EPSON
EXCEED YOUR VISION

CMOS 16-BIT SINGLE CHIP MICROCONTROLLER

S1C17651
Technical Manual

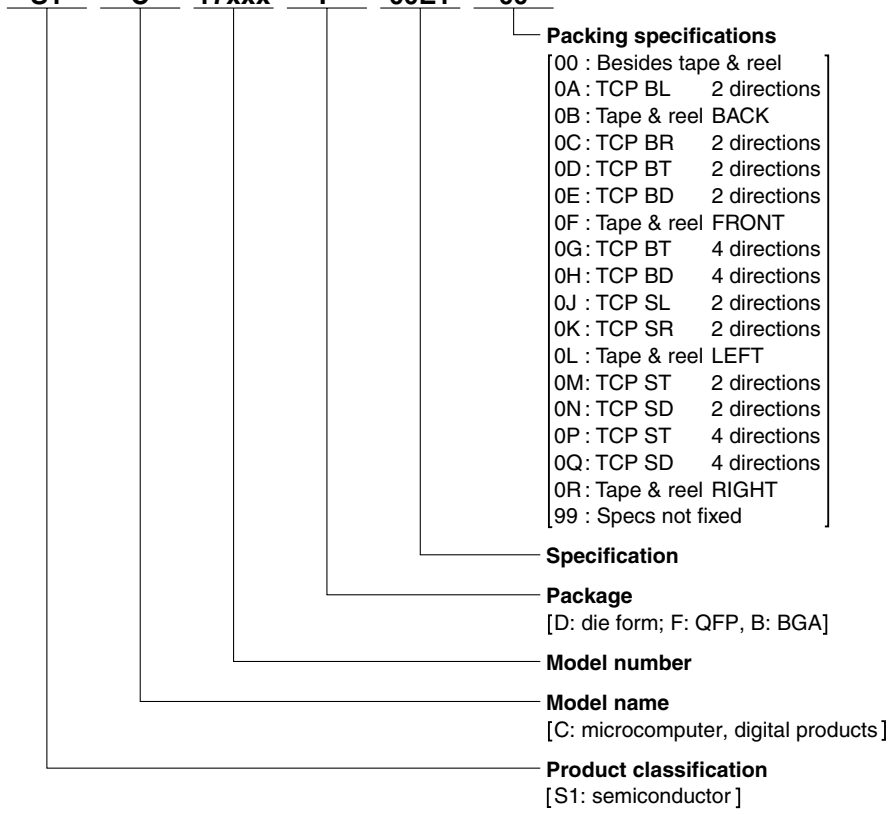
SEIKO EPSON CORPORATION

Rev.1.0

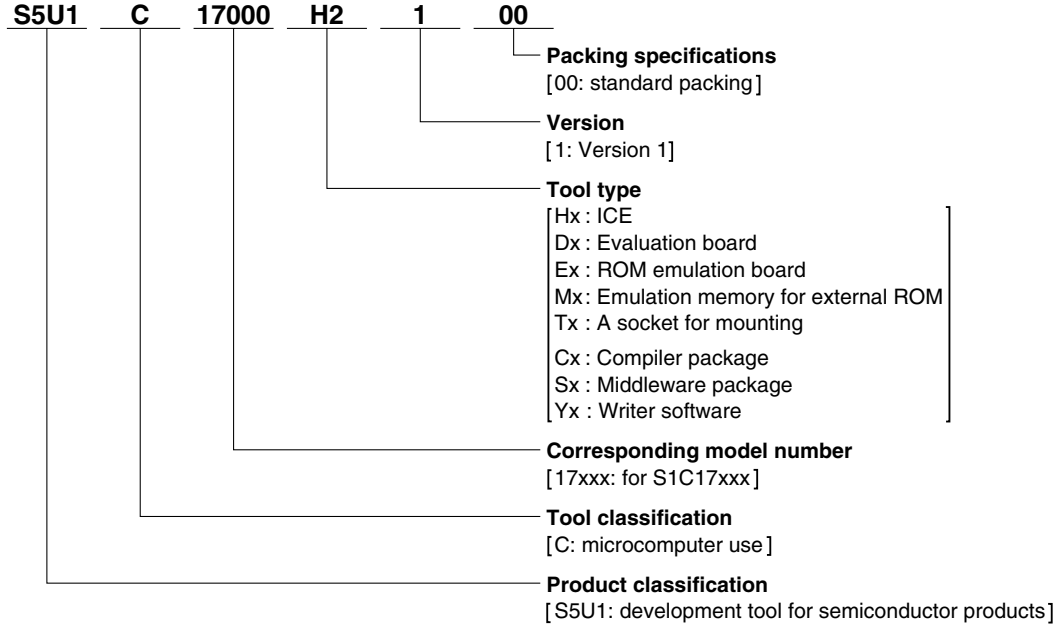
NOTICE

Configuration of product number

Devices



Development tools



– Contents –

1 Overview

1.1 Features

The main features of the S1C17651 are listed below.

Table 1.1.1 Features

CPU	
CPU core	Seiko Epson original 16-bit RISC CPU core S1C17
Multiplier/Divider (COPRO)	• 16-bit × 16-bit multiplier • 16-bit × 16-bit + 32-bit multiply and accumulation unit • 16-bit ÷ 16-bit divider
Embedded Flash memory	
Capacity	16K bytes (for both instructions and data)
Erase/program count	Three times
Other	• Read/program protection function • A programming power supply (V_{PP}) is required. • Allows on-board programming using a debugging tool such as ICDmini.
Embedded RAM	
Capacity	2K bytes
Clock generator	
System clock source	3 sources (OSC3B/OSC3A/OSC1)
OSC3B oscillator circuit	2M/1M/500k Hz (typ.) internal oscillator circuit
OSC3A oscillator circuit	4.2 MHz (max.) crystal or ceramic oscillator circuit
OSC1B oscillator circuit	32 kHz (typ.) internal oscillator circuit
OSC1A oscillator circuit	32.768 kHz (typ.) crystal oscillator circuit
	Oscillation adjustment by theoretical regulation
Other	• Core clock frequency control • Peripheral module clock supply control
LCD driver	
Number of driver outputs	Segment output: 20 pins Common output: 4 pins
Other	• Includes a power supply voltage booster/reducer. • Includes a display data memory.
I/O ports	
Number of general-purpose I/O ports	Max. 12 bits (Pins are shared with the peripheral I/O.)
Other	• Schmitt input • Pull-up control function • Port input interrupt: 8 bits
Serial interfaces	
SPI	1 channel
UART	1 channel (IrDA1.0 supported)
Timers/Counters	
8-bit timer (T8)	1 channel (Generates the SPI clock.)
16-bit PWM timer (T16A2)	1 channel (PWM output, event counter, and count capture functions)
Watchdog timer (WDT)	1 channel (Generates NMI/reset.)
Clock functions	
Real-time clock (RTC)	1 channel (Hour, minute, and second counters) with theoretical regulation support
Clock timer (CT)	1 channel (128 Hz to 1 Hz counters) with theoretical regulation support
Theoretical regulation function (TR)	Time adjustment function in +16/32768 to -15/32768 second units
Sound generator	
Buzzer frequency	8 frequencies selectable
Volume control	8 steps adjustable
Other	• One-shot buzzer • Auto envelope function
Analog circuits	
Supply voltage detection circuit (SVD)	1 channel (Detection voltage: 13 levels)
Interrupts	
Reset interrupt	#RESET pin/watchdog timer
NMI	Watchdog timer
Programmable interrupts	8 systems (8 levels)

Power supply voltage		
Operating voltage (V_{DD})	2.0 V to 3.6 V	
Flash programming/erasing voltage (V_{PP})	7V/7.5V	
Operating temperature		
Operating temperature range	-40°C to 85°C	
Current consumption (Typ value, V_{DD} = 2.0 V to 3.6 V)		
SLEEP state	90 nA	(OSC1 = Off, RTC = Off, OSC3B = Off, OSC3A = Off)
HALT state	0.42 µA	(OSC1 = 32 kHz (OSC1A), RTC = Off, OSC3B = Off, OSC3A = Off)
	0.42 µA	(OSC1 = 32 kHz (OSC1A), RTC = On, OSC3B = Off, OSC3A = Off)
Run state	10 µA	(OSC1 = 32 kHz (OSC1A), RTC = Off, OSC3B = Off, OSC3A = Off)
	1200 µA	(OSC1 = Off, RTC = Off, OSC3B = Off, OSC3A = 4 MHz ceramic)
	650 µA	(OSC1 = Off, RTC = Off, OSC3B = 2 MHz, OSC3A = Off)
Shipping form		
1	TQFP13-64pin (12 mm × 12 mm × 1 mm, lead pitch: 0.5 mm)	
2	Die	

1.2 Block Diagram

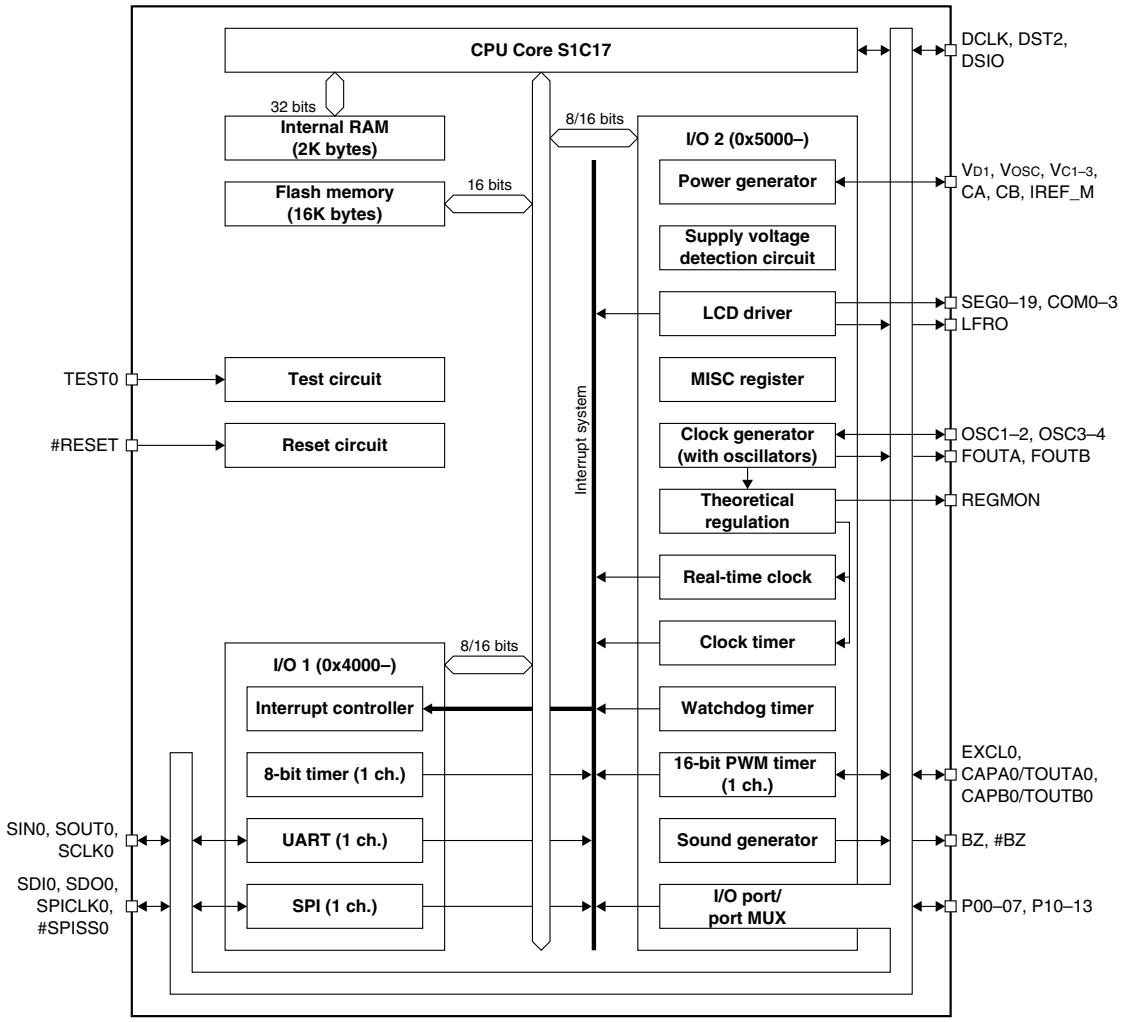


Figure 1.2.1 S1C17651 Block Diagram

1.3 Pins

1.3.1 Pin Configuration Diagram

TQFP13-64pin

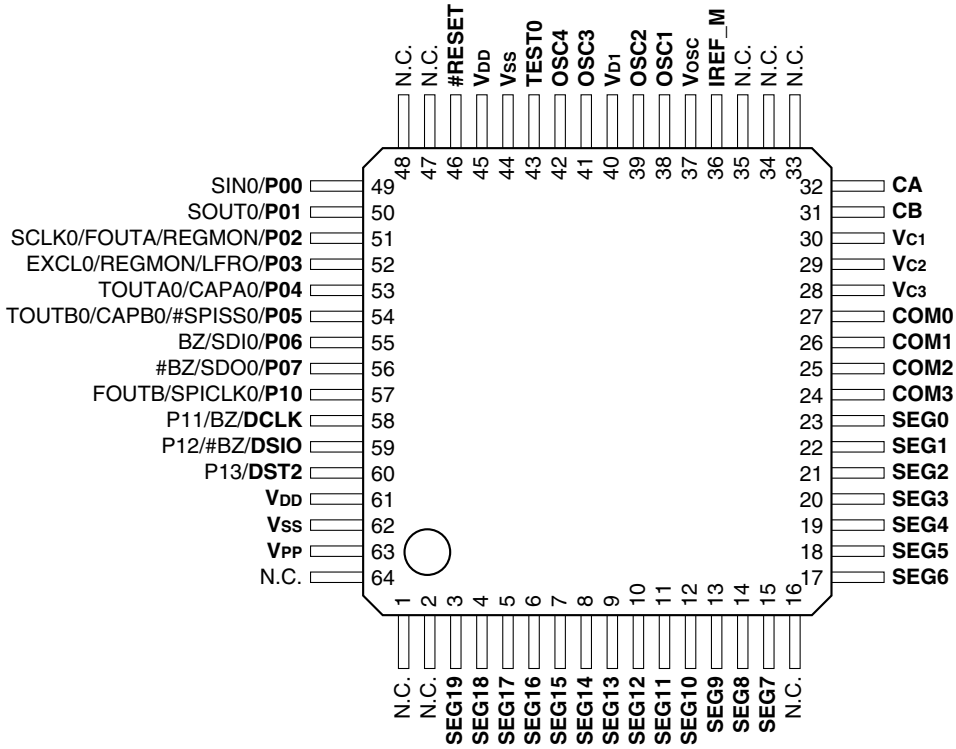


Figure 1.3.1.1 S1C17651 Pin Configuration Diagram (TQFP13-64pin)

Chip

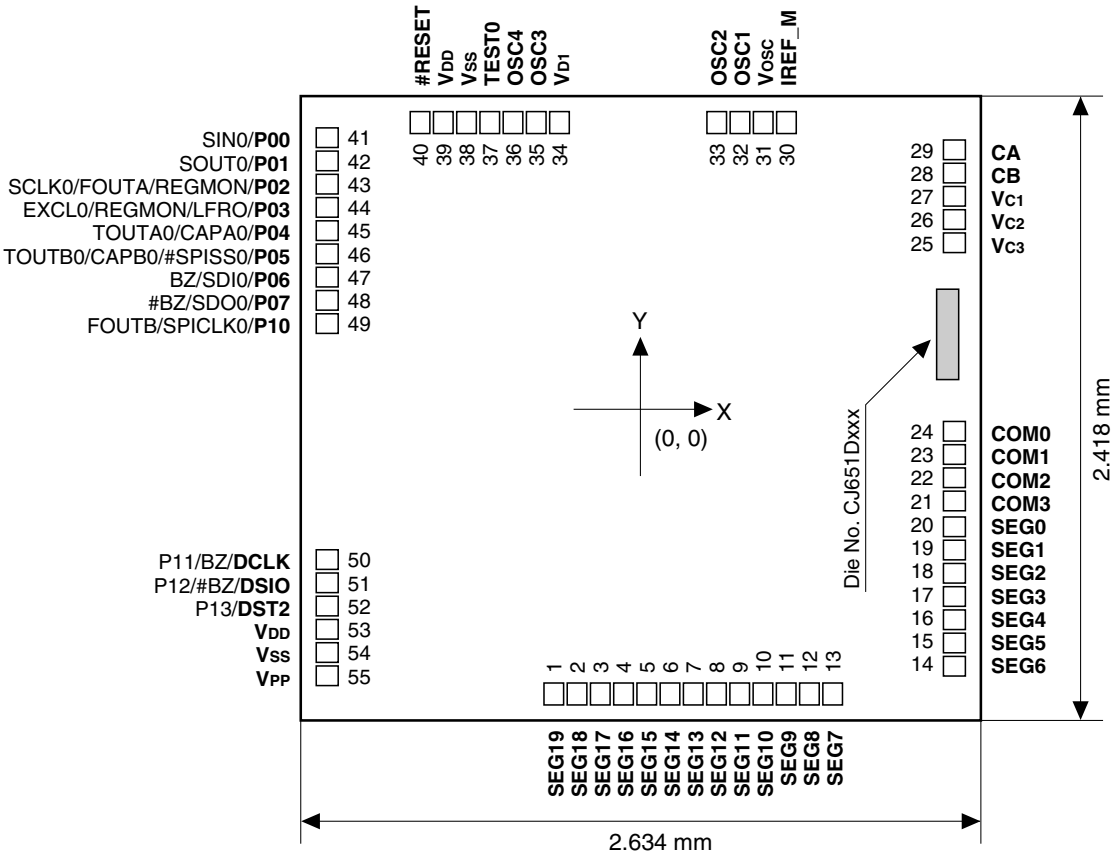


Figure 1.3.1.2 S1C17651 Pad Configuration Diagram

1.3.2 Pin Descriptions

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

Table 1.3.2.1 Pin Descriptions

Pin No.		Name	I/O	Default status	Function
Chip	TQFP				
20–1	23–17, 15–3	**SEG0–SEG19**	O	O (Hi-Z)	LCD segment output pins
24–21	27–24	**COM0–COM3**	O	O (Hi-Z)	LCD common output pins
25	28	**V_{C3}**	–	–	LCD system power supply circuit output pin
26	29	**V_{C2}**	–	–	LCD system power supply circuit output pin
27	30	**V_{C1}**	–	–	LCD system power supply circuit output pin
28	31	**CB**	–	–	Voltage boost capacitor connecting pin for LCD system power supply circuit
29	32	**CA**	–	–	Voltage boost capacitor connecting pin for LCD system power supply circuit
30	36	**IREF_M**	–	–	IREF constant current monitor pin (Leave the pin open during normal operation.)
31	37	**V_{OSC}**	–	–	Oscillation system voltage regulator output pin
32	38	**OSC1**	I	I	OSC1A oscillation input pin
33	39	**OSC2**	O	O	OSC1A oscillation output pin
34	40	**V_{D1}**	–	–	Internal logic system voltage regulator output pin
35	41	**OSC3**	I	I	OSC3A oscillation input pin
36	42	**OSC4**	O	O	OSC3A oscillation output pin
37	43	**TEST0**	I	I (Pull-down)	Test input pin (Connect to V_{SS} for normal operation.)
38	44	**V_{SS}**	–	–	GND pin
39	45	**V_{DD}**	–	–	Power supply pin (2.0 to 3.6 V)
40	46	**#RESET**	I	I (Pull-up)	Initial reset input pin
41	49	**P00**	I/O	I (Pull-up)	I/O port pin (with port input interrupt function)
		SIN0	I		UART Ch.0 data input pin
42	50	**P01**	I/O	I (Pull-up)	I/O port pin (with port input interrupt function)
		SOUT0	O		UART Ch.0 data output pin
43	51	**P02**	I/O	I (Pull-up)	I/O port pin (with port input interrupt function)
		SCLK0	I		UART Ch.0 external clock input pin
		FOUTA	O		Clock output pin
		REGMON	O		Theoretical regulation clock monitor output pin
44	52	**P03**	I/O	I (Pull-up)	I/O port pin (with port input interrupt function)
		EXCL0	I		T16A2 Ch.0 external clock input pin
		REGMON	O		Theoretical regulation clock monitor output pin
		LFRO	O		LCD frame signal output pin
45	53	**P04**	I/O	I (Pull-up)	I/O port pin (with port input interrupt function)
		TOUTA0	O		T16A2 Ch.0 TOUT A signal output pin
		CAPA0	I		T16A2 Ch.0 capture A trigger signal input pin
46	54	**P05**	I/O	I (Pull-up)	I/O port pin (with port input interrupt function)
		TOUTB0	O		T16A2 Ch.0 TOUT B signal output pin
		CAPB0	I		T16A2 Ch.0 capture B trigger signal input pin
		#SPISS0	I		SPI Ch.0 slave select signal input pin
47	55	**P06**	I/O	I (Pull-up)	I/O port pin (with port input interrupt function)
		BZ	O		Buzzer output pin
		SDI0	I		SPI Ch.0 data input pin
48	56	**P07**	I/O	I (Pull-up)	I/O port pin (with port input interrupt function)
		#BZ	O		Buzzer inverted output pin
		SDO0	O		SPI Ch.0 data output pin
49	57	**P10**	I/O	I (Pull-up)	I/O port pin
		FOUTB	O		Clock output pin
		SPICLK0	I/O		SPI Ch.0 clock input/output pin
50	58	**DCLK**	O	O (H)	On-chip debugger clock output pin
		P11	I/O		I/O port pin
		BZ	O		Buzzer output pin
51	59	**DSIO**	I/O	I (Pull-up)	On-chip debugger data input/output pin
		P12	I/O		I/O port pin
		#BZ	O		Buzzer inverted output pin
52	60	**DST2**	O	O (L)	On-chip debugger status output pin
		P13	I/O		I/O port pin
53	61	**V_{DD}**	–	–	Power supply pin (2.0 to 3.6 V)
54	62	**V_{SS}**	–	–	GND pin
55	63	**V_{PP}**	–	–	Flash programming/erasing power supply pin (7.0/7.5 V) (Leave the pin open during normal operation.)

2 CPU

The S1C17651 contains the S1C17 Core as its core processor.
The S1C17 Core is a Seiko Epson original 16-bit RISC-type processor.
It features low power consumption, high-speed operation, large address space, main instructions executable in one clock cycle, and a small sized design. The S1C17 Core is suitable for embedded applications such as controllers and sequencers for which an eight-bit CPU is commonly used.
For details of the S1C17 Core, refer to the "S1C17 Family S1C17 Core Manual."

2.1 Features of the S1C17 Core

Processor type

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

Instruction set

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

Register set

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

Memory space and bus

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

Interrupts

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

Power saving

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

Coprocessor interface

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

2.2 CPU Registers

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

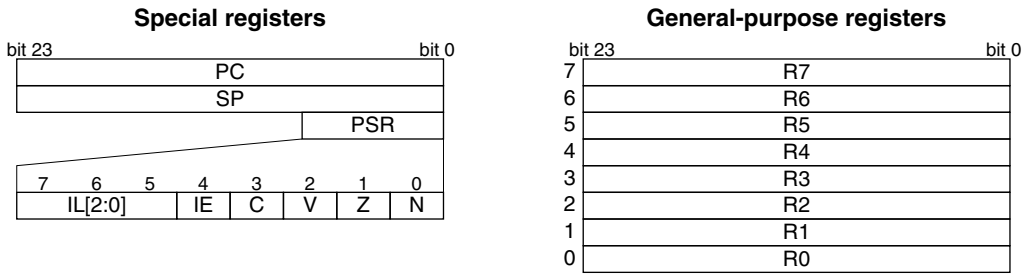


Figure 2.2.1 Registers

2.3 Instruction Set

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

Table 2.3.1 List of S1C17 Core Instructions

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

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

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

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

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

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

Table 2.3.2 Symbol Meanings

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

2.4 Reading PSR

The S1C17651 includes the MISC_PSR register for reading the contents of the PSR (Processor Status Register) in the S1C17 Core. Reading the contents of this register makes it possible to check the contents of the PSR using the application software. Note that data cannot be written to the PSR.

PSR Register (MISC_PSR)

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

D[15:8] Reserved

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

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

D4 PSRIE: PSR Interrupt Enable (IE) Bit

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

1 (R): 1 (interrupt enabled)

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

D3 PSRC: PSR Carry (C) Flag Bit

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

1 (R): 1

0 (R): 0 (default)

D2 PSRV: PSR Overflow (V) Flag Bit

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

1 (R): 1

0 (R): 0 (default)

D1 PSRZ: PSR Zero (Z) Flag Bit

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

1 (R): 1

0 (R): 0 (default)

D0 PSRN: PSR Negative (N) Flag Bit

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

1 (R): 1

0 (R): 0 (default)

2.5 Processor Information

The S1C17651 has the IDIR register shown below that allows the application software to identify CPU core type.

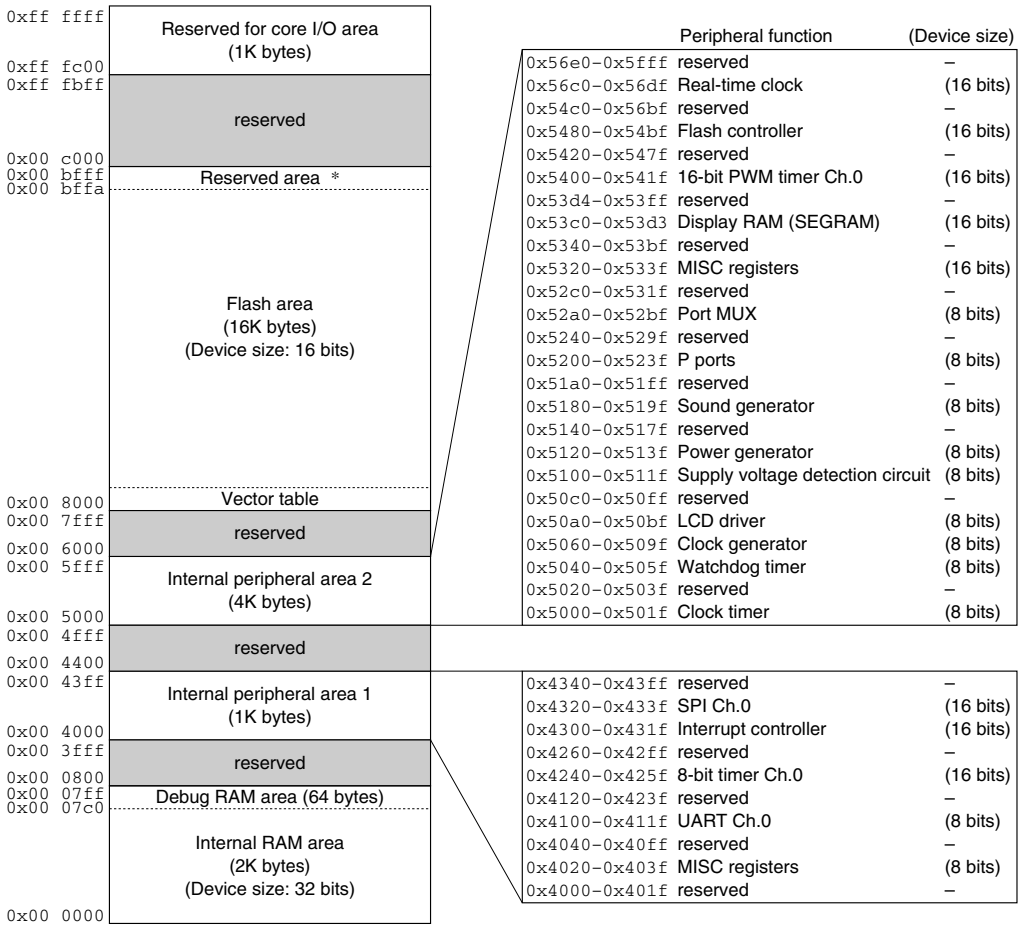
Processor ID Register (IDIR)

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

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

3 Memory Map, Bus Control

Figure 3.1 shows the S1C17651 memory map.



* 0xbffa–0xbffb: Reserved for the theoretical regulation function (See the "Setting Regulation Values" section in the "Theoretical Regulation" chapter.)
0xbffc–0xbfff: Flash protect bit area (See the "Protect Bits" section in this chapter.)

Figure 3.1 S1C17651 Memory Map

3.1 Bus Cycle

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

Table 3.1.1 Number of Bus Accesses

Device size	CPU access size	Number of bus accesses
8 bits	8 bits	1
	16 bits	2
	32 bits*	4
16 bits	8 bits	1
	16 bits	1
	32 bits*	2
32 bits	8 bits	1
	16 bits	1
	32 bits*	1

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

3.1.1 Restrictions on Access Size

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

3.1.2 Restrictions on Instruction Execution Cycles

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

- When the S1C17651 executes the instruction stored in the Flash area and accesses data in the Flash area
- When the S1C17651 executes the instruction stored in the internal RAM area and accesses data in the internal RAM area

3.2 Flash Area

3.2.1 Embedded Flash Memory

The 16K-byte area from address 0x8000 to address 0xbfff contains a Flash memory (4K bytes × 4 sectors) for storing application programs and data. Address 0x8000 is defined as the vector table base address, therefore a vector table (see "Vector Table" in the "Interrupt Controller (ITC)" chapter) must be placed from the beginning of the area. The vector table base address can be modified with the MISC_TTBRL/MISC_TTBRH registers.

3.2.2 Flash Programming

The S1C17651 supports on-board programming of the Flash memory, it makes it possible to program the Flash memory with the application programs/data by using the debugger through an ICDmini.

