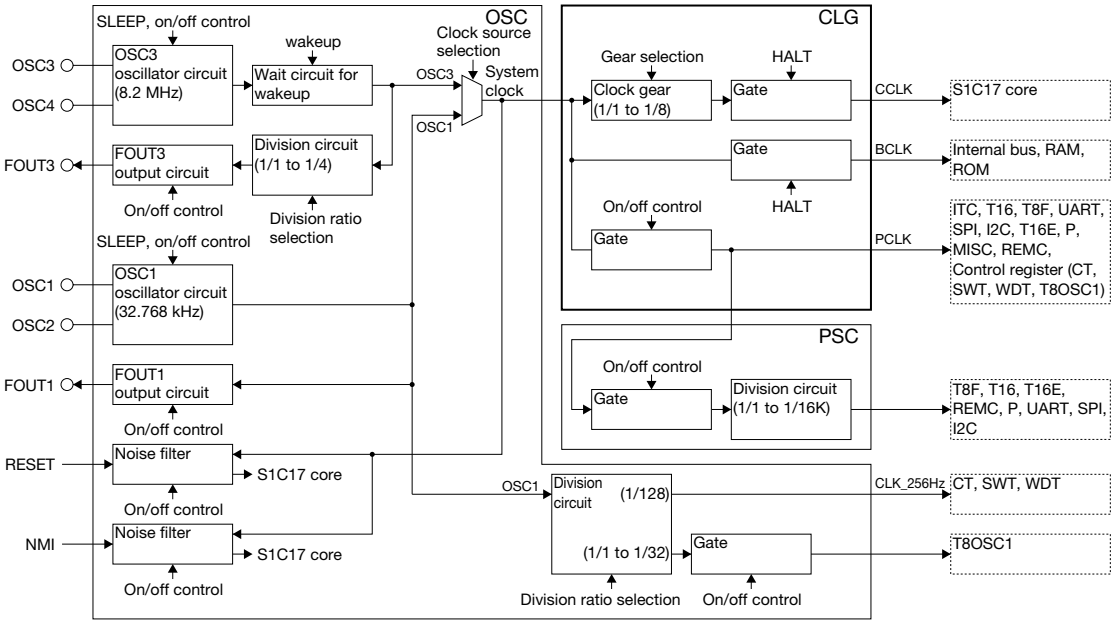


Figure 8.1.1: CLG module configuration

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 command can be executed to stop the clock feed from the CLG to the S1C17 core for power savings.

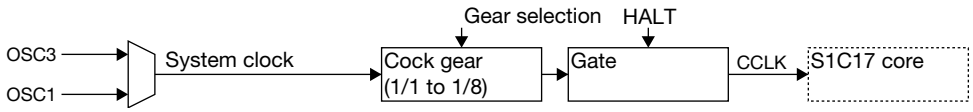


Figure 8.2.1: CCLK feed system

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 command. 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 command 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).

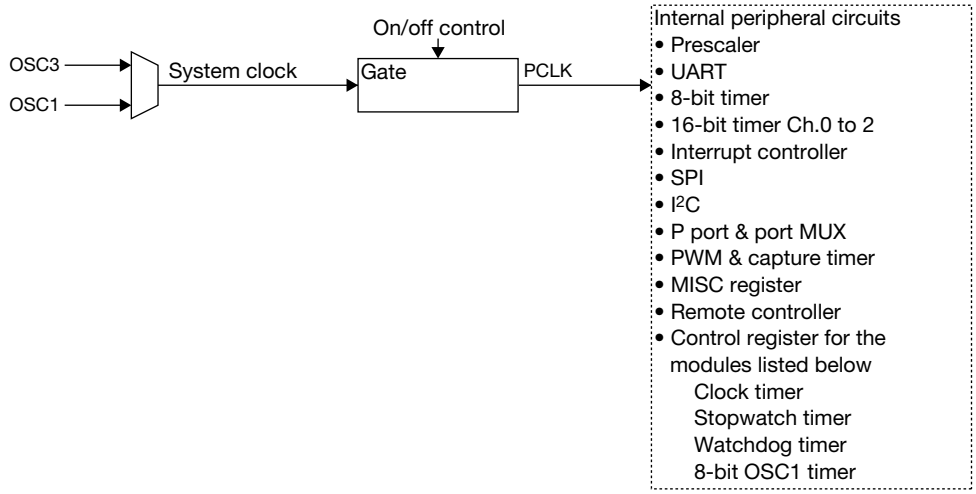


Figure 8.3.1: Peripheral module clock control circuit

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

With the exception of control register access, the clock timer, stopwatch timer, watchdog timer, and 8-bit OSC1 timer operate using the OSC1 division clock. Stopping the PCLK prevents read/write access to/from the control register, but operation will continue.

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 & capture timer, remote controller, P port)
- UART
- 8-bit timer
- 16-bit timer Ch.0 to 2
- Interrupt controller
- SPI
- I^2C
- P port & port MUX
- PWM & capture timer
- MISC register
- Remote controller

Since the following peripheral modules are not operated using PCLK except for control register access, PCLK is not required after setting the control register to start operations.
- 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 & capture timer, remote controller, P port)
- UART
- 8-bit timer
- 16-bit timer Ch.0 to 2
- Interrupt controller
- SPI
- I^2C
- P port & port MUX
- PWM & capture timer
- MISC register
- Remote controller

Since the following peripheral modules are not operated using PCLK except for control register access, PCLK is not required after setting the control register to start operations.
- 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)

This page intentionally left blank.

9. Prescaler (PSC)

9.1 Prescaler Configuration

The S1C17001 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.

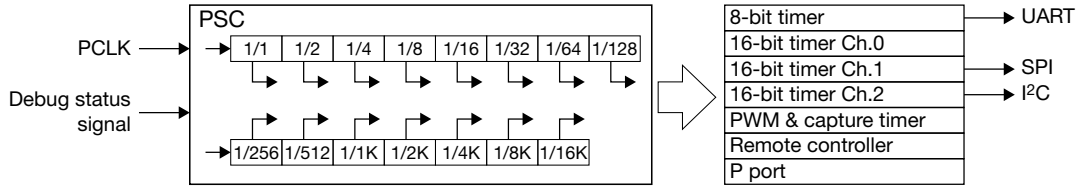


Figure 9.1.1: Prescaler

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.

This page intentionally left blank.

10 Input/Output Port (P)

10.1 Input/Output Port Configuration

The S1C17001 includes 28 input/output ports (P0[7:0], P1[7:0], P2[7:0], P3[3:0]) 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.

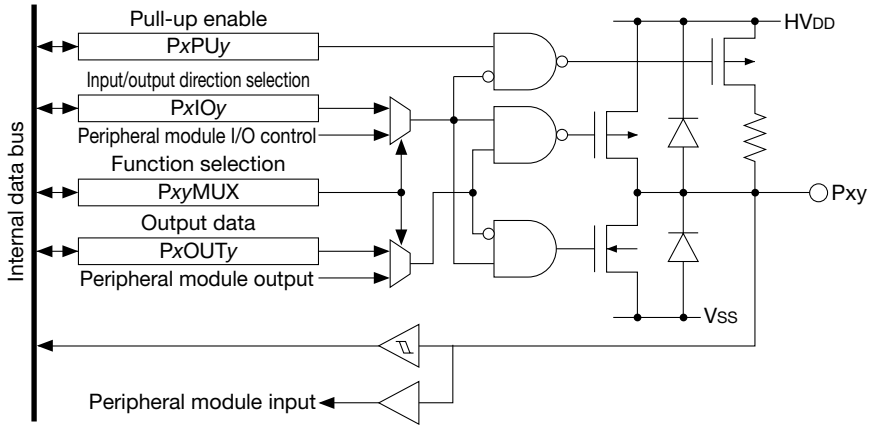


Figure 10.1.1: Input/output port configuration

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 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 = 1	Port function selection bit	Control register
P00	–	–	–
P01	–	–	–
P02	–	–	–
P03	–	–	–
P04	REMI (REMC)	P04MUX (D4)	P0 Port Function Select (P0_PMUX) Register (0x52a0)
P05	REMO (REMC)	P05MUX (D5)	
P06/EXCL2 (T16CH2)	–	–	–
P07/EXCL1 (T16CH1)	–	–	–
P10	–	–	–
P11	–	–	–
P12	–	–	–
P13	FOUT1 (OSC)	P13MUX (D3)	P1 Port Function Select (P1_PMUX) Register (0x52a1)
P14	SDA (I2C)	P14MUX (D4)	
P15	SCL (I2C)	P15MUX (D5)	
P16/EXCL0 (T16CH0)	–	–	–
P17	#SPISS (SPI)	P17MUX (D7)	P1 Port Function Select (P1_PMUX) Register (0x52a1)
P20	SDI (SPI)	P20MUX (D0)	P2 Port Function Select (P2_PMUX) Register (0x52a2)
P21	SDO (SPI)	P21MUX (D1)	
P22	SPICLK (SPI)	P22MUX (D2)	
P23	SIN (UART)	P23MUX (D3)	
P24	SOUT (UART)	P24MUX (D4)	
P25	SCLK (UART)	P25MUX (D5)	
P26	TOUT (T16E)	P26MUX (D6)	
P27	EXCL3 (T16E)	P27MUX (D7)	
P30	FOUT3 (OSC)	P30MUX (D0)	P3 Port Function Select (P3_PMUX) Register (0x52a3)
DCLK (DBG)	P31	P31MUX (D1)	
DST2 (DBG)	P32	P32MUX (D2)	
DSIO (DBG)	P33	P33MUX (D3)	

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

Pins P06, P07, and P16 can also be used as 16-bit timer external clock input pins by setting them to input mode. Note, however, that no port function selection bits are available, since they are simultaneously set to function as general purpose input ports.

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*IO[7:0] (P*x*_IO register).

- **P0IO[7:0]**: P0[7:0] Port I/O Direction Select Bits in the P0 Port I/O Direction Control (P0_IO) Register (D[7:0]/0x5202)
- **P1IO[7:0]**: P1[7:0] Port I/O Direction Select Bits in the P1 Port I/O Direction Control (P1_IO) Register (D[7:0]/0x5212)
- **P2IO[7:0]**: P2[7:0] Port I/O Direction Select Bits in the P2 Port I/O Direction Control (P2_IO) Register (D[7:0]/0x5222)
- **P3IO[3:0]**: P3[3:0] Port I/O Direction Select Bits in the P3 Port I/O Direction Control (P3_IO) Register (D[3:0]/0x5232)

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 (HV_{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[3:0]**: P3[3:0] Port Input Data Bits in the P3 Port Input Data (P3_IN) Register (D[3:0]/0x5230)

Data output

When set to output mode, P*x*IO[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 (HV_{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[3:0]**: P3[3:0] Port Output Data Bits in the P3 Port Output Data (P3_OUT) Register (D[3:0]/0x5231)

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[3:0]**: P3[3:0] Port Pull-up Enable Bits in the P3 Port Pull-up Control (P3_PU) Register (D[3:0]/0x5233)

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. An appropriate wait time must be set for the input/output port loading. The wait time set should be a value not less than that calculated from the following equation.

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

10.5 Input Interface Level

The S1C17001 input interface level is pegged to the CMOS mute level.

10.6 P0 Port Chattering Filter Function

The P0 port includes 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] ports using P0CF*x*[2:0] (P0_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)

Table 10.6.1: Chattering filter function settings

P0CF*x*[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 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 command.
- P0 port interrupts must be blocked when P0_CHAT register (0x5208) settings are being changed. Changing the setting while interrupts are permitted may generate inadvertent P0 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.

10.7 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 10.7.1 illustrates the port input interrupt circuit configuration.

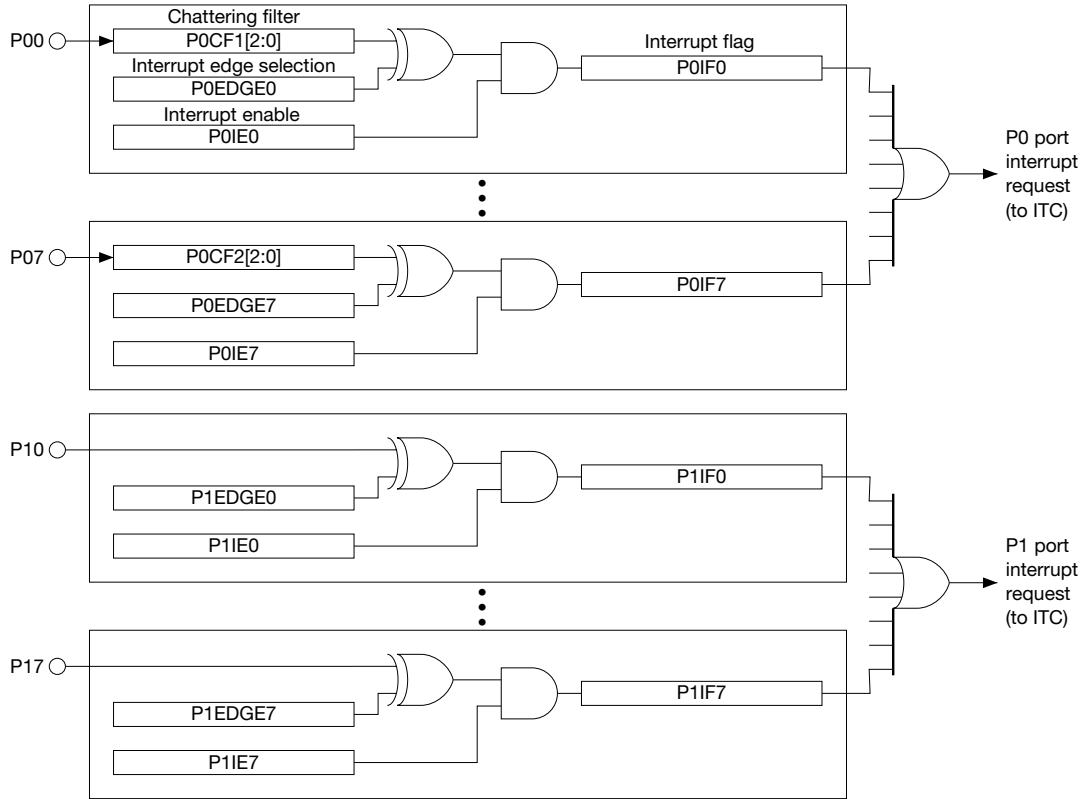


Figure 10.7.1: Port input interrupt circuit configuration

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.
The interrupt controller must also be set to actually generate an interrupt. For more information on making interrupt controller settings, refer to "6. Interrupt Controller (ITC)."

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 in P port module

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. Setting the corresponding P*x*IE[7:0] to 1 sets P*x*IF[7:0] 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. 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.

- **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)

The following processing is needed to manage the interrupt factor occurrence state using the P port module interrupt flags.

1. Set the ITC P0 and P1 interrupt trigger mode to level trigger mode.
2. Reset the P port module interrupt flag P*x*IF[7:0] within the interrupt processing routine after the interrupt occurs (this also resets the ITC interrupt flag).

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

Note: 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).

Port interrupt ITC register

A P0 or P1 port interrupt signal is output to the ITC if the port for which interrupts are permitted as previously set detects the specified edge of an input signal.
The interrupt level and interrupt permission should be set for the ITC register in order to generate a port interrupt.
Table 10.7.1 illustrates the port interrupt ITC control bits.

Table 10.7.1: ITC control bits

Port	Interrupt flag	Interrupt enable	Interrupt level setting	Trigger mode setting
P0	EIFT0 (D0/ITC_IFLG)	EIEN0 (D0/ITC_EN)	EILV0[2:0] (D[2:0]/ITC_ELV0)	EITG0 (D4/ITC_ELV0)
P1	EIFT1 (D1/ITC_IFLG)	EIEN1 (D1/ITC_EN)	EILV1[2:0] (D[10:8]/ITC_ELV0)	EITG1 (D12/ITC_ELV0)

ITC_IFLG register (0x4300)
ITC_EN register (0x4302)
ITC_ELV0 register (0x4306)

The relevant ITC interrupt flag is set to 1 when the P0 or P1 port interrupt signal is activated. When the interrupt enable bit corresponding to that interrupt flag is set to 1, the ITC sends an interrupt request to the S1C17 core. To block port interrupts, set the interrupt enable bit to 0. The interrupt flag is set to 1 by the P0 or P1 port interrupt signal regardless of the interrupt enable bit setting (even if set to 0).
The interrupt level setting bit sets the port interrupt level (0 to 7). The P0 port takes precedence if the same interrupt level is set.
As previously mentioned, the port interrupt trigger mode setting bit must always be set to 1 (level trigger).

The S1C17 core accepts interrupts when all of the following conditions are met:
- Interrupt enable bit is set to 1
- The PSR (S1C17 core internal processor status register) IE (interrupt enable) bit is set to 1.
- The port interrupt has a higher set interrupt level than the PSR IL (interrupt level).
- No other interrupt factors having higher precedence (e.g., NMI) are present.

For more information on these interrupt control registers and procedures for when an interrupt occurs, refer to "6 Interrupt Controller (ITC)."

Interrupt vector

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

Table 10.7.2: Port interrupt vectors

Port	Vector number	Vector address
P0	4 (0x04)	0x8010
P1	5 (0x05)	0x8014

10.8 Control Register Details

Table 10.8.1: 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_IO	P0 Port I/O Direction Control Register	P0 port input/output direction selection
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
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_IO	P1 Port I/O Direction Control Register	P1 port input/output direction selection
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
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_IO	P2 Port I/O Direction Control Register	P2 port input/output direction selection
0x5223	P2_PU	P2 Port Pull-up Control Register	P2 port pull-up control
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_IO	P3 Port I/O Direction Control Register	P3 port input/output direction selection
0x5233	P3_PU	P3 Port Pull-up Control Register	P3 port pull-up control
0x52a0	P0_PMUX	P0 Port Function Select Register	P0 port function selection
0x52a1	P1_PMUX	P1 Port Function Select Register	P1 port function selection
0x52a2	P2_PMUX	P2 Port Function Select Register	P2 port function selection
0x52a3	P3_PMUX	P3 Port Function Select Register	P3 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: 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	
P3 Port Input Data Register (P3_IN)	**0x5230** (8 bits)	D7–4	**–**	reserved	–	–	–	0 when being read.
		D3–0	**P3IN[3:0]**	P3[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 3).

D[7:0] P*x*IN[7:0]: P*x*[7:0] Port Input Data Bits (P3 port is P3IN[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.

0x5201/0x5211/0x5221/0x5231: 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)	D7–4	**–**	reserved	–				–	–	0 when being read.
		D3–0	**P3OUT[3:0]**	P3[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 3).

D[7:0] P*x*OUT[7:0]: P*x*[7:0] Port Output Data Bits (P3 port is P3OUT[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.

0x5202/0x5212/0x5222/0x5232: P*x* Port I/O Direction Control Registers (P*x*_IO)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
P0 Port I/O Direction Control Register (P0_IO)	**0x5202** (8 bits)	D7–0	**P0IO[7:0]**	P0[7:0] port I/O direction select	1	Output	0	Input	0	R/W	
P1 Port I/O Direction Control Register (P1_IO)	**0x5212** (8 bits)	D7–0	**P1IO[7:0]**	P1[7:0] port I/O direction select	1	Output	0	Input	0	R/W	
P2 Port I/O Direction Control Register (P2_IO)	**0x5222** (8 bits)	D7–0	**P2IO[7:0]**	P2[7:0] port I/O direction select	1	Output	0	Input	0	R/W	
P3 Port I/O Direction Control Register (P3_IO)	**0x5232** (8 bits)	D7–4	**–**	reserved	–				–	–	0 when being read.
		D3–0	**P3IO[3:0]**	P3[3:0] port I/O direction select	1	Output	0	Input	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 I/O Direction Select Bits (P3 port is P3IN[3:0])

Set the input/output mode for the input/output port.

1(R/W): Output mode

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

P*x*IO[7:0] are the input/output direction selection bits corresponding directly to the P*x*[7:0] ports. Setting to 1 selects output mode, while setting to 0 selects input mode. The peripheral module function determines the input/output direction for when a pin is used for peripheral modules.

0x5203/0x5213/0x5223/0x5233: 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)	D7–4	–	reserved	–				–	–	0 when being read.
		D3–0	P3PU[3:0]	P3[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 3).

D[7:0] P*x*PU[7:0]: P*x*[7:0] Port Pull-up Enable Bits (P3 port is P3PU[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. An appropriate wait time must be set for the input/output port loading. The wait time set should be a value not less than that calculated from the following equation.

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

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.

To enable interrupt generation, the ITC P0 and P1 port interrupt enable bits must also be set to permit interrupts.

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]. Setting the corresponding P*x*IE[7:0] (P*x*_IMSK register) to 1 sets P*x*IF[7:0] 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. 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.

The following processing is needed to manage the interrupt factor occurrence state using the P*x*IF[7:0].

1. Set the ITC P0 and P1 interrupt trigger mode to level trigger mode.
2. Reset the P port module interrupt flag P*x*IF[7:0] within the interrupt processing routine after the interrupt occurs (this also resets the ITC interrupt flag).

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

Note: 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: P0 Port Chattering Filter Control Register (P0_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	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	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			

D7 **Reserved**

D[6:4] **P0CF2[2:0]: P0[7:4] Chattering Filter Time Select Bits**

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

D3 **Reserved**

D[2:0] **P0CF1[2:0]: P0[3:0] Chattering Filter Time Select Bits**

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

The P0 port includes 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] ports using P0CF*x*[2:0].

Table 10.8.2: Chattering filter function settings

P0CF*x*[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 command.
- P0 port interrupts must be blocked when P0_CHAT register settings are being changed. Changing the setting while interrupts are permitted may generate inadvertent P0 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 10.8.3: 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.

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)	D7–6	–	reserved	–	–	–	0 when being read.
		D5	P05MUX	P05 port function select	1 REMO 0 P05	0	R/W	
		D4	P04MUX	P04 port function select	1 REMI 0 P04	0	R/W	
		D3–0	–	reserved	–	–	–	0 when being read.

The P04 and P05 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] Reserved

D5 P05MUX: P05 Port Function Select Bit
1 (R/W): REMO (REMC)
0 (R/W): P05 port (default)

D4 P04MUX: P04 Port Function Select Bit
1 (R/W): REMI (R EMC)
0 (R/W): P04 port (default)

D[3:0] Reserved

0x52a1: 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)	**0x52a1** (8 bits)	D7	**P17MUX**	P17 port function select	1 #SPISS 0 P17	0	R/W	
		D6	**–**	reserved	–	–	–	0 when being read.
		D5	**P15MUX**	P15 port function select	1 SCL 0 P15	0	R/W	
		D4	**P14MUX**	P14 port function select	1 SDA 0 P14	0	R/W	
		D3	**P13MUX**	P13 port function select	1 FOUT1 0 P13	0	R/W	
		D2–0	**–**	reserved	–	–	–	0 when being read.

The P13 to P15 and P17 input/output port pins are shared with the peripheral module pins. This register is used to select how the pins are used.

D7 P17MUX: P17 Port Function Select Bit
1 (R/W): #SPISS (SPI)
0 (R/W): P17 port (default)

D6 Reserved

D5 P15MUX: P15 Port Function Select Bit
1 (R/W): SCL (I2C)
0 (R/W): P15 port (default)

D4 P14MUX: P14 Port Function Select Bit
1 (R/W): SDA (I2C)
0 (R/W): P14 port (default)

D3 P13MUX: P13 Port Function Select Bit
1 (R/W): FOUT1 (OSC)
0 (R/W): P13 port (default)

D[2:0] Reserved

0x52a2: 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)	**0x52a2** (8 bits)	D7	**P27MUX**	P27 port function select	1	EXCL3	0	P27	0	R/W	
		D6	**P26MUX**	P26 port function select	1	TOUT	0	P26	0	R/W	
		D5	**P25MUX**	P25 port function select	1	SCLK	0	P25	0	R/W	
		D4	**P24MUX**	P24 port function select	1	SOUT	0	P24	0	R/W	
		D3	**P23MUX**	P23 port function select	1	SIN	0	P23	0	R/W	
		D2	**P22MUX**	P22 port function select	1	SPICLK	0	P22	0	R/W	
		D1	**P21MUX**	P21 port function select	1	SDO	0	P21	0	R/W	
		D0	**P20MUX**	P20 port function select	1	SDI	0	P20	0	R/W	

The P20 to P27 input/output port pins are shared with the peripheral module pins. This register is used to select how the pins are used.

D7 P27MUX: P27 Port Function Select Bit
1 (R/W): EXCL3 (T16E)
0 (R/W): P27 port (default)

D6 P26MUX: P26 Port Function Select Bit
1 (R/W): TOUT (T16E)
0 (R/W): P26 port (default)

D5 P25MUX: P25 Port Function Select Bit
1 (R/W): SCLK (UART)
0 (R/W): P25 port (default)

D4 P24MUX: P24 Port Function Select Bit
1 (R/W): SOUT (UART)
0 (R/W): P24 port (default)

D3 P23MUX: P23 Port Function Select Bit
1 (R/W): SIN (UART)
0 (R/W): P23 port (default)

D2 P22MUX: P22 Port Function Select Bit
1 (R/W): SPICLK (SPI)
0 (R/W): P22 port (default)

D1 P21MUX: P21 Port Function Select Bit
1 (R/W): SDO (SPI)
0 (R/W): P21 port (default)

D0 P20MUX: P20 Port Function Select Bit
1 (R/W): SDI (SPI)
0 (R/W): P20 port (default)

0x52a3: 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)	0x52a3 (8 bits)	D7–4	–	reserved	–				–	–	0 when being read.
		D3	P33MUX	P33 port function select	1	P33	0	DSIO	0	R/W	
		D2	P32MUX	P32 port function select	1	P32	0	DST2	0	R/W	
		D1	P31MUX	P31 port function select	1	P31	0	DCLK	0	R/W	
		D0	P30MUX	P30 port function select	1	FOUT3	0	P30	0	R/W	

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[7:4] Reserved

D3 P33MUX: P33 Port Function Select Bit
1 (R/W): P33 port
0 (R/W): DSIO (DBG) (default)

D2 P32MUX: P32 Port Function Select Bit
1 (R/W): P32 port
0 (R/W): DST2 (DBG) (default)

D1 P31MUX: P31 Port Function Select Bit
1 (R/W): P31 port
0 (R/W): DCLK (DBG) (default)

D0 P30MUX: P30 Port Function Select Bit
1 (R/W): FOUT3 (OSC)
0 (R/W): P30 port (default)

10.9 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 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. An appropriate wait time must be set for the input/output port loading. The wait time set should be a value not less than that calculated from the following equation.
  Wait time = R_{IN} x (C_{IN} + load capacitance on board) x 1.6 [s]
  R_{IN}: pull-up resistance maximum value
  C_{IN}: pin capacitance maximum value
- Input/output ports that are not used should be set with pull-up resistance enabled.

P0 and P1 port 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).
- Set the ITC P0 and P1 port interrupt trigger mode to level trigger mode.
  Reset the P port module interrupt flag P0IF[7:0] (0x5207) and P1IF[7:0] (0x5217) within the interrupt processing routine after the interrupt occurs. This also resets the ITC interrupt flag.

P0 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 command.
- P0 port interrupts must be blocked when P0_CHAT register (0x5208) settings are being changed. Changing the setting while interrupts are permitted may generate inadvertent P0 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 S1C17001 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.

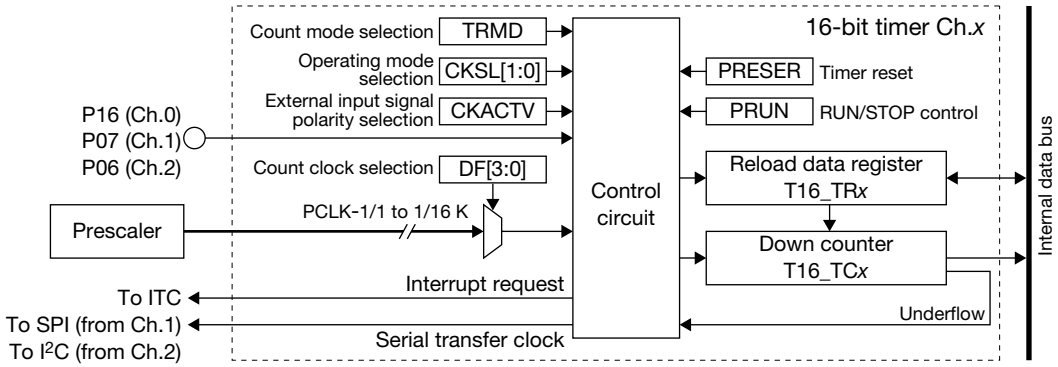


Figure 11.1.1: 16-bit timer configuration (1-channel)

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	P16
Ch.1	EXCL1	P07
Ch.2	EXCL2	P06

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 P07 and P06 ports used by 16-bit timer 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 "10.6 P0 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.

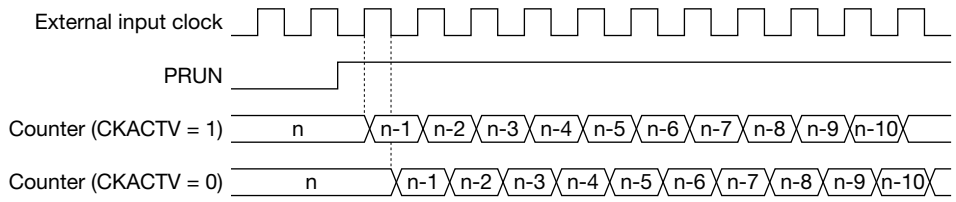


Figure 11.2.2.1: External clock mode count

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 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.