3.2.3 Protect Bits

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

Flash Protect Bits

Address	Bit	Function	Setting				Init.	R/W	Remarks
0xbffc (16 bits)	D15–4	reserved	–				–	–	
	D3	Flash write-protect bit for 0xb000–0xbfff	1	Writable	0	Protected	1	R/W	
	D2	Flash write-protect bit for 0xa000–0xafff	1	Writable	0	Protected	1	R/W	
	D1	Flash write-protect bit for 0x9000–0x9fff	1	Writable	0	Protected	1	R/W	
	D0	Flash write-protect bit for 0x8000–0x8fff	1	Writable	0	Protected	1	R/W	
0xbffe (16 bits)	D15–4	reserved	–				–	–	
	D3	Flash data-read-protect bit for 0xb000–0xbfff	1	Readable	0	Protected	1	R/W	
	D2	Flash data-read-protect bit for 0xa000–0xafff	1	Readable	0	Protected	1	R/W	
	D1	Flash data-read-protect bit for 0x9000–0x9fff	1	Readable	0	Protected	1	R/W	
	D0	reserved	1				1	R/W	Always set to 1.

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

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

3.2.4 Flash Memory Read Wait Cycle Setting

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

FLASHC Read Wait Control Register (FLASHC_WAIT)

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

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

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

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

3.3 Internal RAM Area

3.3.1 Embedded RAM

The S1C17651 contains a RAM in the 2K-byte area from address 0x0 to address 0x7ff. The RAM allows high-speed execution of the instruction codes copied into it as well as storing variables and other data.

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

The S1C17651 enables the RAM size used to apply restrictions to 2KB, 1KB, or 512B. For example, when using the S1C17651 to develop an application for a built-in ROM model, you can set the RAM size to match that of the target model, preventing creating programs that seek to access areas outside the RAM areas of the target product. The RAM size is selected using IRAMSZ[2:0]/MISC_IRAMSZ register.

IRAM Size Register (MISC_IRAMSZ)

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

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

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

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

Selects the internal RAM size used.

Table 3.3.1.1 Selecting Internal RAM Size

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

(Default: 0x3)

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

3.4 Display RAM area

The display RAM for the on-chip LCD driver is located in the 20-byte area beginning with address 0x53c0 in the internal peripheral area. The display RAM is accessed in one cycle as a 16-bit device. See the "Display Memory" section in the "LCD Driver (LCD)" chapter for specific information on the display memory.

3.5 Internal Peripheral Area

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

3.5.1 Internal Peripheral Area 1 (0x4000–)

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

- MISC register (MISC, 8-bit device)
- UART (UART, 8-bit device)
- 8-bit timer (T8, 16-bit device)
- Interrupt controller (ITC, 16-bit device)
- SPI (SPI, 16-bit device)

3.5.2 Internal Peripheral Area 2 (0x5000–)

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

- Clock timer (CT, 8-bit device)
- Watchdog timer (WDT, 8-bit device)
- Clock generator (CLG, 8-bit device)
- LCD driver (LCD, 8-bit device)
- Supply voltage detection circuit (SVD, 8-bit device)
- Power generator (VD1, 8-bit device)
- Sound generator (SND, 8-bit device)
- I/O port & port MUX (P, 8-bit device)
- MISC register (MISC, 16-bit device)
- Display RAM (SEGRAM, 16-bit device)
- 16-bit PWM timer (T16A2, 16-bit device)
- Flash controller (FLASHC, 16-bit device)
- Real-time clock (RTC, 16-bit device)

3.6 S1C17 Core I/O Area

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

Table 3.6.1 I/O Map (S1C17 Core I/O Area)

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

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

4 Power Supply

4.1 Power Supply Voltage (V_{DD})

The S1C17651 operates with a voltage supplied between the V_{DD} and V_{SS} pins. Supply a voltage within the range shown below to the V_{DD} pins with the V_{SS} pins as the GND level.

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

The S1C17651 provides two or more V_{DD} and V_{SS} pins. Do not leave any power supply pins open and be sure to connect them to + power source and GND.

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

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

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

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

4.3 Internal Power Supply Circuit

The S1C17651 has a built-in power supply circuit shown in Figure 4.3.1 to generate the operating voltages required for the internal circuits.

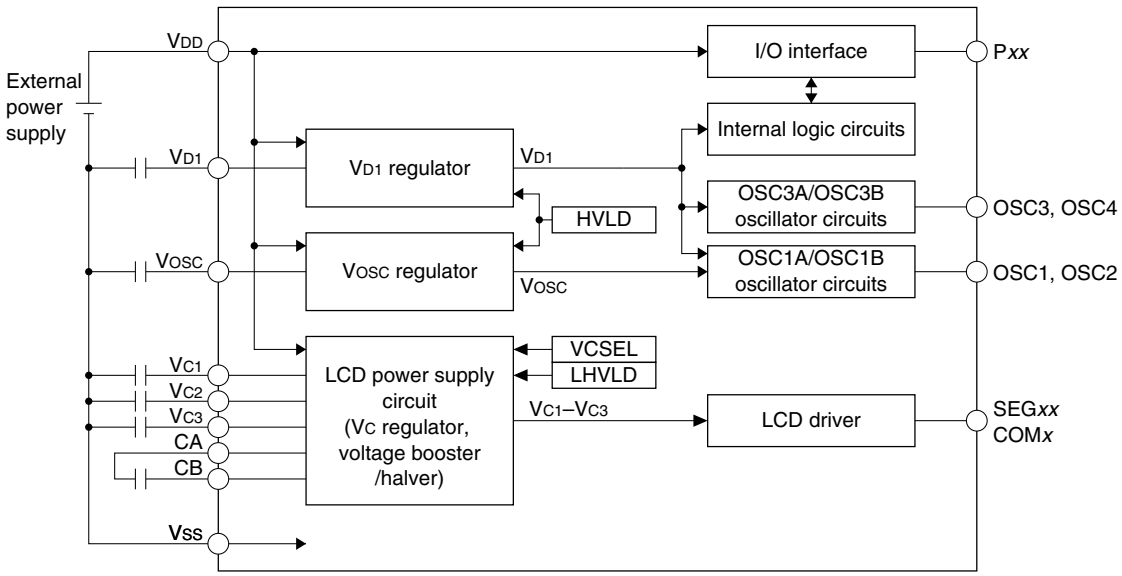


Figure 4.3.1 Configuration of Internal Power Supply Circuit

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

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

4.3.1 V_{D1} and V_{OSC} Regulators

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

4.3.2 LCD Power Supply Circuit

The LCD power supply circuit generates the LCD drive voltages V_{C1} to V_{C3}. These voltages are supplied to the LCD driver to generate LCD drive waveforms. The LCD power supply circuit consists of a V_C regulator and voltage booster/halver.
See the "Electrical Characteristics" chapter for the V_{C1} to V_{C3} values.

V_C regulator

The V_C regulator generates the reference voltage for boosting/halving (V_{C1} or V_{C2}) from V_{DD}.
Either V_{C1} or V_{C2} can be generated and one of them should be selected according to the V_{DD} value using VCSEL/LCD_VREG register.

Table 4.3.2.1 V_C Regulator Output Selection

Power supply voltage V_{DD}	VCSEL setting	Reference voltage
2.0 to 2.2 (3.6) V	0	V_{C1}
2.2 to 3.6 V	1	V_{C2}

(Default: 0)

Notes: • The V_{C1} to V_{C3} voltages cannot be obtained correctly if VCSEL is set to 1 when V_{DD} is 2.2 V or less.

• Although the reference voltage can be set to V_{C1} even if V_{DD} is 2.2 V or higher, current consumption will be increased in comparison with V_{C2}.

Voltage booster/halver

When V_{C1} is selected for the boosting/halving reference voltage, the voltage booster/halver generates V_{C2} and V_{C3} by boosting V_{C1} output from the V_C regulator. When V_{C2} is selected for the reference voltage, the voltage booster/halver generates V_{C1} by halving V_{C2} and V_{C3} by boosting V_{C2}. The boosting operation uses a clock, so it must be supplied to the LCD power supply circuit before LCD can start display.

Booster clock source selection

Select the clock source for the voltage booster/halver from OSC3B, OSC3A, and OSC1 using LCDBCLKSRC[1:0]/LCD_BCLK register.

Table 4.3.2.2 Clock Source Selection

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

(Default: 0x0)

Booster clock division ratio selection

Select the division ratio using LCDBCLKD[2:0]/LCD_BCLK register. Set it so that the clock frequency will be within the range from 512 Hz to 4 kHz.

Table 4.3.2.3 Clock Division Ratio Selection

LCDBCLKD[2:0]	Division ratio		
	Clock source = OSC3B	Clock source = OSC3A	Clock source = OSC1
0x7	Reserved	Reserved	Reserved
0x6	1/4096	1/8192	
0x5	1/2048	1/4096	
0x4	1/1024	1/2048	
0x3	1/512	1/1024	1/64
0x2	1/256	1/512	1/32
0x1	1/128	1/256	1/16
0x0	1/64	1/128	1/8

(Default: 0x0)

Booster clock enable

The booster clock supply is enabled with LCDBCLKE/LCD_BCLK register. The LCDBCLKE default setting is 0, which stops the clock. Setting LCDBCLKE to 1 feeds the clock generated as above to the LCD power supply circuit. If no LCD display is required, stop the clock to reduce current consumption.

4.3.3 Heavy Load Protection Mode

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

Table 4.3.3.1 Heavy Load Protection Mode Control Bits

Regulator	Control bit
V_{D1} regulator	HVLD/VD1_CTL register
V_{OSC} regulator	
V_C regulator	LHVLD/LCD _VREG register

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

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

4.4 Control Register Details

Table 4.4.1 List of Power Control Registers

Address	Register name		Function
0x5071	LCD_BCLK	LCD Booster Clock Control Register	Controls the LCD booster clock.
0x50a3	LCD_VREG	LCD Voltage Regulator Control Register	Controls the V_C regulator.
0x5120	VD1_CTL	V_{D1} Control Register	Controls the V_{D1} regulator.

The power control registers are described in detail below.

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

LCD Booster Clock Control Register (LCD_BCLK)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
LCD Booster Clock Control Register (LCD_BCLK)	**0x5071** (8 bits)	D7	**–**	reserved	–				–	–	0 when being read.
		D6–4	**LCDBCLKD [2:0]**	LCD booster clock division ratio select	LCDB CLKD [2:0]	Division ratio OSC3B	OSC3A	OSC1	0x0	R/W	
					0x7	–	–	–			
					0x6	1/4096	1/8192	–			
					0x5	1/2048	1/4096	–			
					0x4	1/1024	1/2048	–			
					0x3	1/512	1/1024	1/64			
					0x2	1/256	1/512	1/32			
					0x1	1/128	1/256	1/16			
					0x0	1/64	1/128	1/8			
		D3–2	**LCDBCLK SRC[1:0]**	LCD Booster clock source select	LCDBCLK SRC[1:0]		Clock source		0x0	R/W	
					0x3		reserved				
					0x2		OSC3A				
					0x1		OSC1				
					0x0		OSC3B				
		D1	**–**	reserved	–				–	–	0 when being read.
		D0	**LCDBCLKE**	LCD Booster clock enable	1 Enable		0 Disable		0	R/W	

D7 Reserved

D[6:4] LCDBCLKD[2:0]: LCD Booster Clock Division Ratio Select Bits

Selects the division ratio for generating the booster clock.

Table 4.4.2 Clock Division Ratio Selection

LCDBCLKD[2:0]	Division ratio		
	Clock source = OSC3B	Clock source = OSC3A	Clock source = OSC1
0x7	Reserved	Reserved	Reserved
0x6	1/4096	1/8192	
0x5	1/2048	1/4096	
0x4	1/1024	1/2048	
0x3	1/512	1/1024	1/64
0x2	1/256	1/512	1/32
0x1	1/128	1/256	1/16
0x0	1/64	1/128	1/8

(Default: 0x0)

D[3:2] LCDBCLKSRC[1:0]: LCD Booster Clock Source Select Bits

Selects the booster clock source.

Table 4.4.3 Clock Source Selection

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

(Default: 0x0)

D1 Reserved

D0 LCDBCLKE: LCD Booster Clock Enable Bit

Enables or disables the booster clock supply to the LCD power supply circuit.

1 (R/W): Enabled (on)

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

The LCDBCLKE default setting is 0, which disables the clock supply. Setting LCDBCLKE to 1 sends the clock to the LCD power supply circuit. If LCD display is not required, disable the clock supply to reduce current consumption.

LCD Voltage Regulator Control Register (LCD_VREG)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
LCD Voltage Regulator Control Register (LCD_VREG)	0x50a3 (8 bits)	D7–5	–	reserved	–	–	–	0 when being read.
		D4	LHVLD	Vc heavy load protection mode	1 On 0 Off	0	R/W	
		D3–1	–	reserved	–	–	–	0 when being read.
		D0	VCSEL	Reference voltage select	1 V_{C2} 0 V_{C1}	0	R/W	

D[7:5] Reserved

D4 LHVLD: Vc Heavy Load Protection Mode Bit

Sets the Vc regulator into heavy load protection mode.

1 (R/W): Heavy load protection On

0 (R/W): Heavy load protection Off (default)

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

D[3:1] Reserved

D0 VCSEL: Reference Voltage Select Bit

Selects the Vc regulator output voltage (reference voltage for boosting/halving).

1 (R/W): V_{C2}

0 (R/W): V_{C1} (default)

Select either V_{C1} or V_{C2} to be generated by the V_C regulator according to the V_{DD} value.

Table 4.4.4 V_C Regulator Output Selection

Power supply voltage V_{DD}	VCSEL setting	Reference voltage
2.0 to 2.2 (3.6) V	0	V_{C1}
2.2 to 3.6 V	1	V_{C2}

(Default: 0)

Notes: • The V_{C1} to V_{C3} voltages cannot be obtained correctly if VCSEL is set to 1 when V_{DD} is 2.2 V or less.

• Although the reference voltage can be set to V_{C1} even if V_{DD} is 2.2 V or higher, current consumption will be increased in comparison with V_{C2}.

V_{D1} Control Register (VD1_CTL)

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

D[7:6] Reserved

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

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

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

D[4:0] Reserved

5 Initial Reset

5.1 Initial Reset Sources

The S1C17651 has three initial reset sources that initialize the internal circuits.

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

Figure 5.1.1 shows the configuration of the initial reset circuit.

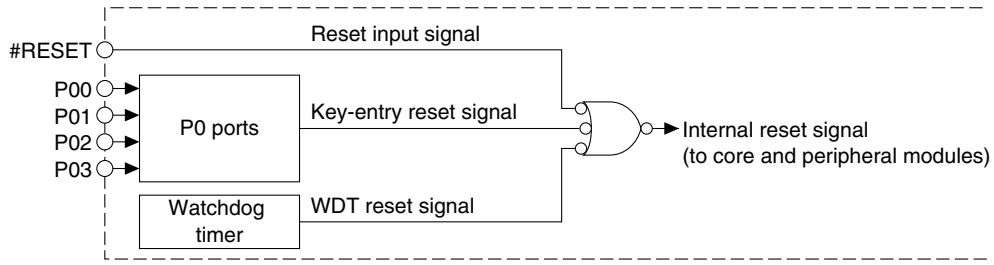


Figure 5.1.1 Configuration of Initial Reset Circuit

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

5.1.1 #RESET Pin

By setting the #RESET pin to low level, the S1C17651 enters initial reset state. In order to initialize the S1C17651 for sure, the #RESET pin must be held at low for more than the prescribed time (see "Input/Output Pin Characteristics" in the "Electrical Characteristics" chapter) after the power supply voltage is supplied.
When the #RESET pin at low level is set to high level, the CPU starts executing the initial reset sequence.
The #RESET pin is equipped with a pull-up resistor.

5.1.2 P0 Port Key-Entry Reset

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

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

5.1.3 Resetting by the Watchdog Timer

The S1C17651 has a built-in watchdog timer to detect runaway of the CPU. The watchdog timer overflows if it is not reset with software (due to CPU runaway) in four-second cycles. The overflow signal can generate either NMI or reset. Write 1 to the WDTMD/WDT_ST register to generate reset (NMI occurs when WDTMD = 0).
For details of the watchdog timer, see the "Watchdog Timer (WDT)" chapter.

Notes: • When using the reset function of the watchdog timer, program the watchdog timer so that it will be reset within four-second cycles to avoid occurrence of an unnecessary reset.

• The reset function of the watchdog timer cannot be used for power-on reset as it must be enabled with software.

5.2 Initial Reset Sequence

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

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

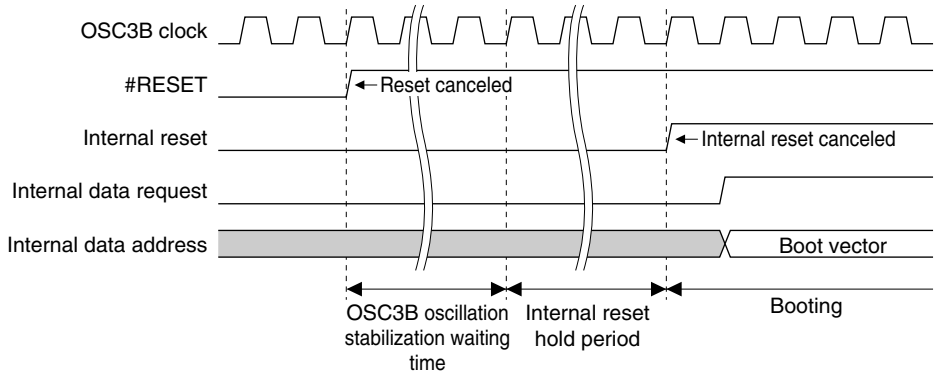


Figure 5.2.1 Operation Sequence Following Cancellation of Initial Reset

5.3 Initial Settings After an Initial Reset

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

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

6 Interrupt Controller (ITC)

6.1 ITC Module Overview

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

- Supports eight maskable interrupt systems.
  1. P00–P07 input interrupt (8 types)
  2. Clock timer interrupt (4 types)
  3. RTC interrupt (10 types)
  4. LCD interrupt (1 type)
  5. 16-bit PWM timer Ch.0 interrupt (6 types)
  6. 8-bit timer Ch.0 interrupt (1 type)
  7. UART Ch.0 interrupt (4 types)
  8. SPI Ch.0 interrupt (2 types)
- Supports eight interrupt levels to prioritize the interrupt sources.

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

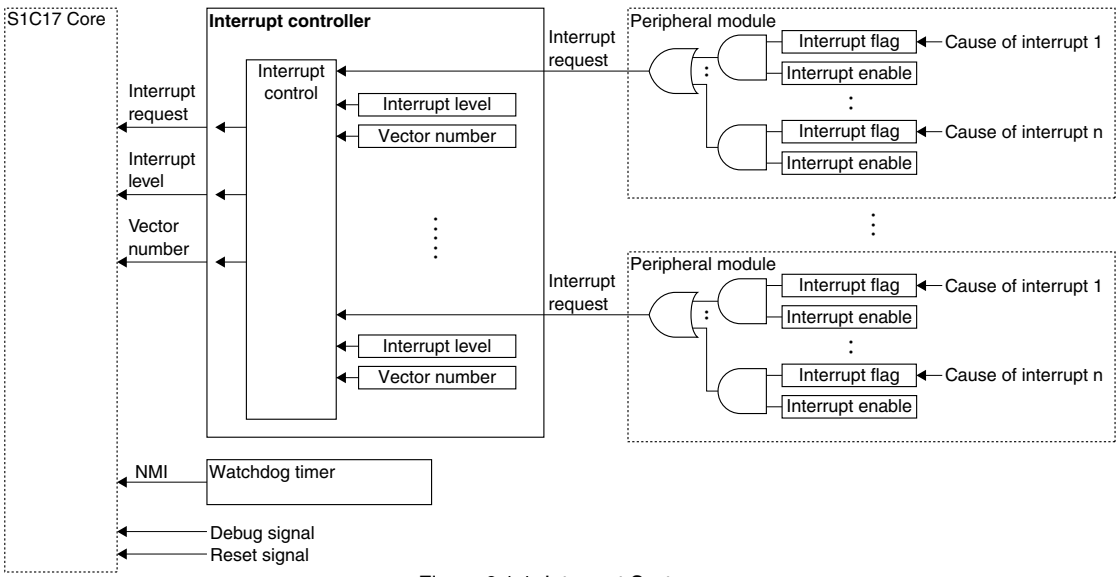


Figure 6.1.1 Interrupt System

6.2 Vector Table

The vector table contains the vectors to the interrupt handler routines (handler routine start address) that will be read by the S1C17 Core to execute the handler when an interrupt occurs.
Table 6.2.1 shows the vector table of the S1C17651.

Table 6.2.1 Vector Table

Vector No. Software interrupt No.	Vector address	Hardware interrupt name	Cause of hardware interrupt	Priority
0 (0x00)	TTBR + 0x00	Reset	• Low input to the #RESET pin • Watchdog timer overflow *2	1
1 (0x01)	TTBR + 0x04	Address misaligned interrupt	Memory access instruction	2
–	(0xfffc00)	Debugging interrupt	`brk` instruction, etc.	3
2 (0x02)	TTBR + 0x08	NMI	Watchdog timer overflow *2	4
3 (0x03)	TTBR + 0x0c	Reserved for C compiler	–	–
4 (0x04)	TTBR + 0x10	P0 port interrupt	P00–P07 port inputs	High *1
5 (0x05)	TTBR + 0x14	reserved	–	↑
6 (0x06)	TTBR + 0x18			
7 (0x07)	TTBR + 0x1c	Clock timer interrupt	• 32 Hz timer signal • 8 Hz timer signal • 2 Hz timer signal • 1 Hz timer signal	
8 (0x08)	TTBR + 0x20	RTC interrupt	• 32 Hz, 8 Hz, 4 Hz, 1 Hz • 10 s, 1 m, 10 m, 1 h • Half-day, one day	
9 (0x09)	TTBR + 0x24	reserved	–	
10 (0x0a)	TTBR + 0x28	LCD interrupt	Frame signal	
11 (0x0b)	TTBR + 0x2c	16-bit PWM timer Ch.0 interrupt	• Compare A/B • Capture A/B • Capture A/B overwrite	
12 (0x0c)	TTBR + 0x30	reserved	–	
13 (0x0d)	TTBR + 0x34			
14 (0x0e)	TTBR + 0x38	8-bit timer Ch. 0 interrupt	Timer underflow	
15 (0x0f)	TTBR + 0x3c	reserved	–	
16 (0x10)	TTBR + 0x40	UART Ch.0 interrupt	• Transmit buffer empty • End of transmission • Receive buffer full • Receive error	
17 (0x11)	TTBR + 0x44	reserved	–	
18 (0x12)	TTBR + 0x48	SPI Ch.0 interrupt	• Transmit buffer empty • Receive buffer full	
19 (0x13)	TTBR + 0x4c	reserved	–	
:	:	:	:	↓
31 (0x1f)	TTBR + 0x7c	reserved	–	Low *1

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

Vector numbers 4, 7, 8, 10, 11, 14, 16, and 18 are assigned to the maskable interrupts supported by the S1C17651.

Vector table base address

The S1C17651 allows the base (starting) address of the vector table to be set using the MISC_TTBRL and MISC_TTBRH registers. “TTBR” described in Table 6.2.1 means the value set to these registers. After an initial reset, the MISC_TTBRL and MISC_TTBRH registers are set to 0x8000. Therefore, even when the vector table location is changed, it is necessary that at least the reset vector be written to the above address. Bits 7 to 0 in the MISC_TTBRL register are fixed at 0, so the vector table starting address always begins with a 256-byte boundary address.

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

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

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

6.3 Control of Maskable Interrupts

6.3.1 Interrupt Control Bits in Peripheral Modules

The peripheral module that generates interrupts includes an interrupt enable bit and an interrupt flag for each interrupt cause. The interrupt flag is set to 1 when the cause of interrupt occurs. By setting the interrupt enable bit to 1 (interrupt enabled), the flag state will be sent to the ITC as an interrupt request signal, generating an interrupt request to the S1C17 Core.
The corresponding interrupt enable bits should be set to 0 for those causes for which interrupts are not desired. In this case, although the interrupt flag is set to 1 if the interrupt cause occurs, the interrupt request signal sent to the ITC will not be asserted.
For specific information on causes of interrupts, interrupt flags, and interrupt enable bits, refer to the respective peripheral module descriptions.

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

6.3.2 ITC Interrupt Request Processing

On receiving an interrupt signal from a peripheral module, the ITC sends the interrupt request, interrupt level, and vector number signals to the S1C17 Core.
Vector numbers are determined by the ITC internal hardware for each interrupt cause, as shown in Table 6.2.1.
The interrupt level is a value used by the S1C17 Core to compare with the IL bits (PSR). This interrupt level is used in the S1C17 Core to disable subsequently occurring interrupts with the same or lower level. (See Section 6.3.3.)
The default ITC settings are level 0 for all maskable interrupts. Interrupt requests are not accepted by the S1C17 Core if the level is 0.
The ITC includes control bits for selecting the interrupt level, and the level can be set to between 0 (low) and 7 (high) interrupt levels for each interrupt type.
If interrupt requests are input to the ITC simultaneously from two or more peripheral modules, the ITC outputs the interrupt request with the highest priority to the S1C17 Core in accordance with the following conditions.

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

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

Table 6.3.2.1 Interrupt Level Setting Bits

Hardware interrupt	Interrupt level setting bits	Register address
P0 port interrupt	ILV0[2:0] (D[2:0]/ITC_LV0 register)	0x4306
Clock timer interrupt	ILV3[2:0] (D[10:8]/ITC_LV1 register)	0x4308
RTC interrupt	ILV4[2:0] (D[2:0]/ITC_LV2 register)	0x430a
LCD interrupt	ILV6[2:0] (D[2:0]/ITC_LV3 register)	0x430c
16-bit PWM timer Ch.0 interrupt	ILV7[2:0] (D[10:8]/ITC_LV3 register)	0x430c
8-bit timer Ch.0 interrupt	ILV10[2:0] (D[2:0]/ITC_LV5 register)	0x4310
UART Ch.0 interrupt	ILV12[2:0] (D[2:0]/ITC_LV6 register)	0x4312
SPI Ch.0 interrupt	ILV14[2:0] (D[2:0]/ITC_LV7 register)	0x4314

6.3.3 Interrupt Processing by the S1C17 Core

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

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

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

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

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

6.4 NMI

In the S1C17651, the watchdog timer can generate a non-maskable interrupt (NMI). The vector number for NMI is 2, with the vector address set to the vector table's starting address + 8 bytes.
This interrupt takes precedence over other interrupts and is unconditionally accepted by the S1C17 Core.

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

6.5 Software Interrupts

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

6.6 HALT and SLEEP Mode Cancellation

HALT and SLEEP modes are cleared by the following signals, which start the CPU.

- Interrupt request signal sent to the CPU from the ITC
- NMI signal output by the watchdog timer
- Debug interrupt signal
- Reset signal

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

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

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

6.7 Control Register Details

Table 6.7.1 List of ITC Registers

Address	Register name		Function
0x4306	ITC_LV0	Interrupt Level Setup Register 0	Sets the P0 interrupt level.
0x4308	ITC_LV1	Interrupt Level Setup Register 1	Sets the CT interrupt level.
0x430a	ITC_LV2	Interrupt Level Setup Register 2	Sets the RTC interrupt level.
0x430c	ITC_LV3	Interrupt Level Setup Register 3	Sets the LCD and T16A2 Ch.0 interrupt levels.
0x4310	ITC_LV5	Interrupt Level Setup Register 5	Sets the T8 Ch.0 interrupt level.
0x4312	ITC_LV6	Interrupt Level Setup Register 6	Sets the UART Ch.0 interrupt level.
0x4314	ITC_LV7	Interrupt Level Setup Register 7	Sets the SPI Ch.0 interrupt level.

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

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

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Interrupt Level Setup Register *x* (ITC_LV*x*)	**0x4306** | **0x4314** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**ILV*n*[2:0]**	*INTn* (1, 3, ... 7) interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**ILV*n*[2:0]**	*INTn* (0, 2, ... 14) interrupt level	0 to 7	0x0	R/W	

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

D[10:8], D[2:0]
ILV*n*[2:0]: *INTn* Interrupt Level Bits (*n* = 0–14)

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

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

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

If multiple interrupt requests enabled by the interrupt enable bit occur simultaneously, the ITC sends the interrupt request with the highest level set by the ITC_LV*x* registers (0x4306 to 0x4314) to the S1C17 Core.

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

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

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

Table 6.7.2 Interrupt Level Bits

Register	Bit	Interrupt
ITC_LV0(0x4306)	ILV0[2:0] (D[2:0])	P0 port interrupt
	(ILV1[2:0] (D[10:8]))	Reserved
ITC_LV1(0x4308)	(ILV2[2:0] (D[2:0]))	Reserved
	ILV3[2:0] (D[10:8])	Clock timer interrupt
ITC_LV2(0x430a)	ILV4[2:0] (D[2:0])	RTC interrupt
	(ILV5[2:0] (D[10:8]))	Reserved
ITC_LV3(0x430c)	ILV6[2:0] (D[2:0])	LCD interrupt
	ILV7[2:0] (D[10:8])	16-bit PWM timer Ch.0 interrupt
ITC_LV5(0x4310)	ILV10[2:0] (D[2:0])	8-bit timer Ch.0 interrupt
	(ILV11[2:0] (D[10:8]))	Reserved
ITC_LV6(0x4312)	ILV12[2:0] (D[2:0])	UART Ch.0 interrupt
	(ILV13[2:0] (D[10:8]))	Reserved
ITC_LV7(0x4314)	ILV14[2:0] (D[2:0])	SPI Ch.0 interrupt
	(ILV15[2:0] (D[10:8]))	Reserved

7 Clock Generator (CLG)

7.1 CLG Module Overview

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

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

Figure 7.1.1 shows the clock system and CLG module configuration.

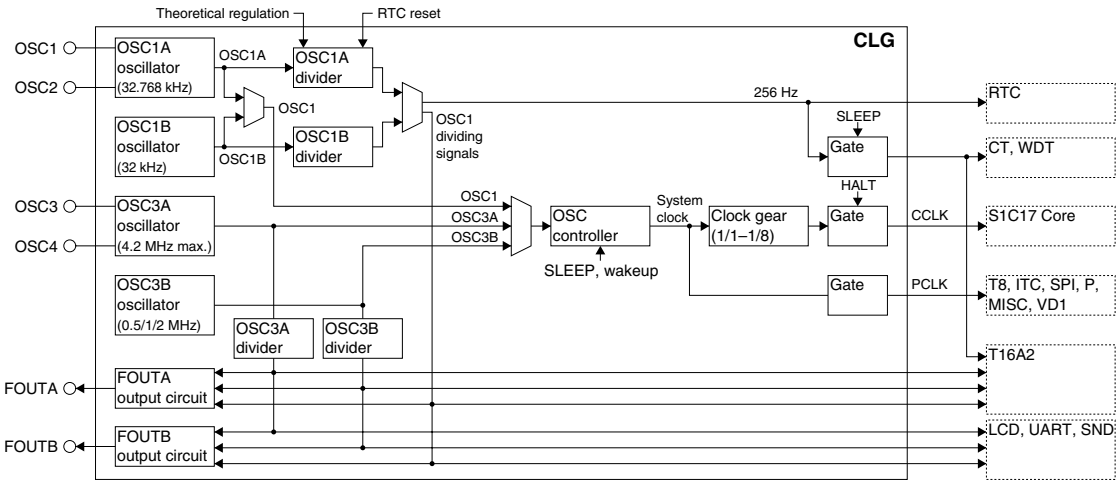


Figure 7.1.1 CLG Module Configuration

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

7.2 CLG Input/Output Pins

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

Table 7.2.1 List of CLG Pins

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

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

7.3 Oscillators

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

7.3.1 OSC3B Oscillator

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

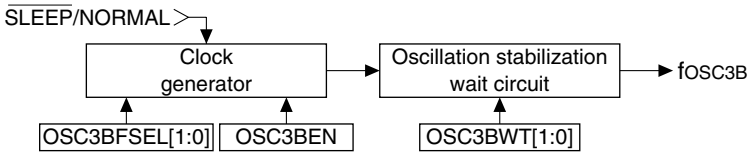


Figure 7.3.1.1 OSC3B Oscillator Circuit

OSC3B oscillation frequency selection

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

Table 7.3.1.1 OSC3B Oscillation Frequency Setting

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

(Default: 0x0)

OSC3B oscillation on/off

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

Stabilization wait time at start of OSC3B oscillation

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

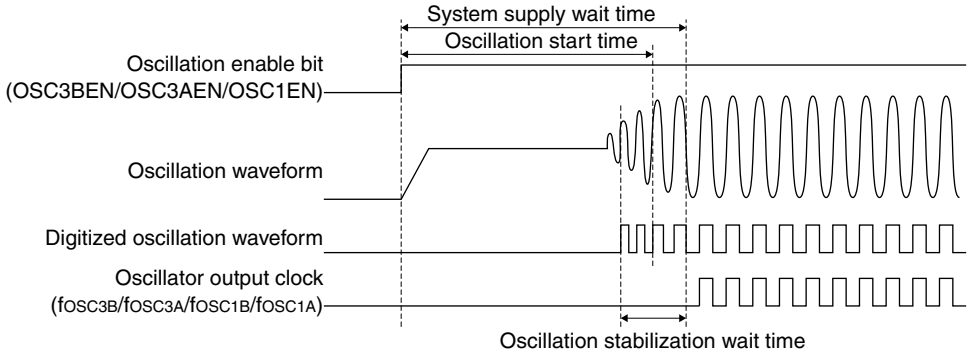


Figure 7.3.1.2 Oscillation Start Time and Oscillation Stabilization Wait Time

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

Table 7.3.1.2 OSC3B Oscillation Stabilization Wait Time Settings

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

(Default: 0x0)

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

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

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

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

7.3.2 OSC3A Oscillator

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

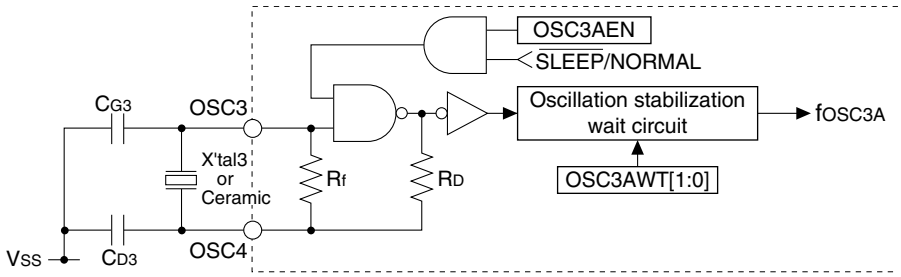


Figure 7.3.2.1 OSC3A Oscillator Circuit

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

OSC3A oscillation on/off

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

Stabilization wait time at start of OSC3A oscillation

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

Table 7.3.2.1 OSC3A Oscillation Stabilization Wait Time Settings

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

(Default: 0x0)

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

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

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

7.3.3 OSC1 Oscillator

The S1C17651 has two low-speed oscillator circuits (OSC1A and OSC1B) and either one can be selected as the OSC1 oscillator.
The OSC1 clock is generally used as the timer operation clock (for the real-time clock, clock timer, watchdog timer, and 16-bit PWM timer) and an operation clock for the UART, sound generator, and LCD driver. It can be used as the system clock instead of the OSC3B or OSC3A clock to reduce power consumption when no high-speed processing is required.

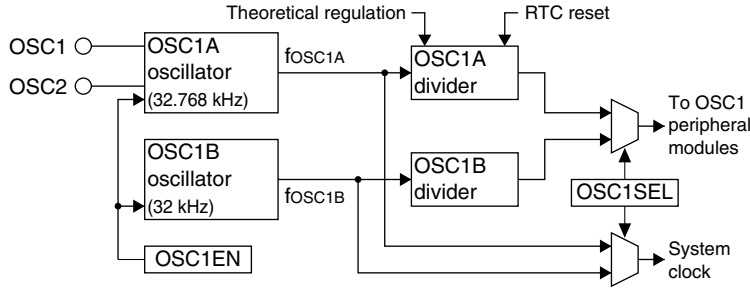


Figure 7.3.3.1 OSC1 Oscillator Configuration

OSC1A oscillator

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

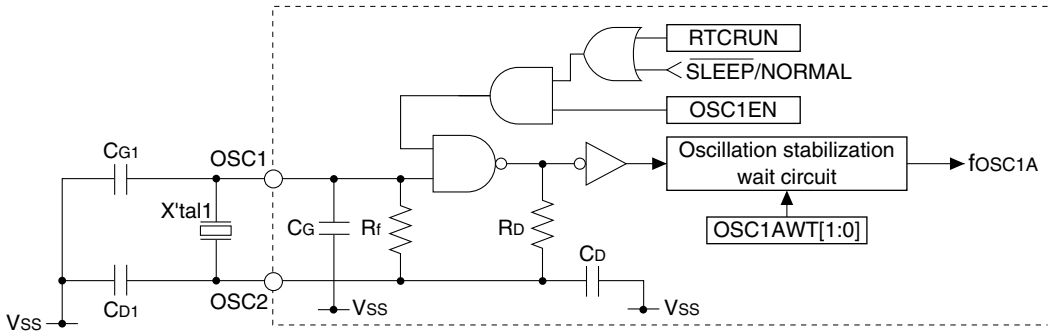


Figure 7.3.3.2 OSC1A Oscillator Circuit

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

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

OSC1B oscillator

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

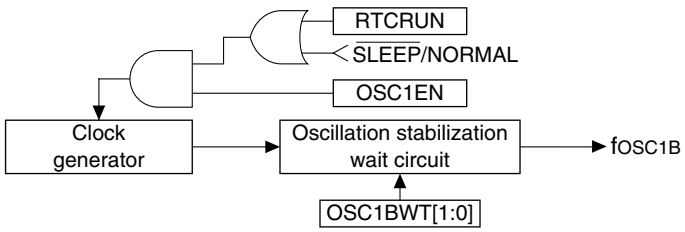


Figure 7.3.3.3 OSC1B Oscillator Circuit

OSC1A/OSC1B oscillator selection

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

OSC1 oscillation on/off

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

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

Table 7.3.3.1 OSC1 Oscillator Operating Status (normal operation)

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

Table 7.3.3.2 OSC1 Oscillator Operating Status (SLEEP mode)

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

Stabilization wait time at start of OSC1 oscillation

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

Table 7.3.3.3 OSC1A Oscillation Stabilization Wait Time Settings

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

(Default: 0x0)

Table 7.3.3.4 OSC1B Oscillation Stabilization Wait Time Settings

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

(Default: 0x0)

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

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

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

7.4 System Clock Switching

The figure below shows the system clock selector.

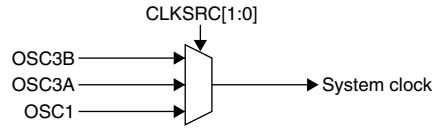


Figure 7.4.1 System Clock Selector

The S1C17651 has three system clock sources (OSC3B, OSC3A, and OSC1) and it start operating with the OSC3B clock after an initial reset. The system clock can be switched to the OSC3A clock when a high-speed clock is required for the processing, or to the OSC1 clock for power saving. Use CLKSRC[1:0]/CLG_SRC register for this switching. Oscillator circuits other than those selected as the system clock source and not used for running peripheral circuits can be shut down to reduce current consumption.

Table 7.4.1 System Clock Selection

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

(Default: 0x0)

The following shows system clock switching procedures:

Switching the system clock to OSC3A from OSC3B or OSC1

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

Switching the system clock to OSC1 from OSC3B or OSC3A

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

Switching the system clock to OSC3B from OSC3A or OSC1

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

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

Table 7.4.2 System Clock Switching Conditions

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

- The oscillator circuit selected as the system clock source cannot be turned off.

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

7.5 CPU Core Clock (CCLK) Control

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

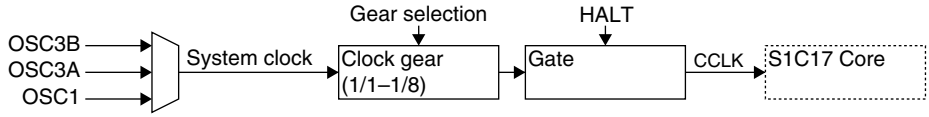


Figure 7.5.1 CCLK Supply System

Clock gear settings

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

Table 7.5.1 CCLK Gear Ratio Selection

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

(Default: 0x0)

Clock supply control

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

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

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

7.6 Peripheral Module Clock (PCLK) Control

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

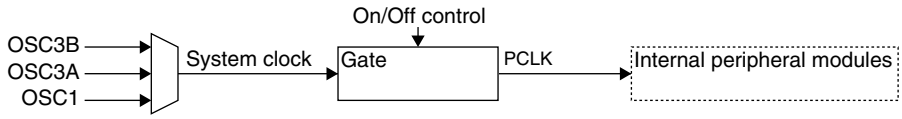


Figure 7.6.1 Peripheral Module Clock Control Circuit

Clock supply control

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

Table 7.6.1 PCLK Control

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

(Default: 0x3)

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

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

Table 7.6.2 Peripheral Modules and Operating Clocks

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

7.7 Clock External Output (FOUTA, FOUTB)

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

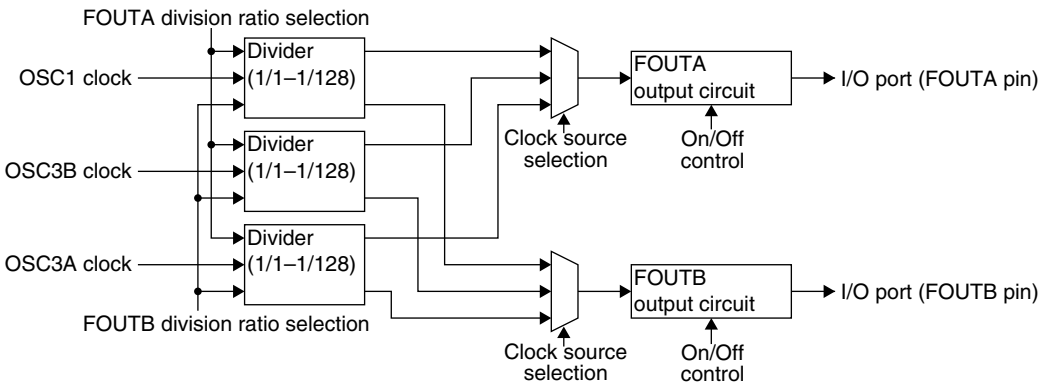


Figure 7.7.1 Clock Output Circuit

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

Output pin setting

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

Clock source selection

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

Table 7.7.1 Clock Source Selection

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

(Default: 0x0)

Clock frequency selection

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

Table 7.7.2 Clock Division Ratio Selection

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

(Default: 0x0)

Clock output control

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

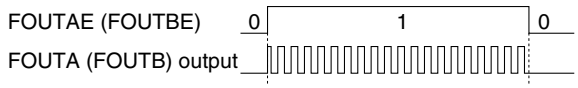


Figure 7.7.2 FOUTA/FOUTB Output

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

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

7.8 Control Register Details

Table 7.8.1 List of CLG Registers

Address	Register name		Function
0x5060	CLG_SRC	Clock Source Select Register	Selects the clock source.
0x5061	CLG_CTL	Oscillation Control Register	Controls oscillation.
0x5064	CLG_FOUTA	FOUTA Control Register	Controls FOUTA clock output.
0x5065	CLG_FOUTB	FOUTB Control Register	Controls FOUTB clock output.
0x507d	CLG_WAIT	Oscillation Stabilization Wait Control Register	Controls oscillation stabilization waiting time.
0x5080	CLG_PCLK	PCLK Control Register	Controls the PCLK supply.
0x5081	CLG_CCLK	CCLK Control Register	Configures the CCLK division ratio.

The CLG module registers are described in detail below.

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

Clock Source Select Register (CLG_SRC)

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

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

Selects the OSC3B oscillation frequency.

Table 7.8.2 OSC3B Oscillation Frequency Setting

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

(Default: 0x0)

D5 Reserved

D4 OSC1SEL: OSC1 Source Select Bit

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

D[3:2] Reserved

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

Selects the system clock source.

Table 7.8.3 System Clock Selection

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

(Default: 0x0)

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

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

Table 7.8.4 System Clock Switching Conditions

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

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

Oscillation Control Register (CLG_CTL)

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

D[7:3] Reserved

D2 OSC3BEN: OSC3B Enable Bit

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

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

D1 OSC1EN: OSC1 Enable Bit

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

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

D0 OSC3AEN: OSC3A Enable Bit

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

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

FOUTA Control Register (CLG_FOUTA)

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

D7 Reserved

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

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

Table 7.8.5 Clock Division Ratio Selection

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

(Default: 0x0)

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

Selects the FOUTA clock source.

Table 7.8.6 FOUTA Clock Source Selection

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

(Default: 0x0)

D1 Reserved

D0 FOUTAE: FOUTA Output Enable Bit

Enables or disables FOUTA clock external output.

1 (R/W): Enabled (on)

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

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

FOUTB Control Register (CLG_FOUTB)

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

D7 Reserved

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

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

Table 7.8.7 Clock Division Ratio Selection

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

(Default: 0x0)

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

Selects the FOUTB clock source.

Table 7.8.8 FOUTB Clock Source Selection

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

(Default: 0x0)

D1 Reserved

D0 FOUTBE: FOUTB Output Enable Bit

Enables or disables FOUTB clock external output.

1 (R/W): Enabled (on)

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

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

Oscillation Stabilization Wait Control Register (CLG_WAIT)

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

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

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

Table 7.8.9 OSC3B Oscillation Stabilization Wait Time Settings

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

(Default: 0x0)

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

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

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

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

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

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

Table 7.8.10 OSC3A Oscillation Stabilization Wait Time Settings

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

(Default: 0x0)

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

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

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

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

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

Table 7.8.11 OSC1B Oscillation Stabilization Wait Time Settings

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

(Default: 0x0)

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

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

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

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

Table 7.8.12 OSC1A Oscillation Stabilization Wait Time Settings

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

(Default: 0x0)

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

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

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

PCLK Control Register (CLG_PCLK)

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

D[7:2] Reserved

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

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

Table 7.8.13 PCLK Control

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

(Default: 0x3)

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

Peripheral modules that use PCLK

- Interrupt controller
- 8-bit timer Ch.0
- SPI Ch.0
- Power generator
- P port & port MUX
- MISC registers

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

Peripheral modules/functions that do not use PCLK

- Real-time clock
- Clock timer
- Watchdog timer
- LCD driver
- Sound generator
- SVD circuit
- 16-bit PWM timer Ch.0
- UART Ch.0
- FOUTA/FOUTB outputs

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

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

CCLK Control Register (CLG_CCLK)

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

D[7:2] Reserved

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

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

Table 7.8.14 CCLK Gear Ratio Selection

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

(Default: 0x0)

8 Theoretical Regulation (TR)

8.1 TR Module Overview

The S1C17651 has a theoretical regulation function that theoretically corrects time clock errors due to deviation in oscillation frequencies.

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

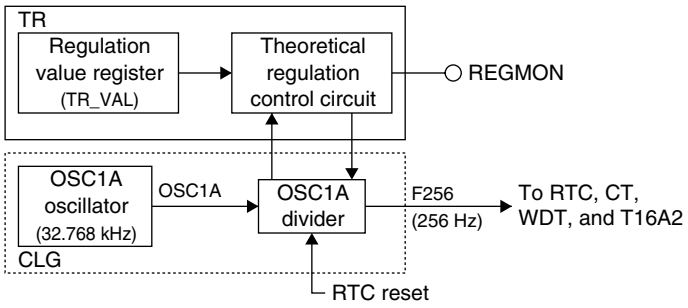


Figure 8.1.1 TR Module Configuration

8.2 TR Output Pin

Table 8.2.1 shows the TR output pin.

Table 8.2.1 TR Output Pin

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

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

8.3 Theoretical Regulation Control

8.3.1 Setting Regulation Values

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

Table 8.3.1.1 Regulation Value Settings

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

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

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

8.3.2 Executing Theoretical Regulation

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

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

8.3.3 Regulated Clock External Monitor

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

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

8.4 Control Register Details

Table 8.4.1 List of TR Registers

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

The TR module registers are described in detail below.

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

TR Control Register (TR_CTL)

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

D[7:4] Reserved

D3 RCLKFSEL: Monitor Clock Frequency Select Bit

Selects the regulated clock to be output from the REGMON pin for monitoring.
1 (R/W): F1 (1 Hz)
0 (R/W): F256 (256 Hz) (default)

D2 RCLKMON: Regulated Clock Monitor Enable Bit

Controls the clock monitor output from the REGMON pin.
1 (R/W): Enabled (On)
0 (R/W): Disabled (Off) (default)

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

D1 Reserved

D0 REGTRIG: Regulation Trigger Bit

Executes theoretical regulation.
1 (W): Trigger
0 (W): Ignored (default)

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

TR Value Register (TR_VAL)

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

D[7:5] Reserved

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

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

Table 8.4.2 Regulation Value Settings

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

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

9 Real-Time Clock (RTC)

9.1 RTC Module Overview

The S1C17651 incorporates a real-time clock (RTC).
The main features of the RTC are outlined below.

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

Figure 9.1.1 shows a block diagram of the RTC.

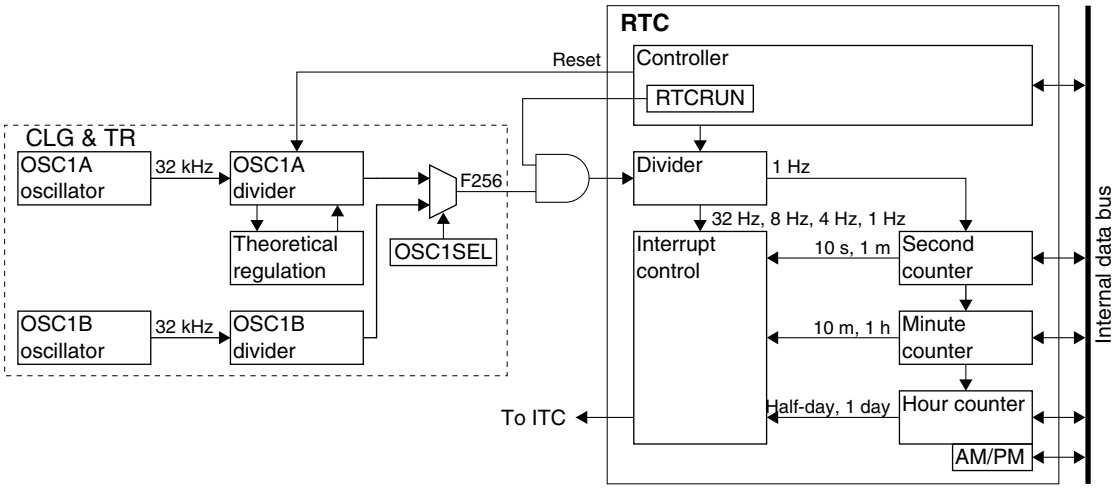


Figure 9.1.1 RTC Block Diagram

9.2 RTC Counters

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

Second counter

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

	RTCSEC[6:0]
Binary mode	0–59

	RTCSEC[6:4]	RTCSEC[3:0]
BCD mode	0–5 (10-second digit)	0–9 (1-second digit)

Figure 9.2.1 Second Counter

Minute counter

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

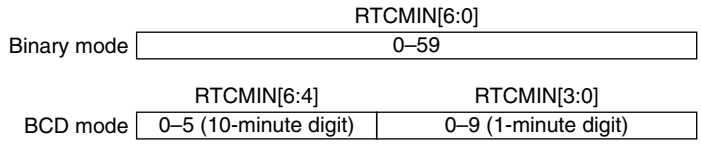


Figure 9.2.2 Minute Counter

Hour counter

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

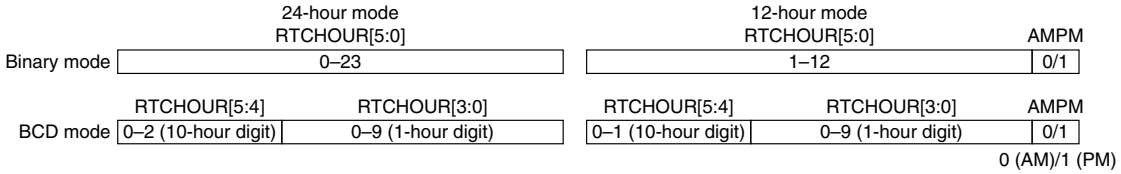


Figure 9.2.3 Hour Counter

Table 9.2.1 Hour Counter Values

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

Initial counter values

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

9.3 RTC Control

9.3.1 Operating Clock Control

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

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

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

9.3.2 12-hour/24-hour mode selection

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

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

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

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

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

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

9.3.3 RTC Start/Stop

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

9.3.4 Counter Settings

Counter values should be set in the procedure shown below.

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

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

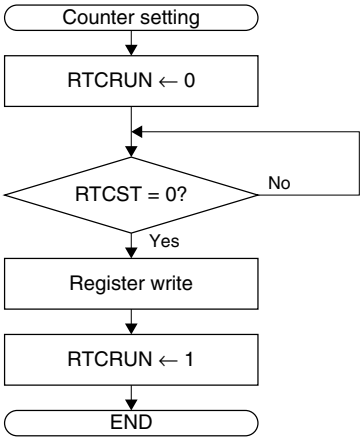


Figure 9.3.4.1 Procedure for Setting Counters

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

9.3.5 Counter Read

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

Read procedure 1

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

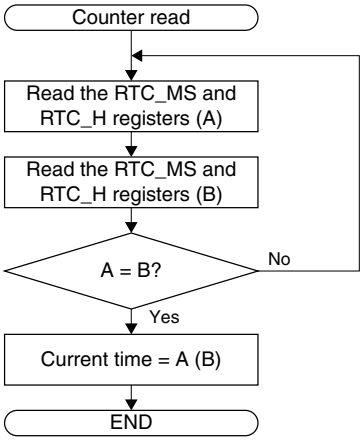


Figure 9.3.5.1 Procedure for Reading Counters

Read procedure 2

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

9.4 RTC Interrupts

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

Table 9.4.1 Interrupt Cycles and Interrupt Control Bits

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

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

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

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

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

9.5 Control Register Details

Table 9.5.1 List of RTC Registers

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

The following describes each RTC register.

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

RTC Control Register (RTC_CTL)

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

D[15:9] Reserved

D8 RTCST: RTC Run/Stop Status Bit

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

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

D[7:6] Reserved

D5 BCDMD: BCD Mode Select Bit

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

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

D4 RTC24H: 24H/12H Mode Select Bit

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

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

D[3:1] Reserved

D0 RTCRUN: RTC Run/Stop Control Bit

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

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

RTC Interrupt Enable Register (RTC_IEN)

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

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

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

D[15:10] Reserved

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

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

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

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

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

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

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

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

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

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

RTC Interrupt Flag Register (RTC_IFLG)

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

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

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

D[15:10] Reserved

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

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

D7 INT1H: 1-Hour Interrupt Flag Bit

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

D6 INT10M: 10-Minute Interrupt Flag Bit

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

D5 INT1M: 1-Minute Interrupt Flag Bit

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

D4 INT10S: 10-Second Interrupt Flag Bit

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

D3 INT1HZ: 1 Hz Interrupt Flag Bit

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

D2 INT4HZ: 4 Hz Interrupt Flag Bit

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

D1 INT8HZ: 8 Hz Interrupt Flag Bit

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

D0 INT32HZ: 32 Hz Interrupt Flag Bit

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

RTC Minute/Second Counter Register (RTC_MS)

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

D15 Reserved

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

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

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

D7 Reserved

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

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

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

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

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

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

RTC Hour Counter Register (RTC_H)

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

D[15:8] Reserved

D7 AMPM: AM/PM Bit

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

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

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

D6 Reserved

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

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

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

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

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

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

10 I/O Ports (P)

10.1 P Module Overview

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

- Maximum 12 I/O ports (P0[7:0], P1[3:0]) are available.
  * The number of ports for general-purpose use depends on the peripheral functions used.
- Each port has a pull-up resistor that can be enabled with software.
- Input interface level: CMOS Schmitt
- The P0 ports can generate input interrupts at the signal edge selected with software.
- The P0 ports include a chattering filter.
- Can generate an initial reset by entering low level simultaneously to the P0 ports selected with software.
- All port provide a port function select bit to configure the pin function (for GPIO or peripheral functions).

Figure 10.1.1 shows the I/O port configuration.

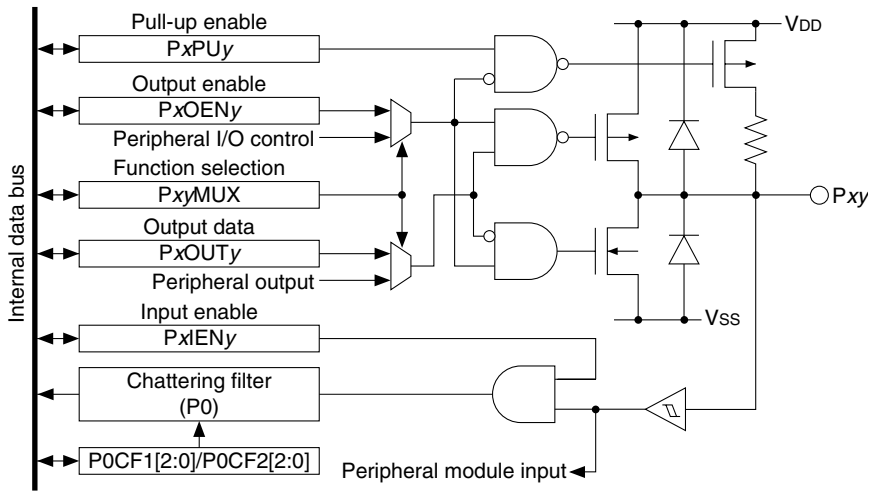


Figure 10.1.1 I/O Port Configuration

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

• The "*xy*" in the register and bit names refers to the port number (P*xy*, *x* = 0 and 1, *y* = 0 to 7).
Example: P*x*IN*y*/P*x*_IN register
P00: P0IN0/P0_IN register
P13: P1IN3/P1_IN register

10.2 Input/Output Pin Function Selection (Port MUX)

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

Table 10.2.1 Input/Output Pin Function Selection

Pin function 1 P*xy*MUX[1:0] = 0x0	Pin function 2 P*xy*MUX[1:0] = 0x1	Pin function 3 P*xy*MUX[1:0] = 0x2	Pin function 4 P*xy*MUX[1:0] = 0x3	Port function select bits
P00	SIN0 (UART)	–	–	P00MUX[1:0]/P00_03PMUX register
P01	SOUT0 (UART)	–	–	P01MUX[1:0]/P00_03PMUX register
P02	SCLK0 (UART)	FOUTA (CLG)	REGMON (TR)	P02MUX[1:0]/P00_03PMUX register
P03	EXCL0 (T16A2)	REGMON (TR)	LFRO (LCD)	P03MUX[1:0]/P00_03PMUX register
P04	TOUTA0/CAPA0 (T16A2)	–	–	P04MUX[1:0]/P04_07PMUX register
P05	TOUTB0/CAPB0 (T16A2)	#SPISS0 (SPI)	–	P05MUX[1:0]/P04_07PMUX register
P06	BZ (SND)	SDI0 (SPI)	–	P06MUX[1:0]/P04_07PMUX register
P07	#BZ (SND)	SDO0 (SPI)	–	P07MUX[1:0]/P04_07PMUX register
P10	FOUTB (CLG)	SPICLK0 (SPI)	–	P10MUX[1:0]/P10_13PMUX register
DCLK (DBG)	P11	BZ (SND)	–	P11MUX[1:0]/P10_13PMUX register
DSIO (DBG)	P12	#BZ (SND)	–	P12MUX[1:0]/P10_13PMUX register
DST2 (DBG)	P13	–	–	P13MUX[1:0]/P10_13PMUX register

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

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

10.3 Data Input/Output

Data input/output control

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

Table 10.3.1 Data Input/Output Status

P*x*OEN*y* output control	P*x*IEN*y* input control	P*x*PU*y* pull-up control	Port status
0	1	0	Functions as an input port (pull-up off). The port pin (external input signal) value can be read out from P*x*IN*y* (input data). Output is disabled.
0	1	1	Functions as an input port (pull-up on). (Default) The port pin (external input signal) value can be read out from P*x*IN*y* (input data). Output is disabled.
1	0	1 or 0	Functions as an output port (pull-up off). Input is disabled. The value read from P*x*IN*y* (input data) is 0.
1	1	1 or 0	Functions as an output port (pull-up off). Input is also enabled. The port pin value (output value) can be read out from P*x*IN*y* (input data).
0	0	0	The pin is placed into high-impedance status (pull-up off). Output and input are both disabled. The value read from P*x*IN*y* (input data) is 0.
0	0	1	The pin is placed into high-impedance status (pull-up on). Output and input are both disabled. The value read from P*x*IN*y* (input data) is 0.