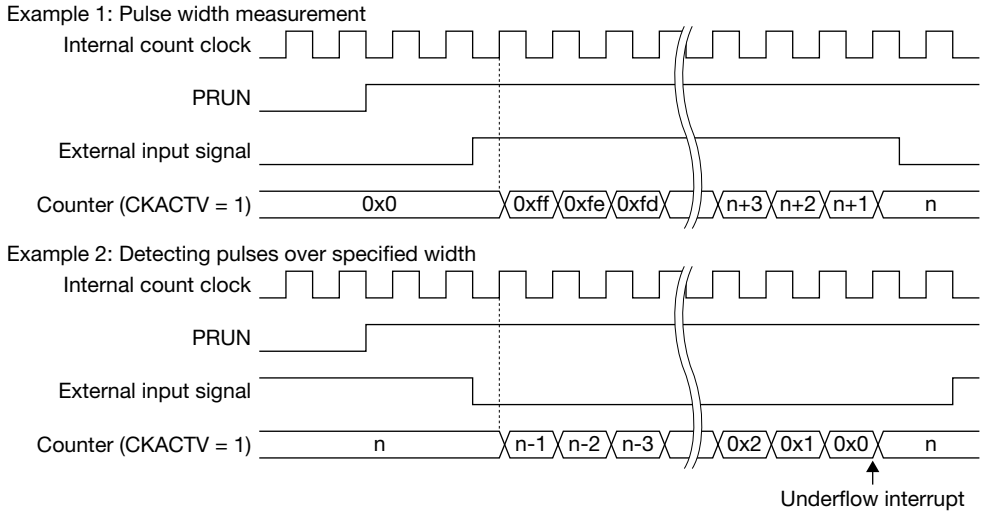


Figure 11.2.3.1: Pulse width measurement mode count operation

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.

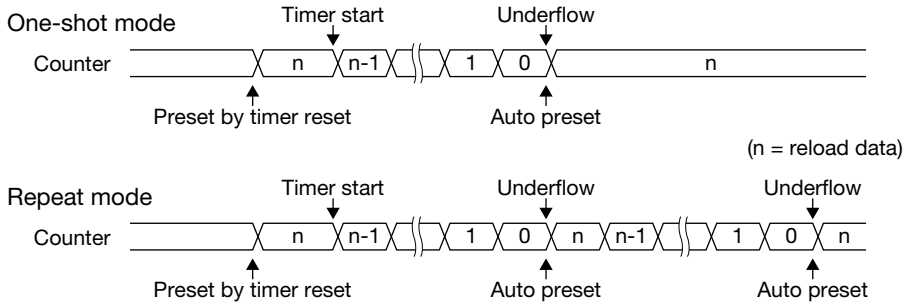


Figure 11.4.1: Preset timing

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.

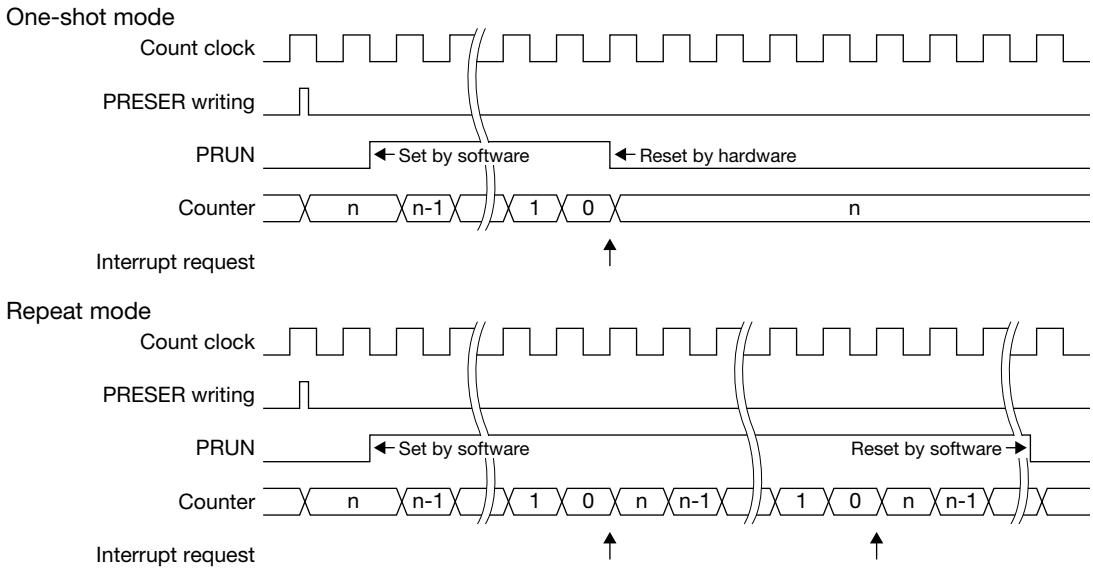


Figure 11.6.1: Count operation

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.

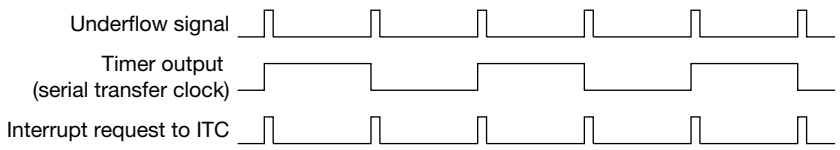


Figure 11.7.1: Timer output clock

The clock generated is sent to the internal serial interface, as shown below.
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 $TR = \frac{clk\_in}{bps \times 2} - 1$

I²C $TR = \frac{clk\_in}{bps \times 4} - 1$

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.
To generate a timer underflow interrupt, the interrupt level and interrupt permission should be set using the ITC registers.

Timer interrupt ITC registers

Table 11.8.1 lists the ITC control registers for each timer channel.

Table 11.8.1: ITC registers

Timer channel	Interrupt flag	Interrupt enable bit	Interrupt level setting bit
Ch.0	IIFT1 (D9/ITC_IFLG)	IIEN1 (D9/ITC_EN)	IILV1[2:0] (D[10:8]/ITC_ILV0)
Ch.1	IIFT2 (D10/ITC_IFLG)	IIEN2 (D10/ITC_EN)	IILV2[2:0] (D[2:0]/ITC_ILV1)
Ch.2	IIFT3 (D11/ITC_IFLG)	IIEN3 (D11/ITC_EN)	IILV3[2:0] (D[10:8]/ITC_ILV1)

ITC_IFLG register (0x4300)
ITC_EN register (0x4302)
ITC_ILV0 register (0x430e)
ITC_ILV1 register (0x4310)

If an underflow occurs in the timer, the corresponding interrupt flag is set to 1.
If the interrupt enable bit corresponding to that interrupt flag is set to 1, the ITC sends an interrupt request to the S1C17 core. To prohibit timer interrupts, set the interrupt enable bit to 0 beforehand. The interrupt flag will be set to 1 by the timer underflow pulse regardless of the interrupt enable bit setting (i.e., even if set to 0).
The interrupt level setting bit sets the timer interrupt level (0 to 7). If set to the same interrupt level, the 16-bit timer Ch.0 takes the highest priority, while the 16-bit timer Ch.2 takes the lowest priority.
The S1C17 core accepts interrupts when all of the following conditions are satisfied:

- The interrupt enable bit has been set to 1.
- The PSR (S1C17 core internal processor status register) IE (interrupt enable) bit has been set to 1.
- The timer interrupt has a higher interrupt level set than that set for the PSR IL (interrupt level).
- No other interrupt factors having higher precedence (e.g., NMI) are present.

For more information on these interrupt control registers and operations when interrupts occur, see "6 Interrupt Controller (ITC)."

Interrupt vectors

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

Table 11.8.2: Timer interrupt vectors

Timer channel	Vector number	Vector address
Timer Ch.0	13 (0x0d)	0x8034
Timer Ch.1	14 (0x0e)	0x8038
Timer Ch.2	15 (0x0f)	0x803c

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
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
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

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 Ch.*x* Input Clock Select Register (T16_CLK*x*)	0x4220 0x4240 0x4260 (16 bits)	D15–4	–	reserved	–		–	–	0 when being read.
		D3–0	DF[3:0]	Timer input clock select (Prescaler output clock)	DF[3:0]	Clock	0x0	R/W	
					0xf	reserved			
					0xe	PCLK-1/16384			
					0xd	PCLK-1/8192			
					0xc	PCLK-1/4096			
					0xb	PCLK-1/2048			
					0xa	PCLK-1/1024			
					0x9	PCLK-1/512			
					0x8	PCLK-1/256			
					0x7	PCLK-1/128			
					0x6	PCLK-1/64			
					0x5	PCLK-1/32			
					0x4	PCLK-1/16			
					0x3	PCLK-1/8			
					0x2	PCLK-1/4			
					0x1	PCLK-1/2			
					0x0	PCLK-1/1			

Note: The "*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	0x0	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: 0x0)
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: P16, Ch.1: P07, Ch.2: P06) 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.

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.

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.

This page intentionally left blank.

12 8-bit Timer (T8F)

12.1 8-bit Timer Overview

The S1C17001 incorporates an 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.

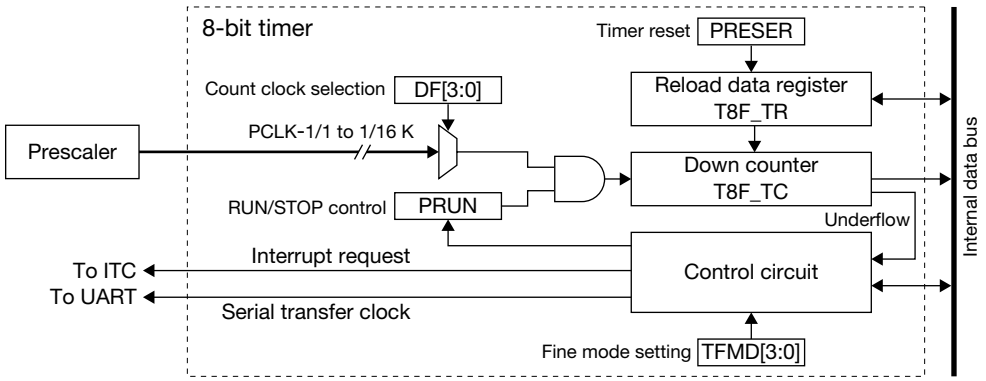


Figure 12.1.1: 8-bit timer configuration

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 Control (T8F_CTL) Register (D4/0x4206)**

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 Input Clock Select (T8F_CLK) Register (D[3:0]/0x4200)**

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) is used to set the initial value for the down counter.あ
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.

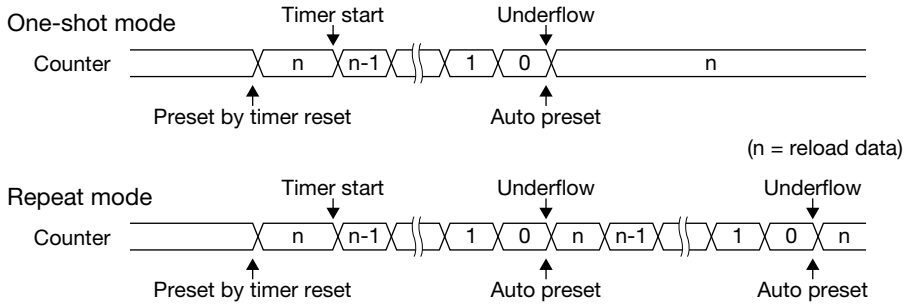


Figure 12.4.1: Preset timing

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 clo ck) 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 register). The reload data is preset and the counter is initialized.

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

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 register).

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

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.

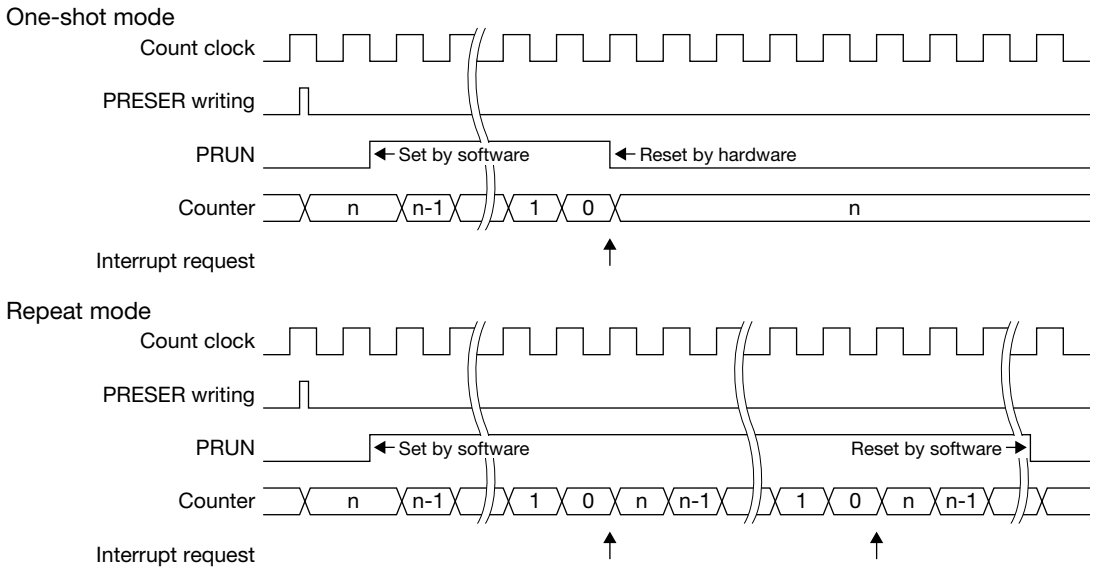


Figure 12.6.1: Count operation

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.

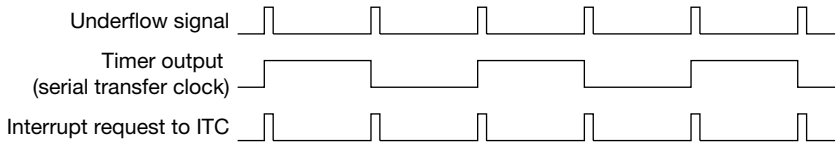


Figure 12.7.1: Timer output clock

The underflow pulses are also used to generate the serial transfer clock and are transmitted to the UART.
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 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 Control (T8F_CTL) Register (D[11:8]/0x4206)**

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.

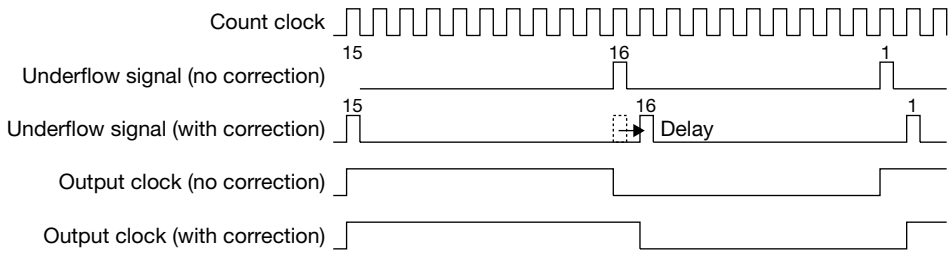


Figure 12.8.1: Delay cycle insertion in Fine mode

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.
To generate a timer underflow interrupt, the interrupt level and interrupt permission should be set using the ITC registers.

Timer interrupt ITC register

The 8-bit timer ITC control bits are listed below.

Interrupt flag IIFT0

* **IIFT0**: 8-bit Timer Interrupt Flag in the Interrupt Flag (ITC_IFLG) Register (D8/0x4300)

Interrupt enable bit IIEN0

* **IIEN0**: 8-bit Timer Interrupt Enable Bit in the Interrupt Enable (ITC_EN) Register (D8/0x4302)

Interrupt level setting bit IILV0

* **IILV0[2:0]**: 8-bit Timer Interrupt Level Bits in the Internal Interrupt Level Setup (ITC_ILV0) Register 0 (D[2:0]/0x430e)

If an underflow occurs in the timer, the corresponding interrupt flag is set to 1.
If the interrupt enable bit corresponding to that interrupt flag is set to 1, the ITC sends an interrupt request to the S1C17 core. To prohibit timer interrupts, set the interrupt enable bit to 0 beforehand. The interrupt flag will be set to 1 by the timer underflow pulse regardless of the interrupt enable bit setting (i.e., even if set to 0).
The interrupt level setting bit sets the timer interrupt level (0 to 7).
The S1C17 core accepts interrupts when all of the following conditions are satisfied:

- The interrupt enable bit has been set to 1.
- The PSR (S1C17 core internal processor status register) IE (interrupt enable) bit has been set to 1.
- The timer interrupt has a higher interrupt level set than that set for the PSR IL (interrupt level).
- No other interrupt factors having higher precedence (e.g., NMI) are present.

For more information on these interrupt control registers and operations when interrupts occur, see "6 Interrupt Controller (ITC)."

Interrupt vectors

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

Vector number: 12 (0x0c)
Vector address: 0x8030

12.10 Control Register Details

Table 12.10.1: 8-bit timer register list

Address	Register name		Function
0x4200	T8F_CLK	8-bit Timer Input Clock Select Register	Prescaler output clock selection
0x4202	T8F_TR	8-bit Timer Reload Data Register	Reload data setting
0x4204	T8F_TC	8-bit Timer Counter Data Register	Counter data
0x4206	T8F_CTL	8-bit Timer Control Register	Timer mode setting and timer RUN/STOP

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: 8-bit Timer Input Clock Select Register (T8F_CLK)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
8-bit Timer Input Clock Select Register (T8F_CLK)	0x4200 (16 bits)	D15–4	–	reserved	–		–	–	0 when being read.
		D3–0	DF[3:0]	Timer input clock select (Prescaler output clock)	DF[3:0]	Clock	0x0	R/W	
					0xf	reserved			
					0xe	PCLK-1/16384			
					0xd	PCLK-1/8192			
					0xc	PCLK-1/4096			
					0xb	PCLK-1/2048			
					0xa	PCLK-1/1024			
					0x9	PCLK-1/512			
					0x8	PCLK-1/256			
					0x7	PCLK-1/128			
					0x6	PCLK-1/64			
					0x5	PCLK-1/32			
					0x4	PCLK-1/16			
					0x3	PCLK-1/8			
					0x2	PCLK-1/4			
					0x1	PCLK-1/2			
					0x0	PCLK-1/1			

D[15:4] Reserved

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

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: 8-bit Timer Reload Data Register (T8F_TR)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
8-bit Timer Reload Data Register (T8F_TR)	**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	

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: 8-bit Timer Counter Data Register (T8F_TC)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
8-bit Timer Counter Data Register (T8F_TC)	**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	0x0	R	

D[15:8] Reserved

D[7:0] TC[7:0]: 8-bit Timer Counter Data

Reads out the counter data. (Default: 0x0)
This register is read-only and cannot be written to.

0x4206: 8-bit Timer Control Register (T8F_CTL)

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

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.

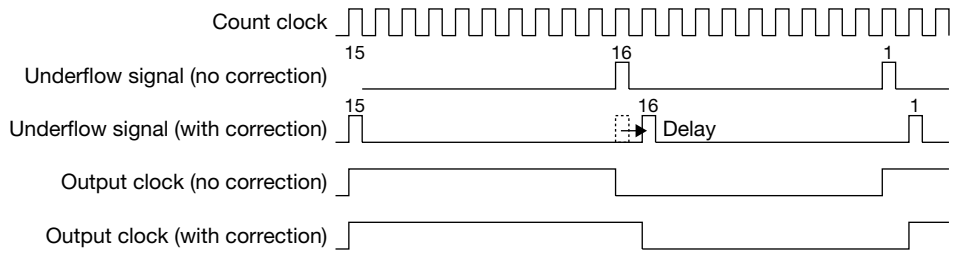


Figure 12.10.1: Delay cycle insertion in Fine mode

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.

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.

13 PWM & Capture Timer (T16E)

13.1 PWM & Capture Timer Overview

The S1C17001 incorporates a single-channel PWM & capture timer.
Figure 13.1.1 illustrates the PWM & capture timer configuration.

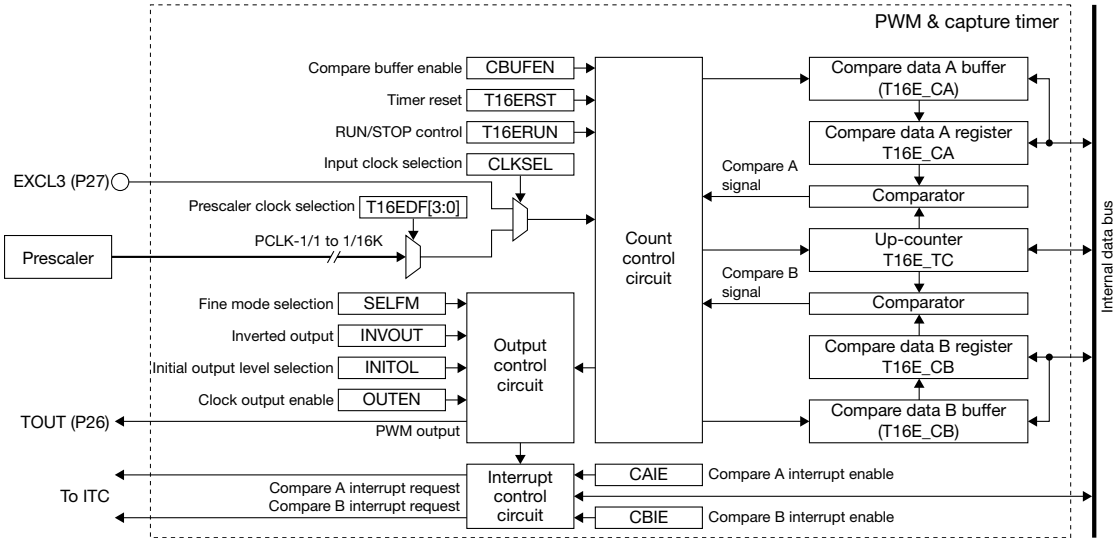


Figure 13.1.1: PWM & capture timer configuration

The PWM & capture 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.

The 16-bit counter can be reset to 0 or set to a counter value by software and counts up for external signals from the prescaler output clock or P27 port pin. The count value can be read by software.

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 & Capture Timer Operating Modes

The PWM & capture 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 & capture timer in internal clock mode.
• Make sure the PWM & capture timer count is halted before changing count clock settings.

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

External clock mode

External clock mode uses the clock and pulses input via the P27 (EXCL3) 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.
To input the EXCL3 clock from the P27 port, P27MUX (D7/P2_PMUX register) must be written as 1 and the P27 pin function must be changed.

∗ **P27MUX**: P27 Port Function Select Bit in the P2 Port Function Select (P2_PMUX) Register

The PWM & capture timer increments counts based on the input signal rising edge.

The PWM & capture 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 & capture 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 & capture 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 (D5/0x5306) 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 (TOUT 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 & Capture Timer RUN/STOP Control

Set the following before starting the PWM & capture 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 & capture 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 & capture 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.

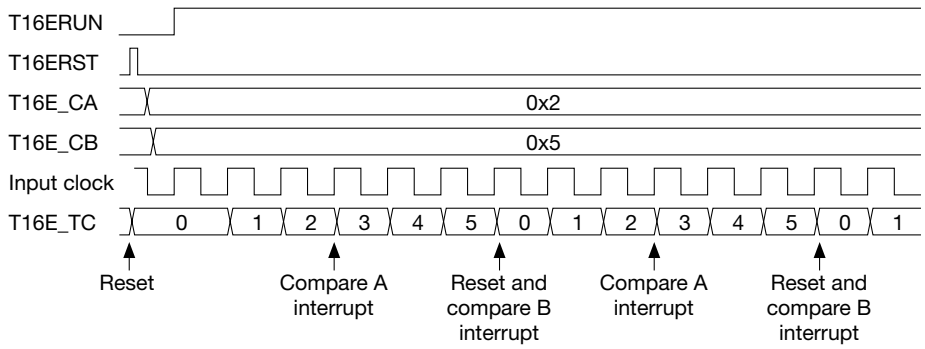


Figure 13.5.1: Basic counter operation timing

13.6 Clock Output Control

The PWM & capture timer can generate a TOUT signal using the compare match signal.
Figure 13.6.1 shows the PWM & capture timer clock output circuit.

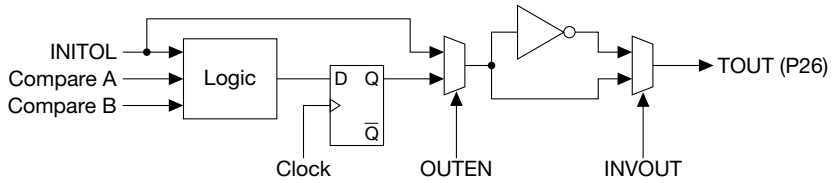


Figure 13.6.1: PWM & capture timer clock output circuit

Initial output level settings

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

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

The initial output level is Low when INITOL is 0 (default). Setting to 1 switches the initial output level to High. The timer output will switch to the initial output level set here even when the timer is reset by writing 1 to T16ERST.

Output signal polarity selection

By default, an active High (normal Low) output signal is generated. 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) signal.

∗ **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.
For detailed information on the output waveform, refer to Figure 13.6.2.

Output pin setting

The TOUT signal generated here can be output from the TOUT (P26) pin. This signal can provide a programmable clock and PWM signal to external devices.
The P26 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 TOUT 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 TOUT 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.

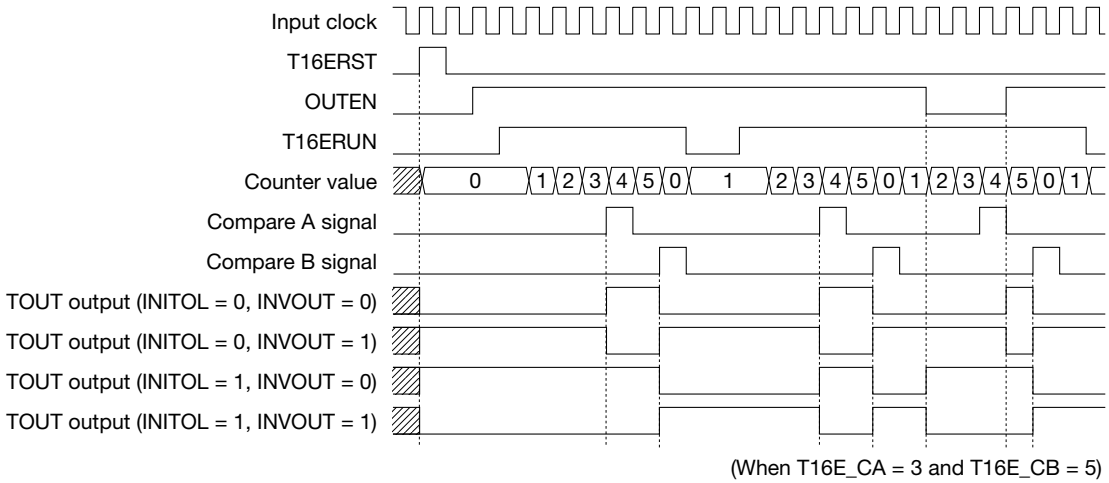


Figure 13.6.2: PWM & capture timer output waveform

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.

When INVOUT = 1 (Active Low)

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.

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.

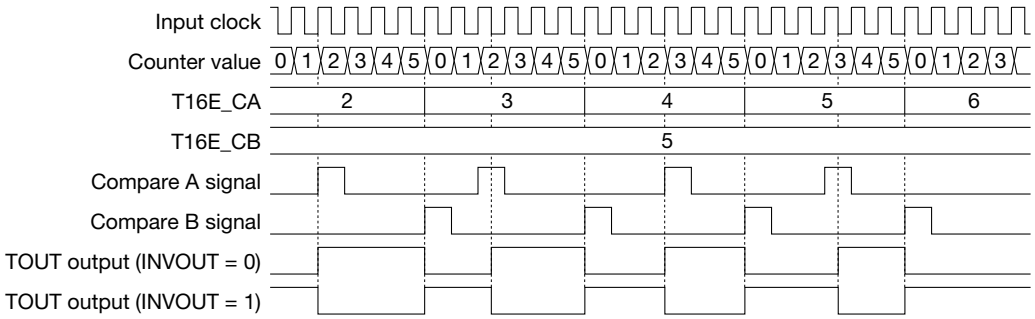


Figure 13.6.3: Fine mode clock output

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 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).

13.7 PWM & Capture 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_IFLG 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. This interrupt request signal sets the PWM & capture timer interrupt flag inside the ITC to 1 and generates an interrupt if the ITC and S1C17 core interrupt conditions are satisfied.
The CAIF should be read and checked in the PWM & capture timer interrupt processing routine to determine whether the PWM & capture timer interrupt is due to compare A matching.
The interrupt factor should be cleared with the interrupt processing routine by resetting the T16E module CAIF (to 1) rather than the ITC PWM & capture timer interrupt flag.

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_IFLG 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_IMSK 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 CBIF is set to 1, the T16E module outputs an interrupt request to the ITC. This interrupt request signal sets the PWM & capture timer interrupt flag inside the ITC to 1 and generates an interrupt if the ITC and S1C17 core interrupt conditions are satisfied.
The CBIF should be read and checked in the PWM & capture timer interrupt processing routine to determine whether the PWM & capture timer interrupt is due to compare B matching.
The interrupt factor should be cleared with the interrupt processing routine by resetting the T16E module CBIF (to 1) rather than the ITC PWM & capture timer interrupt flag.

Note: To prevent generating unnecessary interrupts, reset the corresponding CAIF or CBIF before permitting compare A or compare B interrupts from CAIE or CBIE.

PWM & capture timer interrupt ITC register

The T16E module outputs an interrupt signal to the ITC if a compare match with interrupt permitted is generated by the settings previously described. To generate PWM & capture timer interrupts, the interrupt level and interrupt permission should be set in the ITC register.
The PWM & capture timer ITC control bits are shown below.

Interrupt flag inside ITC

- **EIFT7**: PWM & Capture Timer Interrupt Flag in the Interrupt Flag (ITC_IFLG) Register (D7/0x4300)

Interrupt enable bit inside ITC

- **EIEN7**: PWM & Capture Timer Interrupt Enable Bit in the Interrupt Enable (ITC_EN) Register (D7/0x4302)

Interrupt level setting bit inside ITC

- **EILV7[2:0]**: T16E Interrupt Level Bits in the External Interrupt Level Setup (ITC_ELV3) Register 3 (D[10:8]/0x430c)

Interrupt trigger mode selection bit inside ITC (Fix at 1)

- **EITG7**: T16E Interrupt Trigger Mode Select Bit in the External Interrupt Level Setup (ITC_ELV3) Register 3 (D12/0x430c)

EIFT7 is set to 1 when a compare match is generated with interrupt permitted in the T16E module. If EIEN7 is set to 1 here, the ITC sends an interrupt request to the S1C17 core. To prevent PWM & capture timer interrupts, set the EIEN7 to 0. EIFT7 is set to 1 by the interrupt signal from the T16E module regardless of the EIEN7 setting (even if it is set to 0).
EILV7[2:0] sets the PWM & capture timer interrupt level (0 to 7).

The S1C17 core accepts interrupts when the following conditions are satisfied:

- The interrupt enable bit has been set to 1.
- The PSR (S1C17 core internal processor status register) IE (interrupt enable) bit has been set to 1.
- The PWM & capture timer interrupt has been set to a higher interrupt level than that set for the PSR IL (interrupt level).
- No other interrupt factors having higher precedence (e.g., NMI) are present.

For detailed information on these interrupt control registers and operations when interrupts occur, refer to "6 Interrupt Controller (ITC)."

Note: The following processes must be performed to manage the interrupt factor occurrence state using the T16E module interrupt flag.

1. Set the ITC PWM & capture timer interrupt trigger mode to level trigger mode.
2. Reset the T16E module interrupt flags CAIF and CBIF within the interrupt processing routine after the interrupt occurs (this also resets the ITC interrupt flag).

Interrupt vectors

The PWM & capture timer interrupt vector numbers and vector addresses are listed below.

Vector number: 11 (0x0b)
Vector address: 0x802c

13.8 Control Register Details

Table 13.8.1: PWM & capture timer register list

Address	Register name		Function
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 mask setting
0x530c	T16E_IFLG	PWM Timer Interrupt Flag Register	Interrupt occurrence status display/resetting

The PWM & capture 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 & capture 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 & capture 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): High
0(R/W): Low (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) or when the timer is reset by writing 1 to T16ERST (D1). 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 (active Low)
0 (R/W): Normal (active High) (default)

Writing 1 to INVOUT generates a TOUT 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).

D3 CLKSEL: Input Clock Select Bit

Selects the timer input clock.
1 (R/W): External clock
0 (R/W): Internal clock (default)

If CLKSEL is written as 0, the internal clock (prescaler output) is selected as the timer input clock. If written as 1, the external clock (clock input from EXCL3 (P27) pin) is selected and functions as an event counter.

Note: To input the EXCL3 clock from the P27 port, P27MUX (D7/P2_PMUX register) must be written as 1, and the P27 pin function must be changed beforehand.

∗ **P27MUX**: P27 Port Function Select Bit in the P2 Port Function Select (P2_PMUX) Register

D2 OUTEN: Clock Output Enable Bit

Controls the TOUT signal (timer output clock) output.
1 (R/W): Permitted
0 (R/W): Prohibited (default)

Writing 1 to OUTEN outputs the TOUT signal from the TOUT (P26) output pin. Writing 0 to OUTEN halts output and switches to Off level in accordance with the INVOUT(D4) and INITOL(D8) settings. P26 must be set to the TOUT pin using the P26 port function selection register before outputting the TOUT signal.

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 & capture 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 & capture 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 & capture 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 & capture timer count is halted before changing count clock settings.

0x530a: PWM Timer Interrupt Mask Register (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. The ITC PWM & capture timer interrupt enable bits must also be set to permit interrupts in order to generate 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. The ITC PWM & capture timer interrupt enable bits must also be set to permit interrupts in order to generate interrupts.

0x530c: PWM Timer Interrupt Flag Register (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) to 1 sets this to 1 when the counter matches the compare data B register setting during counting. A PWM & capture timer interrupt request signal is output to the ITC at the same time. This interrupt request signal sets the ITC PWM & capture timer interrupt flag to 1 and generates an interrupt if the ITC and S1C17 core interrupt conditions are satisfied.

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) to 1 sets this to 1 when the counter matches the compare data A register setting during counting. A PWM & capture timer interrupt request signal is output to the ITC at the same time. This interrupt request signal sets the ITC PWM & capture timer interrupt flag to 1 and generates an interrupt if the ITC and S1C17 core interrupt conditions are satisfied.

The following processes must be performed to manage the interrupt factor occurrence state using this register.

1. Set the ITC PWM & capture timer interrupt trigger mode to level trigger mode.
2. Reset the T16E module interrupt flags CAIF and CBIF within the interrupt processing routine after the interrupt occurs (this also resets the ITC interrupt flag).

CAIF and CBIF are reset by writing as 1.

Note: To prevent generating unnecessary interrupts, reset the corresponding CAIF or CBIF before permitting compare A or compare B interrupts from CAIE (D0/T16E_IMSK) or CBIE (D1/T16E_IMSK).

13.9 Precautions

- The prescaler must run before operating the PWM & capture timer.
- Make sure the PWM & capture 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 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 S1C17001 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.

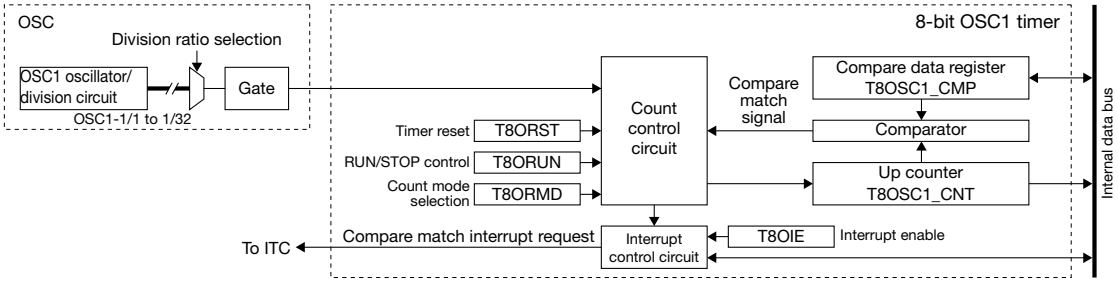


Figure 14.1.1: 8-bit OSC1 timer configuration

The 8-bit OSC1 timer includes an 8-bit up-counter (T8OSC1_CNT register) and 8-bit compare data register (T8OSC1_CMP 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 register contains the data for comparisons against the up counter contents. The comparator outputs a signal to control interrupts if the counter value matches the contents of the compare data register. This means the compare data register can be used to program the interrupt occurrence cycle.

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. Since the interrupt signsl is output at the same time, the 8-bit OSC1 timer should be set to this mode to generate periodic interrupts at desired intervals.

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.

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

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.

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.

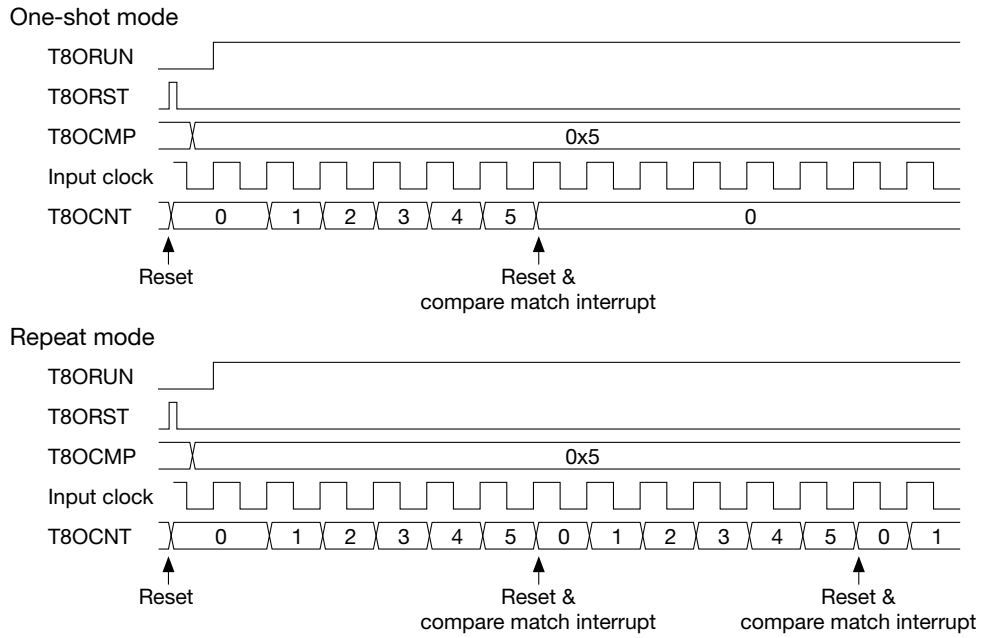


Figure 14.6.1: Basic counter operation timing

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 sets the 8-bit OSC1 timer interrupt flag inside the ITC to 1 and generates an interrupt if the ITC and S1C17 core interrupt conditions are satisfied.
The interrupt factor should be cleared with the interrupt processing routine by resetting the T8OSC1 module T8OIF (to 1) rather than the 8-bit OSC1 timer interrupt flag.

Note: To prevent generating unnecessary interrupts, reset the corresponding T8OIF before permitting compare 8-bit OSC1 interrupts from T8OIE.

8-bit OSC1 timer interrupt ITC register

The 8-bit OSC timer outputs an interrupt signal to the ITC is generated by the settings previously described. To generate 8-bit OSC timer interrupts, the interrupt level and interrupt permission should be set in the ITC register.
The 8-bit OSC timer ITC control bits are shown below.

Interrupt flag inside ITC

* **EIFT4**: 8-bit OSC1 Timer Interrupt Flag in the Interrupt Flag (ITC_IFLG) Register (D4/0x4300)

Interrupt enable bit inside ITC

* **EIEN4**: 8-bit OSC1 Timer Interrupt Enable Bit in the Interrupt Enable (ITC_EN) Register (D4/0x4302)

Interrupt level setting bit inside ITC

* **EILV4[2:0]**: T8OSC1 Interrupt Level Bits in the External Interrupt Level Setup (ITC_ELV2) Register 2 (D[2:0]/0x430a)

Interrupt trigger mode selection bit inside ITC (Fix at 1)

* **EITG4**: T8OSC1 Interrupt Trigger Mode Select Bit in the External Interrupt Level Setup (ITC_ELV2) Register 2 (D4/0x430a)

EIFT4 is set to 1 when a compare match is generated with interrupt permitted in the T8OSC1 module. If EIEN4 is set to 1 here, the ITC sends an interrupt request to the S1C17 core. To prevent 8-bit OSC timer interrupts, set the EIEN4 to 0. EIFT4 is set to 1 by the interrupt signal from the T8OSC1 module regardless of the EIEN4 setting (even if it is set to 0).
EILV4[2:0] sets the 8-bit OSC1 timer interrupt level (0 to 7).

The S1C17 core accepts interrupts when the following conditions are satisfied:

- The interrupt enable bit has been set to 1.
- The PSR (S1C17 core internal processor status register) IE (interrupt enable) bit has been set to 1.
- The 8-bit OSC timer interrupt has been set to a higher interrupt level than that set for the PSR IL (interrupt level).
- No other interrupt factors having higher precedence (e.g., NMI) are present.

For detailed information on these interrupt control registers and operations when interrupts occur, refer to "6 Interrupt Controller (ITC)."

Note: The following processes must be performed to manage the interrupt factor occurrence state using the T8OSC1 module interrupt flag.

1. Set the 8-bit OSC timer interrupt trigger mode to level trigger mode.
2. Reset the 8-bit OSC module interrupt flags T8OIF within the interrupt processing routine after the interrupt occurs (this also resets the ITC interrupt flag).

Interrupt vectors

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

Vector number: 8 (0x08)
Vector address: 0x8020

14.8 Control Register Details

Table 14.8.1: 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

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.

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	0xff	R	

D[7:0] T8OCNT[7:0]: Counter Data

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

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 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	

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.
The ITC 8-bit OSC1 timer interrupt enable bits must also be set to permit interrupts in order to generate 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 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.

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 sets the ITC 8-bit OSC1 timer interrupt flag to 1 and generates an interrupt if the ITC and S1C17 core interrupt conditions are met.
The following processes must be performed to manage the interrupt factor occurrence state using this register.

1. Set the ITC 8-bit OSC1 timer interrupt trigger mode to level trigger mode.
2. Reset the T8OSC1 module interrupt flag T8OIF within the interrupt processing routine after the interrupt occurs (this also resets the ITC interrupt flag).

T8OIF is reset by writing as 1.

Note: To prevent generating unnecessary interrupts, reset T8OIF before permitting compare match interrupts using T8OIE (D0/T8OSC1_IMSK register).

14.9 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 generating unnecessary interrupts, reset T8OIF (D0/T8OSC1_IFLG register) before permitting compare match interrupts using T8OIE (D0/T8OSC1_IMSK register).

This page intentionally left blank.

15 Clock Timer (CT)

15.1 Clock Timer Overview

The S1C17001 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.

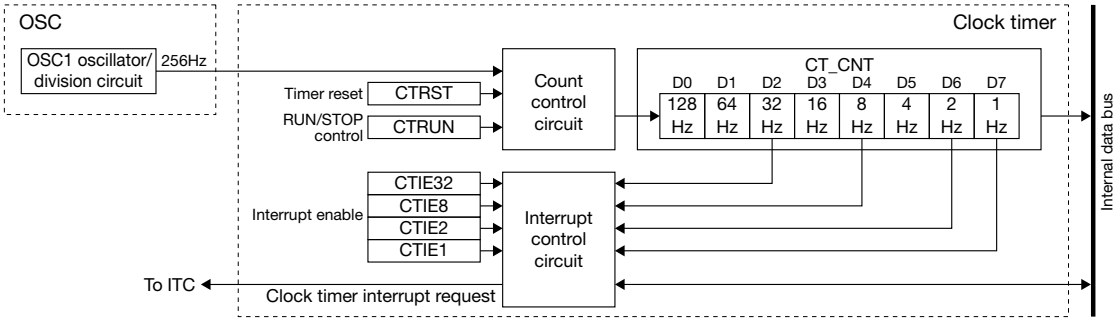


Figure 15.1.1: Clock timer configuration

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).

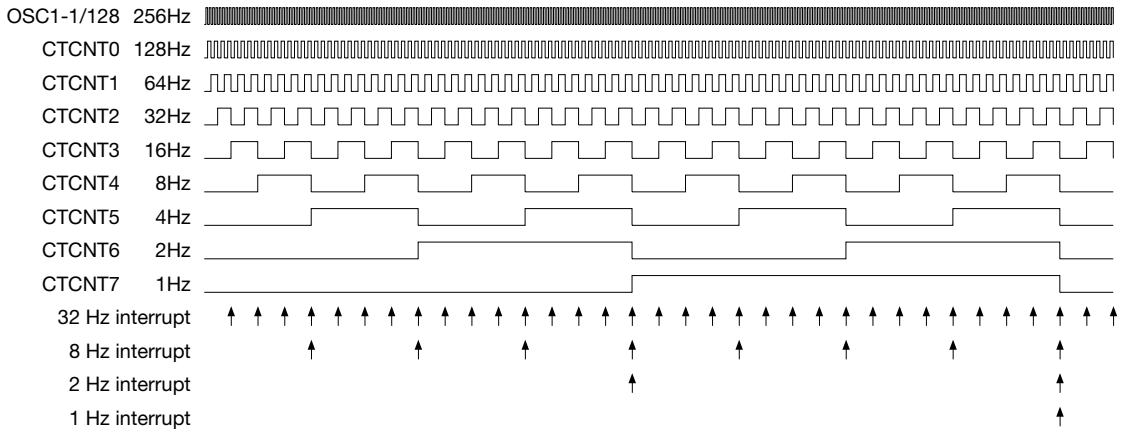


Figure 15.4.1: Clock timer timing chart

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.

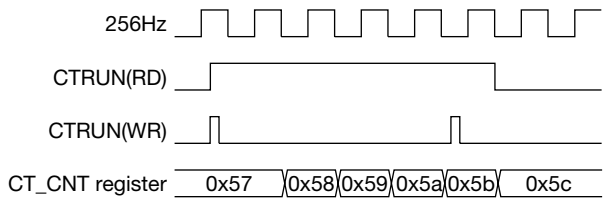


Figure 15.4.2: Run/Stop control timing chart

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.
The interrupt factor should be cleared with the interrupt processing routine by resetting the CT module CTIF* (to 1) rather than the clock timer interrupt flag.

Note: To prevent generating unnecessary interrupts, reset the corresponding CTIF* before permitting clock timer interrupts from CTIE*.

Clock timer interrupt ITC register

The clock timer outputs an interrupt signal to the ITC using the falling edge for the frequency for which interrupts are permitted in the settings previously described. To generate clock timer interrupts, the interrupt level and interrupt permission should be set in the ITC register.
The clock timer ITC control bits are shown below.

Interrupt flag inside ITC

- **EIFT3**: Clock Timer Interrupt Flag in the Interrupt Flag (ITC_IFLG) Register (D3/0x4300)

Interrupt enable bit inside ITC

- **EIEN3**: Clock Timer Interrupt Enable Bit in the Interrupt Enable (ITC_EN) Register (D3/0x4302)

Interrupt level setting bit inside ITC

- **EILV3[2:0]**: CT Interrupt Level Bits in the External Interrupt Level Setup (ITC_ELV1) Register 1 (D[10:8]/0x4308)

Interrupt trigger mode selection bit inside ITC (Fix at 1)

- **EITG3**: CT Interrupt Trigger Mode Select Bit in the External Interrupt Level Setup (ITC_ELV1) Register 1 (D12/0x4308)

EIFT3 is set to 1 at the falling edge of the 32/8/2/1 Hz signals for which interrupts are permitted by the CT module. If EIEN3 is set to 1 here, the ITC sends an interrupt request to the S1C17 core. To prevent clock timer interrupts, set the EIEN3 to 0. EIFT3 is set to 1 by the interrupt signal from the CT module regardless of the EIEN3 setting (even if it is set to 0).
EILV3[2:0] sets the clock timer interrupt level (0 to 7).
The S1C17 core accepts interrupts when the following conditions are satisfied:

- The interrupt enable bit has been set to 1.
- The PSR (S1C17 core internal processor status register) IE (interrupt enable) bit has been set to 1.
- The clock timer interrupt has been set to a higher interrupt level than that set for the PSR IL (interrupt level).
- No other interrupt factors having higher precedence (e.g., NMI) are present.

For detailed information on these interrupt control registers and operations when interrupts occur, refer to "6 Interrupt Controller (ITC)."

Note: The following processes must be performed to manage the interrupt factor occurrence state using the CT module interrupt flag.

1. Set the clock timer interrupt trigger mode to level trigger mode.
2. Reset the CT module interrupt flags CTIF* within the interrupt processing routine after the interrupt occurs (this also resets the ITC interrupt flag).

Interrupt vectors

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

Vector number: 7 (0x07)
Vector address: 0x801c

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

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 sets the clock timer interrupt flag in the ITC to 1 and generates an interrupt if the ITC and S1C17 core interrupt conditions are met.
The following processes must be performed to manage the interrupt factor occurrence state using this register.

1. Set the ITC clock timer interrupt trigger mode to level trigger mode.
2. Reset the CT module interrupt flag within the interrupt processing routine after the interrupt occurs (this also resets the ITC interrupt flag).

CTIF* is reset by writing as 1.

Note: To prevent generating unnecessary interrupts, CTIF* must be reset before permitting clock timer interrupts using CTIE.*

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.

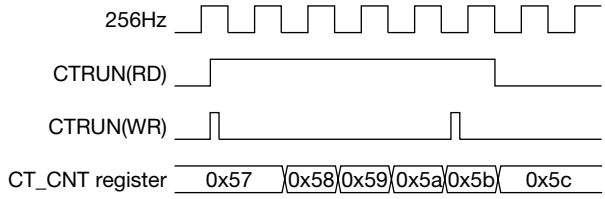


Figure 15.7.1: Run/Stop control timing chart

- Executing the slp command 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 command.

16 Stopwatch Timer (SWT)

16.1 Stopwatch Timer Overview

The S1C17001 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.

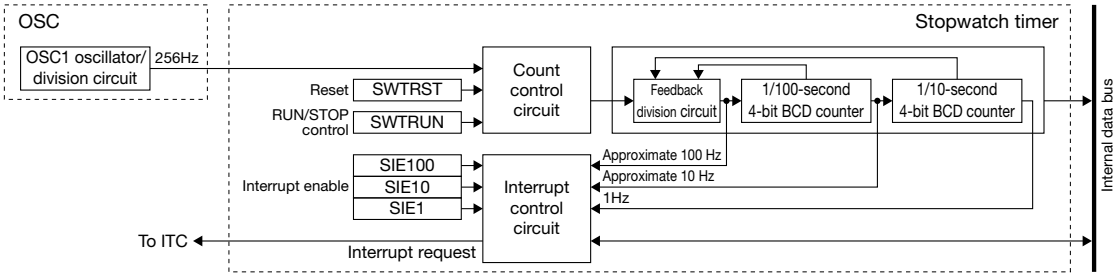


Figure 16.1.1: Stopwatch timer configuration

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.

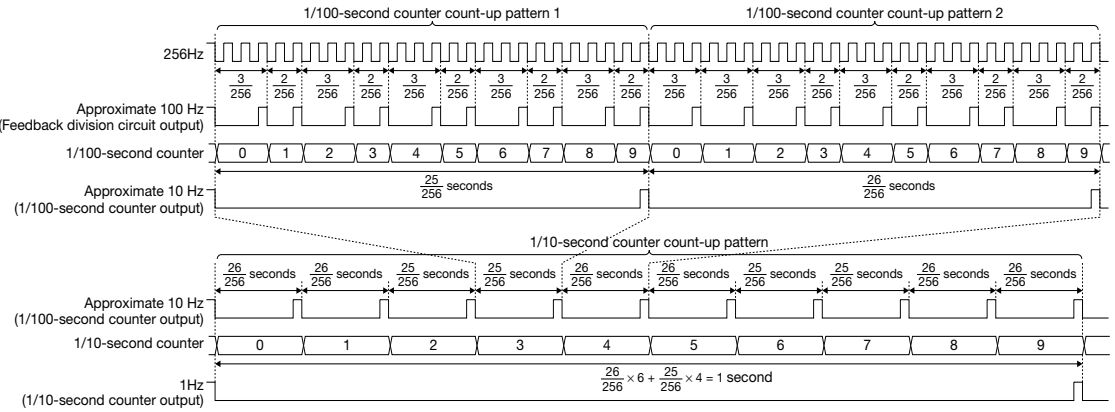


Figure 16.2.1: Stopwatch timer count-up patterns

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).

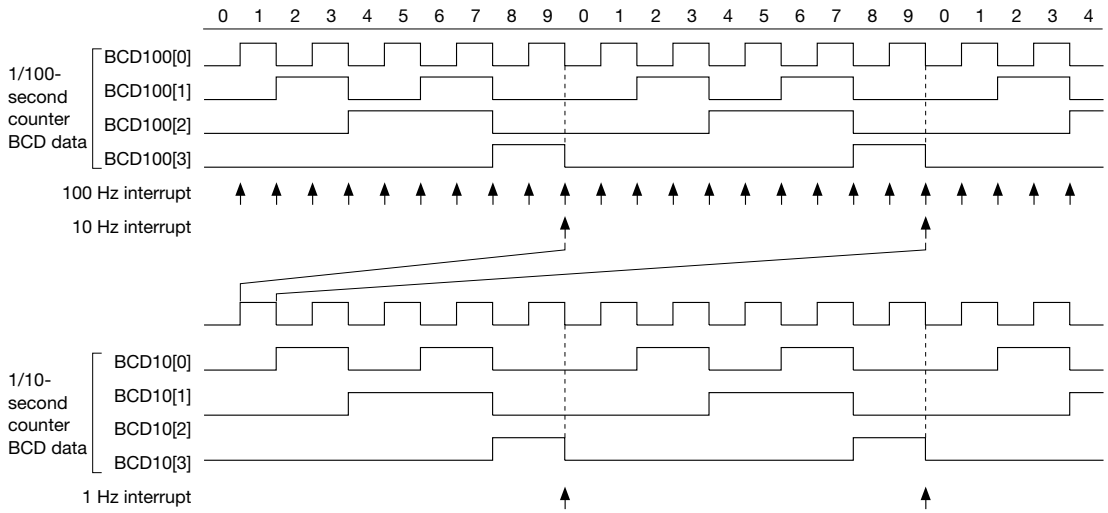


Figure 16.5.1: Stopwatch timer timing chart

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.

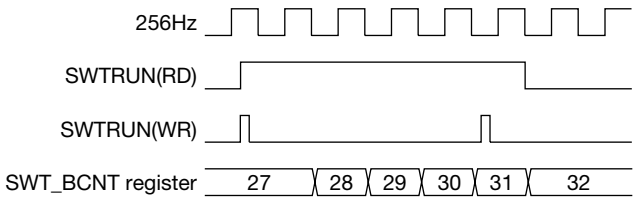


Figure 16.5.2: Run/Stop control timing chart

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 sets the stopwatch 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 stopwatch timer interrupt by reading SIF* as part of the stopwatch timer interrupt processing routine.
The interrupt factor should be cleared with the interrupt processing routine by resetting the SWT module SIF* (to 1) rather than the ITC stopwatch timer interrupt flag.

Note: To prevent generating unnecessary interrupts, reset the corresponding SIF* before permitting stopwatch timer interrupt from SIE*.

Stopwatch timer interrupt ITC register

The stopwatch timer outputs an interrupt signal to the ITC using the falling edge for the frequency for which interrupts are permitted in the settings previously described. To generate stopwatch timer interrupts, the interrupt level and interrupt permission should be set in the ITC register.
The stopwatch timer ITC control bits are shown below.

Interrupt flag inside ITC
* **EIFT2**: Stopwatch Timer Interrupt Flag in the Interrupt Flag (ITC_IFLG) Register (D2/0x4300)

Interrupt enable bit inside ITC
* **EIEN2**: Stopwatch Timer Interrupt Enable Bit in the Interrupt Enable (ITC_EN) Register (D2/0x4302)

Interrupt level setting bit inside ITC
* **EILV2[2:0]**: SWT Interrupt Level Bits in the External Interrupt Level Setup (ITC_ELV1) Register 1 (D[2:0]/0x4308)

Interrupt trigger mode selection bit inside ITC (Fix at 1)
* **EITG2**: SWT Interrupt Trigger Mode Select Bit in the External Interrupt Level Setup (ITC_ELV1) Register 1 (D4/0x4308)

EIFT2 is set to 1 at the falling edge of the 100/10/1 Hz signals for which interrupts are permitted by the SWT module. If EIEN2 is set to 1 here, the ITC sends an interrupt request to the S1C17 core. To prevent stopwatch timer interrupts, set the EIEN2 to 0. EIFT2 is set to 1 by the interrupt signal from the SWT module regardless of the EIEN2 setting (even if it is set to 0).
EILV2[2:0] sets the stopwatch timer interrupt level (0 to 7).

The S1C17 core accepts interrupts when the following conditions are satisfied:
- The interrupt enable bit has been set to 1.
- The PSR (S1C17 core internal processor status register) IE (interrupt enable) bit has been set to 1.
- The stopwatch timer interrupt has been set to a higher interrupt level than that set for the PSR IL (interrupt level).
- No other interrupt factors having higher precedence (e.g., NMI) are present.

For detailed information on these interrupt control registers and operations when interrupts occur, refer to "6 Interrupt Controller (ITC)."

Note: The following processes must be performed to manage the interrupt factor occurrence state using the SWT module interrupt flag.

1. **Set the ITC stopwatch timer interrupt trigger mode to level trigger mode.**
2. **Reset the SWT module interrupt flag SIF* within the interrupt processing routine after the interrupt occurs (this also resets the ITC interrupt flag).**

Interrupt vectors

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

Vector number: 6 (0x06)
Vector address: 0x8018

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.

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 sets the stopwatch timer interrupt flag in the ITC to 1 and generates an interrupt if the ITC and S1C17 core interrupt conditions are met.

The following processes must be performed to manage the interrupt factor occurrence state using this register.

1. Set the ITC stopwatch timer interrupt trigger mode to level trigger mode.
2. Reset the SWT module interrupt flag within the interrupt processing routine after the interrupt occurs (this also resets the ITC interrupt flag).

SIF* is reset by writing as 1.

Note: 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 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.