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

Data input

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

Data output

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

10.4 Pull-up Control

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

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

10.5 Port Input Interrupt

The P0 ports include input interrupt functions.
Select which of the 8 ports are to be used for interrupts based on requirements. You can also select whether interrupts are generated for either the rising edge or falling edge of the input signals.
Figure 10.5.1 shows the port input interrupt circuit configuration.

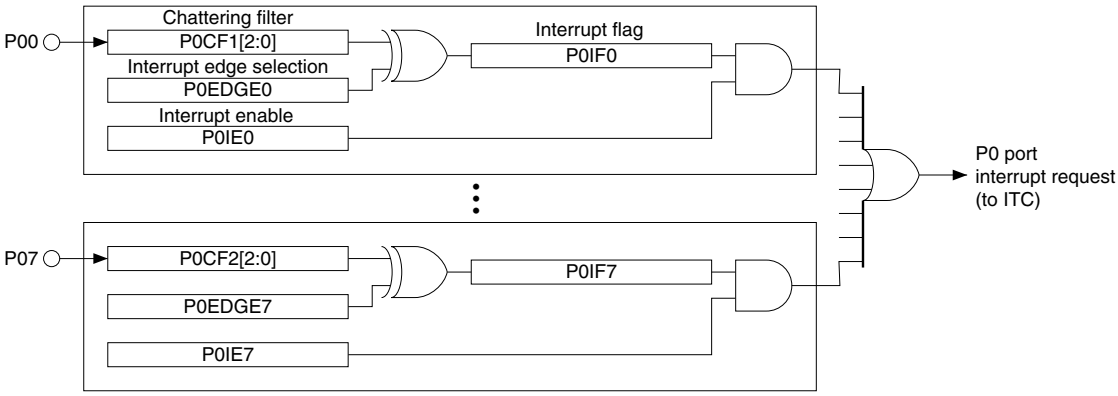


Figure 10.5.1 Port Input Interrupt Circuit Configuration

Interrupt port selection

Select the port generating an interrupt using P0IE*y*/P0_IMSK register.
Setting P0IE*y* to 1 enables interrupt generation by the corresponding port. Setting to 0 (default) disables interrupt generation.

Interrupt edge selection

Port input interrupts can be generated at either the rising edge or falling edge of the input signal. Select the edge used to generate interrupts using P0EDGE*y*/P0_EDGE register.
Setting P0EDGE*y* to 1 generates port input interrupts at the input signal falling edge. Setting it to 0 (default) generates interrupts at the rising edge.

Interrupt flags

The ITC is able to accept one interrupt request from the P0 ports, and the P port module contains interrupt flags P0IF*y*/P0_IFLG register corresponding to the individual eight ports to enable individual control of the eight P0[7:0] port interrupts. P0IF*y* is set to 1 at the specified edge (rising or falling edge) of the input signal. If the corresponding P0IE*y* has been set to 1, an interrupt request signal is also output to the ITC at the same time. An interrupt is generated if the ITC and S1C17 Core interrupt conditions are satisfied.
P0IF*y* is reset by writing 1.
For specific information on interrupt processing, see the "Interrupt Controller (ITC)" chapter.

Notes: • The P port module interrupt flag P0IF*y* must be reset in the interrupt handler routine after a port interrupt has occurred to prevent recurring interrupts.

• To prevent generating unnecessary interrupts, reset the relevant P0IF*y* before enabling interrupts for the required port using P0IE*y*.

10.6 P0 Port Chattering Filter Function

The P0 ports include a chattering filter circuit for key entry that can be disabled or enabled with a check time specified individually for the four P0[3:0] and P0[7:4] ports using P0CF1[2:0]/P0_CHAT register and P0CF2[2:0]/P0_CHAT register, respectively.

Table 10.6.1 Chattering Filter Function Settings

P0CF1[2:0]/P0CF2[2:0]	Check time *
0x7	16384/fPCLK (8 ms)
0x6	8192/fPCLK (4 ms)
0x5	4096/fPCLK (2 ms)
0x4	2048/fPCLK (1 ms)
0x3	1024/fPCLK (512 µs)
0x2	512/fPCLK (256 µs)
0x1	256/fPCLK (128 µs)
0x0	No check time (off)

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

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

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

• The P0 port interrupt must be disabled before setting the P0_CHAT register. Setting the register while the interrupt is enabled may generate inadvertent P0 port interrupt. Also the chattering filter circuit requires a maximum of twice the check time for stabilizing the operation status. Before enabling the interrupt, make sure that the stabilization time has elapsed.

10.7 P0 Port Key-Entry Reset

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

Table 10.7.1 Configuration of P0 Port Key-Entry Reset

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

(Default: 0x0)

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

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

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

10.8 Control Register Details

Table 10.8.1 List of I/O Port Control Registers

Address	Register name		Function
0x5200	P0_IN	P0 Port Input Data Register	P0 port input data
0x5201	P0_OUT	P0 Port Output Data Register	P0 port output data
0x5202	P0_OEN	P0 Port Output Enable Register	Enables P0 port outputs.
0x5203	P0_PU	P0 Port Pull-up Control Register	Controls the P0 port pull-up resistor.
0x5205	P0_IMSK	P0 Port Interrupt Mask Register	Enables P0 port interrupts.
0x5206	P0_EDGE	P0 Port Interrupt Edge Select Register	Selects the signal edge for generating P0 port interrupts.
0x5207	P0_IFLG	P0 Port Interrupt Flag Register	Indicates/resets the P0 port interrupt occurrence status.
0x5208	P0_CHAT	P0 Port Chattering Filter Control Register	Controls the P0 port chattering filter.
0x5209	P0_KRST	P0 Port Key-Entry Reset Configuration Register	Configures the P0 port key-entry reset function.
0x520a	P0_IEN	P0 Port Input Enable Register	Enables P0 port inputs.
0x5210	P1_IN	P1 Port Input Data Register	P1 port input data
0x5211	P1_OUT	P1 Port Output Data Register	P1 port output data
0x5212	P1_OEN	P1 Port Output Enable Register	Enables P1 port outputs.
0x5213	P1_PU	P1 Port Pull-up Control Register	Controls the P1 port pull-up resistor.
0x521a	P1_IEN	P1 Port Input Enable Register	Enables P1 port inputs.
0x52a0	P00_03PMUX	P0[3:0] Port Function Select Register	Selects the P0[3:0] port functions.
0x52a1	P04_07PMUX	P0[7:4] Port Function Select Register	Selects the P0[7:4] port functions.
0x52a2	P10_13PMUX	P1[3:0] Port Function Select Register	Selects the P1[3:0] port functions.

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

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

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

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

Note: P1IN[3:0] only are available for the P1 ports. Other bits are reserved and always read as 0.

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

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

1 (R): High level

0 (R): Low level

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

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

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

Note: P1OUT[3:0] only are available for the P1 ports. Other bits are reserved and always read as 0.

D[7:0] P*x*OUT[7:0]: P*x*[7:0] Port Output Data Bits

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

P*x*OUT*y* corresponds directly to the P*xy* pins. The data written will be output unchanged from the port pins when output is enabled (P*x*OEN*y* = 1). The port pin will be High when the data bit is set to 1 and Low when set to 0.
Port data can also be written when output is disabled (P*x*OEN*y* = 0) (the pin status is unaffected).

P*x* Port Output Enable Registers (P*x*_OEN)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
P*x* Port Output Enable Register (P*x*_OEN)	**0x5202** **0x5212** (8 bits)	D7–0	**P*x*OEN[7:0]**	P*x*[7:0] port output enable	1	Enable	0	Disable	0	R/W	

Note: P1OEN[3:0] only are available for the P1 ports. Other bits are reserved and always read as 0.

D[7:0] P*x*OEN[7:0]: P*x*[7:0] Port Output Enable Bits

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

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

P*x* Port Pull-up Control Registers (P*x*_PU)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
P*x* Port Pull-up Control Register (P*x*_PU)	**0x5203** **0x5213** (8 bits)	D7–0	**P*x*PU[7:0]**	P*x*[7:0] port pull-up enable	1	Enable	0	Disable	1 (0xff)	R/W	

Note: P1PU[3:0] only are available for the P1 ports. Other bits are reserved and always read as 0.

D[7:0] P*x*PU[7:0]: P*x*[7:0] Port Pull-up Enable Bits

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

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

P0 Port Interrupt Mask Register (P0_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	

Note: This register is available for the P0 ports.

D[7:0] P0IE[7:0]: P0[7:0] Port Interrupt Enable Bits

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

Setting P0IE*y* to 1 enables the corresponding P0*y* port input interrupt, while setting to 0 disables the interrupt. Status changes for the input pins with interrupt disabled do not affect interrupt occurrence.

P0 Port Interrupt Edge Select Register (P0_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	

Note: This register is available for the P0 ports.

D[7:0] P0EDGE[7:0]: P0[7:0] Port Interrupt Edge Select Bits

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

Port interrupts are generated at the input signal falling edge when P0EDGE*y* is set to 1 and at the rising edge when set to 0.

P0 Port Interrupt Flag Register (P0_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.

Note: This register is available for the P0 ports.

D[7:0] P0IF[7:0]: P0[7:0] Port Interrupt Flag Bits

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

P0IF*y* is the interrupt flag corresponding to the individual eight P0 ports. It is set to 1 at the specified edge (rising or falling edge) of the input signal. When the corresponding P0IE*y*/P0_IMSK register has been set to 1, a port interrupt request signal is also output to the ITC at the same time. An interrupt is generated if the ITC and S1C17 Core interrupt conditions are satisfied.

P0IF*y* is reset by writing 1.

Notes: • The P port module interrupt flag P0IF*y* must be reset in the interrupt handler routine after a port interrupt has occurred to prevent recurring interrupts.

• To prevent generating unnecessary interrupts, reset the relevant P0IF*y* before enabling interrupts for the required port using P0IE*y*/P0_IMSK register.

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 select	P0CF2[2:0]	Filter time	0x0	R/W	
					0x7	16384/fPCLK			
					0x6	8192/fPCLK			
					0x5	4096/fPCLK			
					0x4	2048/fPCLK			
					0x3	1024/fPCLK			
					0x2	512/fPCLK			
					0x1	256/fPCLK			
					0x0	None			
		D3	–	reserved	–		–	–	0 when being read.
		D2–0	P0CF1[2:0]	P0[3:0] chattering filter time select	P0CF1[2:0]	Filter time	0x0	R/W	
					0x7	16384/fPCLK			
					0x6	8192/fPCLK			
					0x5	4096/fPCLK			
					0x4	2048/fPCLK			
					0x3	1024/fPCLK			
					0x2	512/fPCLK			
					0x1	256/fPCLK			
					0x0	None			

Note: This register is available for the P0 ports.

D7 Reserved

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

Configures the chattering filter circuit for the P0[7:4] ports.

D3 Reserved

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

Configures the chattering filter circuit for the P0[3:0] ports.

The P0 ports include a chattering filter circuit for key entry that can be disabled or enabled with a check time specified individually for the four P0[3:0] and P0[7:4] ports using P0CF1[2:0] and P0CF2[2:0], respectively.

Table 10.8.2 Chattering Filter Function Settings

P0CF1[2:0]/P0CF2[2:0]	Check time *
0x7	16384/fPCLK (8 ms)
0x6	8192/fPCLK (4 ms)
0x5	4096/fPCLK (2 ms)
0x4	2048/fPCLK (1 ms)
0x3	1024/fPCLK (512 µs)
0x2	512/fPCLK (256 µs)
0x1	256/fPCLK (128 µs)
0x0	No check time (off)

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

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

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

- The P0 port interrupt must be disabled before setting the P0_CHAT register. Setting the register while the interrupt is enabled may generate inadvertent P0 interrupt. Also the chattering filter circuit requires a maximum of twice the check time for stabilizing the operation status. Before enabling the interrupt, make sure that the stabilization time has elapsed.

P0 Port Key-Entry Reset Configuration Register (P0_KRST)

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

D[7:2] Reserved

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

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

Table 10.8.3 P0 Port Key-Entry Reset Settings

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

(Default: 0x0)

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

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

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

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

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

Note: P1IEN[3:0] only are available for the P1 ports. Other bits are reserved and always read as 0.

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

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

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

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

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
P0[3:0] Port Function Select Register (P00_03PMUX)	0x52a0 (8 bits)	D7–6	P03MUX[1:0]	P03 port function select	P03MUX[1:0]	Function	0x0	R/W	
					0x3	LFRO			
					0x2	REGMON			
					0x1	EXCL0			
					0x0	P03			
		D5–4	P02MUX[1:0]	P02 port function select	P02MUX[1:0]	Function	0x0	R/W	
					0x3	REGMON			
					0x2	FOUTA			
					0x1	SCLK0			
					0x0	P02			
		D3–2	P01MUX[1:0]	P01 port function select	P01MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	SOUT0			
					0x0	P01			
		D1–0	P00MUX[1:0]	P00 port function select	P00MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	SIN0			
					0x0	P00			

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

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

0x3 (R/W): LFRO (LCD)
0x2 (R/W): REGMON (TR)
0x1 (R/W): EXCL0 (T16A2 Ch.0)
0x0 (R/W): P03 port (default)

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

0x3 (R/W): REGMON (TR)
0x2 (R/W): FOUTA (CLG)
0x1 (R/W): SCLK0 (UART)
0x0 (R/W): P02 port (default)

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

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): SOUT0 (UART)
0x0 (R/W): P01 port (default)

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

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): SIN0 (UART)
0x0 (R/W): P00 port (default)

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

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
P0[7:4] Port Function Select Register (P04_07PMUX)	**0x52a1** (8 bits)	D7–6	**P07MUX[1:0]**	P07 port function select	P07MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	SDO0			
					0x1	#BZ			
					0x0	P07			
		D5–4	**P06MUX[1:0]**	P06 port function select	P06MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	SDI0			
					0x1	BZ			
					0x0	P06			
		D3–2	**P05MUX[1:0]**	P05 port function select	P05MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	#SPISS0			
					0x1	TOUTB0/CAPB0			
					0x0	P05			
		D1–0	**P04MUX[1:0]**	P04 port function select	P04MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	TOUTA0/CAPA0			
					0x0	P04			

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

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

0x3 (R/W): Reserved
0x2 (R/W): SDO0 (SPI Ch.0)
0x1 (R/W): #BZ (SND)
0x0 (R/W): P07 port (default)

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

0x3 (R/W): Reserved
0x2 (R/W): SDI0 (SPI Ch.0)
0x1 (R/W): BZ (SND)
0x0 (R/W): P06 port (default)

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

0x3 (R/W): Reserved
0x2 (R/W): #SPISS0 (SPI Ch.0)
0x1 (R/W): TOUTB0 (T16A2 Ch.0 comparator mode) or CAPB0 (T16A2 Ch.0 capture mode)
0x0 (R/W): P05 port (default)

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

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): TOUTA0 (T16A2 Ch.0 comparator mode) or CAPA0 (T16A2 Ch.0 capture mode)
0x0 (R/W): P04 port (default)

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

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
P1[3:0] Port Function Select Register (P10_13PMUX)	**0x52a2** (8 bits)	D7–6	**P13MUX[1:0]**	P13 port function select	P13MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	P13			
					0x0	DST2			
		D5–4	**P12MUX[1:0]**	P12 port function select	P12MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	#BZ			
					0x1	P12			
					0x0	DSIO			
		D3–2	**P11MUX[1:0]**	P11 port function select	P11MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	BZ			
					0x1	P11			
					0x0	DCLK			
		D1–0	**P10MUX[1:0]**	P10 port function select	P10MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	SPICLK0			
					0x1	FOUTB			
					0x0	P10			

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

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

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): P13 port
0x0 (R/W): DST2 (DBG) (default)

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

0x3 (R/W): Reserved
0x2 (R/W): #BZ (SND)
0x1 (R/W): P12 port
0x0 (R/W): DSIO (DBG) (default)

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

0x3 (R/W): Reserved
0x2 (R/W): BZ (SND)
0x1 (R/W): P11 port
0x0 (R/W): DCLK (DBG) (default)

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

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

11 8-bit Timer (T8)

11.1 T8 Module Overview

The S1C17651 includes an 8-bit timer module (T8).
The features of the T8 module are listed below.

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

Figure 11.1.1 shows the T8 configuration.

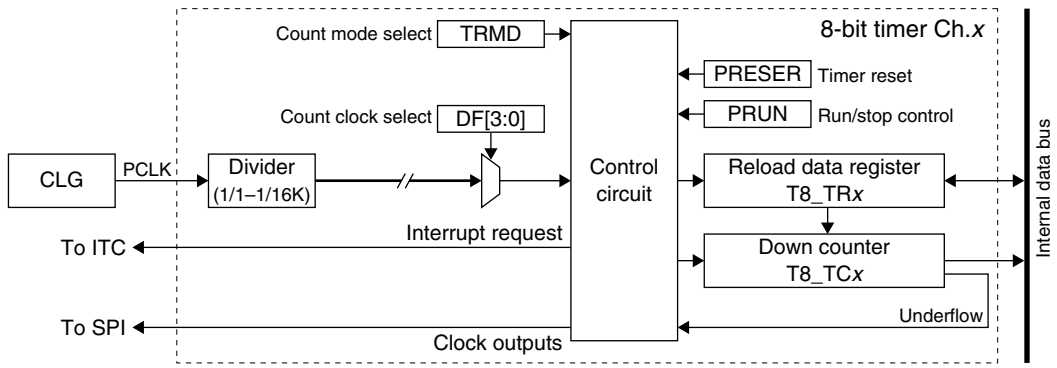


Figure 11.1.1 T8 Configuration

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

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

11.2 Count Clock

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

Table 11.2.1 PCLK Division Ratio Selection

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

(Default: 0x0)

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

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

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

11.3 Count Mode

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

Repeat mode (TRMD = 0, default)

Setting TRMD to 0 sets T8 to repeat mode.

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

One-shot mode (TRMD = 1)

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

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

11.4 Reload Data Register and Underflow Cycle

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

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

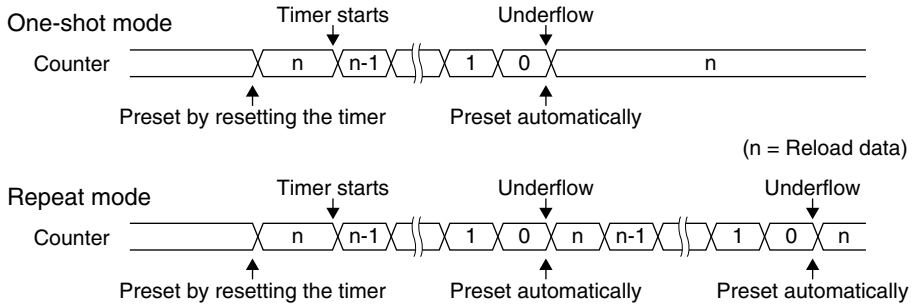


Figure 11.4.1 Preset Timing

The underflow cycle can be calculated as follows:

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

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

11.5 Timer Reset

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

11.6 Timer RUN/STOP Control

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

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

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

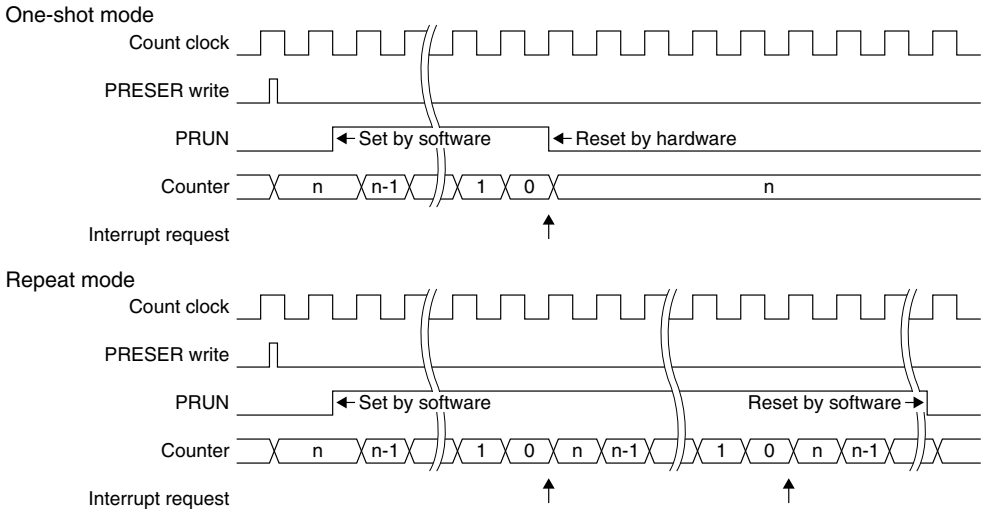


Figure 11.6.1 Count Operation

11.7 T8 Output Signals

The T8 module outputs underflow pulses when the counter underflows.
These pulses are used for timer interrupt requests.
These pulses are also used to generate the serial transfer clock for the internal serial interface.
The clock generated is sent to the SPI module.
Use the following equations to calculate the reload data register value for obtaining the desired transfer rate:

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

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

11.8 T8 Interrupts

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

Underflow interrupt

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

Notes: • The T8 module interrupt flag T8IF must be reset in the interrupt handler routine after a T8 interrupt has occurred to prevent recurring interrupts.

• Reset T8IF before enabling T8 interrupts with T8IE to prevent occurrence of unwanted interrupt. T8IF is reset by writing 1.

11.9 Control Register Details

Table 11.9.1 List of T8 Registers

Address	Register name		Function
0x4240	T8_CLK0	T8 Ch.0 Count Clock Select Register	Selects a count clock.
0x4242	T8_TR0	T8 Ch.0 Reload Data Register	Sets reload data.
0x4244	T8_TC0	T8 Ch.0 Counter Data Register	Counter data
0x4246	T8_CTL0	T8 Ch.0 Control Register	Sets the timer mode and starts/stops the timer.
0x4248	T8_INT0	T8 Ch.0 Interrupt Control Register	Controls the interrupt.

The T8 registers are described in detail below.

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

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

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

D[15:4] Reserved

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

Selects a PCLK division ratio to generate the count clock.

Table 11.9.2 PCLK Division Ratio Selection

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

(Default: 0x0)

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

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

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

D[15:8] Reserved

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

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

The reload data set in this register is preset to the counter when the timer is reset or the counter underflows. If the timer is started after resetting, it counts down from the reload value (initial value).

This means that the reload value and the input clock frequency determine the time elapsed from the point at which the timer starts until the underflow occurs (or between underflows). The time determined is used to obtain the desired wait time, the intervals between periodic interrupts, and the programmable serial interface transfer clock.

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

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

D[15:8] Reserved

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

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

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

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

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

D4 TRMD: Count Mode Select Bit

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

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

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

D[3:2] Reserved

D1 PRESER: Timer Reset Bit

Resets the timer.
1 (W): Reset
0 (W): Ignored
0 (R): Always 0 when read (default)

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

D0 PRUN: Timer Run/Stop Control Bit

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

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

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

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

D[15:9] Reserved

D8 T8IE: T8 Interrupt Enable Bit

Enables or disables interrupts caused by counter underflows.

1 (R/W): Interrupt enabled

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

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

D[7:1] Reserved

D0 T8IF: T8 Interrupt Flag Bit

Indicates whether the cause of counter underflow interrupt has occurred or not.

1 (R): Cause of interrupt has occurred

0 (R): No cause of interrupt has occurred (default)

1 (W): Flag is reset

0 (W): Ignored

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

T8IF is reset by writing 1.

12 16-bit PWM Timer (T16A2)

12.1 T16A2 Module Overview

The S1C17651 includes a 16-bit PWM timer (T16A2) module that consists of counter blocks and comparator/capture blocks. This timer can be used as an interval timer, PWM waveform generator, external event counter and a count capture unit to measure external event periods.
The features of T16A2 are listed below.

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

Figure 12.1.1 shows the T16A2 configuration.

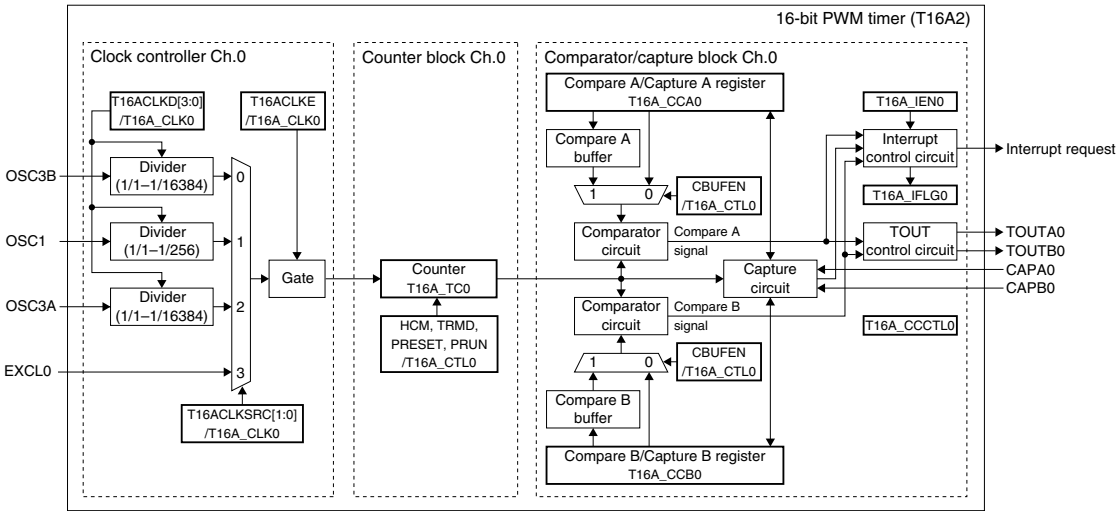


Figure 12.1.1 T16A2 Configuration

Clock controller

T16A2 includes a clock controller that generates the count clock for the counter. The clock source and division ratio can be selected with software.

Counter block

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

Comparator/capture block

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

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

12.2 T16A2 Input/Output Pins

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

Table 12.2.1 List of T16A2 Pins

Pin name	I/O	Qty	Function
EXCL0 (for Ch.0)	I	1	External clock input pins Inputs an external clock for the event counter function.
CAPA0, CAPB0 (for Ch.0)	I	2	Counter-capture trigger signal input pins (effective in capture mode) The specified edge (falling edge, rising edge, or both) of the signal input to the CAPA*x* pin captures the counter data into the capture A register. The CAPB*x* pin input signal captures the counter data into the capture B register.
TOUTA0, TOUTB0 (for Ch.0)	O	2	Timer generating signal output pins (effective in comparator mode) T16A2 has two output pins and the signals generated in different conditions can be output.

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

12.3 Count Clock

The clock controller includes a clock source selector, dividers, and a gate circuit for controlling the count clock.

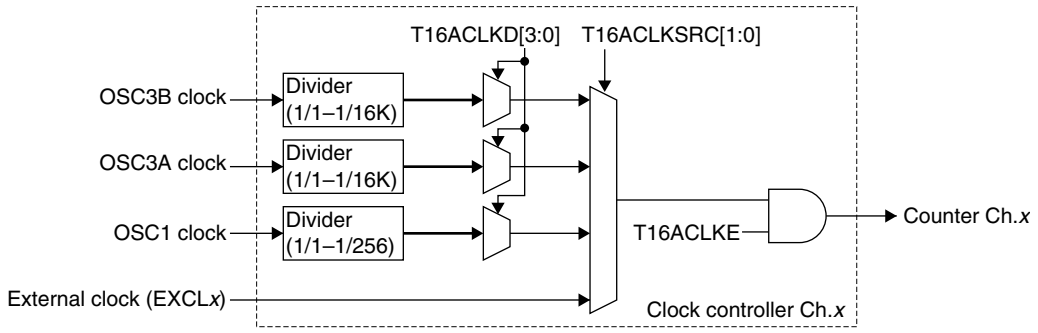


Figure 12.3.1 Clock Controller

Clock source selection

The clock source can be selected from OSC3B, OSC3A, OSC1, or external clock using T16ACLKSRC[1:0]/T16A_CLK*x* register.

Table 12.3.1 Clock Source Selection

T16ACLKSRC[1:0]	Clock source
0x3	External clock (EXCL*x*)
0x2	OSC3A
0x1	OSC1
0x0	OSC3B

(Default: 0x0)

When external clock is selected, the timer can be used as an event counter or for measuring pulse widths by inputting an external clock or pulses. The table below lists the external clock input pins. It is not necessary to switch their pin functions from general-purpose I/O port. However, do not set the I/O port to output mode.

Table 12.3.2 External Clock Input Pins

Channel	External clock input pin
T16A2 Ch.0	EXCL0

Internal clock division ratio selection

When an internal clock (OSC3B, OSC3A, or OSC1) is selected, use T16ACLKD[3:0]/T16A_CLK*x* register to select the division ratio.