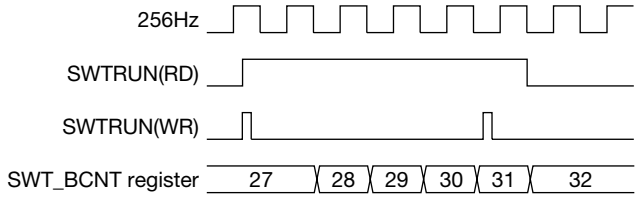


Figure 16.8.1: Run/Stop control timing chart

- Executing the slp command 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 command.

This page intentionally left blank.

17 Watchdog Timer (WDT)

17.1 Watchdog Timer Overview

The S1C17001 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.

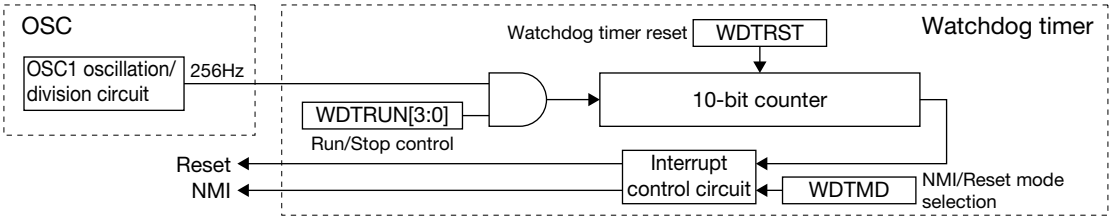


Figure 17.1.1: Watchdog timer block diagram

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 0x8000 and 0x8008.

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 command. 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 command. The watchdog should also be stopped as required using 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 f_{OSC1} = 32.768 kHz).
This resets the up-counter to 0 and starts counting with a new NMI/Reset generation cycle.

D[3:0] WDTRUN[3:0]: Watchdog Timer Run/Stop Control Bits

Controls 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.

This page intentionally left blank.

18 UART

18.1 UART Configuration

The S1C17001 incorporates a single-channel UART. The UART transfers asynchronous data with external devices at a transfer rate of 150 to 460,800 bps (115,200 bps in IrDA mode). It includes a 2-byte receive data buffer and 1-byte transmit data buffer and is capable of full-duplex communications. The transfer clock can be either the internal clock using the timer module or an external clock input via the SCLK (P25) pin. Data length (7 or 8 bits), stop bit length (1 or 2 bits), and parity mode (even, odd, no parity) can be selected via the software. The start bit is fixed at 1 bit. Overrun errors, framing errors, and parity errors can be detected while receiving data. The UART generates three different interrupt types (transmit buffer empty, receive buffer full, and receive error) and enables efficient processing of serial data transfer using interrupt processing.
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.

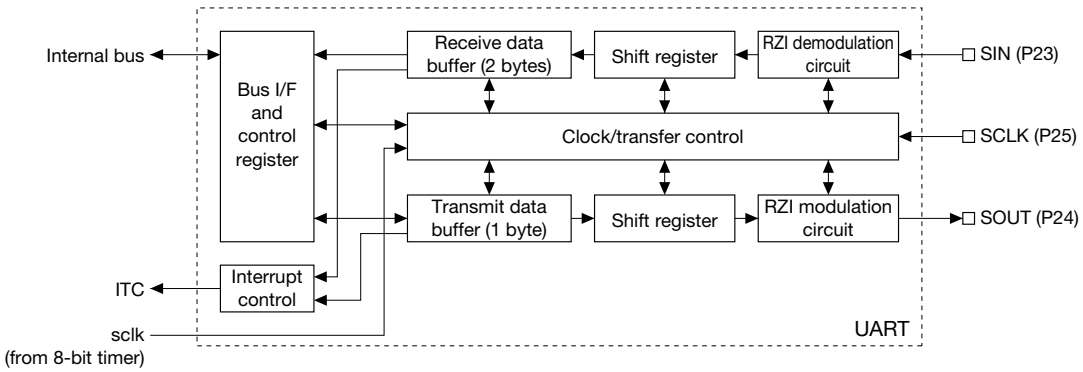


Figure 18.1.1: UART configuration

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
SIN (P23)	I	1	UART data input pin Inputs serial data sent from an external device.
SOUT (P24)	O	1	UART data output pin Outputs serial data sent to an external device.
SCLK (P25)	I	1	UART clock input pin Inputs the external clock when used for the transfer clock.

The UART input/output pins (SIN, SOUT, SCLK) are shared with general purpose input/output port pins (P23, P24, P25) and are initially set as general purpose input/output port pins. The function must be switched using the P2_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.

P23 → SIN
- ∗ **P23MUX**: P23 Port Function Select Bit in the P2 Port Function Select (P2_PMUX) Register (D3/0x52a2)

P24 → SOUT
- ∗ **P24MUX**: P24 Port Function Select Bit in the P2 Port Function Select (P2_PMUX) Register (D4/0x52a2)

P25 → SCLK (only when using external clock)
- ∗ **P25MUX**: P25 Port Function Select Bit in the P2 Port Function Select (P2_PMUX) Register (D5/0x52a2)

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 register).

* **SSCK**: Input Clock Select Bit in the UART Mode (UART_MOD) Register (D0/0x4103)

Note: Make sure the UART is halted (when RXEN/UART_CTL register = 0) before changing SSCK.

* **RXEN**: UART Enable Bit in the UART Control (UART_CTL) Register (D0/0x4104)

Internal clock

Setting SSCK to 0 (default) selects the internal clock. Since the UART uses the 8-bit timer output clock as the transfer clock, the 8-bit timer must be programmed to output a clock suited to the transfer rate.
For detailed information on 8-bit timer control, refer to "12 8-bit Timer (T8F)."

External clock

Setting SSCK to 1 selects the external clock. In this case, set P25 to the SCLK 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 register = 0) before changing transfer data format settings.

∗ **RXEN**: UART Enable Bit in the UART Control (UART_CTL) Register (D0/0x4104)

Data length

The data length is selected by CHLN (D4/UART_MOD 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 Mode (UART_MOD) Register (D4/0x4103)

Stop bit

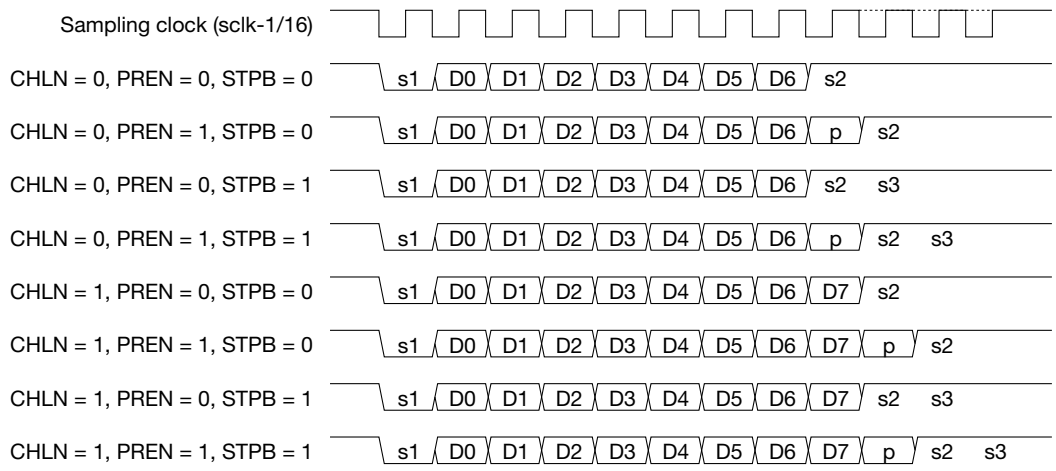
The stop bit length is selected by STPB (D1/UART_MOD register). Setting STPB to 0 (default) sets the stop bit length to 1 bit. Setting STPB to 1 sets the stop bit length to 2 bits.

∗ **STPB**: Stop Bit Select Bit in the UART Mode (UART_MOD) Register (D1/0x4103)

Parity bit

Whether the parity function is enabled or disabled is selected by PREN (D3/UART_MOD 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 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 Mode (UART_MOD) Register (D3/0x4103)
∗ **PMD**: Parity Mode Select Bit in the UART Mode (UART_MOD) Register (D2/0x4103)



s1: Start bit, s2 & s3: Stop bits, p: Parity bit

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 register = 0) before changing the above settings.

∗ **RXEN**: UART Enable Bit in the UART Control (UART_CTL) Register (D0/0x4104)

Permitting data transfers

Set the RXEN bit (D0/UART_CTL 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, write the transmission data to the UART_TXD register (0x4101).

∗ **UART_TXD**: UART Transmit Data Register (0x4101)

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 register) and TRBS (D2/UART_ST register) status flags.

∗ **TDBE**: Transmit Data Buffer Empty Flag in the UART Status (UART_ST) Register (D0/0x4100)
∗ **TRBS**: Transmit Busy Flag in the UART Status (UART_ST) Register (D2/0x4100)

The TDBE flag indicates the transmit data buffer status. This flag switches to 0 when the application program writes data to the transmit data buffer and reverts to 1 when the buffer data is sent to the transmit shift register. 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 overwrite 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.

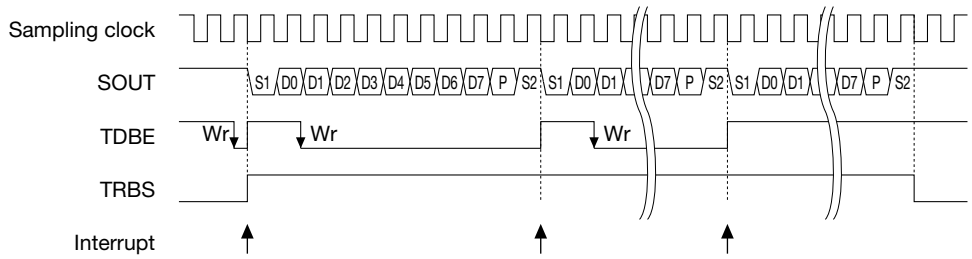


Figure 18.5.1: Data transmission timing chart

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 register (0x4102). The oldest data is read out first, clearing the register.

* **UART_RXD**: UART Receive Data Register (0x4102)

The receiving circuit includes the RDRY (D1/UART_ST register) and RD2B (D3/UART_ST register) buffer status flags.

* **RDRY**: Receive Data Ready Flag in the UART Status (UART_ST) Register (D1/0x4100)
* **RD2B**: Second Byte Receive Flag in the UART Status (UART_ST) Register (D3/0x4100)

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 item 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.

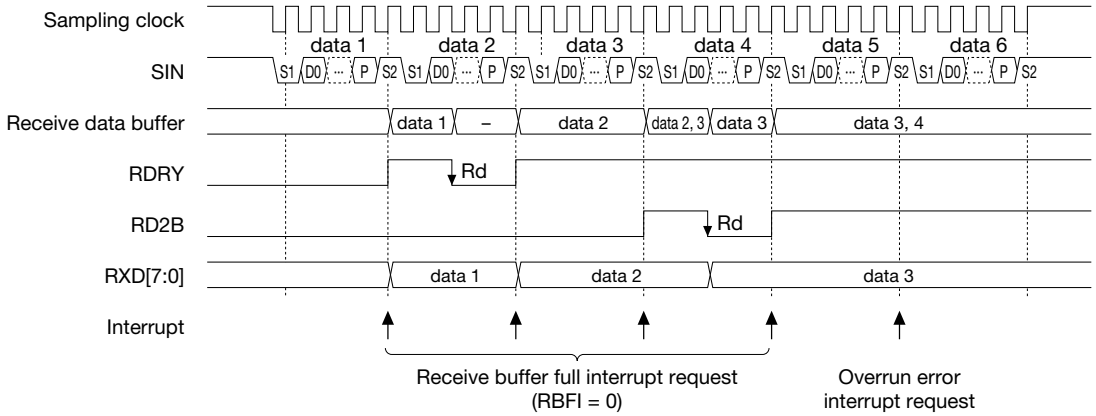
(3) RDRY = 1, RD2B = 1
Two 8-bit data have been received. Read the receive data buffer contents twice. The receive data buffer outputs the oldest data first. This reading resets the RD2B flag. The buffer then reverts to the state in (2) above. The second read outputs the most recent received data, after 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 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 Control (UART_CTL) Register (D1/0x4104)

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 buffer, clearing any remaining data. The data being transferred cannot be guaranteed if RXEN is set to 0 while data is being sent or received.

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 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 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 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 register) is reset to 0 by writing as 1.

- **PREN**: Parity Enable Bit in the UART Mode (UART_MOD) Register (D3/0x4103)
- **PMD**: Parity Mode Select Bit in the UART Mode (UART_MOD) Register (D2/0x4103)
- **PER**: Parity Error Flag in the UART Status (UART_ST) Register (D5/0x4100)

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 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 register) is reset to 0 by writing as 1.

- **FER**: Framing Error Flag in the UART Status (UART_ST) Register (D6/0x4100)

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 register) is set to 1.
The receiving operation continues even if this error occurs.
The OER flag (D4/UART_ST register) is reset to 0 by writing as 1.

- **OER**: Overrun Error Flag in the UART Status (UART_ST) Register (D4/0x4100)

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 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 Control (UART_CTL) Register (D4/0x4104)

When transmission data written to the transmit data buffer is transferred to the shift register, the UART sets the TDBE bit (D0/UART_ST 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 Status (UART_ST) Register (D0/0x4100)

An interrupt occurs if other interrupt conditions are met.
You can inspect the TDBE flag in the UART interrupt processing 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 processing routine.

Receive buffer full interrupt

To use this interrupt, set RIEN (D5/UART_CTL 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 Control (UART_CTL) Register (D5/0x4104)

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 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 register) is set to 1). If RBFI (D1/UART_CTL 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 register) is set to 1).

∗ **RBFI**: Receive Buffer Full Interrupt Condition Setup Bit in the UART Control (UART_CTL) Register (D1/0x4104)
∗ **RDRY**: Receive Data Ready Flag in the UART Status (UART_ST) Register (D1/0x4100)
∗ **RD2B**: Second Byte Receive Flag in the UART Status (UART_ST) Register (D3/0x4100)

An interrupt occurs if other interrupt conditions are met.
You can inspect the RDRY and RD2B flags in the UART interrupt processing 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 processing 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 Status (UART_ST) Register (D5/0x4100)
* **FER**: Framing Error Flag in the UART Status (UART_ST) Register (D6/0x4100)
* **OER**: Overrun Error Flag in the UART Status (UART_ST) Register (D4/0x4100)

If other interrupt conditions are satisfied, an interrupt occurs.
Inspect the error flags above as part of the UART interrupt processing 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 processing routine will proceed with error recovery.

UART interrupt ITC registers

The UART ITC control bits are listed below.

Interrupt flag

* **IIFT4**: UART Interrupt Flag in the Interrupt Flag (ITC_IFLG) Register (D12/0x4300)

Interrupt enable bit

* **IIEN4**: UART Interrupt Enable Bit in the Interrupt Enable (ITC_EN) Register (D12/0x4302)

Interrupt level setting bit

* **IILV4[2:0]**: UART Interrupt Level Bits in the Internal Interrupt Level Setup (ITC_ILV2) Register 2 (D[2:0]/0x4312)

If an interrupt request pulse is output by the UART, the corresponding interrupt flag is set to 1.
If the interrupt enable bit corresponding to this interrupt flag is set to 1, the ITC sends an interrupt request to the S1C17 core. To prohibit UART interrupts, set the interrupt enable bit to 0. The interrupt flag is set to 1 by a UART interrupt request pulse, regardless of the interrupt enable bit setting (i.e., even if set to 0).
The interrupt level setting bit sets the UART interrupt level (0 to 7).
The S1C17 core accepts interrupts when all of the following conditions are met:

- The interrupt enable bit is set to 1.
- The PSR (S1C17 core internal processor status register) IE (interrupt enable) bit is set to 1.
- The UART interrupt has a higher interrupt level set than that set for the PSR IL (interrupt level).
- No other interrupt factors having higher precedence (e.g., NMI) are present.

For detailed information on these interrupt registers and operations when interrupts occur, refer to “6 Interrupt Controller (ITC).”

Interrupt vectors

The UART interrupt vector numbers and vector addresses are as listed below.
Vector number: 16 (0x10)
Vector address: 0x8040

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.

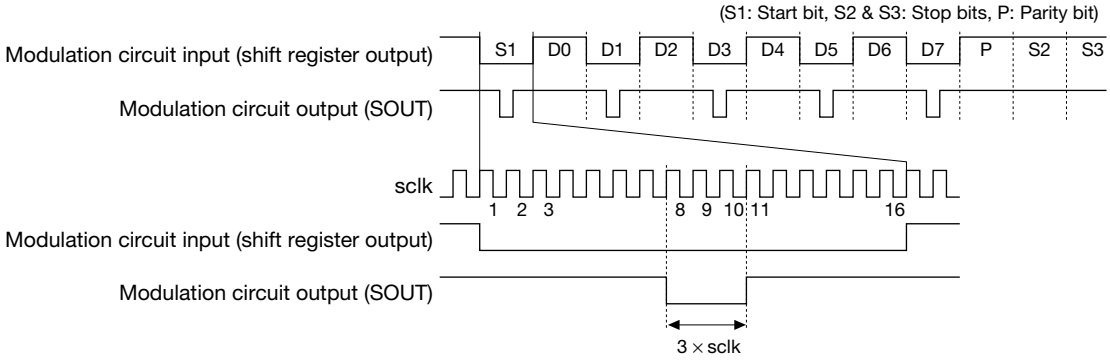


Figure 18.8.1: Transmission signal waveform

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).

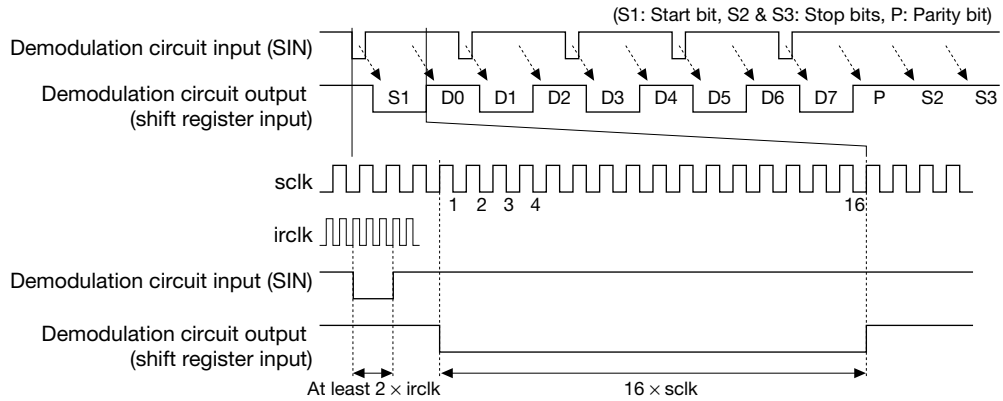


Figure 18.8.2: Receive signal waveform

IrDA enable

To use the IrDA interface function, set IRMD (D0/UART_EXP register) to 1. This enables the RZI modulation/demodulation circuit.

∗ **IRMD**: IrDA Mode Select Bit in the UART Expansion (UART_EXP) Register (D0/0x4105)

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 register)

* **IRCLK[2:0]**: IrDA Receive Detection Clock Select Bits in the UART Expansion (UART_EXP) Register (D[6:4]/0x4105)

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 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_ST	UART Status Register	Transfer, buffer, error status display
0x4101	UART_TXD	UART Transmit Data Register	Transmission data
0x4102	UART_RXD	UART Receive Data Register	Received data
0x4103	UART_MOD	UART Mode Register	Transfer data format setting
0x4104	UART_CTL	UART Control Register	Data transfer control
0x4105	UART_EXP	UART 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	

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 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: UART Transmit Data Register (UART_TXD)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
UART Transmit Data Register (UART_TXD)	**0x4101** (8 bits)	D7–0	**TXD[7:0]**	Transmit data TXD7(6) = MSB TXD0 = LSB	0x0 to 0xff (0x7f)	0x0	R/W	

D[7:0] TXD[7:0]: Transmit Data

Write 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: UART Receive Data Register (UART_RXD)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
UART Receive Data Register (UART_RXD)	**0x4102** (8 bits)	D7–0	**RXD[7:0]**	Receive data in the receive data buffer RXD7(6) = MSB RXD0 = LSB	0x0 to 0xff (0x7f)	0x0	R	Older data in the buffer is read out first.

D[7:0] RXD[7:0]: Receive Data

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 register) and RD2B (D3/UART_ST 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 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: UART Mode Register (UART_MOD)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
UART Mode Register (UART_MOD)	0x4103 (8 bits)	D7–5	–	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	

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)
0 (R/W): Internal clock (default)

Selects whether the internal clock (8-bit timer output clock) or external clock (input via SCLK pin) is used. Writing 1 to SSCK selects the external clock; Writing 0 to it selects the internal clock.

0x4104: UART Control Register (UART_CTL)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
UART Control Register (UART_CTL)	0x4104 (8 bits)	D7	–	reserved	–				–	–	0 when being read.
		D6	REIEN	Receive error int. enable	1	Enable	0	Disable	0	R/W	
		D5	RIEN	Receive buffer full int. enable	1	Enable	0	Disable	0	R/W	
		D4	TIEN	Transmit buffer empty int. enable	1	Enable	0	Disable	0	R/W	
		D3–2	–	reserved	–				–	–	0 when being read.
		D1	RBFI	Receive buffer full int. condition	1	2 bytes	0	1 byte	0	R/W	
		D0	RXEN	UART enable	1	Enable	0	Disable	0	R/W	

D7 Reserved

D6 REIEN: Receive Error Interrupt Enable Bit

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 write data to the transmit data buffer using interrupts.

D[3:2] Reserved

D1 RBFI: Receive Buffer Full Interrupt Condition Setup Bit

Sets the quantity of data in the receive 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 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 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: UART Expansion Register (UART_EXP)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
UART Expansion Register (UART_EXP)	**0x4105** (8 bits)	D7	**–**	reserved	–	–	–	0 when being read.
		D6–4	**IRCLK[2:0]**	IrDA receive detection clock select	IRCLK[2:0] / 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	

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

- Change the UART settings only while transfers are blocked (RXEN = 0).
  - ∗ **RXEN**: UART Enable Bit in the UART Control (UART_CTL) Register (D0/0x4104)
- Do not set RXEN to 0 while the UART is transmitting or receiving data.
- The UART transfer rate is capped at 460,800 bps (115,200 bps in IrDA mode). 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.

This page intentionally left blank.

19 SPI

19.1 SPI Configuration

The S1C17001 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.

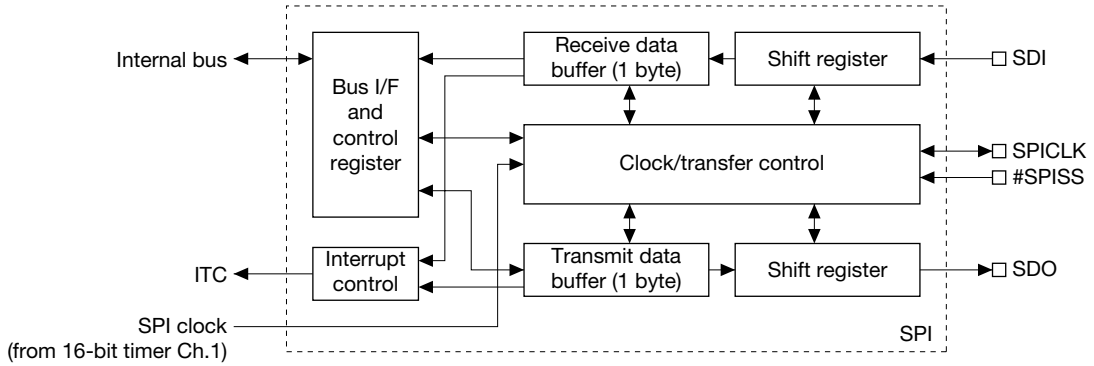


Figure 19.1.1: SPI module configuration

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 (P20)	I	1	SPI data input pin Inputs serial data from SPI bus.
SDO (P21)	O	1	SPI data output pin Outputs serial data to SPI bus.
SPICLK (P22)	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 (P17)	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 (P20, P21, P22, P17) and are initially set as general purpose input/output port pins. The function must be switched using the P2_PMUX and P1_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.

P20 → SDI

- **P20MUX**: P20 Port Function Select Bit in the P2 Port Function Select (P2_PMUX) Register (D0/0x52a2)

P21 → SDO

- **P21MUX**: P21 Port Function Select Bit in the P2 Port Function Select (P2_PMUX) Register (D1/0x52a2)

P22 → SPICLK

- **P22MUX**: P22 Port Function Select Bit in the P2 Port Function Select (P2_PMUX) Register (D2/0x52a2)

P17 → #SPISS

- **P17MUX**: P17 Port Function Select Bit in the P1 Port Function Select (P1_PMUX) Register (D7/0x52a1)

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. It should be programmed to output a clock matching the transfer rate from the 16-bit timer Ch.1. For detailed information on 16-bit timer control, refer to "11 16-bit Timer (T16)."

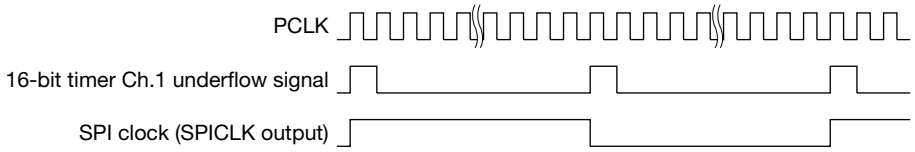


Figure 19.3.1: Master mode SPI clock

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.

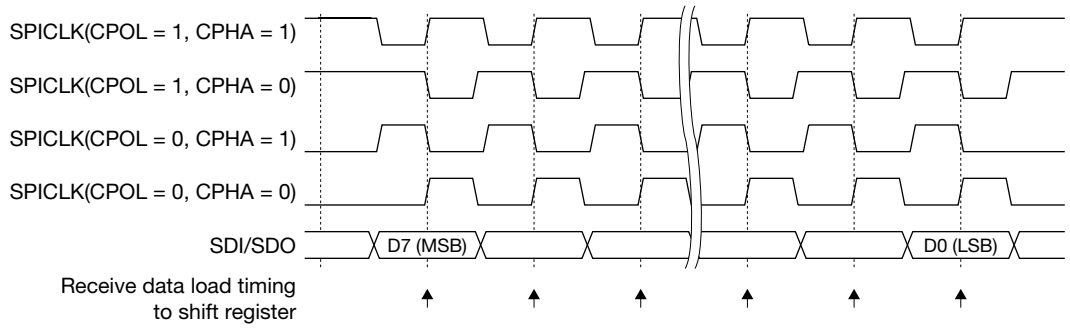


Figure 19.4.1: Clock and data transfer timing

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.

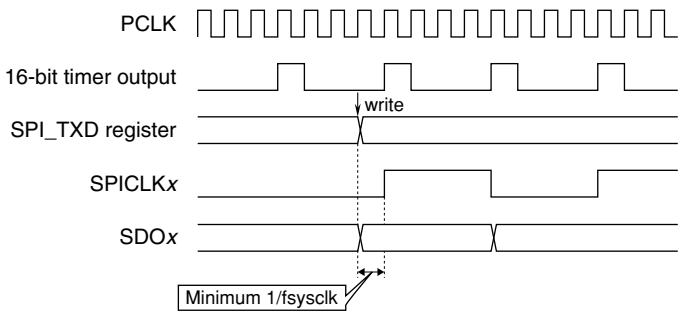


Figure 19.4.2 SDO*x* and SPICLK*x* Change Timings when CPHA = 0

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.

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 loaded into the shift register in sequence with the MSB leading at the clock rising or tailing edge, as 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.

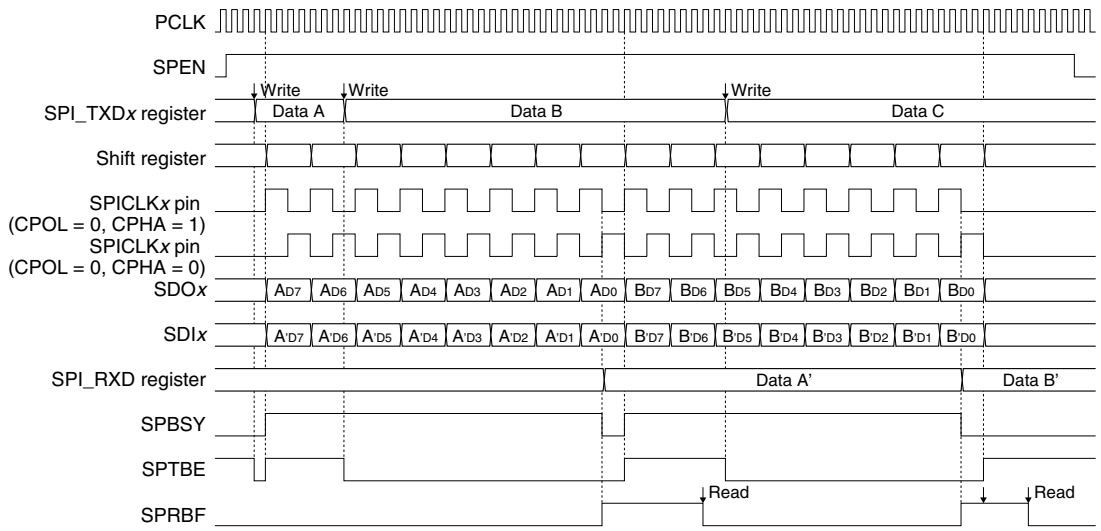


Figure 19.5.1: Data Transmission/Receiving Timing Chart (MSB first)

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.

SPI interrupt ITC registers

The control bits for the SPI module in the ITC are listed below.

Interrupt flag

∗ **IIFT6**: SPI Interrupt Flag in the Interrupt Flag (ITC_IFLG) Register (D14/0x4300)

Interrupt enable bit

∗ **IIEN6**: SPI Interrupt Enable Bit in the Interrupt Enable (ITC_EN) Register (D14/0x4302)

Interrupt level setting bit

∗ **IILV6[2:0]**: SPI Interrupt Level Bits in the Internal Interrupt Level Setup (ITC_ILV3) Register 3 (D[2:0]/0x4314)

If an interrupt request pulse is output by the SPI module, the IIFT6 interrupt flag is set to 1.
If the IIEN6 interrupt enable bit is set to 1, the ITC sends an interrupt request to the S1C17 core. To prohibit SPI interrupts, set IIEN6 to 0.
The IIFT6 flag is set to 1 by a SPI interrupt request pulse, regardless of the IIEN6 bit setting (i.e., even if set to 0).

The IILV6[2:0] interrupt level setting bit sets the SPI interrupt level (0 to 7).
The S1C17 core accepts interrupts when all of the following conditions are met:

- The interrupt enable bit is set to 1.
- The PSR (S1C17 core internal processor status register) IE (interrupt enable) bit is set to 1.
- The SPI interrupt has a higher interrupt level set than that set for the PSR IL (interrupt level).
- No other interrupt factors having higher precedence (e.g., NMI) are present.

For detailed information on these interrupt registers and operations when interrupts occur, refer to "6 Interrupt Controller (ITC)."

Interrupt vectors

The SPI interrupt vector numbers and vector addresses are as listed below.

Vector number: 18 (0x12)
Vector address: 0x8040

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.

• Always use 16-bit access commands to read and write to/from the SPI register. 32-bit and 8-bit access commands cannot be used to read and write to/from the SPI register.

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)	D15–6	–	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[15:6] Reserved

D5 SPRIE: Receive Data Buffer Full Interrupt Enable Bit

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.)

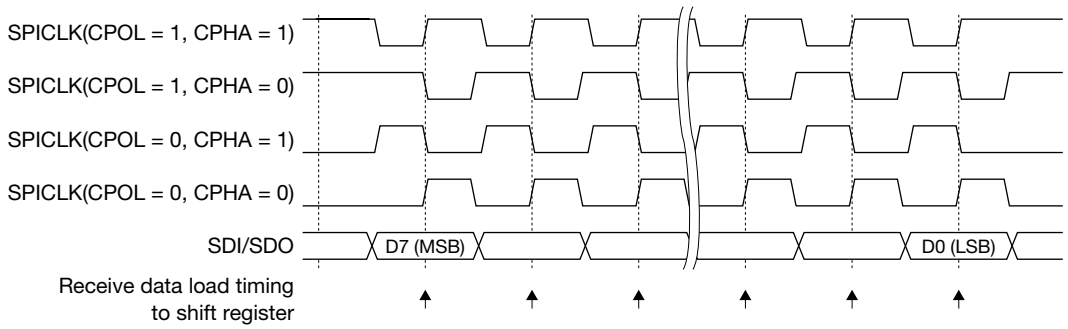


Figure 19.7.1: Clock and data transfer timing

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

- Always use 16-bit access commands to read and write to/from the SPI register (0x4320 to 0x4326). 32-bit and 8-bit access commands cannot be used to read and write to/from the SPI register.
- Do not access the SPI_CTL register (0x4326) while the SPBSY flag (D2/SPI_ST register) is set to 1 (while data is being transferred).

  * **SPBSY**: Transfer Busy Flag in the SPI Status (SPI_ST) Register (D2/0x4320)

20 I²C

20.1 I²C Configuration

The S1C17001 incorporates an I²C bus interface module for high-speed synchronized serial communications. The I²C 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 module configuration.

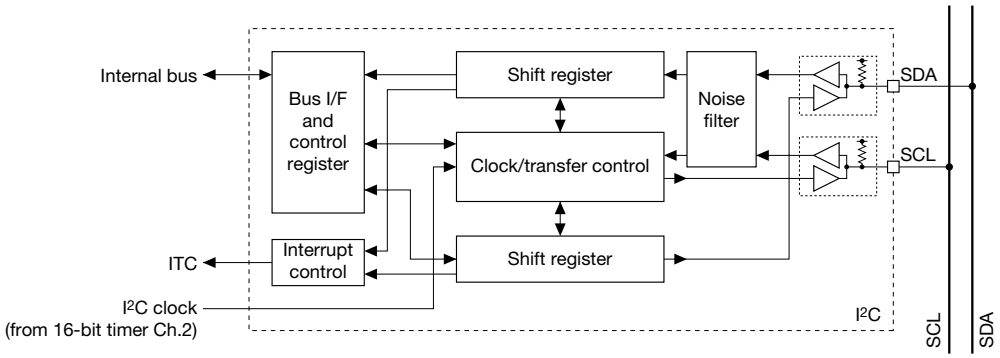


Figure 20.1.1: I²C module configuration

Note: This I²C module does not include a cross-stretch function and does not support I²C slave devices that synchronize communications by cross-stretching.

20.2 I²C Input/Output Pins

Table 20.2.1 lists the I²C pins.

Table 20.2.1: I²C pin list

Pin name	I/O	Qty	Function
SDA (P14)	I/O	1	I²C data input/output pin Inputs serial data from the I²C bus. Also outputs serial data to the I²C bus.
SCL (P15)	I/O	1	I²C clock input/output pin Inputs SCL line status. Also outputs a serial clock.

The I²C input/output pins (SDA, SCL) are shared with general purpose input/output port pins (P14, P15) and are initially set as general purpose input/output port pins. The function must be switched using the P1_PMUX register setting to use general purpose input/output port pins as I²C input/output pins. Switch the pins to I²C mode by setting the following control bits to 1.

P14 → SDA

- **P14MUX: P14 Port Function Select Bit in the P1 Port Function Select (P1_PMUX) Register (D4/0x52a1)**

P15 → SCL

- **P15MUX: P15 Port Function Select Bit in the P1 Port Function Select (P1_PMUX) Register (D5/0x52a1)**

For detailed information on pin function switching, refer to "10.2 Input/Output Pin Function Selection (Port MUX)."

20.3 I²C Clock

The I²C module uses the internal clock output by the 16-bit timer Ch.2 as the synchronizing clock. This clock is output from the SCL 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)."

The I²C module does not function as a slave device. The SCL 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 module includes an optional noise filter function that can be selected via the application program.

Noise filter function

The I²C module incorporates a function for filtering noise from the SDA and SCL 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 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 clock. (See Section 11.)
(2) Select the option function. (See section 20.4.)
(3) Set the interrupt conditions to use I²C 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 transfers and permits clock input/output.

Note: Do not set I2CEN to 0 when the I²C 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) Register setting procedure

To generate a START condition, set the following registers in the order shown below:
1. Set the slave address to RTDT[7:0] (D[7:0]/I2CM_DAT register).
2. Set TXE (D9/I2CM_DAT register) to 1.
3. Set STRT (D0/I2CM_CTL register) to 1.

∗ **RTDT[7:0]**: Receive/Transmit Data Bits in the I²C Master Data (I2CM_DAT) Register (D[7:0]/0x4344)
∗ **TXE**: Transmit Execution Bit in the I²C Master Data (I2CM_DAT) Register (D9/0x4344)
∗ **STRT**: Start Control Bit in the I²C Master Control (I2CM_CTL) Register (D0/0x4342)

This procedure generates the communication waveforms as shown in Items (2) and (3) below. Be sure to follow the register setting procedure.

(2) Generate start condition

The start condition applies when the SCL line is maintained at High and the SDA line is maintained at Low.

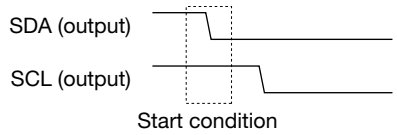


Figure 20.5.1: Start condition

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.

(3) Slave address transmission

Once the start condition has been generated, the I²C master (this module) sends a bit indicating the slave address and transfer direction for communications. I²C slave addresses are either 7-bit or 10-bit. This module uses an 8-bit transfer data register to send the slave address and transfer direction bit, enabling single transfers in 7-bit address mode. In 10-bit mode, data is sent twice or three times under software control. Figure 20.5.2 gives the configuration of the address data.

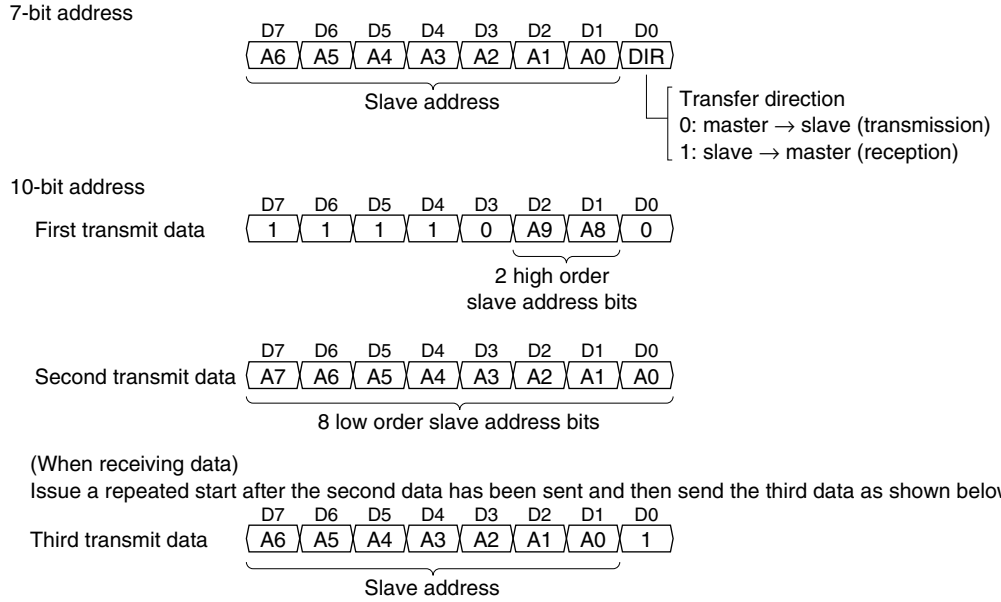


Figure 20.5.2: Slave address and transmission data specifying transfer direction

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.
Transmission is controlled as described below by setting the 8-bit data created as described above to the transfer data register.

The slave address and transfer direction bits can be sent only once after generating the start condition. 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. Slave address transmission involves the same procedure.
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.

- **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)

When TXE is set to 1, the I²C module begins data transmission in sync with the clock. If the start condition or previous data is currently being transmitted, data transmission starts after this has been completed.
The I²C module first transfers the data written to the shift register, then starts outputting the clock from SCL. 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 SDA pin with the MSB leading.
The I²C module outputs 9 clocks with each data transmission. In the 9th clock cycle, an ACK or NACK is received from the slave device with the SDA 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 module interprets to mean an NACK(1) (transmission failed).

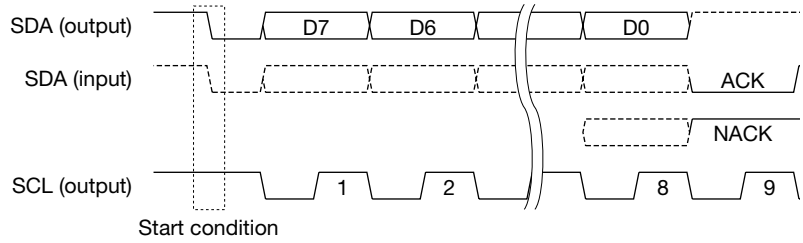


Figure 20.5.3: ACK and NACK

The I²C 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. It also reverts to 0 for Wait state.
Inspect the flag to check whether the I²C 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. The start condition must first be generated before receiving data, and the slave address 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 module starts outputting the clock from the SCL pin with the SDA line at high impedance. The data is loaded to the shift register in sequence at the clock rising edge, with the MSB leading.
RXE is reset to 0 when D6 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 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, so either use this interrupt or read the receive data to determine the presence of valid receive data in RTDT[7:0] by inspecting the RBRDY flag. If the subsequent data is received before RTDT[7:0] is read out, the newly received data overwrites the data already received in RTDT[7:0].
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 module is currently receiving or in standby.
The I²C module outputs nine clocks for one data transmission. In the 9th clock cycle, 0 or 1 set to RTACK(D8/I2C_DAT register) is sent from the SDA pin to the slave as ACK, NACK response, respectively. Do not change RTACK while ACK or NACK is in response.

Note: Receive buffer full interrupt occurs when ACK/NACK is in response (the 9th clock cycle). When the RTACK setting is rewritten without waiting for the end of the ACK/NACK period immediately after this interrupt is occured, ACK/NACK response output to the SDA pin is changed during the response time, the correct communication can not be performed.

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.

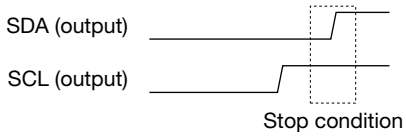


Figure 20.5.4: Stop condition

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)

The stop condition described above is generated if STP is 1 and TXE (D9/I2C_register), RXE (D10/I2C_DAT register), and STRT (D0/I2C_CTL register) are set to 0 when data transfer is complete (including ACK transfer). The I²C bus will then be free. STP is automatically reset to 0 if the stop condition is generated.
STP will be disabled if any of TXE, RXE, or STRT is 1.

The I²C module does not support repeated start condition. The stop condition cannot be omitted before generating the start condition for the subsequent data transfer.

Wait state for TXE, RXE, STRT, and STP settings

The module will switch to Wait state with the SCL output fixed at Low if all of the TXE (D9/I2C_DAT register), RXE (D10/I2C_DAT register), STRT (D0/I2C_CTL register), and STP (D1/I2C_CTL register) bits are 0 on completion of transfer for 1 byte of data and the ACK. This state is canceled either by writing 1 to TXE or RXE to restart data transfer or by generating the stop condition with STP.

Prohibiting data transfer

After the stop condition has been generated, write 0 to I2CEN to disable data transfers. To determine whether the stop condition has been generated, check to see if STP is automatically cleared to 0 after it is set to 1 by polling.
When I2CMEN is set to 0 while the I²C bus is in busy status, the SCL0 and SDA0 output levels and transfer data at that point cannot be guaranteed.

Timing chart

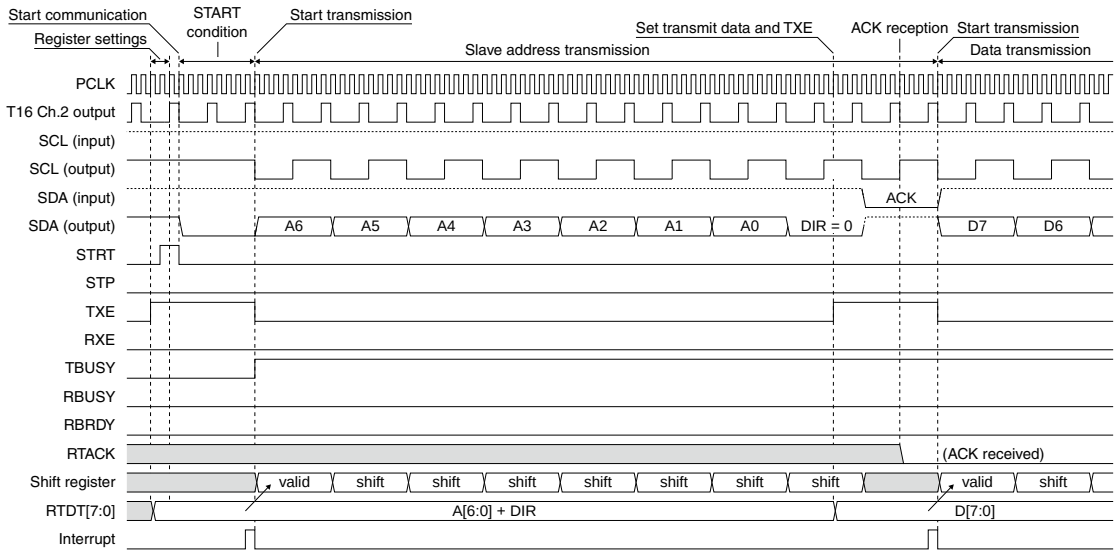


Figure 20.5.5: I²C timing chart 1 (Start condition à Data transmission)

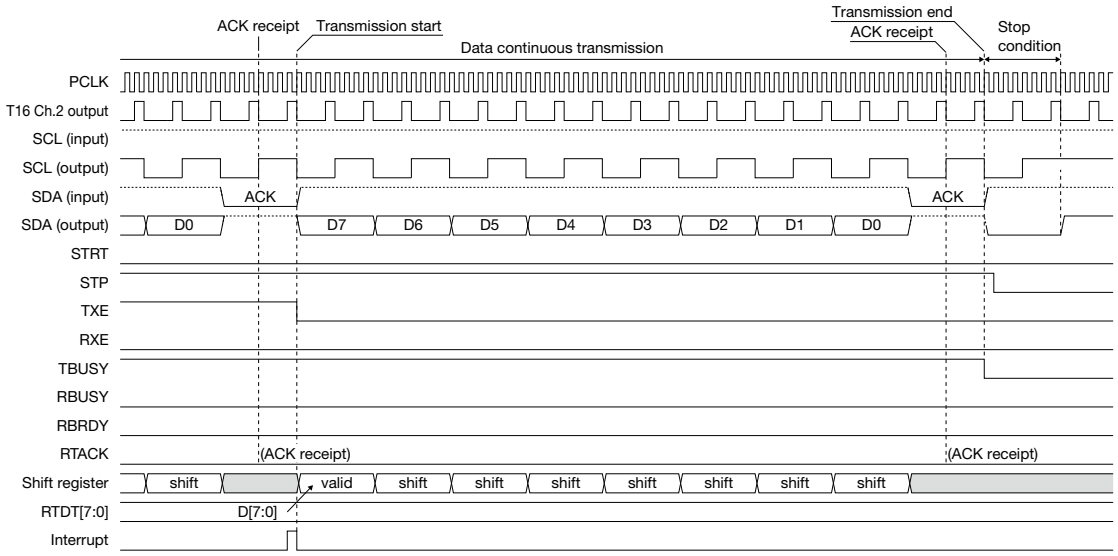


Figure 20.5.6: I²C timing chart 2 (Data transmission à Stop condition)

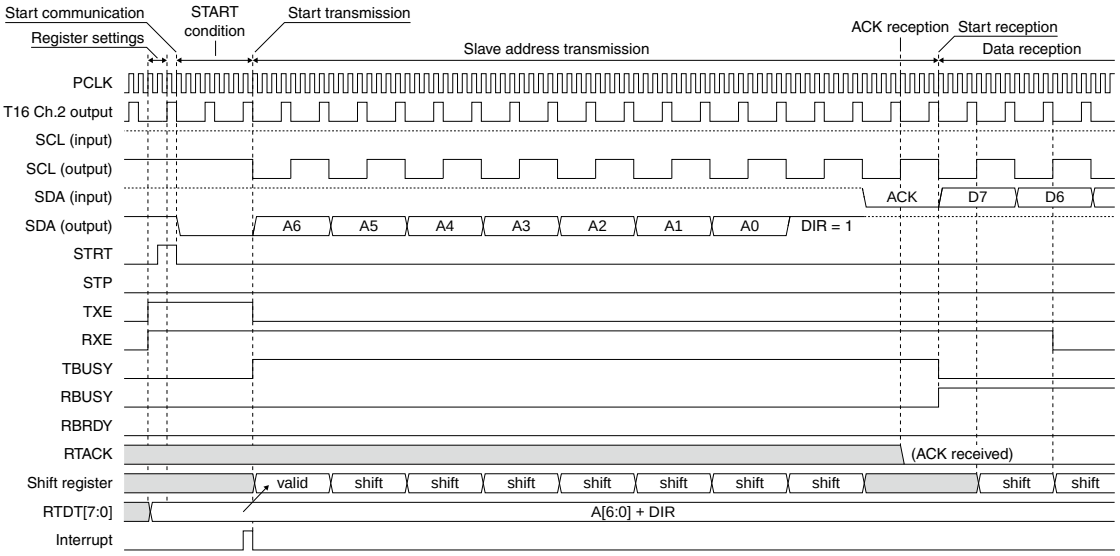


Figure 20.5.7: I²C timing chart 3 (Start condition ➔ Data receipt)

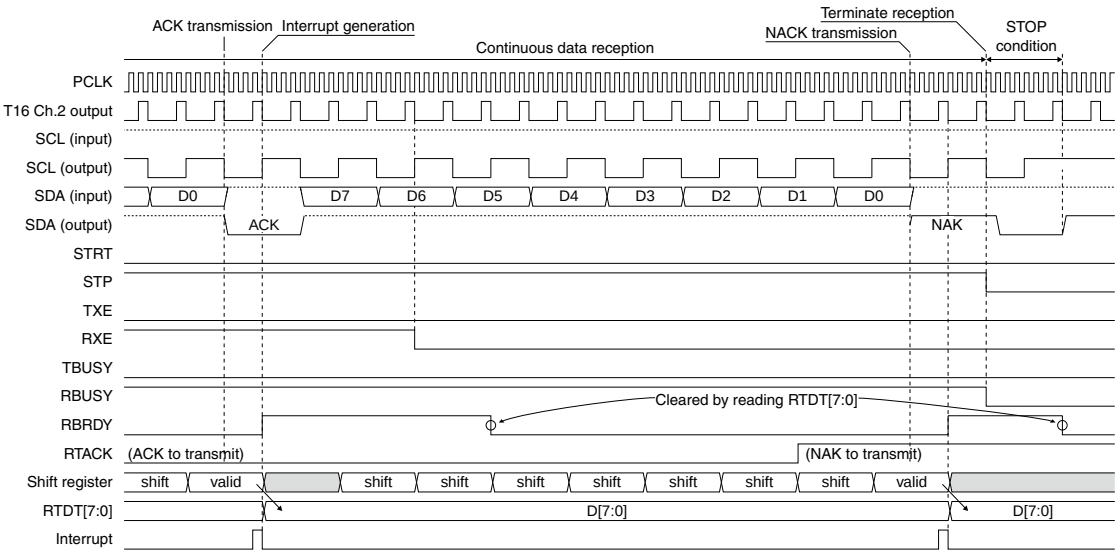


Figure 20.5.8: I²C timing chart 4 (Data receipt ➔ Stop condition)

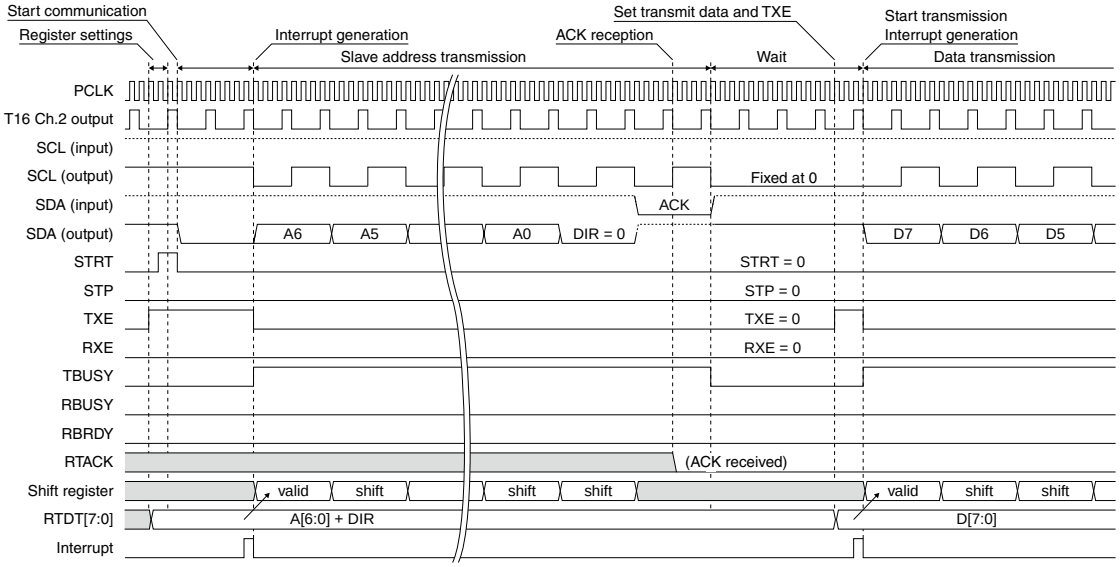


Figure 20.5.9: I²C timing chart 5 (Wait)

20.6 I²C Interrupts

The I²C module includes a function for generating the following two different interrupt types.
- Transmit buffer empty interrupt
- Receive buffer full interrupt

The I²C 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.

I²C interrupt ITC registers

The control bits for the I²C module in the ITC are listed below.

Interrupt flag
∗ **IIFT7**: I²C Interrupt Flag in the Interrupt Flag (ITC_IFLG) Register (D15/0x4300)

Interrupt enable bit
∗ **IIEN7**: I²C Interrupt Enable Bit in the Interrupt Enable (ITC_EN) Register (D15/0x4302)

Interrupt level setting bit
∗ **IILV7[2:0]**: I²C Interrupt Level Bits in the Internal Interrupt Level Setup (ITC_ILV3) Register 3 (D[10:8]/0x4314)

If an interrupt request pulse is output by the I²C module, the IIFT7 interrupt flag is set to 1.
If the IIEN7 interrupt enable bit is set to 1, the ITC sends an interrupt request to the S1C17 core. To prohibit I²C interrupts, set IIEN7 to 0.
The IIFT7 flag is set to 1 by a I²C interrupt request pulse, regardless of the IIEN7 bit setting (i.e., even if set to 0).
The IILV7[2:0] interrupt level setting bit sets the I²C interrupt level (0 to 7).

The S1C17 core accepts interrupts when all of the following conditions are met:
- The interrupt enable bit is set to 1.
- The PSR (S1C17 core internal processor status register) IE (interrupt enable) bit is set to 1.
- The I²C interrupt has a higher interrupt level set than that set for the PSR IL (interrupt level).
- There are no other interrupt factors, including NMI, with higher priority.

For detailed information on these interrupt registers and operations when interrupts occur, refer to "6 Interrupt Controller (ITC)."

Interrupt vectors

The I²C interrupt vector numbers and vector addresses are as listed below.

Vector number: 19 (0x13)
Vector address: 0x804c

20.7 Control Register Details

Table 20.7.1: I²C register list

Address	Register name		Function
0x4340	I2C_EN	I²C Enable Register	I²C module enable
0x4342	I2C_CTL	I²C Control Register	I²C control and transfer status display
0x4344	I2C_DAT	I²C Data Register	Transfer data
0x4346	I2C_ICTL	I²C Interrupt Control Register	I²C interrupt control

The I²C 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.
• Always use 16-bit access commands to read and write to/from the I²C register. 32-bit and 8-bit access commands cannot be used to read and write to/from the I²C register.

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 module operation.
1 (R/W): Permitted
0 (R/W): Prohibited (default)

Setting I2CEN to 1 starts the I²C module operation, enabling data transfer.
Setting I2CEN to 0 stops the I²C 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 the I²C receipt status.

1 (R): Operating

0 (R): Standby (default)

RBUSY is set to 1 when the I²C starts data receiving and is maintained at 1 while receiving is underway. It is cleared to 0 once receipt is complete or in Wait state.

D8 TBUSY: Transmit Busy Flag

Indicates the I²C transmission status.

1 (R): Operating

0 (R): Standby (default)

TBUSY is set to 1 when the I²C starts data transmission and is maintained at 1 while transmission is underway. It is cleared to 0 once transmission is complete. It is also returned to 0 in Wait state.

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 module incorporates a function for filtering noise from the SDA and SCL pin input signals. This function is enabled by setting NSERM to 1.

Note that using this function requires setting the I²C 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)

With STP set at 1, the I²C module generates the stop condition by changing the SDA line from Low to High while maintaining the I²C bus SCL line at High. The I²C bus subsequently becomes free.

Note that the stop condition will be generated only if STP is 1 and TXE (D9/I2C_DAT register), RXE (D10/I2C_DAT register), and STRT (D0) are set to 0 when data transfer is complete (including ACK transfer). STP is disabled if any of TXE, RXE, or STRT is 1.

STP is automatically reset to 0 if the stop condition is generated.

D0 STRT: Start Control Bit

Generates the start condition.
1 (R/W): Start condition generated
0 (R/W): Disabled (default)

With STRT set at 1, the I²C 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.
To generate a START condition, set the following registers in the order shown below:

1. Set the slave address to RTDT[7:0] (D[7:0]/I2CM_DAT register).
2. Set TXE (D9/I2CM_DAT register) to 1.
3. Set STRT to 1.

STRT is automatically reset to 0 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 exists

0 (R): No receive data (default)

The RBRDY flag becomes 1 when the data received in the shift register is loaded to RTDT[7:0] (D[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, so either use this interrupt or read the receive data to determine the presence of valid receive data in RTDT[7:0] by inspecting the RBRDY flag before reading out the receive data.

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 (NACK)

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 (NACK)

0 (R/W): ACK (default)

To return an ACK after data has been received, RTACK should be set to 0 before the I²C module sends the response bit.