Table 12.3.3 Internal Clock Division Ratio Selection

T16ACLKD[3:0]	Division ratio	
	Clock source = OSC3B or OSC3A	Clock source = OSC1
0xf	Reserved	
0xe	1/16384	Reserved
0xd	1/8192	Reserved
0xc	1/4096	Reserved
0xb	1/2048	Reserved
0xa	1/1024	Reserved
0x9	1/512	F256 (Regulated 256 Hz clock)
0x8	1/256	
0x7	1/128	
0x6	1/64	
0x5	1/32	
0x4	1/16	
0x3	1/8	
0x2	1/4	
0x1	1/2	
0x0	1/1	

(Default: 0x0)

Clock enable

Clock supply to the counter is controlled using T16ACLKE/T16A_CLK*x* register. The T16ACLKE default setting is 0, which disables the clock supply. Setting T16ACLKE to 1 sends the clock generated as above to the counter. If T16A2 is not required, disable the clock supply to reduce current consumption.

Note: Make sure the T16A2 count is stopped before setting the count clock.

12.4 T16A2 Operating Modes

The T16A2 module provides some operating modes to support various usages. This section describes the functions of each operating mode and how to enter the mode.

12.4.1 Comparator Mode and Capture Mode

The T16A_CCA*x* and T16A_CCB*x* registers that are embedded in the comparator/capture block can be set to comparator mode or capture mode, individually. The T16A_CCA*x* register mode is selected using CCAMD/T16A_CCCTL*x* register and the T16A_CCB*x* register mode is selected using CCBMD/T16A_CCCTL*x* register.

Comparator mode (CCAMD/CCBMD = 0, default)

The comparator mode compares the counter value and the comparison value set via software. It generates an interrupt and toggles the timer output signal level when the values are matched. The T16A_CCA*x* and T16A_CCB*x* registers function as the compare A and compare B registers that are used for loading compare values in this mode.

When the counter reaches the value set in the compare A register during counting, the comparator asserts the compare A signal. At the same time the compare A interrupt flag is set and the interrupt signal of the timer channel is output to the ITC if the interrupt has been enabled.

When the counter reaches the value set in the compare B register, the comparator asserts the compare B signal. At the same time the compare B interrupt flag is set and the interrupt signal of the timer channel is output to the ITC if the interrupt is enabled. Furthermore, the counter is reset to 0.

The compare A period (time from start of counting to occurrence of a compare A interrupt) and the compare B period (time from start of counting to occurrence of a compare B interrupt) can be calculated as follows:

Compare A period = (CCA + 1) / ct_clk [second]
Compare B period = (CCB + 1) / ct_clk [second]

CCA: Compare A register value set (0 to 65535)
CCB: Compare B register value set (0 to 65535)
ct_clk: Count clock frequency [Hz]

The compare A and compare B signals are also used to generate a timer output waveform (TOUT). See Section 12.6, "Timer Output Control," for more information.

To generate PWM waveform, the T16A_CCA*x* and T16A_CCB*x* registers must be both placed into comparator mode.

Compare buffers

The compare buffer is used to synchronize the comparison data update timings and the counter operation. Setting CBUFEN/T16A_CTL*x* register to 1 enables the compare buffer. The compare A and B signals will be generated by comparing the counter values with the compare A and B buffer values instead of the compare A and B register values. The compare A and B register values written via software are loaded to the compare A and B buffers when the compare B signal is generated.

Note: When writing data to the T16A_CCA*x* or T16A_CCB*x* register successively, data should be written at intervals of one or more T16A2 count clock cycles.

Capture mode (CCAMD/CCBMD = 1)

The capture mode captures the counter value when an external event such as a key entry occurs (at the specified edge of the external input signal). In this mode, the T16A_CCA*x* and/or T16A_CCB*x* registers function as the capture A and/or capture B registers.

The table below lists the input pins of the external trigger signals used for capturing counter values. The pin function of the corresponding ports must be switched for trigger input in advance. See the "I/O Ports (P)" chapter for switching the pin function.

Table 12.4.1.1 List of Counter Capture Trigger Signal Input Pins

Channel	Trigger input pins	
	Capture A	Capture B
T16A2 Ch.0	CAPA0	CAPB0

The trigger edge of the signal can be selected using the CAPATRG[1:0]/T16A_CCCTL*x* register for capture A and CAPBTRG[1:0]/T16A_CCCTL*x* register for capture B.

Table 12.4.1.2 Capture Trigger Edge Selection

CAPATRG[1:0]/ CAPBTRG[1:0]	Trigger edge
0x3	Falling edge and rising edge
0x2	Falling edge
0x1	Rising edge
0x0	Not triggered

(Default: 0x0)

When a specified trigger edge is input during counting, the current counter value is loaded to the capture register. At the same time the capture A or capture B interrupt flag is set and the interrupt signal of the timer channel is output to the ITC if the interrupt has been enabled. This interrupt can be used to read the captured data from the T16A_CCA*x* or T16A_CCB*x* register. For example, external event cycles and pulse widths can be measured from the difference between two captured counter values read.
If the captured data is overwritten by the next trigger when the capture A or capture B interrupt flag has already been set, the overwrite interrupt flag will be set. This interrupt can be used to execute an overwrite error handling. To avoid occurrence of unnecessary overwrite interrupt, the capture A or capture B interrupt flag must be reset after the captured data has been read from the T16A_CCA*x* or T16A_CCB*x* register.

Notes:
- The correct captured data may not be obtained if the captured data is read at the same time the next value is being captured. Read the capture register twice to check if the read data is correct as necessary.
- To capture counter data properly, both the High and Low period of the CAP*x* trigger signal must be longer than the source clock cycle time.

The setting of CAPATRG[1:0] or CAPBTRG[1:0] is ineffective in comparator mode. No counter capturing operation will be performed even if a trigger edge is specified.
The capture mode cannot generate/output the TOUT signal as no compare signal is generated.

12.4.2 Repeat Mode and One-Shot Mode

T16A2 features two count modes: repeat mode and one-shot mode. The count mode is selected using TRMD / T16A_CTL*x* register.

Repeat mode (TRMD = 0, default)

Setting TRMD to 0 sets the corresponding counter to repeat mode.
In this mode, once the count starts, the counter continues running until stopped by the application program. The counter continues the count even if the counter returns to 0 due to a counter overflow. The counter should be set to this mode to generate periodic interrupts at desired intervals or to generate a timer output waveform.

One-shot mode (TRMD = 1)

Setting TRMD to 1 sets the corresponding counter to one-shot mode.
In this mode, the counter stops automatically as soon as the compare B signal is generated. The counter should be set to this mode to set a specific wait time or for pulse width measurement.

12.4.3 Normal Clock Mode and Half Clock Mode

T16A2 supports half clock mode to control the duty ratio of the PWM output waveform with high accuracy. In half clock mode, T16A2 uses the dual-edge counter, which counts at the rising and falling edges of the count clock, to compare with the compare A register. This makes it possible to control the duty ratio with double accuracy as compared to normal clock mode.
Use HCM/T16A_CTL*x* register to select half clock mode.

Normal clock mode (HCM = 0, default)

In normal clock mode, T16A2 generates a compare A signal when the T16A_TC*x* register value matches the T16A_CCA*x* register.

Half clock mode (HCM = 1)

In half clock mode, T16A2 generates a compare A signal when the dual-edge counter value matches the T16A_CCA*x* register.

Notes:
- T16A2 must be placed into comparator mode to set half clock mode, as it is effective only when PWM waveform is generated.
  Be sure to set T16A2 to normal clock mode (HCM = 0) under a condition shown below.
  (1) When T16A2 is placed into capture mode
  (2) When TOUTAMD/T16A_CCCTL*x* register is set to 0x2 or 0x3
  (3) When TOUTBMD/T16A_CCCTL*x* register is set to 0x2 or 0x3
- The dual-edge counter value cannot be read.
- Do not use the compare A interrupt in half clock mode.
- In half clock mode, the T16A_CCB*x* register setting value must be less than [T16A_CCA*x* setting value / 2 + 0x8000].

12.5 Counter Control

12.5.1 Counter Reset

The counter can be reset to 0 by writing 1 to PRESET/T16A_CTL*x* register.

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

12.5.2 Counter RUN/STOP Control

Make the following settings before starting the count operation.
(1) Switch the input/output pin functions to be used for T16A2. Refer to the "I/O Port (P)" chapter.
(2) Select operating modes. See Section 12.4.
(3) Select the clock source. See Section 12.3.
(4) Configure the timer outputs (TOUT). See Section 12.6.
(5) If using interrupts, set the interrupt level and enable the T16A2 interrupts. See Section 12.7.
(6) Reset the counter to 0. See Section 12.5.1.
(7) Set comparison data (in comparator mode). See Section 12.4.1.

T16A2 provides PRUN/T16A_CTL*x* register to control the counter operation.
The counter starts counting when 1 is written to PRUN. Writing 0 to PRUN disables clock input and stops the count.
This control does not affect the counter data. The counter data is retained even when the count is halted, allowing resumption of the count from that data.
If PRUN and PRESET are written as 1 simultaneously, the counter starts counting after reset.

12.5.3 Reading Counter Values

The counter value can be read from T16A2TC[15:0]/T16A_TC*x* register even if the counter is running. However, the counter value should be read at once using a 16-bit transfer instruction. If data is read twice using an 8-bit transfer instruction, the correct value may not be obtained due to occurrence of count up between readings.

12.5.4 Counter Operation and Interrupt Timing Charts

Comparator mode

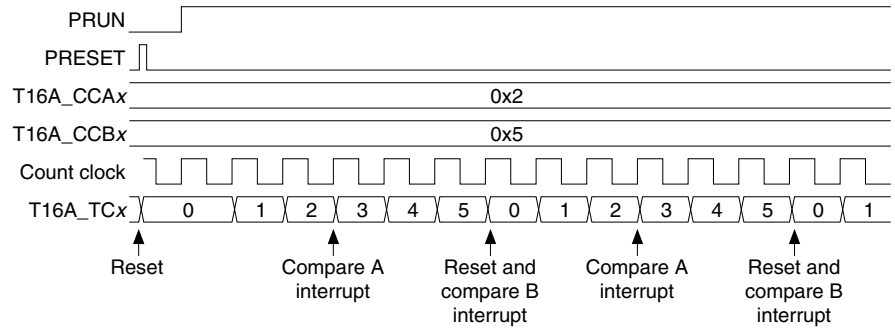


Figure 12.5.4.1 Operation Timing in Comparator Mode

Capture mode

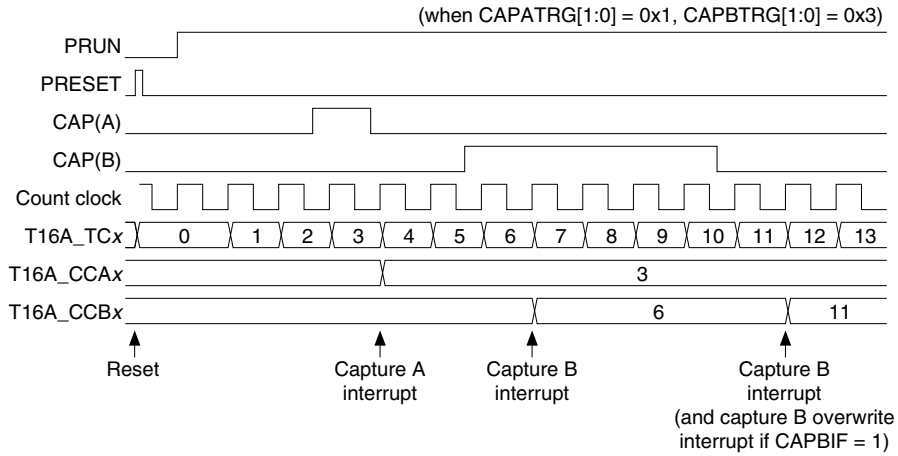


Figure 12.5.4.2 Operation Timing in Capture Mode

12.6 Timer Output Control

The timer that has been set in comparator mode can generate TOUT signals using the compare A and compare B signals and can output it to external devices. T16A2 provides two TOUT outputs, thus the T16A2 module can output up to four TOUT signals. Figure 12.6.1 shows the TOUT output circuit.

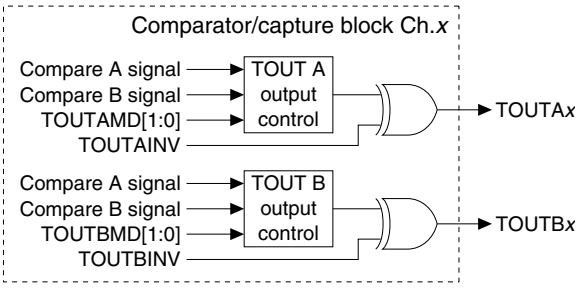


Figure 12.6.1 TOUT Output Circuit

T16A2 includes two TOUT output circuits and their signal generation and output can be controlled individually. Although the output circuit and register names use letters 'A' and 'B' to distinguish two systems, it does not mean that they correspond to compare A and B signals.

TOUT output pins

Table 12.6.1 lists correspondence between the TOUT pins and the timer channels. The pin function of the corresponding ports must be switched for TOUT output in advance. See the "I/O Ports (P)" chapter for switching the pin function.

Table 12.6.1 List of TOUT Output Pins

Channel	TOUT output pin	
	System A	System B
T16A2 Ch.0	TOUTA0	TOUTB0

TOUT generation mode

TOUTAMD[1:0]/T16A_CCCTL*x* register (for system A) or TOUTBMD[1:0]/T16A_CCCTL*x* register (for system B) is used to set how the TOUT signal is changed by the compare A and compare B signals.

Table 12.6.2 TOUT Generation Mode

TOUTAMD[1:0]/ TOUTBMD[1:0]	When compare A occurs	When compare B occurs
0x3	No change	Toggle
0x2	Toggle	No change
0x1	Rise	Fall
0x0	Disable output	

(Default: 0x0)

TOUTAMD[1:0] and TOUTBMD[1:0] are also used to turn the TOUT outputs On and Off.

TOUT signal polarity selection

By default, an active High output signal is generated. This logic can be inverted using TOUTAINV/T16A_CCCTL*x* register (for system A) or TOUTBINV/T16A_CCCTL*x* register (for system B). Writing 1 to TOUTAINV/TOUTBINV sets the timer to generate an active Low TOUT signal.

Resetting the counter sets the TOUT signal to the inactive level.

Figure 12.6.2 illustrates the TOUT output waveform.

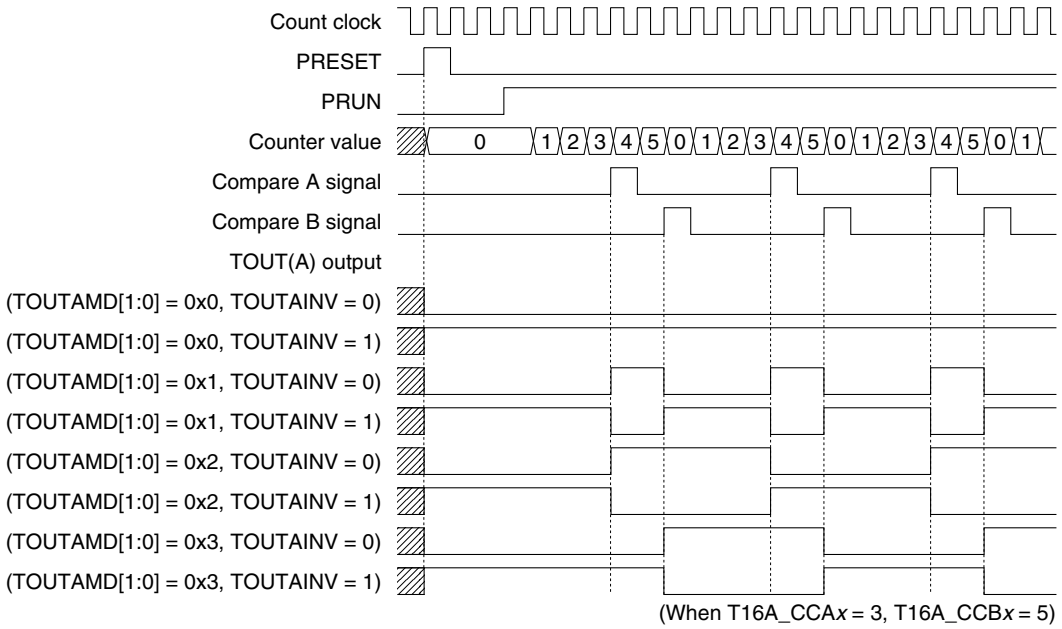
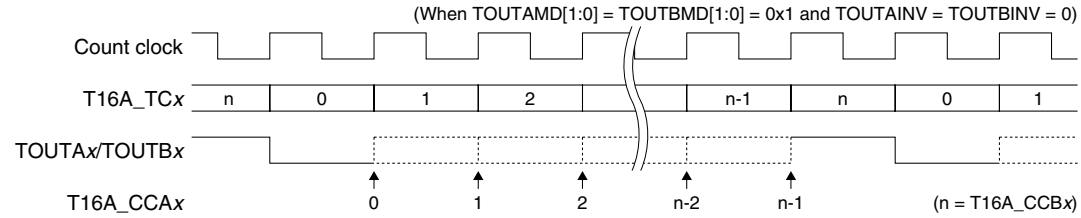


Figure 12.6.2 TOUT Output Waveform

PWM waveform output timings

Normal clock mode (HCM = 0)



Example: HCM = 0, T16A_CCA*x* = 1, and T16A_CCB*x* = 5

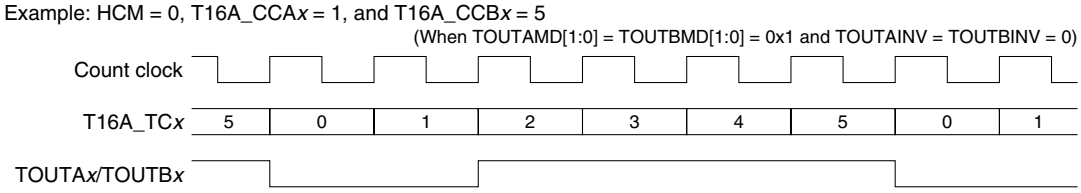
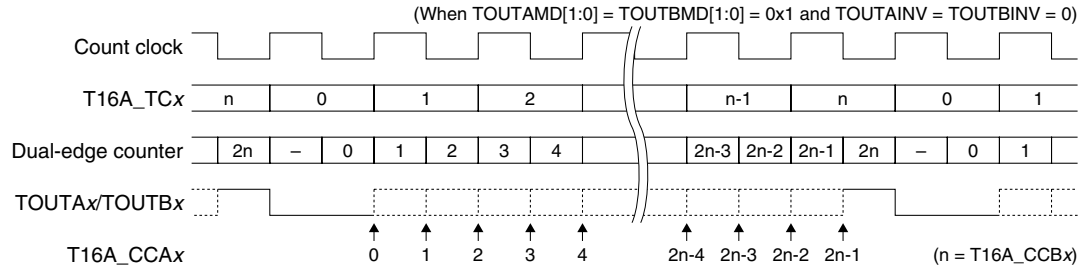


Figure 12.6.3 PWM Waveform Output Timings in Normal Clock Mode

Half clock mode (HCM = 1)



Example: HCM = 1, T16A_CCA*x* = 1, and T16A_CCB*x* = 5

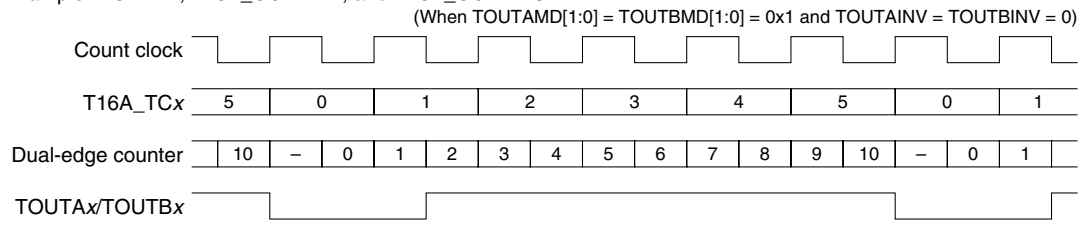


Figure 12.6.4 PWM Waveform Output Timings in Half Clock Mode

12.7 T16A2 Interrupts

The T16A2 module can generate the following six kinds of interrupts:

- Compare A interrupt (in comparator mode)
- Compare B interrupt (in comparator mode)
- Capture A interrupt (in capture mode)
- Capture B interrupt (in capture mode)
- Capture A overwrite interrupt (in capture mode)
- Capture B overwrite interrupt (in capture mode)

The T16A2 module outputs a single interrupt signal shared by the above interrupt causes to the interrupt controller (ITC). Read the interrupt flags in the T16A2 module to identify the interrupt cause that has been occurred.

Interrupts in comparator mode

Compare A interrupt

This interrupt request is generated when the counter matches the compare A register value during counting in comparator mode. It sets the interrupt flag CAIF/T16A_IFLG*x* register in the T16A2 module to 1.
To use this interrupt, set CAIE/T16A_IEN*x* register to 1. If CAIE is set to 0 (default), interrupt requests for this cause is not sent to the ITC.

Compare B interrupt

This interrupt request is generated when the counter matches the compare B register value during counting in comparator mode. It sets the interrupt flag CBIF/T16A_IFLG*x* register in the T16A2 module to 1.
To use this interrupt, set CBIE/T16A_IEN*x* register to 1. If CBIE is set to 0 (default), interrupt requests for this cause is not sent to the ITC.

Interrupts in capture mode

Capture A interrupt

This interrupt request is generated when the counter value is captured in the capture A register by an external trigger during counting in capture mode. It sets the interrupt flag CAPAIF/T16A_IFLG*x* register in the T16A2 module to 1.
To use this interrupt, set CAPAIE/T16A_IEN*x* register to 1. If CAPAIE is set to 0 (default), interrupt requests for this cause is not sent to the ITC.

Capture B interrupt

This interrupt request is generated when the counter value is captured in the capture B register by an external trigger during counting in capture mode. It sets the interrupt flag CAPBIF/T16A_IFLG*x* register in the T16A2 module to 1.
To use this interrupt, set CAPBIE/T16A_IEN*x* register to 1. If CAPBIE is set to 0 (default), interrupt requests for this cause is not sent to the ITC.

Capture A overwrite interrupt

This interrupt request is generated if the capture A register is overwritten by a new external trigger when the capture A interrupt flag CAPAIF has been set (a counter value has already been loaded to the capture A register). It sets the interrupt flag CAPAOWIF/T16A_IFLG*x* register in the T16A2 module to 1.
To use this interrupt, set CAPAOWIE/T16A_IEN*x* register to 1. If CAPAOWIE is set to 0 (default), interrupt requests for this cause is not sent to the ITC.
CAPAOWIF will be set if the capture A register is overwritten when CAPAIF has been set regardless of whether the capture A register has been read or not. Therefore, be sure to reset CAPAIF immediately after the capture A register is read.

Capture B overwrite interrupt

This interrupt request is generated if the capture B register is overwritten by a new external trigger when the capture B interrupt flag CAPBIF has been set (a counter value has already been loaded to the capture B register). It sets the interrupt flag CAPBOWIF/T16A_IFLG*x* register in the T16A2 module to 1.
To use this interrupt, set CAPBOWIE/T16A_IEN*x* register to 1. If CAPBOWIE is set to 0 (default), interrupt requests for this cause is not sent to the ITC.
CAPBOWIF will be set if the capture B register is overwritten when CAPBIF has been set regardless of whether the capture B register has been read or not. Therefore, be sure to reset CAPBIF immediately after the capture B register is read.

If the interrupt flag is set to 1 when the interrupt has been enabled, the T16A2 module outputs an interrupt request to the ITC. An interrupt is generated if the ITC and S1C17 core interrupt conditions are satisfied.
For more information on interrupt control registers and the operation when an interrupt occurs, see the “Interrupt Controller (ITC)” chapter.

Notes: • Reset the interrupt flag before enabling interrupts with the interrupt enable bit to prevent occurrence of unwanted interrupt. The interrupt flag is reset by writing 1.

• After an interrupt occurs, the interrupt flag in the T16A2 module must be reset in the interrupt handler routine.

12.8 Control Register Details

Table 12.8.1 List of T16A2 Registers

Address	Register name		Function
0x5068	T16A_CLK0	T16A Clock Control Register Ch.0	Controls the T16A2 Ch.0 clock.
0x5400	T16A_CTL0	T16A Counter Ch.0 Control Register	Controls the counter.
0x5402	T16A_TC0	T16A Counter Ch.0 Data Register	Counter data
0x5404	T16A_CCCTL0	T16A Comparator/Capture Ch.0 Control Register	Controls the comparator/capture block and TOUT.
0x5406	T16A_CCA0	T16A Compare/Capture Ch.0 A Data Register	Compare A/capture A data
0x5408	T16A_CCB0	T16A Compare/Capture Ch.0 B Data Register	Compare B/capture B data
0x540a	T16A_IEN0	T16A Compare/Capture Ch.0 Interrupt Enable Register	Enables/disables interrupts.
0x540c	T16A_IFLG0	T16A Compare/Capture Ch.0 Interrupt Flag Register	Displays/sets interrupt occurrence status.

The T16A2 registers are described in detail below.

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

T16A Clock Control Register Ch.*x* (T16A_CLK*x*)

Register name	Address	Bit	Name	Function	Setting			Init.	R/W	Remarks
T16A Clock Control Register Ch.*x* (T16A_CLK*x*)	**0x5068** (8 bits)	D7–4	**T16ACLKD [3:0]**	Clock division ratio select	T16ACLKD[3:0]	Division ratio		0x0	R/W	
						OSC3A or OSC3B	OSC1			
					0xf	–	–			
					0xe	1/16384	–			
					0xd	1/8192	–			
					0xc	1/4096	–			
					0xb	1/2048	–			
					0xa	1/1024	–			
					0x9	1/512	F256			
					0x8	1/256	1/256			
					0x7	1/128	1/128			
					0x6	1/64	1/64			
					0x5	1/32	1/32			
					0x4	1/16	1/16			
					0x3	1/8	1/8			
					0x2	1/4	1/4			
					0x1	1/2	1/2			
					0x0	1/1	1/1			
		D3–2	**T16ACLK SRC[1:0]**	Clock source select	T16ACLKSRC [1:0]	Clock source		0x0	R/W	
					0x3	External clock				
					0x2	OSC3A				
					0x1	OSC1				
					0x0	OSC3B				
		D1	–	reserved	–			–	–	0 when being read.
		D0	**T16ACLKE**	Count clock enable	1 Enable	0 Disable		0	R/W	

D[7:4] T16ACLKD[3:0]: Clock Division Ratio Select Bits

Selects the division ratio for generating the count clock when an internal clock (OSC3B, OSC3A, or OSC1) is used.

Table 12.8.2 Internal Clock Division Ratio Selection

T16ACLKD[3:0]	Division ratio	
	Clock source = OSC3B or OSC3A	Clock source = OSC1
0xf	Reserved	
0xe	1/16384	Reserved
0xd	1/8192	Reserved
0xc	1/4096	Reserved
0xb	1/2048	Reserved
0xa	1/1024	Reserved
0x9	1/512	F256 (Regulated 256 Hz clock)
0x8	1/256	
0x7	1/128	
0x6	1/64	
0x5	1/32	
0x4	1/16	
0x3	1/8	
0x2	1/4	
0x1	1/2	
0x0	1/1	

(Default: 0x0)

D[3:2] T16ACLKSRC[1:0]: Clock Source Select Bits

Selects the count clock source.

Table 12.8.3 Clock Source Selection

T16ACLKSRC[1:0]	Clock source
0x3	External clock (EXCL*x*)
0x2	OSC3A
0x1	OSC1
0x0	OSC3B

(Default: 0x0)

When using an external clock as the count clock, supply the clock to the EXCL*x* pin.

D1 Reserved

D0 T16ACLKE: Count Clock Enable Bit

Enables or disables the count clock supply to the counter.

1 (R/W): Enabled (on)

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

The T16ACLKE default setting is 0, which disables the clock supply. Setting T16ACLKE to 1 sends the clock selected as above to the counter. If timer operation is not required, disable the clock supply to reduce current consumption.

T16A Counter Ch.*x* Control Register (T16A_CTL*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16A Counter Ch.*x* Control Register (T16A_CTL*x*)	0x5400 (16 bits)	D15–7	–	reserved	–	–	–	0 when being read.
		D6	HCM	Half clock mode enable	1 Enable 0 Disable	0	R/W	
		D5–4	–	reserved	–	–	–	0 when being read.
		D3	CBUFEN	Compare buffer enable	1 Enable 0 Disable	0	R/W	
		D2	TRMD	Count mode select	1 One-shot 0 Repeat	0	R/W	
		D1	PRESET	Counter reset	1 Reset 0 Ignored	0	W	0 when being read.
		D0	PRUN	Counter run/stop control	1 Run 0 Stop	0	R/W	

D[15:7] Reserved

D6 HCM: Half Clock Mode Enable Bit

Sets T16A2 to half clock mode.

1 (R/W): Enabled (half clock mode)

0 (R/W): Disabled (normal clock mode) (default)

Setting HCM to 1 places T16A2 into half clock mode. In half clock mode, T16A2 uses the dual-edge counter, which counts at the rising and falling edges of the count clock, to generate a compare A signal when the dual-edge counter value matches the T16A_CCA*x* register. This makes it possible to control the duty ratio with double accuracy as compared to normal clock mode.

Setting HCM to 0 places T16A2 into normal clock mode. In normal clock mode, T16A2 generates a compare A signal when the T16A_TC*x* register value matches the T16A_CCA*x* register.

Notes: • T16A2 must be placed into comparator mode to set half clock mode, as it is effective only when PWM waveform is generated.
Be sure to set T16A2 to normal clock mode under a condition shown below.
(1) When T16A2 is placed into capture mode
(2) When TOUTAMD/T16A_CCCTL*x* register is set to 0x2 or 0x3
(3) When TOUTBMD/T16A_CCCTL*x* register is set to 0x2 or 0x3

- The dual-edge counter value cannot be read.
- Do not use the compare A interrupt in half clock mode.
- In half clock mode, the T16A_CCB*x* register setting value must be less than [T16A_CCA*x* setting value / 2 + 0x8000].