To return an NACK, 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 SDA 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 SDA 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 interrupts.
1 (R/W): Permitted
0 (R/W): Prohibited (default)

Setting RINTE to 1 permits the output of I²C 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 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 interrupts.
1 (R/W): Permitted
0 (R/W): Prohibited (default)

Setting TINTE to 1 permits the output of I²C 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 interrupts are not generated by transmit buffer empty if TINTE is set to 0.

This page intentionally left blank.

21 Remote Controller (REMC)

21.1 REMC Configuration

The S1C17001 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 21.1.1 shows the REMC module configuration.

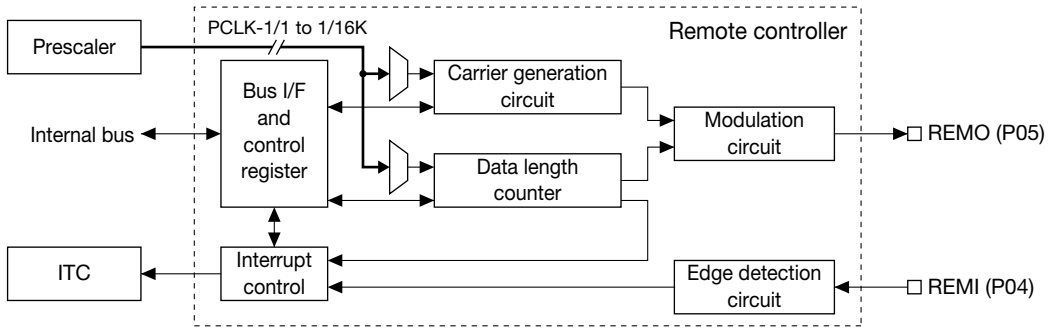


Figure 21.1.1: REMC module configuration

21.2 REMC Input/output Pin

Table 21.2.1 lists the REMC input/output pins.

Table 21.2.1: REMC input/output pin list

Pin name	I/O	Qty	Function
REMI (P04)	I	1	Remote control transmit data input pin Inputs receive data.
REMO (P05)	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 (P04, P05) 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.

P04 → REMI

* **P04MUX**: P04 Port Function Select Bit in the P0 Port Function Select (P0_PMUX) Register (D4/0x52a0)

P05 → REMO

* **P05MUX**: P05 Port Function Select Bit in the P0 Port Function Select (P0_PMUX) Register (D5/0x52a0)

For detailed information on pin function switching, refer to "10.2 Input/output Pin Function Selection (Port MUX)."

21.3 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[7:4]/REMC_PSC register).

* **CGCLK[3:0]**: Carrier Generator Clock Select Bits in the REMC Prescaler Clock Select (REMC_PSC) Register (D[7:4]/0x5341)

Table 21.3.1: 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[5:0]/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[5:0]/0x5343)

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 21.3.1.

Example: CGCLK[3:0] = 0x2 (PCLK-1/4), REMCH[5:0] = 2, REMCL[5:0] = 1

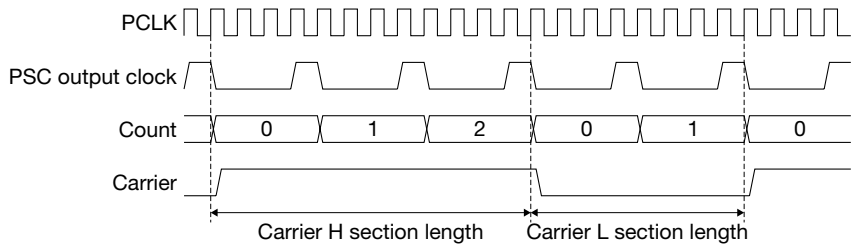


Figure 21.3.1: Carrier signal generation

21.4 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[3:0]/REMC_PSC register) provided separately to the carrier generation clock.

* **LCCLK[3:0]**: Length Counter Clock Select Bits in the REMC Prescaler Clock Select (REMC_PSC) Register (D[3:0]/0x5341)

Table 21.4.1: 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.

21.5 Data Transfer Control

Make the following settings before starting data transfers.
(1) Set the carrier signal. (See Section 21.3.)
(2) Select the data length counter clock. (See Section 21.4.)
(3) Set the interrupt conditions. (See Section 21.6.)

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

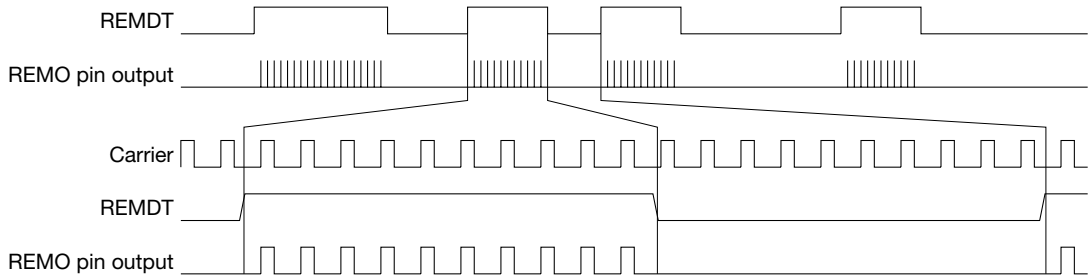


Figure 21.5.1: Data transmission

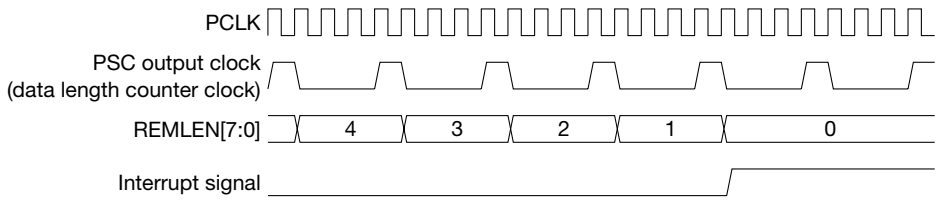


Figure 21.5.2: Underflow interrupt generation timing

(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[7:0]/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[7:0]/0x5345)

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

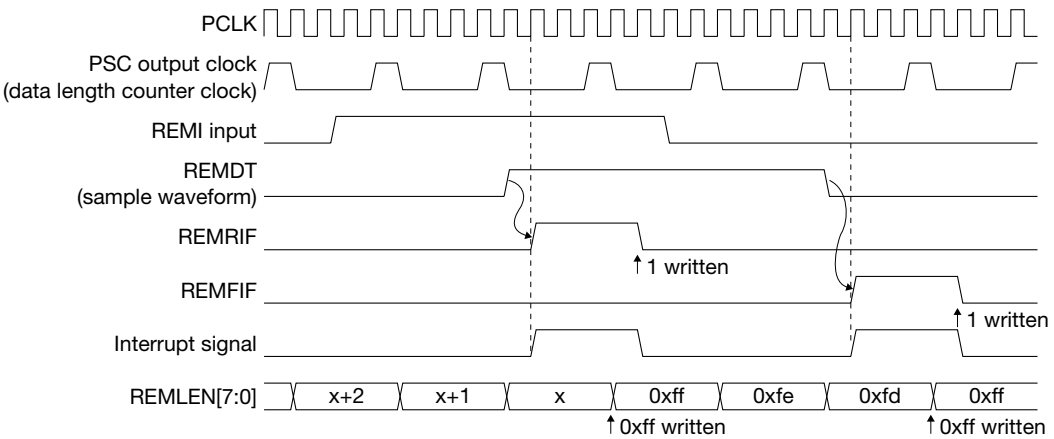


Figure 21.5.3: Data receipt

(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[7:0]/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_ST 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.

21.6 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_IFLG) Register (D0/0x5347)

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_IFLG) Register (D1/0x5347)

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_IMSK) 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_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.

∗ **REMFIF**: Falling Edge Interrupt Flag in the REMC Interrupt Flag (REMC_IFLG) Register (D2/0x5347)

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_IMSK) 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).

REMC interrupt ITC register

The REMC module outputs an interrupt signal to the ITC if an interrupt factor permitted by the previous settings occurs. To generate an REMC interrupt, the interrupt level and interrupt permission should be set using the ITC register.
The ITC control bits for the REMC are given below.

ITC internal interrupt flag

∗ **IIFT5**: Remote Controller Interrupt Flag in the Interrupt Flag (ITC_IFLG) Register (D13/0x4300)

ITC internal interrupt enable bit

∗ **IIEN5**: Remote Controller Interrupt Enable Bit in the Interrupt Enable (ITC_EN) Register (D13/0x4302)

ITC internal interrupt level setting bit

∗ **IILV5[2:0]**: REMC Interrupt Level Bits in the Internal Interrupt Level Setup (ITC_ILV2) Register 2 (D[10:8]/0x4312)

Interrupt signals from the REMC module set IIFT5 to 1. If IIEN5 is set to 1 here, the ITC sends an interrupt request to the S1C17 core. To block interrupts by the REMC module, set IIEN5 to 0. IIFT5 is set to 1 by the interrupt signal from the REMC module, regardless of the IIEN5 setting (even if set to 0).
IILV5[2:0] sets the REMC interrupt level (0 to 7).

The S1C17 core accepts interrupts when all of the following conditions are met:
- The interrupt enable bit is set to 1.
- The PSR (S1C17 core internal processor status register) IE (interrupt enable) bit is set to 1.
- The REMC interrupt has a higher interrupt level set than that set for the PSR IL (interrupt level).
- There are no other interrupt factors, including NMI, with higher priority.

For detailed information on these interrupt registers and operations when interrupts occur, refer to "6 Interrupt Controller (ITC)."

Interrupt vectors

The REMC interrupt vector numbers and vector addresses are as listed below.

Vector number: 17 (0x11)
Vector address: 0x8040

21.7 Control Register Details

Table 21.7.1: REMC register list

Address	Register name		Function
0x5340	REMC_CFG	REMC Configuration Register	Transfer selection and permission
0x5341	REMC_PSC	REMC Prescaler Clock Select Register	Prescaler output clock selection
0x5342	REMC_CARH	REMC H Carrier Length Setup Register	Carrier H section length setting
0x5343	REMC_CARL	REMC L Carrier Length Setup Register	Carrier L section length setting
0x5344	REMC_ST	REMC Status Register	Transfer bit
0x5345	REMC_LCNT	REMC Length Counter Register	Transfer data length setting
0x5346	REMC_IMSK	REMC Interrupt Mask Register	Interrupt mask setting
0x5347	REMC_IFLG	REMC Interrupt Flag Register	Interrupt occurrence status display and resetting

The REMC 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.

0x5340: REMC Configuration Register (REMC_CFG)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
REMC Configuration Register (REMC_CFG)	**0x5340** (8 bits)	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[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.

0x5341: REMC Prescaler Clock Select Register (REMC_PSC)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
REMC Prescaler Clock Select Register (REMC_PSC)	**0x5341** (8 bits)	D7–4	**CGCLK[3:0]**	Carrier generator clock select (Prescaler output clock)	CGCLK[3:0] LCCLK[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			
		D3–0	**LCCLK[3:0]**	Length counter clock select (Prescaler output clock)	0x7	PCLK-1/128	0x0	R/W	
					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[7:4] CGCLK[3:0]: Carrier Generator Clock Select Bits

Select a carrier generation clock from the 15 prescaler output clocks.

Table 21.7.2: 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[3:0] LCCLK[3:0]: Length Counter Clock Select Bits

Select a data length counter clock from the 15 prescaler output clocks.

Table 21.7.3: 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/REMC_CFG register = 0).

0x5342: REMC H Carrier Length Setup Register (REMC_CARH)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
REMC H Carrier Length Setup Register (REMC_CARH)	**0x5342** (8 bits)	D7–6	**–**	reserved	–	–	–	0 when being read.
		D5–0	**REMCH[5:0]**	H carrier length setup	0x0 to 0x3f	0x0	R/W	

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[7:4]/REMC_PSC 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[5:0]/REMC_CARL register).
The carrier signal is generated from these settings as shown in Figure 21.7.1.

Example: CGCLK[3:0] = 0x2 (PCLK-1/4), REMCH[5:0] = 2, REMCL[5:0] = 1

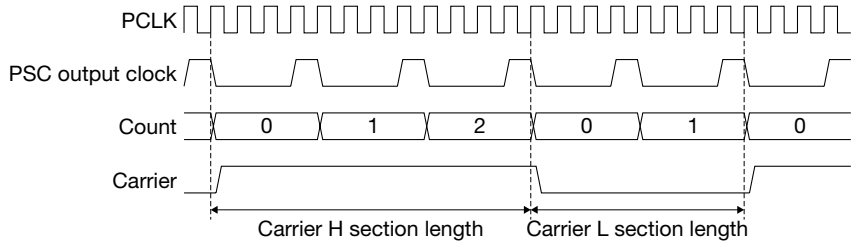


Figure 21.7.1: Carrier signal generation

0x5343: REMC L Carrier Length Setup Register (REMC_CARL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
REMC L Carrier Length Setup Register (REMC_CARL)	**0x5343** (8 bits)	D7–6	**–**	reserved	–	–	–	0 when being read.
		D5–0	**REMCL[5:0]**	L carrier length setup	0x0 to 0x3f	0x0	R/W	

D[7:6] Reserved

D[5:0] 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 CG-CLK[3:0] (D[7:4]/REMC_PSC register) + 1.

Calculate carrier L section length as follows:

$$\text{Carrier L section length} = \frac{\text{REMCL} + 1}{\text{clk\_in}} \text{ [s]}$$

REMCH: REMCL[5:0] settings

clk_in: Prescaler output clock frequency

The H section length is specified by REMCH[5:0] (D[5:0]/REMC_CARH register).

The carrier signal is generated from these settings as shown in Figure 21.7.1.

0x5344: REMC Status Register (REMC_ST)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
REMC Status Register (REMC_ST)	**0x5344** (8 bits)	D7–1	**–**	reserved	–	–	–	0 when being read.
		D0	**REMDT**	Transmit/receive data	1 1 (H) 0 0 (L)	0	R/W	

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.

0x5345: REMC Length Counter Register (REMC_LCNT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
REMC Length Counter Register (REMC_LCNT)	**0x5345** (8 bits)	D7–0	**REMLEN[7:0]**	Transmit/receive data length count (down counter)	0x0 to 0xff	0x0	R/W	

D[7:0] 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.

0x5346: REMC Interrupt Mask Register (REMC_IMSK)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
REMC Interrupt Mask Register (REMC_IMSK)	0x5346 (8 bits)	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 permits or blocks individual interrupt requests due to data length counter underflow, input signal rising edge, or input signal falling edge. Setting the interrupt enable bit to 1 permits interrupt requests from corresponding factors; setting it to 0 prevents interrupts. To generate interrupts, note that the ITC REMC interrupt enable bit must also be set to permit interrupts.

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)

0x5347: REMC Interrupt Flag Register (REMC_IFLG)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
REMC Interrupt Flag Register (REMC_IFLG)	0x5347 (8 bits)	D7–3	–	reserved	–				–	–	0 when being read.
		D2	REMFIF	Falling edge interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D1	REMRIF	Rising edge interrupt flag					0	R/W	
		D0	REMUIF	Underflow interrupt flag					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 generating unnecessary interrupts, reset the interrupt flag before permitting interrupts by the interrupt enable bit.

D2 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.

D1 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.

D0 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.

21.8 Precautions

The prescaler must run before operating the REMC module.

This page intentionally left blank.

22 On-chip Debugger (DBG)

22.1 Resource Requirements and Debugging Tool

Debugging work area

Debugging requires a 64-byte debugging work area. In the S1C17001, 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 S1C17701 debug pin and inputting the debug command 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 22.1.1: Debug pin list

Pin name	I/O	Qty	Function
DCLK (P31)	O	1	On-chip debugger clock output pin Outputs a clock to the ICD.
DSIO (P33)	I/O	1	On-chip debugger data input/output pin Used for inputting/outputting debugging data and inputting break signals.
DST2 (P32)	O	1	On-chip debugger status signal output pin Outputs the processor status during debugging.

Shared with general purpose input/output port pins (P31, P32, P33), 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 P3_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 → P31

∗ **P31MUX**: P31 Port Function Select Bit in the P3 Port Function Select (P3_PMUX) Register (D1/0x52a3)

DST2 →P32

∗ **P32MUX**: P32 Port Function Select Bit in the P3 Port Function Select (P3_PMUX) Register (D2/0x52a3)

DSIO → P33

∗ **P33MUX**: P33 Port Function Select Bit in the P3 Port Function Select (P3_PMUX) Register (D3/0x52a3)

For more information on pin function and switching, refer to “10.2 Input/Output Pin Function Selection (Port MUX).”

22.2 Debug Break Operation Status

The S1C17 core switches to debug mode when the `brk` command is executed or a debug interrupt is generated by a break signal (Low) input to the DSIO pin. This state persists until the `retd` command 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 & capture timer
- Remote controller
- P port
- UART
- SPI
- I^2C

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

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 (D0/0x5322)

The 8-bit OSC1 timer does not stop in debug mode, even if O1DBG is set to 1.

22.3 Control Register Details

Table 22.3.1: Debug register list

Address	Register name		Function
0x5322	MISC_OSC1	OSC1 Peripheral Control Register	OSC1 operation peripheral function setting for debugging
0xffff90	DBRAM	Debug RAM Base Register	Debug RAM base address display

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.

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 (8 bits)	D7–1	–	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

The 8-bit OSC1 timer does not stop in debug mode, even if O1DBG is set to 1.

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	0x7c0	0x7c0	R	

D[31:24] Not used (Fixed at 0)

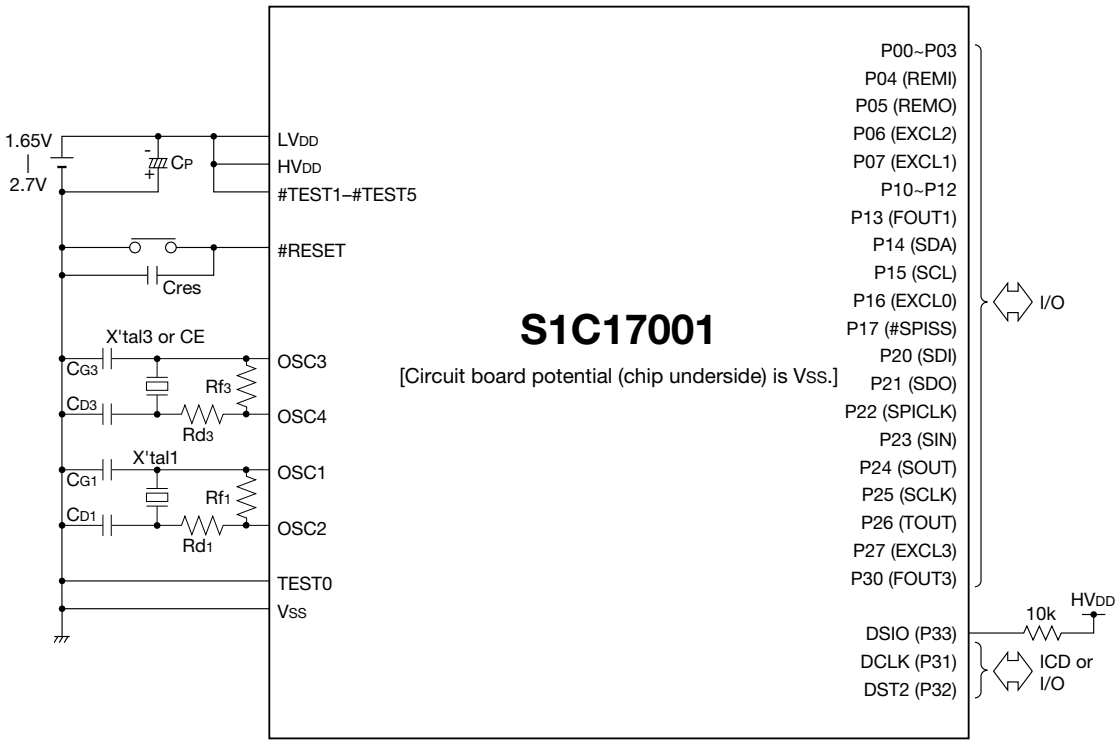
D[23:0] DBRAM[23:0]: Debug RAM Base Address Bits

Read-only register containing the initial address of the debugging work area (64 bytes).

This page intentionally left blank.

23 Basic External Connection Diagram

For single power supply (LV_{DD} = HV_{DD})

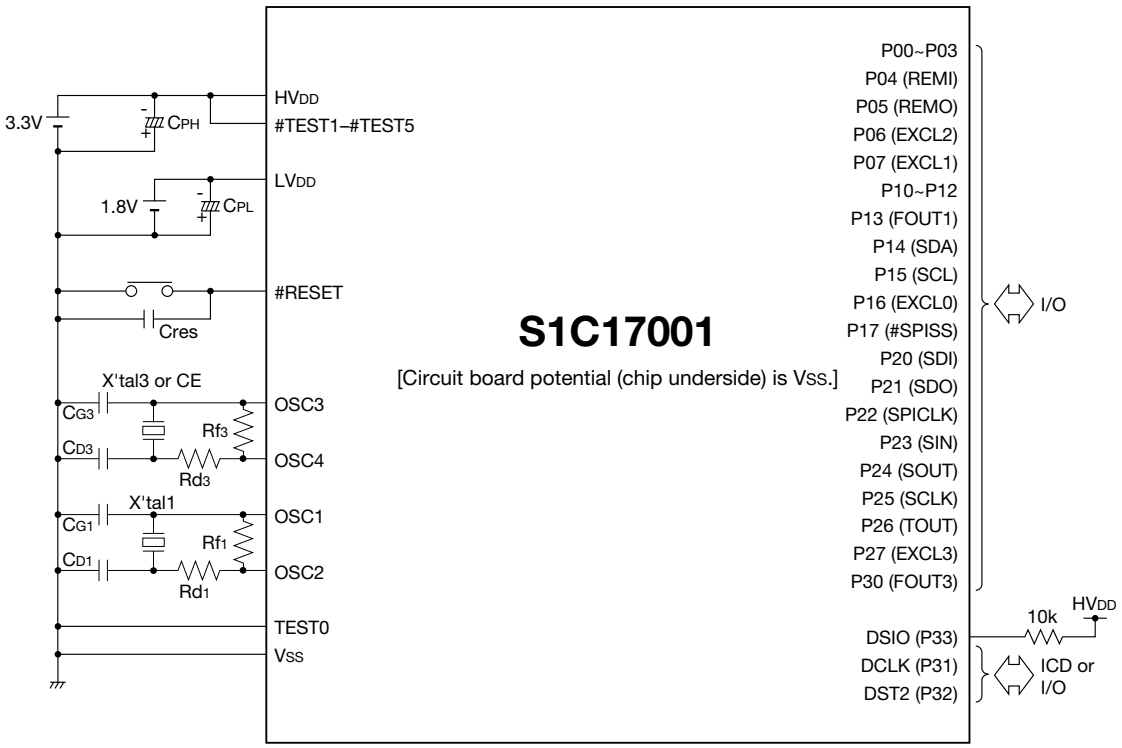


Recommended values for external components

Symbol	Name	Recommended value
X'tal1	Crystal oscillator	32.768kHz (MC-146, Epson Toyocom)
C_{G1}	Gate capacitor	7pF
C_{D1}	Drain capacitor	7pF
R_{f1}	Feedback resistor	10MΩ
R_{d1}	Drain resistor	0Ω
X'tal3	Crystal oscillator	8MHz (CA-301, Epson Toyocom)
CE	Ceramic oscillator	0.2~8MHz
C_{G3}	Gate capacitor	27pF
C_{D3}	Drain capacitor	27pF
R_{f3}	Feedback resistor	1MΩ
R_{d3}	Drain resistor	0Ω
C_P	Inter-power supply capacitor	3.3μF
Cres	#RESET pin capacitor	0.47μF

Note: The values given here are intended to serve as examples.
They do not represent performance guarantees.

For multiple power supplies ($LV_{DD} \neq HV_{DD}$)



Recommended values for external components

Symbol	Name	Recommended value
X'tal1	Crystal oscillator	32.768kHz (MC-146, Epson Toyocom)
C_{G1}	Gate capacitor	7pF
C_{D1}	Drain capacitor	7pF
Rf_1	Feedback resistor	10MΩ
Rd_1	Drain resistor	0Ω
X'tal3	Crystal oscillator	8MHz (CA-301, Epson Toyocom)
CE	Ceramic oscillator	0.2~8MHz
C_{G3}	Gate capacitor	27pF
C_{D3}	Drain capacitor	27pF
Rf_3	Feedback resistor	1MΩ
Rd_3	Drain resistor	0Ω
C_P	Inter-power supply capacitor	3.3µF
Cres	#RESET pin capacitor	0.47µF

Note: The values given here are intended to serve as examples.
They do not represent performance guarantees.

24 Electrical Characteristics

24.1 Absolute Maximum Ratings

(Vss = 0V)

Item	Code	Condition	Rating	Units
Power supply voltage	HV_{DD}	$HV_{DD} \geq LV_{DD}$	Vss - 0.3 to 4.0	V
	LV_{DD}		Vss - 0.3 to 3.0	V
Input voltage	HV_I		Vss - 0.3 to HV_{DD} + 0.5 *1	V
	LV_I		Vss - 0.3 to LV_{DD} + 0.5 *1	V
Output voltage	HV_O		Vss - 0.3 to HV_{DD} + 0.5 *1	V
	LV_O		Vss - 0.3 to LV_{DD} + 0.5 *1	V
Output current	I_{OUT}	1 pin	±30 (±50 *2)	mA
Storage temperature	Tstg		-65 to 150	°C

*1: With N channel as open drain (-0.3 V to 4.0 V permissible for input buffer)
*2: Applies to 24 mA output current buffer

24.2 Recommended Operating Conditions

(Vss = 0V)

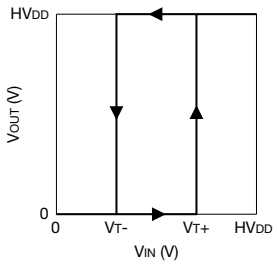
Item	Code	Condition	Min.	Typ.	Max.	Units
Power supply voltage	HV_{DD}		1.65		3.6	V
	LV_{DD}		1.65		2.7	V
Input voltage	HV_I		-0.3		HV_{DD} + 0.3	V
	LV_I		-0.3		LV_{DD} + 0.3 *1	V
Operating frequency	f_{OSC3}	Crystal/ceramic oscillation	0.2		8.2	MHz
	f_{OSC1}	Crystal oscillation		32.768	100	kHz
Ambient temperature	Ta		-40		85 *2	°C

*1: With N channel as open drain (Up to 3.9 V permissible for input buffer)
Do not apply voltages exceeding this value to fail-safe cell for H level output from external sources.
*2: Recommended ambient temperature assuming Tj = -40°C to 125°C.

24.3 DC Characteristics

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

Item	Code	Condition	Min.	Typ.	Max.	Units
High level Schmitt input voltage	V_{T+}	P***xx***, #RESET, #TEST1 to 5	$0.5HV_{DD}$		$0.9HV_{DD}$	V
Low level Schmitt input voltage	V_{T-}	P***xx***, #RESET, #TEST1 to 5	$0.1HV_{DD}$		$0.4HV_{DD}$	V
Hysteresis voltage	ΔV_1	P***xx***, #RESET, #TEST1 to 5 (HV_{DD} = 3V)	0.3			V
High level output current	I_{OH}	P***xx***, $V_{OH} = HV_{DD}$ - 0.2V			-0.9	mA
Low level output current	I_{OL}	P***xx***, V_{OL} = 0.2V	0.9			mA
Input leakage current	I_{LI}	P***xx***, #RESET, #TEST1 to 5	-1		1	µA
Output leakage current	I_{LO}	P***xx***, #RESET, #TEST1 to 5	-1		1	µA
Input pull-up resistance	R_{PU}	P***xx***, #RESET, #TEST1 to 5	30		480	kΩ



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

Item	Code	Condition	Min.	Typ.	Max.	Units
High level input current	V_{IH}	TEST0, OSC1, OSC3	$0.9LV_{DD}$			V
Low level input current	V_{IL}	TEST0, OSC1, OSC3			$0.1LV_{DD}$	V
Input pull-down resistance	R_{PD}	TEST0	20		240	kΩ

24.4 Consumption Current

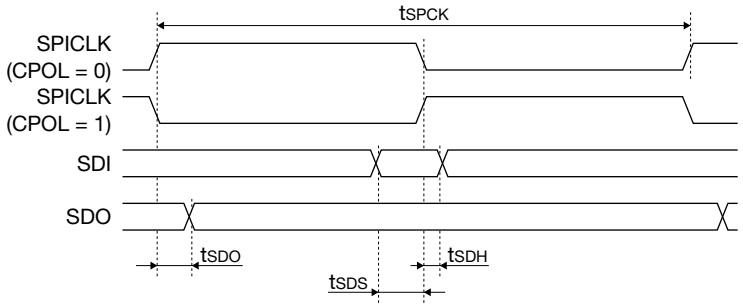
Unless otherwise specified: 1.8V, LV_{DD} = 1.8V, V_{SS} = 0V, Ta = 25°C, PCKEN = 3

Item	Code	Condition	Min.	Typ.	Max.	Units
Current consumption at SLEEP	I_{SLP}	OSC1 = OFF, OSC3 = OFF		0.5		µA
Current consumption at HALT	I_{HALT1}	OSC1 = 32kHz, OSC3 = OFF, PCKEN = 0		2.5		µA
	I_{HALT2}	OSC1 = 32kHz, OSC3 = 8MHz (ceramic)		580		µA
Current consumption during execution *1	I_{EXE1}	OSC1 = 32kHz, OSC3 = OFF, FLCYC = 4 (1 cycle)		10		µA
	I_{EXE2}	OSC1 = 32kHz, OSC3 = 8MHz (ceramic), FLCYC = 4 (1 cycle)		1800		µA
	I_{EXE3}	OSC1 = 32kHz, OSC3 = 2MHz (ceramic), FLCYC = 4 (1 cycle)		500		µA

*1: Current consumption during execution is the value exhibited when operating continuously while fetching the following test program from ROM: ALU commands 60.5%, branching commands 17%, memory reading 12%, and memory writing 10.5%.