D[5:4] Reserved

D3 CBUFEN: Compare Buffer Enable Bit

Enables or disables writing to the compare buffer.
1 (R/W): Enabled
0 (R/W): Disabled (default)

Setting CBUFEN to 1 enables the compare buffer. The compare A and B signals will be generated by comparing the counter values with the compare A and B buffer values instead of the compare A and B register values. The compare A and B register values written via software are loaded to the compare A and B buffers when the compare B signal is generated.
Setting CBUFEN to 0 disables the compare buffer. The compare A and B signals will be generated by comparing the counter values with the compare A and B register values.

Note: Make sure the counter is halted (PRUN = 0) before setting CBUFEN.

D2 TRMD: Count Mode Select Bit

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

Setting TRMD to 0 sets the counter to repeat mode. In this mode, once the count starts, the counter continues counting until stopped by the application program.
Setting TRMD to 1 sets the counter to one-shot mode. In this mode, the counter stops counting automatically as soon as the compare B signal is generated.

D1 PRESET: Counter Reset Bit

Resets the counter.
1 (W): Reset
0 (W): Ignored
0 (R): Normally 0 when read out (default)

Writing 1 to this bit resets the counter to 0.

D0 PRUN: Counter Run/Stop Control Bit

Starts/stops the count.
1 (W): Run
0 (W): Stop
1 (R): Counting
0 (R): Stopped (default)

The counter starts counting when PRUN is written as 1 and stops when written as 0. The counter data is retained even if the counter is stopped.

T16A Counter Ch.*x* Data Register (T16A_TC*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16A Counter Ch.*x* Data Register (T16A_TC*x*)	**0x5402** (16 bits)	D15–0	**T16A2TC [15:0]**	Counter data T16A2TC15 = MSB T16A2TC0 = LSB	0x0 to 0xffff	0x0	R	

D[15:0] T16A2TC[15:0]: Counter Data Bits

Counter data can be read out. (Default: 0x0)

The counter value can be read out even if the counter is running. However, the counter value should be read at once using a 16-bit transfer instruction. If data is read twice using an 8-bit transfer instruction, the correct value may not be obtained due to occurrence of count up between readings.

T16A Comparator/Capture Ch.*x* Control Register (T16A_CCCTL*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16A Comparator/ Capture Ch.*x* Control Register (T16A_CCCTL*x*)	**0x5404** (16 bits)	D15–14	**CAPBTRG [1:0]**	Capture B trigger select	CAPBTRG[1:0] Trigger edge 0x3 ↑ and ↓ 0x2 ↓ 0x1 ↑ 0x0 None	0x0	R/W	
		D13–12	**TOUTBMD [1:0]**	TOUT B mode select	TOUTBMD[1:0] Mode 0x3 cmp B: ↑ or ↓ 0x2 cmp A: ↑ or ↓ 0x1 cmp A: ↑, B: ↓ 0x0 Off	0x0	R/W	
		D11–10	–	reserved	–	–	–	0 when being read.
		D9	**TOUTBINV**	TOUT B invert	1 Invert 0 Normal	0	R/W	
		D8	**CCBMD**	T16A_CCB register mode select	1 Capture 0 Comparator	0	R/W	
		D7–6	**CAPATRG [1:0]**	Capture A trigger select	CAPATRG[1:0] Trigger edge 0x3 ↑ and ↓ 0x2 ↓ 0x1 ↑ 0x0 None	0x0	R/W	
		D5–4	**TOUTAMD [1:0]**	TOUT A mode select	TOUTAMD[1:0] Mode 0x3 cmp B: ↑ or ↓ 0x2 cmp A: ↑ or ↓ 0x1 cmp A: ↑, B: ↓ 0x0 Off	0x0	R/W	
		D3–2	–	reserved	–	–	–	0 when being read.
		D1	**TOUTAINV**	TOUT A invert	1 Invert 0 Normal	0	R/W	
		D0	**CCAMD**	T16A_CCA register mode select	1 Capture 0 Comparator	0	R/W	

D[15:14] CAPBTRG[1:0]: Capture B Trigger Select Bits

Selects the trigger edge(s) of the external signal (CAPB*x*) at which the counter value is captured in the capture B register.

Table 12.8.4 Capture B Trigger Edge Selection

CAPBTRG[1:0]	Trigger edge
0x3	Falling edge and rising edge
0x2	Falling edge
0x1	Rising edge
0x0	Not triggered

(Default: 0x0)

CAPBTRG[1:0] are control bits for capture mode and are ineffective in comparator mode.

D[13:12] TOUTBMD[1:0]: TOUT B Mode Select Bits

Configures how the TOUT B signal waveform (TOUTB*x* output) is changed by the compare A and compare B signals. These bits are also used to turn the TOUT B output On and Off.

Table 12.8.5 TOUT B Generation Mode

TOUTBMD[1:0]	When compare A occurs	When compare B occurs
0x3	No change	Toggle
0x2	Toggle	No change
0x1	Rise	Fall
0x0	Disable output	

(Default: 0x0)

TOUTBMD[1:0] are control bits for comparator mode and are ineffective in capture mode.

D[11:10] Reserved

D9 TOUTBINV: TOUT B Invert Bit

Selects the TOUT B signal (TOUTB*x* output) polarity.
1 (R/W): Inverted (active Low)
0 (R/W): Normal (active High) (default)

Writing 1 to TOUTBINV generates an active Low signal (Off level = High) for the TOUT B output. When TOUTBINV is 0, an active High signal (Off level = Low) is generated.
TOUTBINV is a control bit for comparator mode and is ineffective in capture mode.

D8 CCBMD: T16A_CCB Register Mode Select Bit

Selects the T16A_CCB*x* register function (comparator mode or capture mode).
1 (R/W): Capture mode
0 (R/W): Comparator mode (default)

Writing 1 to CCBMD configures the T16A_CCB*x* register as the capture B register (capture mode) to which the counter data will be loaded by the external trigger signal. When CCBMD is 0, the T16A_CCB*x* register functions as the compare B register (comparator mode) for writing a comparison value to generate the compare B signal.

D[7:6] CAPATRG[1:0]: Capture A Trigger Select Bits

Selects the trigger edge(s) of the external signal (CAPA*x*) at which the counter value is captured in the capture A register.

Table 12.8.6 Capture A Trigger Edge Selection

CAPATRG[1:0]	Trigger edge
0x3	Falling edge and rising edge
0x2	Falling edge
0x1	Rising edge
0x0	Not triggered

(Default: 0x0)

CAPATRG[1:0] are control bits for capture mode and are ineffective in comparator mode.

D[5:4] TOUTAMD[1:0]: TOUT A Mode Select Bits

Configures how the TOUT A signal waveform (TOUTA*x* output) is changed by the compare A and compare B signals. These bits are also used to turn the TOUT A output On and Off.

Table 12.8.7 TOUT A Generation Mode

TOUTAMD[1:0]	When compare A occurs	When compare B occurs
0x3	No change	Toggle
0x2	Toggle	No change
0x1	Rise	Fall
0x0	Disable output	

(Default: 0x0)

TOUTAMD[1:0] are control bits for comparator mode and are ineffective in capture mode.

D[3:2] Reserved

D1 TOUTAINV: TOUT A Invert Bit

Selects the TOUT A signal (TOUTA*x* output) polarity.
1 (R/W): Inverted (active Low)
0 (R/W): Normal (active High) (default)

Writing 1 to TOUTAINV generates an active Low signal (Off level = High) for the TOUT A output. When TOUTAINV is 0, an active High signal (Off level = Low) is generated.
TOUTAINV is a control bit for comparator mode and is ineffective in capture mode.

D0 CCAMD: T16A_CCA Register Mode Select Bit

Selects the T16A_CCA*x* register function (comparator mode or capture mode).
1 (R/W): Capture mode
0 (R/W): Comparator mode (default)

Writing 1 to CCAMD configures the T16A_CCA*x* register as the capture A register (capture mode) to which the counter data will be loaded by the external trigger signal. When CCAMD is 0, the T16A_CCA*x* register functions as the compare A register (comparator mode) for writing a comparison value to generate the compare A signal.

T16A Comparator/Capture Ch.*x* A Data Register (T16A_CCA*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16A Comparator/ Capture Ch.*x* A Data Register (T16A_CCA*x*)	**0x5406** (16 bits)	D15–0	**CCA[15:0]**	Compare/capture A data CCA15 = MSB CCA0 = LSB	0x0 to 0xffff	0x0	R/W	

D[15:0] CCA[15:0]: Compare/Capture A Data Bits

In comparator mode (CCAMD/ T16A_CCCTL*x* register = 0)
Sets a compare A data, which will be compared with the counter value, through this register.
When CBUFEN/T16A_CTL*x* register is set to 0, compare A data will be set to the compare A register after a lapse of two T16A2 count clock cycles from the time when it is written to this register.
When CBUFEN is set to 1, the data written to this register is loaded to the compare A buffer. The buffer contents are loaded into the compare A register when the compare B signal is generated.
The compare A register is always directly accessed when being read regardless of the CBUFEN setting.
The data set is compared with the counter data. When the counter reaches the comparison value set, the compare A signal is asserted and a cause of compare A interrupt occurs. Furthermore, the TOUT output waveform changes when TOUTAMD[1:0]/T16A_CCCTL*x* register or TOUTBMD[1:0]/T16A_CCCTL*x* register is set to 0x2 or 0x1. These processes do not affect the counter data and the count up operation.

In capture mode (CCAMD = 1)
When the counter value is captured at the external trigger signal (CAPA*x*) edge selected using CAPATRG[1:0]/T16A_CCCTL*x* register, the captured value is loaded to this register. At the same time a capture A interrupt can be generated, thus the captured counter value can be read out in the interrupt handler.

T16A Comparator/Capture Ch.*x* B Data Register (T16A_CCB*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16A Comparator/ Capture Ch.*x* B Data Register (T16A_CCB*x*)	**0x5408** (16 bits)	D15–0	**CCB[15:0]**	Compare/capture B data CCB15 = MSB CCB0 = LSB	0x0 to 0xffff	0x0	R/W	

D[15:0] CCB[15:0]: Compare/Capture B Data Bits

Sets a compare B data, which will be compared with the counter value, through this register.
When CBUFEN/T16A_CTL*x* register is set to 0, compare B data will be set to the compare B register after a lapse of two T16A2 count clock cycles from the time when it is written to this register.
When CBUFEN is set to 1, the data written to this register is loaded to the compare B buffer. The buffer contents are loaded into the compare B register when the compare B signal is generated.
The compare B register is always directly accessed when being read regardless of the CBUFEN setting.
The data set is compared with the counter data. When the counter reaches the comparison value set, the compare B signal is asserted and a cause of compare B interrupt occurs. The counter is reset to 0. Furthermore, the TOUT output waveform changes when TOUTAMD[1:0]/T16A_CCCTL*x* register or TOUTBMD[1:0]/T16A_CCCTL*x* register is set to 0x3 or 0x1.

In capture mode (CCBMD = 1)

When the counter value is captured at the external trigger signal (CAPB*x*) edge selected using CAPBTRG[1:0]/T16A_CCCTL*x* register, the captured value is loaded to this register. At the same time a capture B interrupt can be generated, thus the captured counter value can be read out in the interrupt handler.

T16A Comparator/Capture Ch.*x* Interrupt Enable Register (T16A_IEN*x*)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
T16A Comparator/ Capture Ch.*x* Interrupt Enable Register (T16A_IEN*x*)	0x540a (16 bits)	D15–6	–	reserved	–				–	–	0 when being read.
		D5	CAPBOWIE	Capture B overwrite interrupt enable	1	Enable	0	Disable	0	R/W	
		D4	CAPAOWIE	Capture A overwrite interrupt enable	1	Enable	0	Disable	0	R/W	
		D3	CAPBIE	Capture B interrupt enable	1	Enable	0	Disable	0	R/W	
		D2	CAPAIE	Capture A interrupt enable	1	Enable	0	Disable	0	R/W	
		D1	CBIE	Compare B interrupt enable	1	Enable	0	Disable	0	R/W	
		D0	CAIE	Compare A interrupt enable	1	Enable	0	Disable	0	R/W	

D[15:6] Reserved

D5 CAPBOWIE: Capture B Overwrite Interrupt Enable Bit

Enables or disables capture B overwrite interrupts.

1 (R/W): Interrupt enabled

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

Setting CAPBOWIE to 1 enables capture B overwrite interrupt requests to the ITC. Setting it to 0 disables interrupts.

D4 CAPAOWIE: Capture A Overwrite Interrupt Enable Bit

Enables or disables capture A overwrite interrupts.

1 (R/W): Interrupt enabled

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

Setting CAPAOWIE to 1 enables capture A overwrite interrupt requests to the ITC. Setting it to 0 disables interrupts.

D3 CAPBIE: Capture B Interrupt Enable Bit

Enables or disables capture B interrupts.

1 (R/W): Interrupt enabled

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

Setting CAPBIE to 1 enables capture B interrupt requests to the ITC. Setting it to 0 disables interrupts.

D2 CAPAIE: Capture A Interrupt Enable Bit

Enables or disables capture A interrupts.

1 (R/W): Interrupt enabled

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

Setting CAPAIE to 1 enables capture A interrupt requests to the ITC. Setting it to 0 disables interrupts.

D1 CBIE: Compare B Interrupt Enable Bit

Enables or disables compare B interrupts.

1 (R/W): Interrupt enabled

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

Setting CBIE to 1 enables compare B interrupt requests to the ITC. Setting it to 0 disables interrupts.

D0 CAIE: Compare A Interrupt Enable Bit

Enables or disables compare A interrupts.

1 (R/W): Interrupt enabled

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

Setting CAIE to 1 enables compare A interrupt requests to the ITC. Setting it to 0 disables interrupts.

T16A Comparator/Capture Ch.*x* Interrupt Flag Register (T16A_IFLG*x*)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
T16A Comparator/ Capture Ch.*x* Interrupt Flag Register (T16A_IFLG*x*)	0x540c (16 bits)	D15–6	–	reserved	–				–	–	0 when being read.
		D5	**CAPBOWIF**	Capture B overwrite interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D4	**CAPAOWIF**	Capture A overwrite interrupt flag					0	R/W	
		D3	**CAPBIF**	Capture B interrupt flag					0	R/W	
		D2	**CAPAIF**	Capture A interrupt flag					0	R/W	
		D1	**CBIF**	Compare B interrupt flag					0	R/W	
		D0	**CAIF**	Compare A interrupt flag					0	R/W	

D[15:6] Reserved

D5 CAPBOWIF: Capture B Overwrite Interrupt Flag Bit

Indicates whether the cause of capture B overwrite interrupt has occurred or not.

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

CAPBOWIF is a T16A2 interrupt flag that is set to 1 when the capture B register is overwritten.
CAPBOWIF is reset by writing 1.

D4 CAPAOWIF: Capture A Overwrite Interrupt Flag Bit

Indicates whether the cause of capture A overwrite interrupt has occurred or not.

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

CAPAOWIF is a T16A2 interrupt flag that is set to 1 when the capture A register is overwritten.
CAPAOWIF is reset by writing 1.

D3 CAPBIF: Capture B Interrupt Flag Bit

Indicates whether the cause of capture B interrupt has occurred or not.

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

CAPBIF is a T16A2 interrupt flag that is set to 1 when the counter value is captured in the capture B register.
CAPBIF is reset by writing 1.

D2 CAPAIF: Capture A Interrupt Flag Bit

Indicates whether the cause of capture A interrupt has occurred or not.

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

CAPAIF is a T16A2 interrupt flag that is set to 1 when the counter value is captured in the capture A register.
CAPAIF is reset by writing 1.

D1 CBIF: Compare B Interrupt Flag Bit

Indicates whether the cause of compare B interrupt has occurred or not.

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

CBIF is a T16A2 interrupt flag that is set to 1 when the counter reaches the value set in the compare B register.
CBIF is reset by writing 1.

D0 CAIF: Compare A Interrupt Flag Bit

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

CAIF is a T16A2 interrupt flag that is set to 1 when the counter reaches the value set in the compare A register.
CAIF is reset by writing 1.

13 Clock Timer (CT)

13.1 CT Module Overview

The S1C17651 includes a clock timer module (CT) that uses the OSC1 oscillator as its clock source. This timer can be used for generating cyclic interrupts to implement a software clock function.
The features of the CT module are listed below.

- 8-bit binary counter (128 Hz to 1 Hz)
- 32 Hz, 8 Hz, 2 Hz, and 1 Hz interrupts can be generated.

Figure 13.1.1 shows the CT configuration.

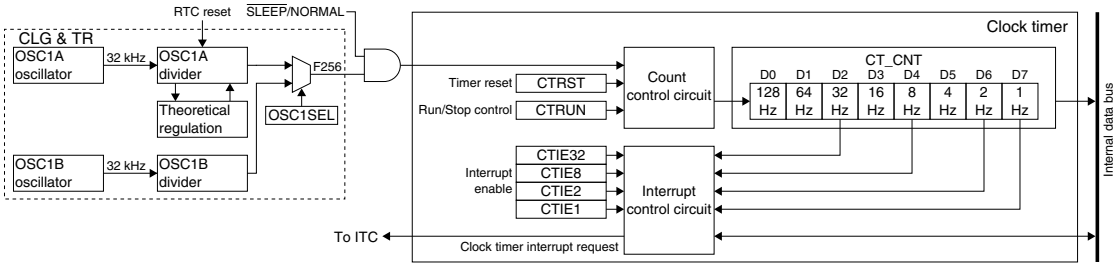


Figure 13.1.1 CT Configuration

The CT module consists of an 8-bit binary counter that uses the 256 Hz signal divided from the OSC1 clock as the input clock and allows data for each bit (128 Hz to 1 Hz) to be read out by software. The clock timer can also generate interrupts using the 32 Hz, 8 Hz, 2 Hz, and 1 Hz signals. This clock timer is normally used for various timing functions, such as a clock.

13.2 Operation Clock

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

Notes:
- The CT module input clock frequency is 256 Hz only when the OSC1 clock frequency is 32.768 kHz. The frequency described in this chapter will vary accordingly for other OSC1 clock frequencies.
- The CT module can also be operated with the OSC1B divider clock (about 256 Hz) even if OSC1B is selected as the OSC1 clock source in the CLG. However, the CT module cannot be used as an accurate clock.
- The OSC1A divider is reset when the RTC starts running (when 1 is written to RTCRUN/RTC_CTL register). This affects the count operations of the CT module, as new 256 Hz cycle begins from that point.

13.3 Timer Reset

Reset the timer by writing 1 to CTRST/CT_CTL register. This clears the counter to 0.
Apart from this operation, the counter is also cleared by an initial reset.

13.4 Timer RUN/STOP Control

Make the following settings before starting CT.
(1) If using interrupts, set the interrupt level and enable interrupts for the clock timer. See Section 13.5.
(2) Reset the timer. See Section 13.3.

The clock timer includes CTRUN/CT_CTL register for Run/Stop control.
The clock timer starts operating when 1 is written to CTRUN. Writing 0 to CTRUN disables clock input and stops the operation.
This control does not affect the counter (CT_CNT register) data. The counter data is retained even when the count is halted, allowing resumption of the count from that data.
If 1 is written to both CTRUN and CTRST simultaneously, the clock timer starts counting after resetting.
A cause of interrupt occurs during counting at the 32 Hz, 8 Hz, 2 Hz, and 1 Hz signal falling edges. If interrupts are enabled, an interrupt request is sent to the interrupt controller (ITC).

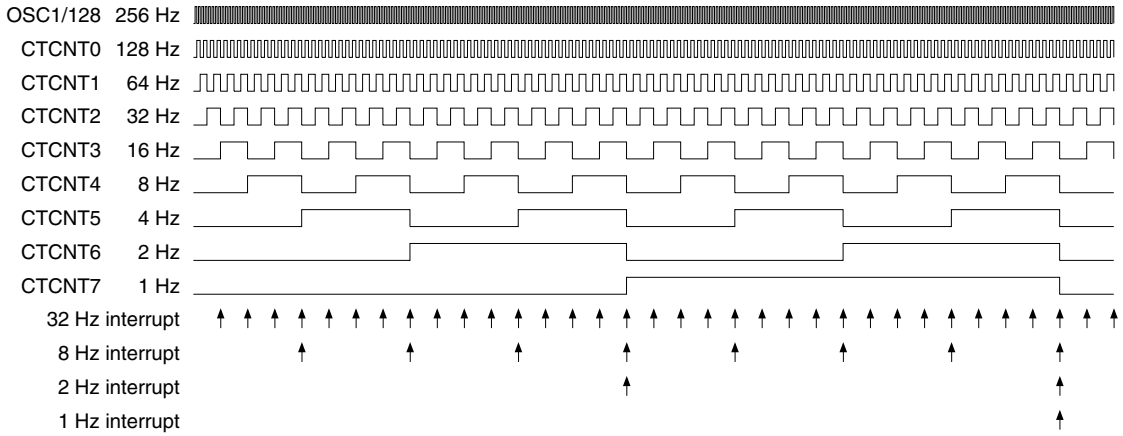


Figure 13.4.1 Clock Timer Timing Chart

Notes: • The clock timer switches to Run/Stop status synchronized with the 256 Hz signal falling edge after data is written to CTRUN. When 0 is written to CTRUN, the timer stops after counting an additional "+1." 1 is retained for CTRUN reading until the timer actually stops.
Figure 13.4.2 shows the Run/Stop control timing chart.

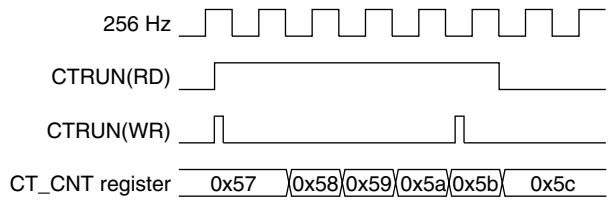


Figure 13.4.2 Run/Stop Control Timing Chart

• Executing the `slp` instruction while the timer is running (CTRUN = 1) will destabilize the timer operation during restarting from SLEEP status. When switching to SLEEP status, stop the timer (CTRUN = 0) before executing the `slp` instruction.

13.5 CT Interrupts

The CT module includes functions for generating the following four kinds of interrupts:
32 Hz, 8 Hz, 2 Hz, and 1 Hz interrupts

The CT module outputs a single interrupt signal shared by the above four interrupt causes to the interrupt controller (ITC). The interrupt flag in the CT module should be read to identify the cause of interrupt that occurred.

32 Hz, 8 Hz, 2 Hz, and 1 Hz interrupts

The 32 Hz, 8 Hz, 2 Hz, and 1 Hz signal falling edges set the corresponding interrupt flag in the CT module to 1. At the same time, an interrupt request is sent to the ITC if the corresponding interrupt enable bit has been set to 1 (interrupt enabled). An interrupt is generated if the ITC and S1C17 Core interrupt conditions are satisfied.
If the interrupt enable bit is set to 0 (interrupt disabled, default), no interrupt request will be sent to the ITC.

Table 13.5.1 CT Interrupt Flags and Interrupt Enable Bits

Cause of interrupt	Interrupt flag	Interrupt enable bit
32 Hz Interrupt	CTIF32/CT_IFLG register	CTIE32/CT_IMSK register
8 Hz Interrupt	CTIF8/CT_IFLG register	CTIE8/CT_IMSK register
2 Hz Interrupt	CTIF2/CT_IFLG register	CTIE2/CT_IMSK register
1 Hz Interrupt	CTIF1/CT_IFLG register	CTIE1/CT_IMSK register

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

Notes: • The CT module interrupt flag must be reset in the interrupt handler routine after a CT interrupt has occurred to prevent recurring interrupts.

• Reset the interrupt flag before enabling CT interrupts with the interrupt enable bit to prevent occurrence of unwanted interrupt. The interrupt flag is reset by writing 1.

13.6 Control Register Details

Table 13.6.1 List of CT Registers

Address	Register name		Function
0x5000	CT_CTL	Clock Timer Control Register	Resets and starts/stops the timer.
0x5001	CT_CNT	Clock Timer Counter Register	Counter data
0x5002	CT_IMSK	Clock Timer Interrupt Mask Register	Enables/disables interrupt.
0x5003	CT_IFLG	Clock Timer Interrupt Flag Register	Indicates/resets interrupt occurrence status.

The CT registers are described in detail below.

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

Clock Timer Control Register (CT_CTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Clock Timer Control Register (CT_CTL)	**0x5000** (8 bits)	D7–5	–	reserved	–	–	–	0 when being read.
		D4	**CTRST**	Clock timer reset	1 Reset 0 Ignored	0	W	
		D3–1	–	reserved	–	–	–	
		D0	**CTRUN**	Clock timer run/stop control	1 Run 0 Stop	0	R/W	

D[7:5] Reserved

D4 CTRST: Clock Timer Reset Bit

Resets the clock timer.
1 (W): Reset
0 (W): Ignored
0 (R): Always 0 when read (default)

Writing 1 to this bit resets the counter to 0x0. When reset in Run state, the clock timer restarts immediately after resetting. The reset data 0x0 is retained when in Stop state.

D[3:1] Reserved

D0 CTRUN: Clock Timer Run/Stop Control Bit

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

The clock timer starts counting when CTRUN is written as 1 and stops when written as 0. The counter data is retained at Stop state until a reset or the next Run state.

Clock Timer Counter Register (CT_CNT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Clock Timer Counter Register (CT_CNT)	0x5001 (8 bits)	D7–0	CTCNT[7:0]	Clock timer counter value	0x0 to 0xff	0x0	R	

D[7:0] CTCNT[7:0]: Clock Timer Counter Value Bits

The counter data can be read out. (Default: 0x0)
This register is read-only and cannot be written to.
The bits correspond to various frequencies, as follows:
D7: 1 Hz, D6: 2 Hz, D5: 4 Hz, D4: 8 Hz, D3: 16 Hz, D2: 32 Hz, D1: 64 Hz, D0: 128 Hz

Note: The correct counter value may not be read out (reading is unstable) if the register is read while counting is underway. Read the counter register twice in succession and treat the value as valid if the values read are identical.

Clock Timer Interrupt Mask Register (CT_IMSK)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Clock Timer Interrupt Mask Register (CT_IMSK)	0x5002 (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.
		D3	CTIE32	32 Hz interrupt enable	1 Enable 0 Disable	0	R/W	
		D2	CTIE8	8 Hz interrupt enable	1 Enable 0 Disable	0	R/W	
		D1	CTIE2	2 Hz interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	CTIE1	1 Hz interrupt enable	1 Enable 0 Disable	0	R/W	

This register enables or disables interrupt requests individually for the 32 Hz, 8 Hz, 2 Hz, and 1 Hz signals. Setting CTIE* to 1 enables CT interrupts for the corresponding frequency signal falling edge, while setting to 0 disables interrupts.

D[7:4] Reserved

D3 CTIE32: 32 Hz Interrupt Enable Bit

Enables or disables 32 Hz interrupts.
1 (R/W): Interrupt enabled
0 (R/W): Interrupt disabled (default)

D2 CTIE8: 8 Hz Interrupt Enable Bit

Enables or disables 8 Hz interrupts.
1 (R/W): Interrupt enabled
0 (R/W): Interrupt disabled (default)

D1 CTIE2: 2 Hz Interrupt Enable Bit

Enables or disables 2 Hz interrupts.
1 (R/W): Interrupt enabled
0 (R/W): Interrupt disabled (default)

D0 CTIE1: 1 Hz Interrupt Enable Bit

Enables or disables 1 Hz interrupts.
1 (R/W): Interrupt enabled
0 (R/W): Interrupt disabled (default)

Clock Timer Interrupt Flag Register (CT_IFLG)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Clock Timer Interrupt Flag Register (CT_IFLG)	0x5003 (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.
		D3	CTIF32	32 Hz interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D2	CTIF8	8 Hz interrupt flag		0	R/W	
		D1	CTIF2	2 Hz interrupt flag		0	R/W	
		D0	CTIF1	1 Hz interrupt flag		0	R/W	

This register indicates the occurrence state of interrupt causes due to 32 Hz, 8 Hz, 2 Hz, and 1 Hz signals. If a CT interrupt occurs, identify the interrupt cause (frequency) by reading the interrupt flag in this register. CTIF* is a CT module interrupt flag that is set to 1 at the falling edge of the corresponding 32 Hz, 8 Hz, 2 Hz, or 1 Hz interrupt. CTIF* is reset by writing 1.

D[7:4] Reserved

D3 CTIF32: 32 Hz Interrupt Flag Bit

Indicates whether the cause of 32 Hz interrupt has occurred or not.

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

D2 CTIF8: 8 Hz Interrupt Flag Bit

Indicates whether the cause of 8 Hz interrupt has occurred or not.

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

D1 CTIF2: 2 Hz Interrupt Flag Bit

Indicates whether the cause of 2 Hz interrupt has occurred or not.

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

D0 CTIF1: 1 Hz Interrupt Flag Bit

Indicates whether the cause of 1 Hz interrupt has occurred or not.

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

14 Watchdog Timer (WDT)

14.1 WDT Module Overview

The S1C17651 includes a watchdog timer module (WDT) that uses the OSC1 oscillator as its clock source. This timer is used to detect CPU runaway.
The features of WDT are listed below.

- 10-bit up counter
- Either reset or NMI can be generated if the counter overflows.

Figure 14.1.1 shows the WDT configuration.

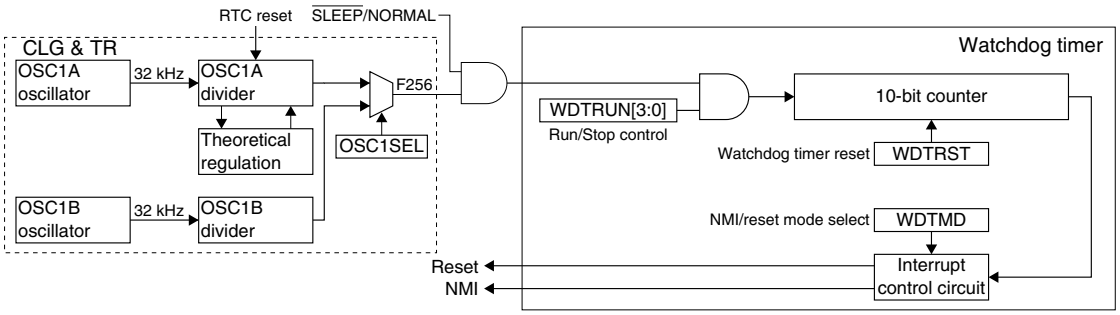


Figure 14.1.1 WDT Configuration

The WDT module generates an NMI or reset (selectable via software) to the CPU if not reset within 131,072/fosc1 seconds (4 seconds when fosc1 = 32.768 kHz).
Reset WDT via software within this cycle to prevent NMI/resets, which in turn enables runaway detection for programs that do not pass through the handler routine.

14.2 Operation Clock

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

Notes: • The WDT module input clock frequency is 256 Hz only when the OSC1 clock frequency is 32.768 kHz. The frequency and time described in this chapter will vary accordingly for other OSC1 clock frequencies.

- The WDT module can also be operated with the OSC1B divider clock (about 256 Hz) even if OSC1B is selected as the OSC1 clock source in the CLG.
- The OSC1A divider is reset when the RTC starts running (when 1 is written to RTCRUN/RTC_CTL register). This affects the count operations of the WDT module, as new 256 Hz cycle begins from that point.

14.3 WDT Control

14.3.1 NMI/Reset Mode Selection

WDTMD/WDT_ST register is used to select whether an NMI signal or a reset signal is output when WDT has not been reset within the NMI/reset generation cycle.
To generate an NMI, set WDTMD to 0 (default). Set to 1 to generate a reset.

14.3.2 WDT Run/Stop Control

WDT starts counting when a value other than 0b1010 is written to WDTRUN[3:0]/WDT_CTL register and stops when 0b1010 is written.
At initial reset, WDTRUN[3:0] is set to 0b1010 to stop the watchdog timer.
Since an NMI or reset may be generated immediately after running depending on the counter value, WDT should also be reset concurrently (before running the watchdog timer), as explained in the following section.

14.3.3 WDT Reset

To reset WDT, write 1 to WDTRST/WDT_CTL register.
A location should be provided for periodically processing the routine for resetting WDT before an NMI or reset is generated when using WDT. Process this routine within 131,072/fosc1 second (4 seconds when fosc1 = 32.768 kHz) cycle.
After resetting, WDT starts counting with a new NMI/Reset generation cycle.
If WDT is not reset within the NMI/Reset generation cycle for any reason, the CPU is switched to interrupt processing by NMI or reset, the interrupt vector is read out, and the interrupt handler routine is executed. The reset and NMI vector addresses are TTBR + 0x0 and TTBR + 0x08.

If the counter overflows and generates an NMI without WDT being reset, WDTST/WDT_ST register is set to 1. This bit is provided to confirm that WDT was the source of the NMI. The WDTST set to 1 is cleared to 0 by resetting WDT.

14.3.4 Operations in HALT and SLEEP Modes

HALT mode

The WDT module operates in HALT mode, as the clock is supplied. HALT mode is therefore cleared by an NMI or reset if it continues for more than the NMI/reset generation cycle. To disable WDT while in HALT mode, stop WDT by writing 0b1010 to WDTRUN[3:0]/WDT_CTL register before executing the `halt` instruction. Reset WDT before resuming operations after HALT mode is cleared.

SLEEP mode

The clock supplied from the CLG module is stopped in SLEEP mode, which also stops WDT. To prevent generation of an unnecessary NMI or reset after clearing SLEEP mode, reset WDT before executing the `slp` instruction. WDT should also be stopped as required using WDTRUN[3:0].

14.4 Control Register Details

Table 14.4.1 List of WDT Registers

Address	Register name		Function
0x5040	WDT_CTL	Watchdog Timer Control Register	Resets and starts/stops the timer.
0x5041	WDT_ST	Watchdog Timer Status Register	Sets the timer mode and indicates NMI status.

The WDT registers are described in detail below.

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

Watchdog Timer Control Register (WDT_CTL)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Watchdog Timer Control Register (WDT_CTL)	0x5040 (8 bits)	D7–5	–	reserved	–				–	–	0 when being read.
		D4	WDTRST	Watchdog timer reset	1	Reset	0	Ignored	0	W	
		D3–0	WDTRUN[3:0]	Watchdog timer run/stop control	Other than 1010	Run	1010	Stop	1010	R/W	

D[7:5] Reserved

D4 WDTRST: Watchdog Timer Reset Bit

Resets WDT.

1 (W): Reset

0 (W): Ignored

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

Note: To use WDT, it must be reset by writing 1 to this bit within the NMI/reset generation cycle (4 seconds when f_{OSC1} = 32.768 kHz). This resets the up-counter to 0 and starts counting with a new NMI/reset generation cycle.

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

Controls WDT Run/Stop.

Values other than 0b1010 (R/W): Run

0b1010 (R/W): Stop (default)

Note: WDT must also be reset to prevent generation of an unnecessary NMI or Reset before starting WDT.

Watchdog Timer Status Register (WDT_ST)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Watchdog Timer Status Register (WDT_ST)	0x5041 (8 bits)	D7–2	–	reserved	–				–	–	0 when being read.
		D1	WDTMD	NMI/Reset mode select	1	Reset	0	NMI	0	R/W	
		D0	WDTST	NMI status	1	NMI occurred	0	Not occurred	0	R	

D[7:2] Reserved

D1 WDTMD: NMI/Reset Mode Select Bit

Selects NMI or reset generation on counter overflow.

1 (R/W): Reset

0 (R/W): NMI (default)

Setting this bit to 1 outputs a reset signal when the counter overflows. Setting to 0 outputs an NMI signal.

D0 WDTST: NMI Status Bit

Indicates a counter overflow and NMI occurrence.

1 (R): NMI occurred (counter overflow)

0 (R): NMI not occurred (default)

This bit confirms that WDT was the source of the NMI. The WDTST set to 1 is cleared to 0 by resetting WDT.

This is also set by a counter overflow if reset output is selected, but is cleared by initial reset and cannot be confirmed.

15 UART

15.1 UART Module Overview

The S1C17651 includes a UART module for asynchronous communication. It includes a 2-byte receive data buffer and 1-byte transmit data buffer allowing successive data transfer. The UART module also includes an RZI modulator/demodulator circuit that enables IrDA 1.0-compatible infrared communications simply by adding basic external circuits.
The following shows the main features of the UART:

- Number of channels: 1 channel
- Transfer rate: 150 to 230,400 bps (150 to 115,200 bps in IrDA mode)
- Transfer clock: Internal clock (baud rate generator output) or an external clock (SCLK input) can be selected.
- Character length: 7 or 8 bits (LSB first)
- Parity mode: Even, odd, or no parity
- Stop bit: 1 or 2 bits
- Start bit: 1 bit fixed
- Supports full-duplex communications.
- Includes a 2-byte receive data buffer and a 1-byte transmit data buffer.
- Includes a baud rate generator with fine adjustment function.
- Includes an RZI modulator/demodulator circuit to support IrDA 1.0-compatible infrared communications.
- Can detect parity error, framing error, and overrun error during receiving.
- Can generate receive buffer full, transmit buffer empty, end of transmission and receive error interrupts.

Figure 15.1.1 shows the UART configuration.

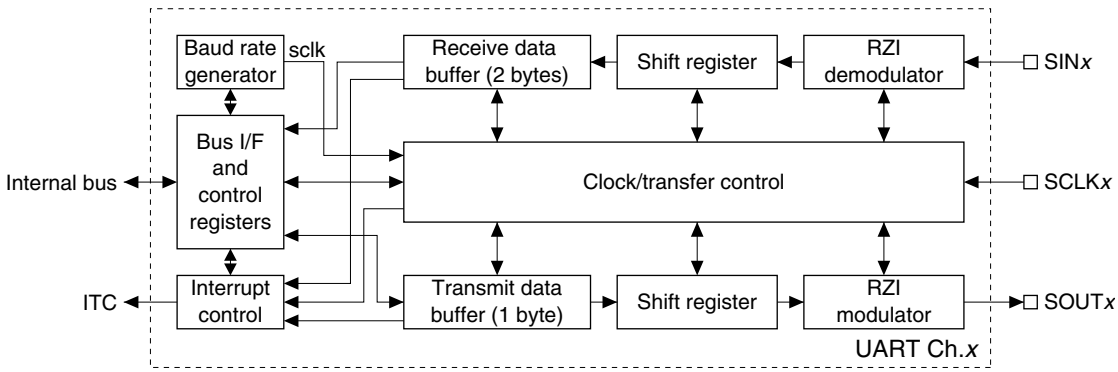


Figure 15.1.1 UART Configuration

Note: The letter '*x*' in register and pin names refers to a channel number (0).
Example: UART_CTL*x* register
Ch.0: UART_CTL0 register

15.2 UART Input/Output Pins

Table 15.2.1 lists the UART input/output pins.

Table 15.2.1 List of UART Pins

Pin name	I/O	Qty	Function
SIN0 (Ch.0)	I	1	UART data input pin Inputs serial data sent from an external serial device.
SOUT0 (Ch.0)	O	1	UART data output pin Outputs serial data sent to an external serial device.
SCLK0 (Ch.0)	I	1	UART clock input pin Inputs the transfer clock when an external clock is used.

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

15.3 Baud Rate Generator

The UART module includes a baud rate generator to generate the transfer (sampling) clock. It consists of an 8-bit programmable timer with fine mode. The timer counts down from the initial value set via software and outputs an underflow signal when the counter underflows. The underflow signal is used to generate the transfer clock. The underflow cycle can be programmed by selecting the clock source and initial data, enabling the application program to obtain serial transfer rates as required. Fine mode provides a function that minimizes transfer rate errors.

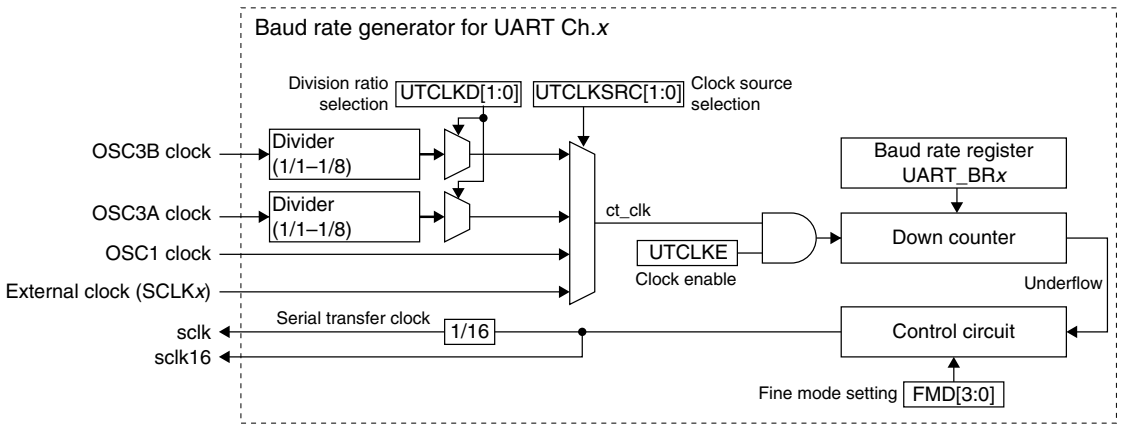


Figure 15.3.1 Baud Rate Generator

Clock source settings

The clock source can be selected from OSC3B, OSC3A, OSC1, or external clock using UTCLKSRC[1:0]/UART_CLK*x* register.

Table 15.3.1 Clock Source Selection

UTCLKSRC[1:0]	Clock source
0x3	External clock (SCLK*x*)
0x2	OSC3A
0x1	OSC1
0x0	OSC3B

(Default: 0x0)

Note: When inputting the external clock via the SCLK*x* pin, the clock duty ratio must be 50%.

When OSC3B or OSC3A is selected as the clock source, use UTCLKD[1:0]/UART_CLK*x* register to select the division ratio.

Table 15.3.2 OSC3B/OSC3A Division Ratio Selection

UTCLKD[1:0]	Division ratio
0x3	1/8
0x2	1/4
0x1	1/2
0x0	1/1

(Default: 0x0)

Clock supply to the counter is controlled using UTCLKE/UART_CLK*x* register. The UTCLKE default setting is 0, which disables the clock supply. Setting UTCLKE to 1 sends the clock selected to the counter.

Initial counter value setting

BR[7:0]/UART_BR*x* register is used to set the initial value for the down counter.
The initial counter value is preset to the down counter if the counter underflows. This means that the initial counter value and the count clock frequency determine the time elapsed between underflows.

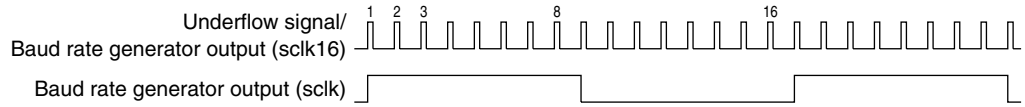


Figure 15.3.2 Counter Underflow and Clock Generated

Use the following equations to calculate the initial counter value for obtaining the desired transfer rate.

$$bps = \frac{ct\_clk}{\{(BR + 1) \times 16 + FMD\}}$$

$$BR = \left(\frac{ct\_clk}{bps} - FMD - 16\right) \div 16$$

ct_clk: Count clock frequency (Hz)
BR: BR[7:0] setting (0 to 255)
bps: Transfer rate (bit/s)
FMD: FMD[3:0] (fine mode) setting (0 to 15)

Note: The UART transfer rate is capped at 230,400 bps (115,200 bps in IrDA mode). Do not set faster transfer rates.

Fine Mode

Fine mode provides a function that minimizes transfer rate errors. The baud rate generator output clock can be set to the required frequency by selecting the appropriate clock source and initial counter data. Note that errors may occur, depending on the transfer rate. Fine mode extends the output clock cycle by delaying the underflow pulse from the counter. This delay can be specified with the FMD[3:0]/UART_FMD*x* register. FMD[3:0] specifies the delay pattern to be inserted into a 16 underflow period. Inserting one delay extends the output clock cycle by one count clock cycle.

Table 15.3.3 Delay Patterns Specified by FMD[3:0]

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

D: Indicates the insertion of a delay cycle.

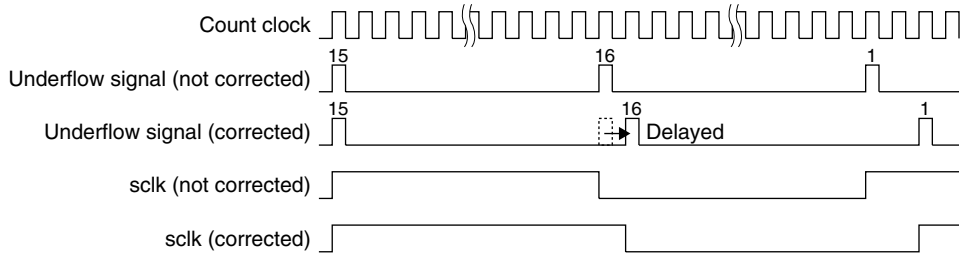


Figure 15.3.3 Delay Cycle Insertion in Fine Mode

At initial reset, FMD[3:0] is set to 0x0, preventing insertion of delay cycles.

Note: Make sure the UART is halted (RXEN/UART_CTL*x* register = 0) before setting the baud rate generator.

15.4 Transfer Data Settings

Set the following conditions to configure the transfer data format.
- Data length: 7 or 8 bits
- Start bit: Fixed at 1 bit
- Stop bit: 1 or 2 bits
- Parity bit: Even, odd, or no parity

Note: Make sure the UART is halted (RXEN/UART_CTL*x* register = 0) before changing transfer data format settings.

Data length

The data length is selected by CHLN/UART_MOD*x* register. Setting CHLN to 0 (default) configures the data length to 7 bits. Setting CHLN to 1 configures it to 8 bits.

Stop bit

The stop bit length is selected by STPB/UART_MOD*x* register. Setting STPB to 0 (default) configures the stop bit length to 1 bit. Setting STPB to 1 configures it to 2 bits.

Parity bit

Whether the parity function is enabled or disabled is selected by PREN/UART_MOD*x* register. Setting PREN to 0 (default) disables the parity function. In this case, no parity bit is added to the transfer data and the data is not checked for parity when received. Setting PREN to 1 enables the parity function. In this case, a parity bit is added to the transfer data and the data is checked for parity when received. When the parity function is enabled, the parity mode is selected by PMD/UART_MOD*x* register. Setting PMD to 0 (default) adds a parity bit and checks for even parity. Setting PMD to 1 adds a parity bit and checks for odd parity.

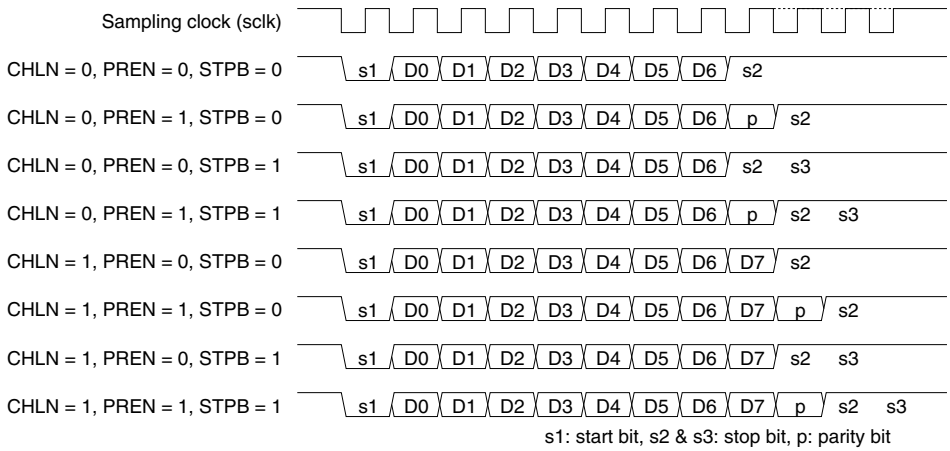


Figure 15.4.1 Transfer Data Format

15.5 Data Transfer Control

Make the following settings before starting data transfers.
(1) Select the input clock. (See Section 15.3.)
(2) Program the baud rate generator to output the transfer clock. (See Section 15.3.)
(3) Set the transfer data format. (See Section 15.4.)
(4) To use the IrDA interface, set IrDA mode. (See Section 15.8.)
(5) Set interrupt conditions to use UART interrupts. (See Section 15.7.)

Note: Make sure the UART is halted (RXEN/UART_CTL*x* register = 0) before changing the above settings.

Enabling data transfers

Set RXEN/UART_CTL*x* register to 1 to enable data transfers. This puts the transmitter/receiver circuit in ready-to-transmit/receive status.

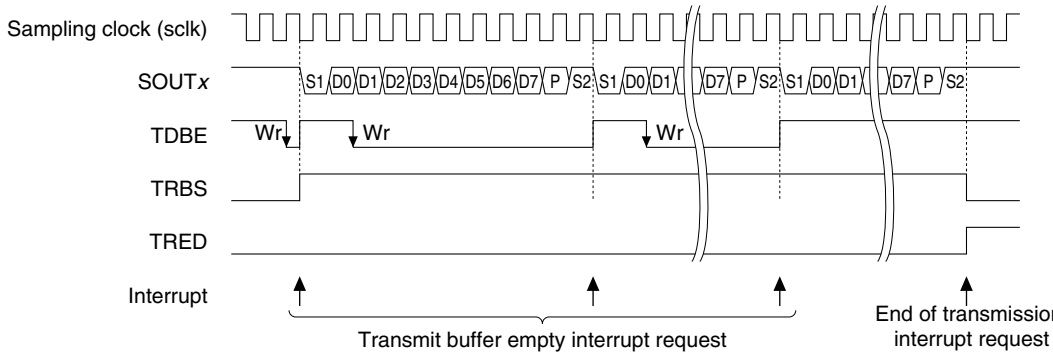
Note: Do not set RXEN to 0 while the UART is sending or receiving data.

Data transmission control

To start data transmission, write the transmit data to TXD[7:0]/UART_TXD*x* register.
The data is written to the transmit data buffer, and the transmitter circuit starts sending data.
The buffer data is sent to the transmit shift register, and the start bit is output from the SOUT*x* pin. The data in the shift register is then output from the LSB. The transfer data bit is shifted in sync with the sampling clock rising edge and output in sequence via the SOUT*x* pin. Following output of MSB, the parity bit (if parity is enabled) and the stop bit are output.

The transmitter circuit includes three status flags: TDBE/UART_ST*x* register, TRBS/UART_ST*x* register, and TRED/UART_ST*x* register.
The TDBE flag indicates the transmit data buffer status. This flag switches to 0 when the application program writes data to the transmit data buffer and reverts to 1 when the buffer data is sent to the transmit shift register. An interrupt can be generated when this flag is set to 1 (see Section 15.7). Subsequent data is sent after confirming that the transmit data buffer is empty either by using this interrupt or by reading the TDBE flag. The transmit data buffer size is 1 byte, but a shift register is provided separately to allow data to be written while the previous data is being sent. Always confirm that the transmit data buffer is empty before writing transmit data. Writing data while the TDBE flag is 0 will overwrite earlier transmit data inside the transmit data buffer.

The TRBS flag indicates the shift register status. This flag switches to 1 when transmit data is loaded from the transmit data buffer to the shift register and reverts to 0 once the data is sent. Read this flag to check whether the transmitter circuit is operating or at standby.
The TRED switches to 1 when the TRBS flag reverts to 0 from 1, indicating that transmit operation has completed. An interrupt can be generated when this flag is set to 1 (see Section 15.7). Use this interrupt for transmission end processing. The TRED flag is reset to 0 by writing 1.



S1: Start bit, S2: Stop bit, P: Parity bit, Wr: Data write to transmit data buffer

Figure 15.5.1 Data Transmission Timing Chart

Data reception control

The receiver circuit is activated by setting RXEN to 1, enabling data to be received from an external serial device.
When the external serial device sends a start bit, the receiver circuit detects its Low level and starts sampling the following data bits. The data bits are sampled at the sampling clock rising edge, and the lead bit is loaded into the receive shift register as LSB. Once the MSB has been received into the shift register, the received data is loaded into the receive data buffer. If parity checking is enabled, the receiver circuit checks the received data at the same time by checking the parity bit received immediately after the MSB.
The receive data buffer, a 2-byte FIFO, receives data until full.

Received data in the buffer can be read from RXD[7:0]/UART_RXD*x* register. The oldest data is read out first and data is cleared by reading.

The receiver circuit includes two buffer status flags: RDRY/UART_ST*x* register and RD2B/UART_ST*x* register.
The RDRY flag indicates that the receive data buffer still contains data. The RD2B flag indicates that the receive data buffer is full.

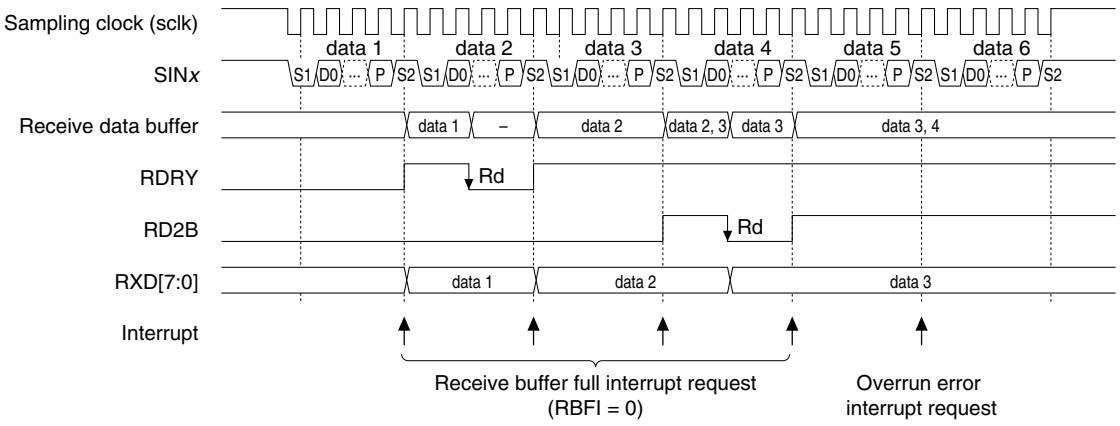
(1) RDRY = 0, RD2B = 0
The receive data buffer contents need not be read, since no data has been received.

(2) RDRY = 1, RD2B = 0
One 8-bit data has been received. Read the receive data buffer contents once. This resets the RDRY flag. The buffer reverts to state (1) above.
If the receive data buffer contents are read twice, the second data read will be invalid.

(3) RDRY = 1, RD2B = 1
Two 8-bit data have been received. Read the receive data buffer contents twice. The receive data buffer outputs the oldest data first. This resets the RD2B flag. The buffer then reverts to the state in (2) above. The second read outputs the most recent received data, after which the buffer reverts to the state in (1) above.
Even when the receive data buffer is full, the shift register can start receiving 8-bit data one more time. An overrun error will occur if receiving is finished before the receive data buffer has been read. In this case, the last received data cannot be read. The contents of the receive data buffer must be read out before an overrun error occurs. For detailed information on overrun errors, refer to Section 15.6.

The volume of data received can be checked by reading these flags.
The UART allows receive buffer full interrupts to be generated once data has been received in the receive data buffer. These interrupts can be used to read the receive data buffer. By default, a receive buffer full interrupt occurs when the receive data buffer receives one 8-bit data (status (2) above). This can be changed by setting RBFI/UART_CTL*x* register to 1 so that an interrupt occurs when the receive data buffer receives two 8-bit data.

Three error flags are also provided in addition to the flags previously mentioned. See Section 15.6 for detailed information on flags and receive errors.



S1: Start bit, S2: Stop bit, P: Parity bit, Rd: Data read from RXD[7:0]

Figure 15.5.2 Data Receiving Timing Chart

Disabling data transfers

Write 0 to RXEN to disable data transfers. The data being transferred cannot be guaranteed if RXEN is set to 0 while data is being sent or received. Before setting RXEN to 0, check the data transfer status with software in consideration of the communication procedure. The data transmit status can be checked using the TRBS flag.

Note: Setting RXEN to 0 empties the transmit data buffer, clearing any remaining data. The data being transferred cannot be guaranteed if RXEN is set to 0 while data is being sent or received.
Make sure that the TDBE flag is 1 and the TRBS and RDRY flags are both 0 before disabling data transfer.

15.6 Receive Errors

Three different receive errors may be detected while receiving data.
Since receive errors are interrupt causes, they can be processed by generating interrupts. For more information on UART interrupt control, see Section 15.7.

Parity error

If PREN/UART_MOD*x* register has been set to 1 (parity enabled), data received is checked for parity.
Data received in the shift register is checked for parity when sent to the receive data buffer. The matching is checked against the PMD/UART_MOD*x* register setting (odd or even parity). If the result is a non-match, a parity error is issued, and the parity error flag PER/UART_ST*x* register is set to 1. Even if this error occurs, the data received is sent to the receive data buffer, and the receiving operation continues. However, the received data cannot be guaranteed if a parity error occurs. The PER flag is reset to 0 by writing 1.

Framing error

A framing error occurs if the stop bit is received as 0 and the UART determines loss of sync. If the stop bit is set to two bits, only the first bit is checked.
The framing error flag FER/UART_ST*x* register is set to 1 if this error occurs. The received data is still transferred to the receive data buffer if this error occurs and the receiving operation continues, but the data cannot be guaranteed, even if no framing error occurs for subsequent data receiving. The FER flag is reset to 0 by writing 1.

Overrun error

Even if the receive data buffer is full (two 8-bit data already received), the third data can be received in the shift register. However, if the receive data buffer is not emptied (by reading out data received) by the time this data has been received, the third data received in the shift register will not be sent to the buffer and generate an overrun error. If an overrun error occurs, the overrun error flag OER/UART_ST*x* register is set to 1. The receiving operation continues even if this error occurs. The OER flag is reset to 0 by writing 1.

15.7 UART Interrupts

The UART includes a function for generating the following four different types of interrupts.
- Transmit buffer empty interrupt
- End of transmission interrupt
- Receive buffer full interrupt
- Receive error interrupt

The UART outputs one interrupt signal shared by the four above interrupt causes to the interrupt controller (ITC). Inspect the status flag and error flag to determine the interrupt cause occurred.

Transmit buffer empty interrupt

To use this interrupt, set TIEN/UART_CTL*x* register to 1. If TIEN is set to 1 while TDBE/UART_ST*x* register is 1 (transmit data buffer empty) or if TDBE is set to 1 (when the transmit data buffer becomes empty by loading the transmit data written to it to the shift register) while TIEN is 1, an interrupt request is sent to the ITC. An interrupt occurs if other interrupt conditions are met.

If TIEN is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

You can inspect the TDBE flag in the UART interrupt handler routine to determine whether the UART interrupt is attributable to a transmit buffer empty. If TDBE is 1, the next transmit data can be written to the transmit data buffer by the interrupt handler routine.

End of transmission interrupt

To use this interrupt, set TEIEN/UART_CTL*x* register to 1. If TEIEN is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

When the TRBS flag is reset to 0, the UART sets TRED/UART_ST*x* register to 1, indicating that the transmit operation has completed. If end of transmission interrupts are enabled (TEIEN = 1), an interrupt request is sent simultaneously to the ITC.

An interrupt occurs if other interrupt conditions are met. You can inspect the TRED flag in the UART interrupt handler routine to determine whether the UART interrupt is attributable to an end of transmission. If TRED is 1, the transmission processing can be terminated.

Receive buffer full interrupt

To use this interrupt, set RIEN/UART_CTL*x* register to 1. If RIEN is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

If the specified volume of received data is loaded into the receive data buffer when a receive buffer full interrupt is enabled (RIEN = 1), the UART outputs an interrupt request to the ITC. If RBFI/UART_CTL*x* register is 0, an interrupt request is output as soon as one received data is loaded into the receive data buffer (when RDRY/UART_ST*x* register is set to 1). If RBFI is 1, an interrupt request is output as soon as two received data are loaded into the receive data buffer (when RD2B/UART_ST*x* register is set to 1).