24.5 AC Characteristics

24.5.1 SPI AC Characteristics



Master mode

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

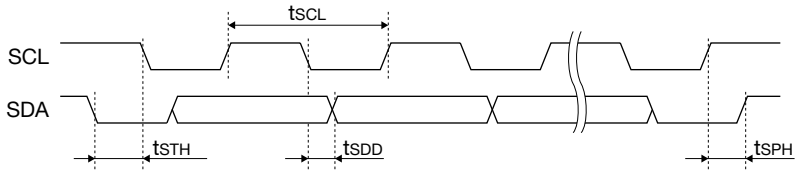
Item	Code	Min.	Typ.	Max.	Units
SPICLK cycle time	t_{SPCK}	500			ns
SDI setup time	t_{SDS}	70			ns
SDI hold time	t_{SDH}	10			ns
SDO output delay time	t_{SDO}			20	ns

Slave mode

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

Item	Code	Min.	Typ.	Max.	Units
SPICLK cycle time	t_{SPCK}	500			ns
SDI setup time	t_{SDS}	10			ns
SDI hold time	t_{SDH}	10			ns
SDO output delay time	t_{SDO}			80	ns

24.5.2 I²C AC Characteristics

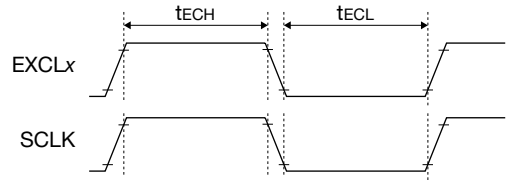


Unless otherwise specified: HV_{DD} = 1.65 to 3.6V, LV_{DD} = 1.65 to 2.7V, V_{SS} = 0V, Ta = -40 to 85°C

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

24.5.3 External Clock Input AC Characteristics



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

Item	Code	Min.	Typ.	Max.	Units
EXCL*x* input High pulse width	t_{ECH}	2/f_{SYS}			s
EXCL*x* input Low pulse width	t_{ECL}	2/f_{SYS}			s
UART transfer rate	R_U			460800	bps
UART transfer rate (IrDA mode)	R_{UIrDA}			115200	bps

∗f_{SYS}: System operation clock frequency

24.5.4 System AC Characteristics

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

Item	Code	Min.	Typ.	Max.	Units
Reset Low pulse width	t_{SR}	100			µs

24.6 Oscillation Characteristics

Oscillation characteristics vary depending on various parameters such as components used (oscillator, R_f, R_d, C_G, C_D) and circuit board patterns. The characteristics below should be used as reference values. For ceramic or crystal oscillators, select external resistors (R_f, R_d) and capacitors (C_G, C_D) only after fully evaluating the components when actually mounted on the circuit board.

OSC1 oscillator circuit (crystal oscillator)

Unless otherwise specified: HV_{DD} = 1.65 to 3.6V, LV_{DD} = 1.65 to 2.7V, V_{SS} = 0V, Ta = 25°C

Item	Code	Condition	Min.	Typ.	Max.	Units
Oscillation start time	t_{sta}				3	s

OSC3 oscillator circuit (crystal/ceramic oscillator)

Note: Use a crystal fundamental wave oscillator for the OSC3 crystal oscillator circuit.

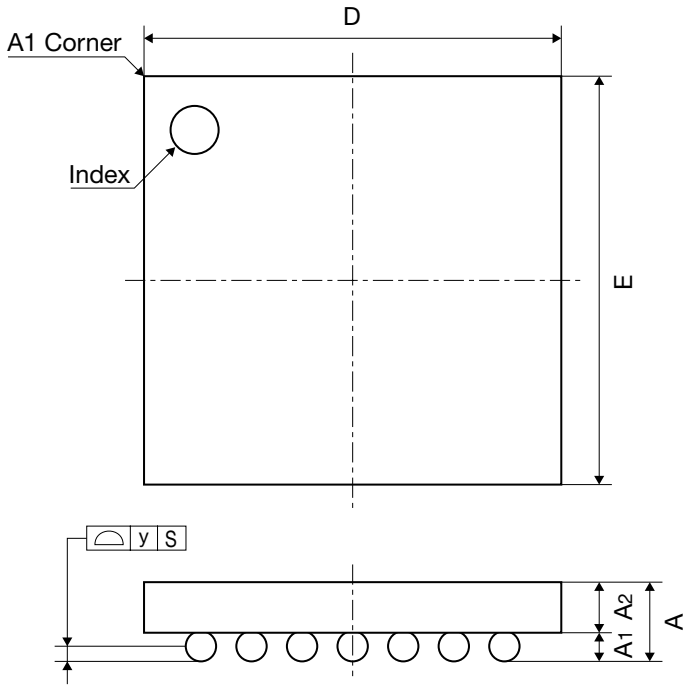
Unless otherwise specified: HV_{DD} = 1.65 to 3.6V, LV_{DD} = 1.65 to 2.7V, V_{SS} = 0V, Ta = 25°C

Item	Code	Condition	Min.	Typ.	Max.	Units
Oscillation start time	t_{sta}				25	ms

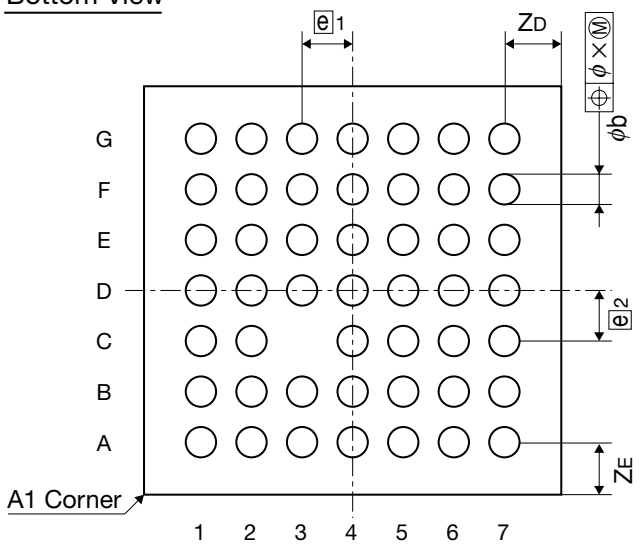
25 Package

WCSP-48pin

Top View



Bottom View



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	–
e1	–	0.40	–
e2	–	0.40	–
b	0.23	0.26	0.29
X	–	–	0.08
y	–	–	0.05
Z_D	–	0.362	–
Z_E	–	0.362	–

This page intentionally left blank.

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 to 0x403f	–	–	Reserved
UART (with IrDA) (8-bit device)	0x4100	UART_ST	UART Status Register	Transfer, buffer, error status display
	0x4101	UART_TXD	UART Transmit Data Register	Transmission data
	0x4102	UART_RXD	UART Receive Data Register	Receiving data
	0x4103	UART_MOD	UART Mode Register	Transfer data format setting
	0x4104	UART_CTL	UART Control Register	Data transfer control
	0x4105	UART_EXP	UART Expansion Register	IrDA mode setting
	0x4106 to 0x411f	–	–	Reserved
8-bit timer (with F mode) (16-bit device)	0x4200	T8F_CLK	8-bit Timer Input Clock Select Register	Prescaler output clock selection
	0x4202	T8F_TR	8-bit Timer Reload Data Register	Reload data setting
	0x4204	T8F_TC	8-bit Timer Counter Data Register	Counter data
	0x4206	T8F_CTL	8-bit Timer Control Register	Timer mode setting and timer RUN/STOP
	0x4208 to 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 to 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 to 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 to 0x427f	–	–	Reserved
Interrupt controller (16-bit device)	0x4300	ITC_IFLG	Interrupt Flag Register	Interrupt occurrence status display/reset
	0x4302	ITC_EN	Interrupt Enable Register	Maskable interrupt permission/prevention
	0x4304	ITC_CTL	ITC Control Register	ITC operation permission/prevention
	0x4306	ITC_ELV0	External Interrupt Level Setup Register 0	P0/P1 port interrupt level and trigger mode setting
	0x4308	ITC_ELV1	External Interrupt Level Setup Register 1	Stopwatch timer and clock timer interrupt level and trigger mode setting
	0x430a	ITC_ELV2	External Interrupt Level Setup Register 2	8-bit OSC1 timer interrupt level and trigger mode setting
	0x430c	ITC_ELV3	External Interrupt Level Setup Register 3	PWM & capture timer interrupt level and trigger mode setting
	0x430e	ITC_ILV0	Internal Interrupt Level Setup Register 0	8-bit timer and 16-bit timer Ch.0 interrupt level setting
	0x4310	ITC_ILV1	Internal Interrupt Level Setup Register 1	16-bit timer Ch.1 and 16-bit timer Ch.2 interrupt level setting
	0x4312	ITC_ILV2	Internal Interrupt Level Setup Register 2	UART and remote controller interrupt level setting
	0x4314	ITC_ILV3	Internal Interrupt Level Setup Register 3	SPI and I²C interrupt level setting
	0x4316 to 0x431f	–	–	Reserved
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 to 0x433f	–	–	Reserved
I²C (16-bit device)	0x4340	I2C_EN	I²C Enable Register	I²C module enable
	0x4342	I2C_CTL	I²C Control Register	I²C control and transfer status display
	0x4344	I2C_DAT	I²C Data Register	Transfer data
	0x4346	I2C_ICTL	I²C Interrupt Control Register	I²C interrupt control
	0x4348 to 0x435f	–	–	Reserved

Peripheral circuit	Address	Register name		Function
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 to 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 to 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 to 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 to 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 to 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 data
	0x50c4	T8OSC1_IFLG	8-bit OSC1 Timer Interrupt Flag Register	Interrupt occurrence status display/reset
	0x50c5 to 0x50df	–	–	Reserved
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_IO	P0 Port I/O Direction Control Register	P0 port input/output direction selection
	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 to 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_IO	P1 Port I/O Direction Control Register	P1 port input/output direction selection
	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 to 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_IO	P2 Port I/O Direction Control Register	P2 port input/output direction selection
	0x5223	P2_PU	P2 Port Pull-up Control Register	P2 port pull-up control
	0x5224 to 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_IO	P3 Port I/O Direction Control Register	P3 port input/output direction selection
	0x5233	P3_PU	P3 Port Pull-up Control Register	P3 port pull-up control
	0x5234 to 0x527f	–	–	Reserved
	0x52a0	P0_PMUX	P0 Port Function Select Register	P0 port function selection
	0x52a1	P1_PMUX	P1 Port Function Select Register	P1 port function selection
	0x52a2	P2_PMUX	P2 Port Function Select Register	P2 port function selection
	0x52a3	P3_PMUX	P3 Port Function Select Register	P3 port function selection
	0x52a4 to 0x52bf	–	–	Reserved

Peripheral circuit	Address	Register name		Function
PWM & capture timer (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 mask setting
	0x530c	T16E_IFLG	PWM Timer Interrupt Flag Register	Interrupt occurrence status display/reset
	0x530e to 0x531f	–	–	Reserved
MISC register (8-bit device)	0x5320	MISC_FL	ROM Control Register	ROM access condition setting
	0x5321	–	–	Reserved
	0x5322	MISC_OSC1	OSC1 Peripheral Control Register	OSC1 operation peripheral function setting for debugging
	0x5323 to 0x533f	–	–	Reserved
Remote controller (8-bit device)	0x5340	REMC_CFG	REMC Configuration Register	Transfer selection and permission
	0x5341	REMC_PSC	REMC Prescaler Clock Select Register	Prescaler output clock selection
	0x5342	REMC_CARH	REMC H Carrier Length Setup Register	Carrier H section length setting
	0x5343	REMC_CARL	REMC L Carrier Length Setup Register	Carrier L section length setting
	0x5344	REMC_ST	REMC Status Register	Transfer bit
	0x5345	REMC_LCNT	REMC Length Counter Register	Transfer data length setting
	0x5346	REMC_IMSK	REMC Interrupt Mask Register	Interrupt mask setting
	0x5347	REMC_IFLG	REMC Interrupt Flag Register	Interrupt occurrence status display/reset
	0x5348 to 0x535f	–	–	Reserved

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	

0x4100–0x4105 UART (with IrDA)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
UART Status Register (UART_ST)	**0x4100** (8 bits)	D7	**–**	reserved	–	–	–	0 when being read.
		D6	**FER**	Framing error flag	1 Error 0 Normal	0	R/W	Reset by writing 1.
		D5	**PER**	Parity error flag	1 Error 0 Normal	0	R/W	
		D4	**OER**	Overrun error flag	1 Error 0 Normal	0	R/W	
		D3	**RD2B**	Second byte receive flag	1 Ready 0 Empty	0	R	
		D2	**TRBS**	Transmit busy flag	1 Busy 0 Idle	0	R	Shift register status
		D1	**RDRY**	Receive data ready flag	1 Ready 0 Empty	0	R	
		D0	**TDBE**	Transmit data buffer empty flag	1 Empty 0 Not empty	1	R	
UART Transmit Data Register (UART_TXD)	**0x4101** (8 bits)	D7–0	**TXD[7:0]**	Transmit data TXD7(6) = MSB TXD0 = LSB	0x0 to 0xff (0x7f)	0x0	R/W	
UART Receive Data Register (UART_RXD)	**0x4102** (8 bits)	D7–0	**RXD[7:0]**	Receive data in the receive data buffer RXD7(6) = MSB RXD0 = LSB	0x0 to 0xff (0x7f)	0x0	R	Older data in the buffer is read out first.
UART Mode Register (UART_MOD)	**0x4103** (8 bits)	D7–5	**–**	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 Control Register (UART_CTL)	**0x4104** (8 bits)	D7	**–**	reserved	–	–	–	0 when being read.
		D6	**REIEN**	Receive error int. enable	1 Enable 0 Disable	0	R/W	
		D5	**RIEN**	Receive buffer full int. enable	1 Enable 0 Disable	0	R/W	
		D4	**TIEN**	Transmit buffer empty int. enable	1 Enable 0 Disable	0	R/W	
		D3–2	**–**	reserved	–	–	–	0 when being read.
		D1	**RBFI**	Receive buffer full int. condition	1 2 bytes 0 1 byte	0	R/W	
		D0	**RXEN**	UART enable	1 Enable 0 Disable	0	R/W	
UART Expansion Register (UART_EXP)	**0x4105** (8 bits)	D7	**–**	reserved	–	–	–	0 when being read.
		D6–4	**IRCLK[2:0]**	IrDA receive detection clock select	IRCLK[2:0] Clock 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	

0x4200–0x4206 8-bit Timer (with Fine Mode)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
8-bit Timer Input Clock Select Register (T8F_CLK)	**0x4200** (16 bits)	D15–4	**–**	reserved	–	–	–	0 when being read.
		D3–0	**DF[3:0]**	Timer input clock select (Prescaler output clock)	DF[3:0] / Clock 0xf reserved 0xe PCLK-1/16384 0xd PCLK-1/8192 0xc PCLK-1/4096 0xb PCLK-1/2048 0xa PCLK-1/1024 0x9 PCLK-1/512 0x8 PCLK-1/256 0x7 PCLK-1/128 0x6 PCLK-1/64 0x5 PCLK-1/32 0x4 PCLK-1/16 0x3 PCLK-1/8 0x2 PCLK-1/4 0x1 PCLK-1/2 0x0 PCLK-1/1	0x0	R/W	
8-bit Timer Reload Data Register (T8F_TR)	**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 Counter Data Register (T8F_TC)	**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	0x0	R	
8-bit Timer Control Register (T8F_CTL)	**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	

0x4220–0x4246 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	0x0	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.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	0x0	R	
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	

0x4260–0x4266 16-bit Timer

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
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	0x0	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	

0x4300–0x4310 Interrupt Controller

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Interrupt Flag Register (ITC_IFLG)	**0x4300** (16 bits)	D15	**IIFT7**	I²C interrupt flag	1 Cause of interrupt occurred / 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D14	**IIFT6**	SPI interrupt flag		0	R/W	
		D13	**IIFT5**	Remote controller interrupt flag		0	R/W	
		D12	**IIFT4**	UART interrupt flag		0	R/W	
		D11	**IIFT3**	16-bit timer Ch.2 interrupt flag		0	R/W	
		D10	**IIFT2**	16-bit timer Ch.1 interrupt flag		0	R/W	
		D9	**IIFT1**	16-bit timer Ch.0 interrupt flag		0	R/W	
		D8	**IIFT0**	8-bit timer interrupt flag		0	R/W	
		D7	**EIFT7**	PWM&capture timer interrupt flag	1 Cause of interrupt occurred / 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1 in pulse trigger mode. Cannot be reset by software in level trigger mode.
		D6	**EIFT6**	reserved		0	R/W	
		D5	**EIFT5**	reserved		0	R/W	
		D4	**EIFT4**	8-bit OSC1 timer interrupt flag		0	R/W	
		D3	**EIFT3**	Clock timer interrupt flag		0	R/W	
		D2	**EIFT2**	Stopwatch timer interrupt flag		0	R/W	
		D1	**EIFT1**	P1 port interrupt flag		0	R/W	
		D0	**EIFT0**	P0 port interrupt flag		0	R/W	
Interrupt Enable Register (ITC_EN)	**0x4302** (16 bits)	D15	**IIEN7**	I²C interrupt enable	1 Enable / 0 Disable	0	R/W	
		D14	**IIEN6**	SPI interrupt enable		0	R/W	
		D13	**IIEN5**	Remote controller interrupt enable		0	R/W	
		D12	**IIEN4**	UART interrupt enable		0	R/W	
		D11	**IIEN3**	16-bit timer Ch.2 interrupt enable		0	R/W	
		D10	**IIEN2**	16-bit timer Ch.1 interrupt enable		0	R/W	
		D9	**IIEN1**	16-bit timer Ch.0 interrupt enable		0	R/W	
		D8	**IIEN0**	8-bit timer interrupt enable		0	R/W	
		D7	**EIEN7**	PWM&capture timer interrupt enable		0	R/W	
		D6	**EIEN6**	reserved		0	R/W	
		D5	**EIEN5**	reserved		0	R/W	
		D4	**EIEN4**	8-bit OSC1 timer interrupt enable		0	R/W	
		D3	**EIEN3**	Clock timer interrupt enable		0	R/W	
		D2	**EIEN2**	Stopwatch timer interrupt enable		0	R/W	
		D1	**EIEN1**	P1 port interrupt enable		0	R/W	
		D0	**EIEN0**	P0 port interrupt enable		0	R/W	
ITC Control Register (ITC_CTL)	**0x4304** (16 bits)	D15–1	**–**	reserved	–	–	–	0 when being read.
		D0	**ITEN**	ITC enable	1 Enable / 0 Disable	0	R/W	
External Interrupt Level Setup Register 0 (ITC_ELV0)	**0x4306** (16 bits)	D15–13	**–**	reserved	–	–	–	0 when being read.
		D12	**EITG1**	P1 interrupt trigger mode	1 Level / 0 Pulse	0	R/W	Be sure to set to 1.
		D11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**EILV1[2:0]**	P1 interrupt level	0 to 7	0x0	R/W	
		D7–5	**–**	reserved	–	–	–	0 when being read.
		D4	**EITG0**	P0 interrupt trigger mode	1 Level / 0 Pulse	0	R/W	Be sure to set to 1.
		D3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**EILV0[2:0]**	P0 interrupt level	0 to 7	0x0	R/W	
External Interrupt Level Setup Register 1 (ITC_ELV1)	**0x4308** (16 bits)	D15–13	**–**	reserved	–	–	–	0 when being read.
		D12	**EITG3**	CT interrupt trigger mode	1 Level / 0 Pulse	0	R/W	Be sure to set to 1.
		D11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**EILV3[2:0]**	CT interrupt level	0 to 7	0x0	R/W	
		D7–5	**–**	reserved	–	–	–	0 when being read.
		D4	**EITG2**	SWT interrupt trigger mode	1 Level / 0 Pulse	0	R/W	Be sure to set to 1.
		D3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**EILV2[2:0]**	SWT interrupt level	0 to 7	0x0	R/W	
External Interrupt Level Setup Register 2 (ITC_ELV2)	**0x430a** (16 bits)	D15–5	**–**	reserved	–	–	–	0 when being read.
		D4	**EITG4**	T8OSC1 interrupt trigger mode	1 Level / 0 Pulse	0	R/W	Be sure to set to 1.
		D3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**EILV4[2:0]**	T8OSC1 interrupt level	0 to 7	0x0	R/W	
External Interrupt Level Setup Register 3 (ITC_ELV3)	**0x430c** (16 bits)	D15–13	**–**	reserved	–	–	–	0 when being read.
		D12	**EITG7**	T16E interrupt trigger mode	1 Level / 0 Pulse	0	R/W	Be sure to set to 1.
		D11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**EILV7[2:0]**	T16E interrupt level	0 to 7	0x0	R/W	
		D7–0	**–**	reserved	–	–	–	0 when being read.
Internal Interrupt Level Setup Register 0 (ITC_ILV0)	**0x430e** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**IILV1[2:0]**	T16 Ch.0 interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**IILV0[2:0]**	T8 interrupt level	0 to 7	0x0	R/W	
Internal Interrupt Level Setup Register 1 (ITC_ILV1)	**0x4310** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**IILV3[2:0]**	T16 Ch.2 interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**IILV2[2:0]**	T16 Ch.1 interrupt level	0 to 7	0x0	R/W	

0x4312–0x4314 Interrupt Controller

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Internal Interrupt Level Setup Register 2 (ITC_ILV2)	**0x4312** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**IILV5[2:0]**	REMC interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**IILV4[2:0]**	UART interrupt level	0 to 7	0x0	R/W	
Internal Interrupt Level Setup Register 3 (ITC_ILV3)	**0x4314** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**IILV7[2:0]**	I^2C interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**IILV6[2:0]**	SPI 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)	D15–6	**–**	reserved	–				–	–	0 when being read.
		D5	**SPRIE**	Receive data buffer full int. enable	1	Enable	0	Disable	0	R/W	
		D4	**SPTIE**	Transmit data buffer empty int. enable	1	Enable	0	Disable	0	R/W	
		D3	**CPHA**	Clock phase select	1	Data out	0	Data in	0	R/W	These bits must be set before setting SPEN to 1.
		D2	**CPOL**	Clock polarity select	1	Active L	0	Active H	0	R/W	
		D1	**MSSL**	Master/slave mode select	1	Master	0	Slave	0	R/W	
		D0	**SPEN**	SPI enable	1	Enable	0	Disable	0	R/W	

0x4340–0x4346 I²C

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	

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)	D7–1	**–**	reserved	–	–	–	0 when being read.
		D0	**CLKSRC**	System clock source select	1 OSC1 0 OSC3	0	R/W	
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	1	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	1	R/W	
FOUT Control Register (OSC_FOUT)	**0x5064** (8 bits)	D7–4	**–**	reserved	–	–	–	0 when being read.
		D3–2	**FOUT3D[1:0]**	FOUT3 clock division ratio select	FOUT3D[1:0] / Division ratio: 0x3 reserved; 0x2 OSC3-1/4; 0x1 OSC3-1/2; 0x0 OSC3-1/1	0x0	R/W	
		D1	**FOUT3E**	FOUT3 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	
		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	**CCLK-GR[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			

0x50c0–0x50c4 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	0xff	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.

0x5200–0x5216 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 I/O Direction Control Register (P0_IO)	**0x5202** (8 bits)	D7–0	**P0IO[7:0]**	P0[7:0] port I/O direction select	1: Output 0: Input	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	
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 I/O Direction Control Register (P1_IO)	**0x5212** (8 bits)	D7–0	**P1IO[7:0]**	P1[7:0] port I/O direction select	1: Output 0: Input	0	R/W	
P1 Port Pull-up Control Register (P1_PU)	**0x5213** (8 bits)	D7–0	**P1PU[7:0]**	P1[7:0] port pull-up enable	1: Enable 0: Disable	1 (0xff)	R/W	
P1 Port 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	

0x5217–0x52a3 P Port & Port MUX

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
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.
P2 Port Input Data Register (P2_IN)	0x5220 (8 bits)	D7–0	P2IN[7:0]	P2[7:0] port input data	1	1 (H)	0	0 (L)	×	R	
P2 Port Output Data Register (P2_OUT)	0x5221 (8 bits)	D7–0	P2OUT[7:0]	P2[7:0] port output data	1	1 (H)	0	0 (L)	0	R/W	
P2 Port I/O Direction Control Register (P2_IO)	0x5222 (8 bits)	D7–0	P2IO[7:0]	P2[7:0] port I/O direction select	1	Output	0	Input	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	
P3 Port Input Data Register (P3_IN)	0x5230 (8 bits)	D7–4	–	reserved	–				–	–	0 when being read.
		D3–0	P3IN[3:0]	P3[3:0] port input data	1	1 (H)	0	0 (L)	×	R	
P3 Port Output Data Register (P3_OUT)	0x5231 (8 bits)	D7–4	–	reserved	–				–	–	0 when being read.
		D3–0	P3OUT[3:0]	P3[3:0] port output data	1	1 (H)	0	0 (L)	0	R/W	
P3 Port I/O Direction Control Register (P3_IO)	0x5232 (8 bits)	D7–4	–	reserved	–				–	–	0 when being read.
		D3–0	P3IO[3:0]	P3[3:0] port I/O direction select	1	Output	0	Input	0	R/W	
P3 Port Pull-up Control Register (P3_PU)	0x5233 (8 bits)	D7–4	–	reserved	–				–	–	0 when being read.
		D3–0	P3PU[3:0]	P3[3:0] port pull-up enable	1	Enable	0	Disable	1 (0xff)	R/W	
P0 Port Function Select Register (P0_PMUX)	0x52a0 (8 bits)	D7–6	–	reserved	–				–	–	0 when being read.
		D5	P05MUX	P05 port function select	1	REMO	0	P05	0	R/W	
		D4	P04MUX	P04 port function select	1	REMI	0	P04	0	R/W	
		D3–0	–	reserved	–				–	–	0 when being read.
P1 Port Function Select Register (P1_PMUX)	0x52a1 (8 bits)	D7	P17MUX	P17 port function select	1	#SPISS	0	P17	0	R/W	
		D6	–	reserved	–				–	–	0 when being read.
		D5	P15MUX	P15 port function select	1	SCL	0	P15	0	R/W	
		D4	P14MUX	P14 port function select	1	SDA	0	P14	0	R/W	
		D3	P13MUX	P13 port function select	1	FOUT1	0	P13	0	R/W	
		D2–0	–	reserved	–				–	–	0 when being read.
P2 Port Function Select Register (P2_PMUX)	0x52a2 (8 bits)	D7	P27MUX	P27 port function select	1	EXCL3	0	P27	0	R/W	
		D6	P26MUX	P26 port function select	1	TOUT	0	P26	0	R/W	
		D5	P25MUX	P25 port function select	1	SCLK	0	P25	0	R/W	
		D4	P24MUX	P24 port function select	1	SOUT	0	P24	0	R/W	
		D3	P23MUX	P23 port function select	1	SIN	0	P23	0	R/W	
		D2	P22MUX	P22 port function select	1	SPICLK	0	P22	0	R/W	
		D1	P21MUX	P21 port function select	1	SDO	0	P21	0	R/W	
		D0	P20MUX	P20 port function select	1	SDI	0	P20	0	R/W	
P3 Port Function Select Register (P3_PMUX)	0x52a3 (8 bits)	D7–4	–	reserved	–				–	–	0 when being read.
		D3	P33MUX	P33 port function select	1	P33	0	DSIO	0	R/W	
		D2	P32MUX	P32 port function select	1	P32	0	DST2	0	R/W	
		D1	P31MUX	P31 port function select	1	P31	0	DCLK	0	R/W	
		D0	P30MUX	P30 port function select	1	FOUT3	0	P30	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**	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	
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 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	

0x5320–0x5322 MISC Registers

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
ROM Control Register (MISC_FL)	**0x5320** (8 bits)	D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**FLCYC[2:0]**	ROM 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** (8 bits)	D7–1	**–**	reserved	–	–	–	0 when being read.
		D0	**O1DBG**	OSC1 peripheral control in debug mode	1 Run 0 Stop	0	R/W	

0x5340–0x5347 Remote Controller

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
REMC Configuration Register (REMC_CFG)	**0x5340** (8 bits)	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 Prescaler Clock Select Register (REMC_PSC)	**0x5341** (8 bits)	D7–4	**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	
		D3–0	**LCCLK[3:0]**	Length counter clock select (Prescaler output clock)		0x0	R/W	
REMC H Carrier Length Setup Register (REMC_CARH)	**0x5342** (8 bits)	D7–6	**–**	reserved	–	–	–	0 when being read.
		D5–0	**REMCH[5:0]**	H carrier length setup	0x0 to 0x3f	0x0	R/W	
REMC L Carrier Length Setup Register (REMC_CARL)	**0x5343** (8 bits)	D7–6	**–**	reserved	–	–	–	0 when being read.
		D5–0	**REMCL[5:0]**	L carrier length setup	0x0 to 0x3f	0x0	R/W	
REMC Status Register (REMC_ST)	**0x5344** (8 bits)	D7–1	**–**	reserved	–	–	–	0 when being read.
		D0	**REMDT**	Transmit/receive data	1 1 (H) / 0 0 (L)	0	R/W	
REMC Length Counter Register (REMC_LCNT)	**0x5345** (8 bits)	D7–0	**REMLEN[7:0]**	Transmit/receive data length count (down counter)	0x0 to 0xff	0x0	R/W	
REMC Interrupt Mask Register (REMC_IMSK)	**0x5346** (8 bits)	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	
REMC Interrupt Flag Register (REMC_IFLG)	**0x5347** (8 bits)	D7–3	**–**	reserved	–	–	–	0 when being read.
		D2	**REMFIF**	Falling edge interrupt flag	1 Cause of interrupt occurred / 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D1	**REMRIF**	Rising edge interrupt flag		0	R/W	
		D0	**REMUIF**	Underflow interrupt flag		0	R/W	

0xffff80–0xffff90 S1C17 Core I/O

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Vector Table Base Register (TTBR)	**0xffff80** (32 bits)	D31–24	**–**	Unused (fixed at 0)	0x0	0x0	R	
		D23–0	**TTBR[23:0]**	Vector table base address	0x8000	0x8000	R	
Processor ID Register (IDIR)	**0xffff84** (8 bits)	D7–0	**IDIR[7:0]**	Processor ID 0x10: S1C17 Core	0x10	0x10	R	
Debug RAM Base Register (DBRAM)	**0xffff90** (32 bits)	D31–24	**–**	Unused (fixed at 0)	0x0	0x0	R	
		D23–0	**DBRAM[23:0]**	Debug RAM base address	0x7c0	0x7c0	R	

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 S1C17001 clock system.

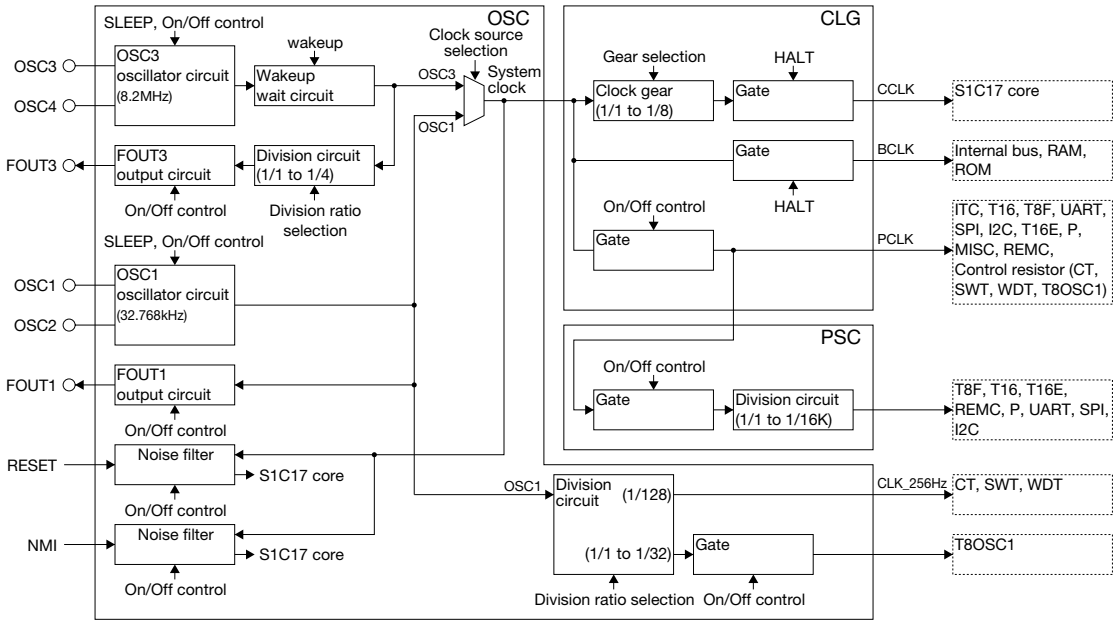


Figure B.1.1 Clock system

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` command
  Execute the `slp` command 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)
  Operate the oscillation circuit comprising the system clock source. Where possible, stop the other circuit. You can reduce current consumption by using OSC1 as the system clock and stopping the OSC3 oscillation circuit.

CPU clock (CCLK)

- Execute the `halt` command
  Execute the `halt` command 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` command, 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 command. 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
  - Interrupt controller
  - SPI
  - I^2C
  - P port and port MUX (control register, chattering filter)
  - PWM & capture timer
  - MISC register
  - Remote controller

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

Maskable interrupts do not occur while PCLK is stopped. Interrupts are retained until the PCLK feed is restarted.

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** command	1
	Oscillation (system CLK)	Stop	Stop	Stop	Operation	Execute **halt** command	1, 2
	Oscillation (system CLK)	Stop	Stop	Operation	Operation	Execute **halt** command	1, 2, 3
	Oscillation	Stop	Operation(1/1)	Operation	Operation		
	Oscillation	Oscillation (system CLK)	Stop	Operation	Operation	Execute **halt** command	1, 2, 3
	Oscillation	Oscillation (system CLK)	Operation (Low gear)	Operation	Operation		
High ↓	Oscillation	Oscillation (system CLK)	Operation(1/1)	Operation	Operation		

Canceling HALT and SLEEP modes (CPU startup methods)

1. Port startup
   Started by input/output port interrupt factors or debug interrupts (ICD forced breaking). If the interrupt controller or CPU IE flag blocks input/output port interrupts, the CPU executes commands following the **halt** or **slp** commands without accepting the interrupt. If interrupts are permitted and PCLK was running before the **halt** or **slp** commands were executed, the CPU branches to an interrupt processing routine. If PCLK was stopped before the **halt** or **slp** commands were executed, branching to the interrupt processing routine is retained until PCLK runs, even when interrupts are permitted.
2. OSC1 peripheral circuit startup
   Started by clock timer, stopwatch timer, watchdog timer, or 8-bit OSC1 timer interrupt factors. If the interrupt controller or CPU IE flag blocks these interrupts, the CPU executes commands following the **halt** command without accepting the interrupt. If interrupts are permitted and PCLK was running before the **halt** command was executed, the CPU branches to an interrupt processing routine. If PCLK was stopped before the **halt** command was executed, branching to the interrupt processing routine is retained until PCLK runs, even when interrupts are permitted.
3. PCLK peripheral circuit startup
   Started by PCLK peripheral circuit interrupt factors permitted by the interrupt controller. If the CPU IE flag is 0, the CPU executes commands following the **halt** command, rejecting the interrupt. If the IE flag is 1, the CPU branches to an interrupt processing routine.

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).
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 FOUT3 pins.
You can check the quality of the OSC3 output waveform via the FOUT3 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

Sudden noise-induced power supply fluctuations will cause malfunctions. Consider the following precautions to avoid such malfunctions.

(1) Form connections from the power supply to LVDD, HVDD, and the VSS pin with the shortest, thickest patterns possible.

(2) If a bypass capacitor is connected between LVDD/HVDD and VSS, the connection between the LVDD/HVDD pins and the VSS pin should be as short as possible.

Signal line location

- To prevent electromagnetically-induced noise caused by mutual inductance, avoid locating large-current signal lines near circuits susceptible to noise (e.g., clock inputs).
- Locating signal lines in parallel over significant distances or crossing high-speed signal lines will result in mutual interference, noise, and malfunctions.
  In particular, avoid positioning high-speed signal lines close to circuits susceptible to noise (e.g., clock input components).

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 HVDD 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) LVDD, HVDD, VSS 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 mm.

(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).

Miscellaneous

This product series is manufactured by a 0.25 μm microscopic process.
The design is intended to ensure basic IC reliability meeting EIAJ and MIL standards. Nevertheless, keep in mind the following precautions when mounting the product.
In addition to physical damage, mounting the product may result in electrical damage caused by voltages exceeding the absolute maximum rating (2.5 V) with random fluctuations over time:

(1) electromagnetically-induced noise from industrial power supplies used in mounting reflow processes, reworking after mounting, and individual characteristic evaluation (testing) processes;
(2) electromagnetically-induced noise generated by solder irons during soldering.

In particular, the soldering iron GND (tip potential) must be at the same potential as the IC GND during soldering.

Appendix D: Initialization Routine

This section lists typical vector tables and initialization routines.

boot.s

```
.org    0x8000
.section .rodata                                                          ...(1)

; ======================================================================
;       Vector table
; ======================================================================
                        ; interrupt  vector  interrupt
                        ; number     offset  source

.long BOOT              ; 0x00       0x00    reset                        ...(2)
.long unalign_handler   ; 0x01       0x04    unalign
.long nmi_handler       ; 0x02       0x08    NMI
.long int03_handler     ; 0x03       0x0c    -
.long p0_handler        ; 0x04       0x10    P0 port
.long p1_handler        ; 0x05       0x14    P1 port
.long swt_handler       ; 0x06       0x18    SWT
.long ct_handler        ; 0x07       0x1c    CT
.long t8osc1_handler    ; 0x08       0x20    T8OSC1
.long int09_handler     ; 0x09       0x24    -
.long int0a_handler     ; 0x0a       0x28    -
.long t16e_handler      ; 0x0b       0x2c    T16E
.long t8f_handler       ; 0x0c       0x30    T8F
.long t16_0_handler     ; 0x0d       0x34    T16 ch0
.long t16_1_handler     ; 0x0e       0x38    T16 ch1
.long t16_2_handler     ; 0x0f       0x3c    T16 ch2
.long uart_handler      ; 0x10       0x40    UART
.long remc_handler      ; 0x11       0x44    REMC
.long spi_handler       ; 0x12       0x48    SPI
.long i2c_handler       ; 0x13       0x4c    I2C
.long int14_handler     ; 0x14       0x50    -
.long int15_handler     ; 0x15       0x54    -
.long int16_handler     ; 0x16       0x58    -
.long int17_handler     ; 0x17       0x5c    -
.long int18_handler     ; 0x18       0x60    -
.long int19_handler     ; 0x19       0x64    -
.long int1a_handler     ; 0x1a       0x68    -
.long int1b_handler     ; 0x1b       0x6c    -
.long int1c_handler     ; 0x1c       0x70    -
.long int1d_handler     ; 0x1d       0x74    -
.long int1e_handler     ; 0x1e       0x78    -
.long int1f_handler     ; 0x1f       0x7c    -

; ======================================================================
;       Program code
; ======================================================================

.text                                                                     ...(3)
.align 1

BOOT:
        ; ===== Initialize ============================================

        ; ----- Stack pointer --------------------
        Xld.a   %sp, 0x7c0                                                ...(4)

        ; ----- Memory controller ----------------
        Xld.a   %r1, 0x5320     ; MISC register base address

        ; ROM
        Xld.a   %r0, 0x04       ; 1 cycle access
        ld.b    [%r1], %r0      ; [0x5320] <= 0x04                        ...(5)
```

```
        ; ----- ITC (interrupt controller) ------
        Xld.a   %r7, 0x4300     ; ITC register base address

        Xld.a   %r0, 0x1010     ; P0, P1 interrupt level & trigger mode
        ext     0x06
        ld      [%r7], %r0      ; [0x4306] <= 0x1010                        ...(6)

        Xld.a   %r0, 0x1010     ; SWT, CT interrupt level & trigger mode
        ext     0x08
        ld      [%r7], %r0      ; [0x4308] <= 0x1010                        ...(6)

        Xld.a   %r0, 0x0010     ; T8OSC1 interrupt level & trigger mode
        ext     0x0a
        ld      [%r7], %r0      ; [0x430a] <= 0x0010                        ...(6)

        Xld.a   %r0, 0x1000     ; T16E interrupt level & trigger mode
        ext     0x0c
        ld      [%r7], %r0      ; [0x430c] <= 0x1000                        ...(6)

        ; ===== Main routine ========================================
      ...
; ========================================================================
;       Interrupt handler
; ========================================================================

; ----- Address unalign --------------------------
unalign_handler:
        ...

; ----- NMI --------------------------------------
nmi_handler:
        ...
```

(1) `.rodata` section is declared to position vector table in `.vector` section.

(2) Interrupt processing routine address is defined as vector.
`Int`*`XX`*`_handler` can be used as software interrupt.

(3) Program code is written in `.text` section.

(4) Sets stack pointer.

(5) Sets ROM read access cycles.
(See "3 Memory Map and Bus Control.")

(6) Sets interrupt trigger mode to level trigger for the following peripheral circuits.
P0 port, P1 port, stopwatch timer, clock timer, 8-bit OSC1 timer, PWM & capture timer
(See "6 Interrupt Controller (ITC).")

Appendix E: S1C17001 Mask ROM Code Development

(1) The S1C17001 mask ROM code was developed using the S1C17704 Flash microprocessor.

(2) ROM data is provided to Epson in "*file*.PAn" (winmdc output) format. Final user verification of ROM data should use "*file*.psa" (sconv32 output) format.

(3) Set the following arguments when executing moto2ff in the S1C17001.
- Data start address = 8,000
- Data block size = 8,000

(4) Use S1C17704 to check operations, confirming that the following functional differences are fully understood:
- The ROM size differs. Remove 32 KB of code in step (3) for final checking using "*file*.psa" (sconv 32 output).
- The RAM size differs. Note the stack pointer addresses. Confirm that the data in addresses 0x800 to 0xfef has not changed after checking program operations.
- Do not access LCD driver and SVD related registers that do not exist in the S1C17001.

Table E.1: Comparison of functions between S1C17704 and S1C17001.

Circuit/function	S1C17704	S1C17001
Flash	64 Kbytes	None
ROM (Mask ROM)	None	32 Kbytes
RAM	4 Kbytes	2 Kbytes
Display RAM	572 bytes	None
Operating frequency	32kHz to 8.2MHz	←
OSC3 oscillator circuit	Crystal/CR	←
OSC1 oscillator circuit	Crystal	←
Operation clock input (OSC3)	For evaluation use only	○
Operation clock input (OSC1)	For evaluation use only	○
Input/output port	28	←
External port interrupt	16	←
SPI (master/slave)	1ch.	←
I²C (master)	1ch.	←
UART (IrDA1.0 compatible)	1ch.	←
8-bit timer (T8F)	1ch.	←
16-bit timer (T16)	3ch.	←
PWM & capture timer (T16E)	1ch.	←
Clock timer (CT)	1ch.	←
Stopwatch timer (SWT)	1ch.	←
Watchdog timer (WDT)	1ch.	←
8-bit OSC1 timer (T8OSC1)	1ch.	←
LCD driver	○	None
Power supply voltage detection (SVD) circuit	○	None
Power supply voltage	1.8V to 3.6V	Core voltage (LVDD): 1.65 to 2.7V I/O voltage (HVDD): 1.65 to 3.6V

Appendix F: Revision History

Rev. No.	Date	Page	Section	Details
0.8	2007.9.21	1-3	1.3.1 Pinout Diagram	Figure 1.3.1.1 changed
		1-4	1.3.2 Pin Descriptions	Table 1.3.2.1 changed
		23-1	23 Basic External Connection Diagram	Figure changed and added
		AP-31	Appendix D: Initialization Routine	Sample program changed
0.9	2007.10.28	1-2	1-2 Block Diagram	Figure 1.2.1 changed
		2-1	2.1 S1C17 Core Features	Description changed "It features low power consumption, … are widely used."
		2-3 to 2-5	2.3 Command Set	Table 2.3.1 changed Note (*2) added
		2-7	2.4 Vector Table	Table 2.4.1 changed
		3-1	3 Memory Map and Bus Control	Figure 3.1 changed
		3-2	3.1 Bus Cycle	Description changed "Interrupt processing stack operation involves reading … in the last 24 bits."
		3-2	3.1.2 Command Execution Cycle Restrictions	Section changed
		3-3	3.2 Internal ROM Area	Section title changed
		3-4	3.3 Internal RAM Area	Section title changed
		3-5	3.4 Internal Peripheral Circuit Area	Description changed "The 1 Kbyte area starting at address 0x4000 … and control registers."
		5-3	5.2 Initial Reset Sequence	Description changed "CPU startup waits for … OSC3 clock frequency"
		6-1	6.1 ITC Configuration	Figure 6.1.1 changed
		6-2	6.2 Vector Table	Table 6.2.1 changed
		6-3	6.3.2 Interrupt Request from Peripheral Module and Interrupt Flag	Table 6.3.2.1 changed
		6-8	6.3.6 S1C17 Core Interrupt Processing	Description added "- ITEN … has been set to 1."
		6-11	6.6 HALT and SLEEP Mode Cancellation by Interrupt Factors	Section title changed Description changed "HALT and SLEEP modes are … in Appendix B."
		6-13	6.7 Control Register Details	Table 6.7.2 changed
		6-15		Table 6.7.4 changed
		7-1	7.1 OSC Module Configuration	Figure 7.1.1 changed
		8-1	8.1 Clock Generator Configuration	Figure 8.1.1 changed
		10-1	10.1 Input/Output Port Configuration	Description (note) changed "Switch on the … using this function."
		10-2	10.2 Input/Output Port Pin Function Selection (Port MUX)	Description changed "Resetting the … in Table 10.2.1)."
		10-3	10.3 Data Input/Output	Description changed "The input/output ports … PxIO[7:0] (Px_IO register)."
		10-4	10.4 Pull-up Control	Description changed "The input/output port contains a pull-up resistor, … (Px_PU register)."
		10-7	10.7 Port Input Interrupt	Description changed "The interrupt controller … Interrupt Controller (ITC)."
		10-24	10.9 Precautions	Description added Pull-up "- Input/output ports that are not used … enabled."
		11-3	11.2.2 External Clock Mode	Description changed "For instructions on controlling chattering filter circuits, … Function."
		11-8	11.6 16-bit Timer RUN/STOP Control	Description changed (one location only) ITC → interrupt controller (ITC)
		11-10	11.8 16-bit Timer Interrupts	Description changed (one location only) ITC → interrupt controller (ITC)
		12-2	12.2 8-bit Timer Count Mode	Description added "Note: Make sure … count mode settings."
		12-6	12.6 8-bit Timer RUN/STOP Control	Description changed (one location only) ITC → interrupt controller (ITC)
		12-9	12.9 8-bit Timer Interrupts	Description changed (one location only) ITC → interrupt controller (ITC)
		12-15	12.10 Control Register Details	Description added (0x4206, D4) "Note: Make sure … count mode settings."
		13-4	13.4 Compare Data Settings	Counter reset cycle calculation equation added
		13-5	13.5 PWM & Capture Timer RUN/STOP Control	Description changed (one location only) ITC → interrupt controller (ITC)

Rev. No.	Date	Page	Section	Details
0.9	2007.10.28	13-8	13.6 Clock Output Control	Description changed "If the counter data register … A bit 0 (T16ECA0) value." ------ "Precautions (1) Compare data … (High when INVOUT = 1)."
		13-9	13.7 PWM & Capture Timer Interrupts	Description changed (one location only) ITC → interrupt controller (ITC)
		13-20	13.9 Precautions	Description changed "- Setting compare data with A > B … (High when INVOUT = 1)."
		14-2	14.2 8-bit OSC1 Timer Count Mode	Description added "Note: Make sure… count mode settings."
		14-5	14.5 Compare Data Settings	Compare match cycle calculation equation added
		14-6	14.6 8-bit OSC1 Timer RUN/STOP Control	Description changed (one location only) ITC → interrupt controller (ITC)
		14-7	14.7 8-bit OSC1 Timer Interrupts	Description changed (one location only) ITC → interrupt controller (ITC)
		14-10	14.8 Control Register Details	Description added (0x50c0, D1) "Note: Set the count mode … stopped."
		15-4	15.4 Clock Timer RUN/STOP Control	Description changed (one location only) ITC → interrupt controller (ITC)
		15-5	15.5 Clock Timer Interrupts	Description changed (one location only) ITC → interrupt controller (ITC)
		16-5	16.5 Stopwatch Timer RUN/STOP Control	Description changed (one location only) ITC → interrupt controller (ITC)
		16-6	16.6 Stopwatch Timer Interrupts	Description changed (one location only) ITC → interrupt controller (ITC)
		18-9	18.7 UART Interrupts	Description changed (one location only) ITC → interrupt controller (ITC)
		19-8	19.6 SPI Interrupts	Description changed (one location only) ITC → interrupt controller (ITC)
		20-1, 20-11, 20-18 to 20-19	20 I^2C	Interrupt name changed Receive interrupt à Receive buffer full interrupt Transmit interrupt à Transmit buffer empty interrupt
		20-11	20.6 I^2C Interrupts	Description changed (one location only) ITC → interrupt controller (ITC)
		21-3	21.3 Carrier Generation	Carrier H/L section length calculation equation added
		21-6	21.5 Data Transfer Control	Description changed (one location only) ITC → interrupt controller (ITC)
		21-8	21.6 REMC Interrupts	Description changed (one location only) ITC → interrupt controller (ITC)
		21-13	21.7 Control Register Details	Carrier H section length calculation equation added
		21-14		Carrier L section length calculation equation added
		22-1	22.1 Resource Requirements and Debugging Tool	ICD name changed ICD Mini → ICD (S5U1C17001H etc.)
		22-2	22.2 Debug Break Operation Status	Description added "The 8-bit OSC1 timer… even if O1DBG is set to 1."
		22-4	22.3 Control Register Details	Description added "The 8-bit OSC1 timer … even if O1DBG is set to 1."
		23-1 to 23-2	23 Basic External Connection Diagram	Table changed
		24-1	24.2 Recommended Operating Conditions	Table changed
		24-2 to 24-3	24.5 AC Characteristics	Sections 24.5.1 to 24.5.4 and tables added
		24-6 to 24-7		Graph (TBD) added
		AP-23	Appendix A: I/O Register List	Error corrected 0x5340-0x5347 Remote Controller
		AP-25	B.1 Clock Control Power Saving	Figure B.1.1 changed
		AP-26		Description changed "- Execute slp command … to startup using ports (described later)."
		AP-27		Table B.1.1 and description changed "Canceling HALT and SLEEP modes … processing routine."
1.0	2008.1.30	1-1	1.1 Features	TBD section fixed
		7-2	7.2 OSC3 Oscillator Circuit	Figure 7.2.1 changed Description added "A drain resistor (R_d) should … if required."
		7.3	7.3 OSC1 Oscillator Circuit	Figure 7.3.1 changed Description added "A drain resistor (R_d) should … if required."
		11-3	11.2.2 External Clock Mode	(TBD) deleted

Rev. No.	Date	Page	Section	Details
1.0	2008.1.30	11-6	11.4 16-bit Timer Reload Register and Underflow Cycle	TBD section fixed
		11-9	11.7 16-bit Timer Output Signal	TBD section fixed
		12-4	12.4 8-bit Timer Reload Register and Underflow Cycle	TBD section fixed
		12-8	12.8 Fine Mode	(TBD) deleted
		12-14	12.10 Control Register Details 0x4206	(TBD) deleted
		17-3	17.3.3 Watchdog Timer Resetting	(TBD) deleted
			17.4 Control Register Details	(TBD) deleted
		23-1	23 Basic External Connection Diagram	Figure and table changed
		24-1 to 24-6	24 Electrical Characteristics	Characteristics tables changed Characteristics graph deleted
	2008.2.8	1-1	1 Overview	Description deleted “On-chip ICE function ...”
		1-1	1.1 Features	Description changed • Main (OSC3) oscillator circuit • Sub (OSC1) oscillator circuit • Serial interface Description added • Mask ROM code development Flash memory
		1-4	1.3.2 Pin Descriptions	Table 1.3.2.1 changed
		7-2	7.2 OSC3 Oscillator Circuit	Figure 7.2.1 changed Description added “When used with external clock … input to the OSC3 pin.”
		7-4	7.3 OSC1 Oscillator Circuit	Figure 7.3.1 changed Description added “When used with external clock … input to the OSC1 pin.”
		12-4	12.4 8-bit Timer Reload Register and Underflow Cycle	(TBD) deleted
		18-1	18.1 UART Configuration	Description changed “The UART transfers … 150 to 460,800 bps (115,200 bps in IrDA mode).”
		18-21	18.10 Precautions	Description changed “The UART transfer … 460,800 bps (115,200 bps in IrDA mode).”
		24-1	24.2 Recommended Operating Conditions	Table changed (operating frequency added)
		24-3	24.4 Consumption Current	Description changed “*1: Current consumption … and memory writing 10.5%.”
		24-5	24.5.3 External Clock Input AC Characteristics	Table changed (UART transfer rate)
		AP-31	Appendix D: Initialization Routine	Sample code comment deleted (under 3.3 MHZ system clock)
		AP-32		Description deleted (System clock is 3.3 MHz ...)
		AP-33	Appendix E: S1C17001 Mask ROM Code Development	Section added

Revision History

Code No.	Page	Contents
411412301	4, 5, 6	Descriptions added. Package, Pin Descriptions
	11	Descriptions modified. Branch ipa.d→jpa.d
	63, 64	Figure 7.6.2 , Figure 7.6.3 modified.
	218	Descriptions modified. (2) RDRY = 1, RD2B = 0...cannot be read. The contents of the receive data buffer must be read out before an overrun error occurs.
	219, 231, 233	Descriptions modified. Setting the RXEN bit to 0 empties the transmission data buffer, clearing any remaining data. Preventing transfers by writing 0 to RXEN also clears transmit data buffer. •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.
	220	Descriptions modified. 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...will not be sent to the buffer and generate an overrun error.
	226	Descriptions modified. FER is reset by writing 1....PER is reset by writing 1....OER is reset by writing 1.
	237	Descriptions deleted. Since the internal circuit operates in sync with the PCLK...synchronize the differentiated PCLK clock. Descriptions modified. Note: The duty ratio of the clock input via the SPICLK pin must be 50%. Figure 19.3.2 deleted.
	238	Descriptions added. Note: When the SPI module is used in...during continuous transfer. Figure 19.4.2 added.
	240, 241	Figure 19.5.1, Figure 19.5.2 deleted. Figure 19.5.1 added.
	241	Descriptions modified. After a data transfer is completed (both transmission and reception)...be guaranteed if SPEN is set to 0 while data is being sent or received.
	246	Descriptions added. Note: Make sure that SPEN is set to 1 before writing data to the...to start data transmission/reception.
	255	Descriptions modified. (1)Register setting procedure... control. Figure 20.5.2 gives the configuration of the address data.
	256	Figure 20.5.2 modified.
	257	Descriptions modified. In the 9th clock cycle, 0 or 1 set by ...response time, the correct communication can not be performed.
	258	Descriptions modified. After the stop condition has been generated... and transfer data at that point cannot be guaranteed.
	259	Figure 20.5.5 modified.
	260	Figure 20.5.7, Figure 20.5.8 modified.
	261	Figure 20.5.9 modified.
	262	Descriptions modified. Transmit buffer empty interrupt...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.
	266	Descriptions modified. To generate a START condition, set the following registers...3. Set STRT (D0/I2C_CTL register) to 1.
	303	EXCLx input High pulse width 1/fSYS→2/fSYS EXCLx input Low pulse width 1/fSYS→2/fSYS

EPSON

International Sales Operations

AMERICA

EPSON ELECTRONICS AMERICA, INC.
2580 Orchard Parkway,
San Jose, CA 95131, USA
Phone: +1-800-228-3964 Fax: +1-408-922-0238

EUROPE

EPSON EUROPE ELECTRONICS GmbH
Riesstrasse 15, 80992 Munich,
GERMANY
Phone: +49-89-14005-0 Fax: +49-89-14005-110

ASIA

EPSON (CHINA) CO., LTD.
7F, Jinbao Bldg., No.89 Jinbao St.,
Dongcheng District,
Beijing 100005, CHINA
Phone: +86-10-8522-1199 Fax: +86-10-8522-1125

SHANGHAI BRANCH
7F, Block B, Hi-Tech Bldg., 900 Yishan Road,
Shanghai 200233, CHINA
Phone: +86-21-5423-5577 Fax: +86-21-5423-4677

SHENZHEN BRANCH
12F, Dawning Mansion, Keji South 12th Road,
Hi-Tech Park, Shenzhen 518057, CHINA
Phone: +86-755-2699-3828 Fax: +86-755-2699-3838

EPSON HONG KONG LTD.
20/F, Harbour Centre, 25 Harbour Road,
Wanchai, Hong Kong
Phone: +852-2585-4600 Fax: +852-2827-4346
Telex: 65542 EPSCO HX

EPSON TAIWAN TECHNOLOGY & TRADING LTD.
14F, No. 7, Song Ren Road,
Taipei 110, TAIWAN
Phone: +886-2-8786-6688 Fax: +886-2-8786-6660

EPSON SINGAPORE PTE., LTD.
1 HarbourFront Place,
#03-02 HarbourFront Tower One, Singapore 098633
Phone: +65-6586-5500 Fax: +65-6271-3182

SEIKO EPSON CORP.
KOREA OFFICE
5F, KLI 63 Bldg., 60 Yoido-dong,
Youngdeungpo-Ku, Seoul 150-763, KOREA
Phone: +82-2-784-6027 Fax: +82-2-767-3677

SEIKO EPSON CORP.
MICRODEVICES OPERATIONS DIVISION

Device Sales & Marketing Dept.
421-8, Hino, Hino-shi, Tokyo 191-8501, JAPAN
Phone: +81-42-587-5814 Fax: +81-42-587-5117

Document Code: 411412303
First Issue April 2008
Revised March 2011Ⓗ