An interrupt occurs if other interrupt conditions are met. You can inspect the RDRY and RD2B flags in the UART interrupt handler routine to determine whether the UART interrupt is attributable to a receive buffer full. If RDRY or RD2B is 1, the received data can be read from the receive data buffer by the interrupt handler routine.

Receive error interrupt

To use this interrupt, set REIEN/UART_CTL*x* register to 1. If REIEN is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

The UART sets an error flag, PER, FER, or OER/UART_ST*x* register to 1 if a parity error, framing error, or overrun error is detected when receiving data. If receive error interrupts are enabled (REIEN = 1), an interrupt request is sent simultaneously to the ITC.

If other interrupt conditions are satisfied, an interrupt occurs. You can inspect the PER, FER, and OER flags in the UART interrupt handler routine to determine whether the UART interrupt was caused by a receive error. If any of the error flags has the value 1, the interrupt handler routine will proceed with error recovery.

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

15.8 IrDA Interface

This UART module includes an RZI modulator/demodulator circuit enabling implementation of IrDA 1.0-compatible infrared communication function simply by adding basic external circuits.

The transmit data output from the UART transmit shift register is input to the modulator circuit and output from the SOUT*x* pin after the Low pulse has been modulated to a 3 × sclk16 cycle.

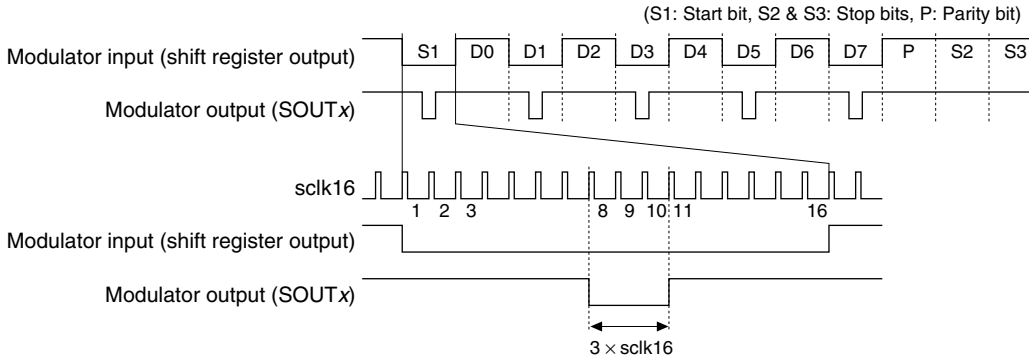


Figure 15.8.1 Transmission Signal Waveform

The received IrDA signal is input to the demodulator circuit and the Low pulse width is converted to 16 × sclk16 cycles before entry to the receive shift register. The demodulator circuit uses the pulse detection clock selected separately from the transfer clock to detect Low pulses input (when minimum pulse width = 1.41 µs/115,200 bps).

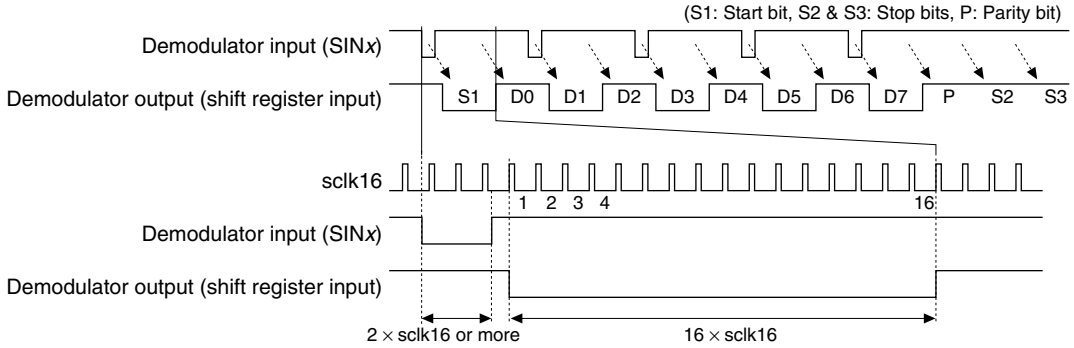


Figure 15.8.2 Receive Signal Waveform

IrDA enable

To use the IrDA interface function, set IRMD/UART_EXP*x* register to 1. This enables the RZI modulator/demodulator circuit.

Note: This setting must be performed before setting other UART conditions.

Serial data transfer control

Data transfer control in IrDA mode is identical to that for normal interfaces. For detailed information on data format settings and data transfer and interrupt control methods, refer to the preceding sections.

15.9 Control Register Details

Table 15.9.1 List of UART Registers

Address	Register name		Function
0x4100	UART_ST0	UART Ch.0 Status Register	Indicates transfer, buffer and error statuses.
0x4101	UART_TXD0	UART Ch.0 Transmit Data Register	Transmit data
0x4102	UART_RXD0	UART Ch.0 Receive Data Register	Receive data
0x4103	UART_MOD0	UART Ch.0 Mode Register	Sets transfer data format.
0x4104	UART_CTL0	UART Ch.0 Control Register	Controls data transfer.
0x4105	UART_EXP0	UART Ch.0 Expansion Register	Sets IrDA mode.
0x4106	UART_BR0	UART Ch.0 Baud Rate Register	Sets baud rate.
0x4107	UART_FMD0	UART Ch.0 Fine Mode Register	Sets fine mode.
0x506c	UART_CLK0	UART Ch.0 Clock Control Register	Selects the baud rate generator clock.

The UART registers are described in detail below.

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

- The following UART bits should be set with transfers disabled (RXEN = 0).
  - All UART_MOD*x* register bits (STPB, PMD, PREN, CHLN)
  - RBFI bit in the UART_CTL*x* register
  - All UART_EXP*x* register bits (IRMD)
  - All UART_BR*x* register bits (BR[7:0])
  - All UART_FMD*x* register bits (FMD[3:0])
  - All UART_CLK*x* register bits (UTCLKD[1:0], UTCLKSRC[1:0], UTCLKE)

UART Ch.*x* Status Register (UART_ST*x*)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
UART Ch.*x* Status Register (UART_ST*x*)	**0x4100** (8 bits)	D7	**TRED**	End of transmission flag	1	Completed	0	Not completed	0	R/W	Reset by writing 1.
		D6	**FER**	Framing error flag	1	Error	0	Normal	0	R/W	
		D5	**PER**	Parity error flag	1	Error	0	Normal	0	R/W	
		D4	**OER**	Overrun error flag	1	Error	0	Normal	0	R/W	
		D3	**RD2B**	Second byte receive flag	1	Ready	0	Empty	0	R	
		D2	**TRBS**	Transmit busy flag	1	Busy	0	Idle	0	R	Shift register status
		D1	**RDRY**	Receive data ready flag	1	Ready	0	Empty	0	R	
		D0	**TDBE**	Transmit data buffer empty flag	1	Empty	0	Not empty	1	R	

D7 TRED: End of Transmission Flag Bit

Indicates whether the transmit operation has completed or not.

1 (R): Completed
0 (R): Not completed (default)
1 (W): Reset to 0
0 (W): Ignored

TRED is set to 1 when the TRBS flag is reset to 0 (when transmission has completed).
TRED is reset by writing 1.

D6 FER: Framing Error Flag Bit

Indicates whether a framing error has occurred or not.

1 (R): Error occurred
0 (R): No error (default)
1 (W): Reset to 0
0 (W): Ignored

FER is set to 1 when a framing error occurs. Framing errors occur when data is received with the stop bit set to 0. FER is reset by writing 1.

D5 PER: Parity Error Flag Bit

Indicates whether a parity error has occurred or not.

1 (R): Error occurred
0 (R): No error (default)
1 (W): Reset to 0
0 (W): Ignored

PER is set to 1 when a parity error occurs. Parity checking is enabled only when PREN/ UART_MOD*x* register is set to 1 and is performed when received data is transferred from the shift register to the receive data buffer. PER is reset by writing 1.

D4 OER: Overrun Error Flag Bit

Indicates whether an overrun error has occurred or not.

1 (R): Error occurred
0 (R): No error (default)
1 (W): Reset to 0
0 (W): Ignored

OER is set to 1 when an overrun error occurs. Overrun errors occur if the receive data buffer is full when data is received in the shift register. The receive data buffer is not overwritten even if this error occurs. The shift register is overwritten as soon as the error occurs.
OER is reset by writing 1.

D3 RD2B: Second Byte Receive Flag Bit

Indicates that the receive data buffer contains two received data.

1 (R): Second byte can be read
0 (R): Second byte not received (default)

RD2B is set to 1 when the second byte of data is loaded into the receive data buffer and is reset to 0 when the first data is read from the receive data buffer.

D2 TRBS: Transmit Busy Flag Bit

Indicates the transmit shift register status.

1 (R): Operating
0 (R): Standby (default)

TRBS is set to 1 when transmit data is loaded from the transmit data buffer into the shift register and is reset to 0 when the data transfer is completed. Inspect TRBS to determine whether the transmit circuit is operating or at standby.

D1 RDRY: Receive Data Ready Flag Bit

Indicates that the receive data buffer contains valid received data.

1 (R): Data can be read
0 (R): Buffer empty (default)

RDRY is set to 1 when received data is loaded into the receive data buffer and is reset to 0 when all data has been read from the receive data buffer.

D0 TDBE: Transmit Data Buffer Empty Flag Bit

Indicates the transmit data buffer status.

1 (R): Buffer empty (default)
0 (R): Data exists

TDBE is reset to 0 when transmit data is written to the transmit data buffer and is set to 1 when the data is transferred to the shift register.

UART Ch.*x* Transmit Data Register (UART_TXD*x*)

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

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

Write transmit data to be set in the transmit data buffer. (Default: 0x0)

The UART starts transmitting when data is written to this register. Data written to TXD[7:0] is retained until sent to the transmit data buffer.

Transmitting data from within the transmit data buffer generates a cause of transmit buffer empty interrupt.

TXD7 (MSB) is invalid in 7-bit mode.

Serial converted data is output from the SOUT*x* pin beginning with the LSB, in which the bits set to 1 are output as High level and bits set to 0 as Low level signals.

This register can also be read.

UART Ch.*x* Receive Data Register (UART_RXD*x*)

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

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

Data in the receive data buffer is read out in sequence, starting with the oldest. Received data is placed in the receive data buffer. The receive data buffer is a 2-byte FIFO that allows proper data reception until it fills, even if data is not read out. If the buffer is full and the shift register also contains received data, an overrun error will occur, unless the data is read out before reception of the subsequent data starts.

The receive circuit includes two receive buffer status flags: RDRY/UART_ST*x* register and RD2B/UART_ST*x* register. The RDRY flag indicates the presence of valid received data in the receive data buffer, while the RD2B flag indicates the presence of two received data in the receive data buffer.

A receive buffer full interrupt occurs when the received data in the receive data buffer reaches the number specified by RBFI/UART_CTL*x* register.

0 is loaded into RXD7 in 7-bit mode.

Serial data input via the SIN*x* pin is converted to parallel, with the initial bit as LSB, the High level bit as 1, and the Low level bit as 0. This data is then loaded into the receive data buffer.

This register is read-only. (Default: 0x0)

UART Ch.*x* Mode Register (UART_MOD*x*)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
UART Ch.*x* Mode Register (UART_MOD*x*)	**0x4103** (8 bits)	D7–5	**–**	reserved	–				–	–	0 when being read.
		D4	**CHLN**	Character length select	1	8 bits	0	7 bits	0	R/W	
		D3	**PREN**	Parity enable	1	With parity	0	No parity	0	R/W	
		D2	**PMD**	Parity mode select	1	Odd	0	Even	0	R/W	
		D1	**STPB**	Stop bit select	1	2 bits	0	1 bit	0	R/W	
		D0	**–**	reserved	–				–	–	0 when being read.

D[7:5] Reserved

D4 CHLN: Character Length Select Bit

Selects the serial transfer data length.

1 (R/W): 8 bits

0 (R/W): 7 bits (default)

D3 PREN: Parity Enable Bit

Enables the parity function.
1 (R/W): With parity
0 (R/W): No parity (default)

PREN is used to select whether received data parity checking is performed and whether a parity bit is added to transmit data. Setting PREN to 1 parity-checks the received data. A parity bit is automatically added to the transmit data. If PREN is set to 0, no parity bit is checked or added.

D2 PMD: Parity Mode Select Bit

Selects the parity mode.
1 (R/W): Odd parity
0 (R/W): Even parity (default)

Writing 1 to PMD selects odd parity; writing 0 to it selects even parity. Parity checking and parity bit addition are enabled only when PREN is set to 1. The PMD setting is disabled if PREN is 0.

D1 STPB: Stop Bit Select Bit

Selects the stop bit length.
1 (R/W): 2 bits
0 (R/W): 1 bit (default)

Writing 1 to STPB selects 2 stop bits; writing 0 to it selects 1 bit. The start bit is fixed at 1 bit.

D0 Reserved

UART Ch.*x* Control Register (UART_CTL*x*)

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

D7 TEIEN: End of Transmission Interrupt Enable Bit

Enables interrupt requests to the ITC when transmit operation has completed.
1 (R/W): Enabled
0 (R/W): Disabled (default)

Set this bit to 1 to terminate transmit processing using interrupts.

D6 REIEN: Receive Error Interrupt Enable Bit

Enables interrupt requests to the ITC when a receive error occurs.
1 (R/W): Enabled
0 (R/W): Disabled (default)

Set this bit to 1 to process receive errors using interrupts.

D5 RIEN: Receive Buffer Full Interrupt Enable Bit

Enables interrupt requests to the ITC caused when the received data quantity in the receive data buffer reaches the quantity specified in RBFI.
1 (R/W): Enabled
0 (R/W): Disabled (default)

Set this bit to 1 to read received data using interrupts.

D4 TIEN: Transmit Buffer Empty Interrupt Enable Bit

Enables interrupt requests to the ITC caused when transmission data in the transmit data buffer is sent to the shift register (i.e. when data transmission begins).
1 (R/W): Enabled
0 (R/W): Disabled (default)

Set this bit to 1 to write data to the transmit data buffer using interrupts.

D[3:2] Reserved

D1 RBFI: Receive Buffer Full Interrupt Condition Setup Bit

Sets the quantity of data in the receive data buffer to generate a receive buffer full interrupt.
1 (R/W): 2 bytes
0 (R/W): 1 byte (default)

If receive buffer full interrupts are enabled (RIEN = 1), the UART outputs an interrupt request to the ITC when the quantity of received data specified by RBFI is loaded into the receive data buffer.
If RBFI is 0, an interrupt request is output as soon as one received data is loaded into the receive data buffer (when RDRY/UART_ST*x* register is set to 1). If RBFI is 1, an interrupt request is output as soon as two received data are loaded into the receive data buffer (when RD2B/UART_ST*x* register is set to 1).

D0 RXEN: UART Enable Bit

Enables data transfer by the UART.
1 (R/W): Enabled
0 (R/W): Disabled (default)

Set RXEN to 1 before starting UART transfers. Setting RXEN to 0 disables data transfers. Set the transfer conditions while RXEN is 0.
Disabling transfers by writing 0 to RXEN also clears the transmit data buffer.

UART Ch.*x* Expansion Register (UART_EXP*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
UART Ch.*x* Expansion Register (UART_EXP*x*)	0x4105 (8 bits)	D7–1	–	reserved	–	–	–	0 when being read.
		D0	IRMD	IrDA mode select	1 On, 0 Off	0	R/W	

D[7:1] Reserved

D0 IRMD: IrDA Mode Select Bit

Switches the IrDA interface function on and off.
1 (R/W): On
0 (R/W): Off (default)

Set IRMD to 1 to use the IrDA interface. When IRMD is set to 0, this module functions as a normal UART, with no IrDA functions.

UART Ch.*x* Baud Rate Register (UART_BR*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
UART Ch.*x* Baud Rate Register (UART_BR*x*)	0x4106 (8 bits)	D7–0	BR[7:0]	Baud rate setting	0x0 to 0xff	0x0	R/W	

D[7:0] BR[7:0]: Baud Rate Setting Bits

Sets the initial counter value of the baud rate generator. (Default: 0x0)
The counter in the baud rate generator repeats counting from the value set in this register to occurrence of counter underflow to generate the transfer (sampling) clock.
Use the following equations to calculate the initial counter value for obtaining the desired transfer rate.

$$bps = \frac{ct\_clk}{\{(BR + 1) \times 16 + FMD\}}$$

$$BR = \left(\frac{ct\_clk}{bps} - FMD - 16\right) \div 16$$

ct_clk: Count clock frequency (Hz)
BR: BR[7:0] setting (0 to 255)
bps: Transfer rate (bit/s)
FMD: FMD[3:0] (fine mode) setting (0 to 15)

UART Ch.*x* Fine Mode Register (UART_FMD*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
UART Ch.*x* Fine Mode Register (UART_FMD*x*)	**0x4107** (8 bits)	D7–4	**–**	reserved	–	–	–	0 when being read.
		D3–0	**FMD[3:0]**	Fine mode setup	0x0 to 0xf	0x0	R/W	Set a number of times to insert delay into a 16-underflow period.

D[7:4] Reserved

D[3:0] FMD[3:0]: Fine Mode Setup Bits

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

FMD[3:0] specifies the delay pattern to be inserted into a 16 underflow period of the baud rate generator output clock. Inserting one delay extends the output clock cycle by one count clock cycle.

Table 15.9.2 Delay Patterns Specified by FMD[3:0]

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

D: Indicates the insertion of a delay cycle.

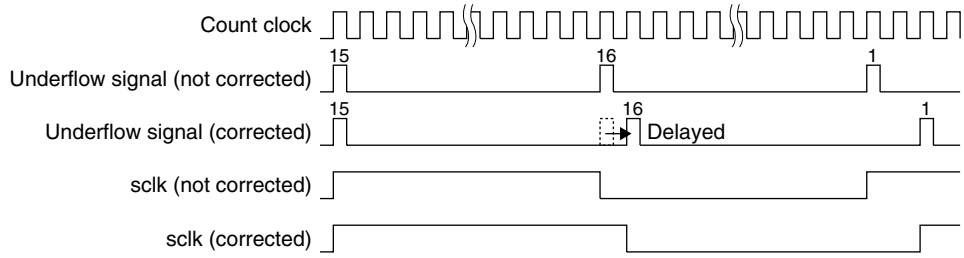


Figure 15.9.1 Delay Cycle Insertion in Fine Mode

UART Ch.*x* Clock Control Register (UART_CLK*x*)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
UART Ch.*x* Clock Control Register (UART_CLK*x*)	**0x506c** (8 bits)	D7–6	**–**	reserved	–		–	–	0 when being read.
		D5–4	**UTCLKD [1:0]**	Clock division ratio select	UTCLKD[1:0]	Division ratio	0x0	R/W	When the clock source is OSC3B or OSC3A
					0x3	1/8			
					0x2	1/4			
					0x1	1/2			
					0x0	1/1			
		D3–2	**UTCLKSRC [1:0]**	Clock source select	UTCLKSRC [1:0]	Clock source	0x0	R/W	
					0x3	External clock			
					0x2	OSC3A			
					0x1	OSC1			
					0x0	OSC3B			
		D1	**–**	reserved	–		–	–	0 when being read.
		D0	**UTCLKE**	Count clock enable	1 Enable	0 Disable	0	R/W	

D[7:6] Reserved

D[5:4] UTCLKD[1:0]: Clock Division Ratio Select Bits

Selects the division ratio for generating the count clock of the baud rate generator when OSC3B or OSC3A is used as the clock source.

Table 15.9.3 OSC3B/OSC3A Division Ratio Selection

UTCLKD[1:0]	Division ratio
0x3	1/8
0x2	1/4
0x1	1/2
0x0	1/1

(Default: 0x0)

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

Selects the count clock source for the baud rate generator.

Table 15.9.4 Clock Source Selection

UTCLKSRC[1:0]	Clock source
0x3	External clock (SCLK*x*)
0x2	OSC3A
0x1	OSC1
0x0	OSC3B

(Default: 0x0)

D1 Reserved

D0 UTCLKE: Count Clock Enable Bit

Enables or disables the count clock supply to the counter of the baud rate generator.
1 (R/W): Enabled (on)
0 (R/W): Disabled (off) (default)

The UTCLKE default setting is 0, which disables the clock supply. Setting UTCLKE to 1 sends the clock selected to the counter.

16 SPI

16.1 SPI Module Overview

The S1C17651 includes a synchronized serial interface module (SPI).
The following shows the main features of the SPI:

- Number of channels: 1 channel
- Supports both master and slave modes.
- Data length: 8 bits fixed
- Supports both MSB first and LSB first modes.
- Contains one-byte receive data buffer and one-byte transmit data buffer.
- Supports full-duplex communications.
- Data transfer timing (clock phase and polarity variations) is selectable from among 4 types.
- Can generate receive buffer full and transmit buffer empty interrupts.

Figure 16.1.1 shows the SPI module configuration.

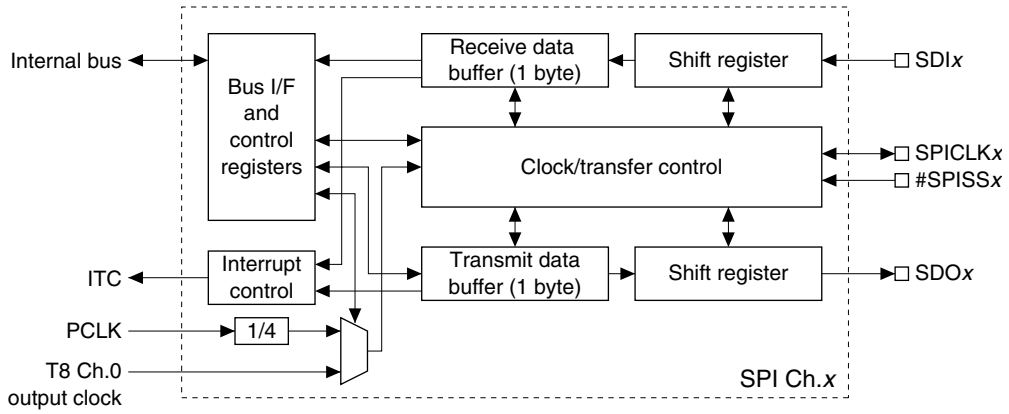


Figure 16.1.1 SPI Module Configuration

Note: The letter '*x*' in register and pin names refers to a channel number (0).
Example: SPI_CTL*x* register
Ch.0: SPI_CTL0 register

16.2 SPI Input/Output Pins

Table 16.2.1 lists the SPI pins.

Table 16.2.1 List of SPI Pins

Pin name	I/O	Qty	Function
SDI0 (Ch.0)	I	1	SPI data input pin Inputs serial data from SPI bus.
SDO0 (Ch.0)	O	1	SPI data output pin Outputs serial data to SPI bus.
SPICLK0 (Ch.0)	I/O	1	SPI external clock input/output pin Outputs SPI clock when SPI is in master mode. Inputs external clock when SPI is used in slave mode.
#SPISS0 (Ch.0)	I	1	SPI slave select signal (active Low) input pin SPI (Slave mode) is selected as a slave device by Low input to this pin.

Note: Use an I/O (P) port to output the slave select signal when the SPI module is configured to master mode.

The SPI input/output pins (SDI*x*, SDO*x*, SPICLK*x*, #SPISS*x*) are shared with I/O ports and are initially set as general purpose I/O port pins. The pin functions must be switched using the port function select bits to use the general purpose I/O port pins as SPI input/output pins.
For detailed information on pin function switching, see the "I/O Ports (P)" chapter.

16.3 SPI Clock

The master mode SPI uses the 8-bit timer (T8) Ch.0 output clock or a PCLK/4 clock to generate the SPI clock. This clock is output from the SPICLK*x* pin to the slave device while also driving the shift register.
Use MCLK/SPI_CTL*x* register to select whether the T8 Ch.0 output clock or PCLK/4 clock is used.
Setting MCLK to 1 selects the T8 Ch.0 output clock; setting to 0 selects the PCLK/4 clock.
Using the T8 Ch.0 output clock enables programmable transfer rates. For more information on T8 control, see the "8-bit timer (T8)" chapter.

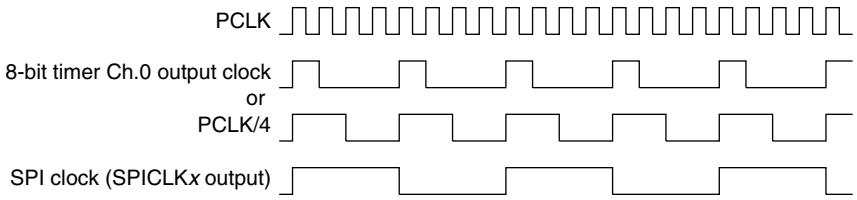


Figure 16.3.1 Master Mode SPI Clock

In slave mode, the SPI clock is input via the SPICLK*x* pin.

16.4 Data Transfer Condition Settings

The SPI module can be set to master or slave modes. The SPI clock polarity/phase and bit direction (MSB first/LSB first) can also be set via the SPI_CTL*x* register. The data length is fixed at 8 bits.

Note: Make sure the SPI module is halted (SPEN/SPI_CTL*x* register = 0) before master/slave mode selection and clock condition settings.

Master/slave mode selection

MSSL/SPI_CTL*x* register is used to set the SPI module to master mode or slave mode. Setting MSSL to 1 sets master mode; setting it to 0 (default) sets slave mode. In master mode, data is transferred using the internal clock. In slave mode, data is transferred by inputting the master device clock.

SPI clock polarity and phase settings

The SPI clock polarity is selected by CPOL/SPI_CTL*x* register. Setting CPOL to 1 treats the SPI clock as active Low; setting it to 0 (default) treats it as active High.
The SPI clock phase is selected by CPHA/SPI_CTL*x* register.

As shown below, these control bits set transfer timing.

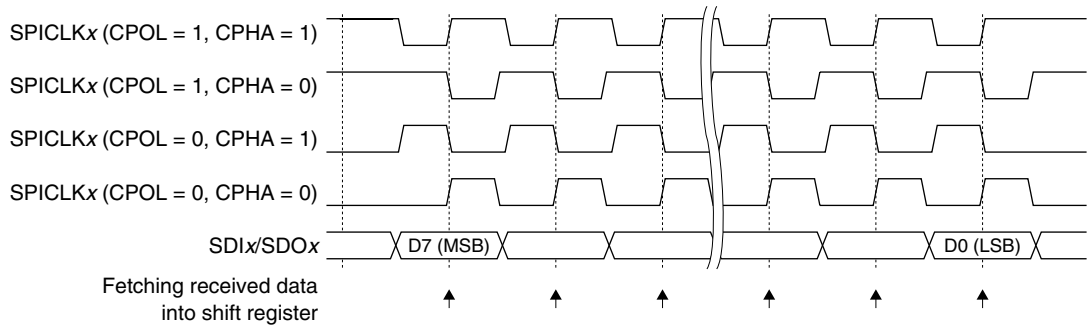


Figure 16.4.1 Clock and Data Transfer Timing

MSB first/LSB first settings

Use MLSB/SPI_CTL*x* register to select whether the data MSB or LSB is input/output first.
MSB first is selected when MLSB is 0 (default); LSB first is selected when MLSB is 1.

16.5 Data Transfer Control

Make the following settings before starting data transfers.
(1) Select the SPI clock source. (See Section 16.3.)
(2) Select master mode or slave mode. (See Section 16.4.)
(3) Set clock conditions. (See Section 16.4.)
(4) Set the interrupt conditions to use SPI interrupts. (See Section 16.6.)

Note: Make sure the SPI is halted (SPEN/SPI_CTL*x* register = 0) before setting the above conditions.

Enabling data transfers

Set SPEN/SPI_CTL*x* register to 1 to enable SPI operations. This enables SPI transfers and clock input/output.

Note: Do not set SPEN to 0 when the SPI module is transferring data.

Data transmission control

To start data transmission, write the transmit data to SPTDB[7:0]/SPI_TXD*x* register.
The data is written to the transmit data buffer, and the SPI module starts sending data. The buffer data is sent to the transmit shift register. In master mode, the module starts clock output from the SPICLK*x* pin. In slave mode, the module awaits clock input from the SPICLK*x* pin. The data in the shift register is shifted in sequence at the clock rising or falling edge, as determined by CPHA/SPI_CTL*x* register and CPOL/SPI_CTL*x* register (see Figure 16.4.1) and sent from the SDO*x* pin.

Note: Make sure that SPEN is set to 1 before writing data to the SPI_TXD*x* register.

The SPI module includes two status flags for transfer control: SPTBE/SPI_ST*x* register and SPBSY/SPI_ST*x* register.
The SPTBE flag indicates the transmit data buffer status. This flag switches to 0 when the application program writes data to the SPI_TXD*x* register (transmit data buffer) and reverts to 1 when the buffer data is sent to the transmit shift register. An interrupt can be generated when this flag is set to 1 (see Section 16.6). Subsequent data is sent after confirming that the transmit data buffer is empty either by using this interrupt or by inspecting the SPTBE flag. The transmit data buffer size is 1 byte, but a shift register is provided separately to allow data to be written while the previous data is being sent. Always confirm that the transmit data buffer is empty before writing transmit data. Writing data while the SPTBE flag is 0 will overwrite earlier transmit data inside the transmit data buffer.

In master mode, the SPBSY flag indicates the shift register status. This flag switches to 1 when transmit data is loaded from the transmit data buffer to the shift register and reverts to 0 once the data is sent. Read this flag to check whether the SPI module is operating or at standby.
In slave mode, SPBSY flag indicates the SPI slave selection signal (#SPISS*x* pin) status. The flag is set to 1 when the SPI module is selected as a slave module and is set to 0 when the module is not selected.

Note: When the SPI module is used in master mode with CPHA set to 0, the clock may change a minimum of one system clock (PCLK) cycle time from change of the first transmit data bit.

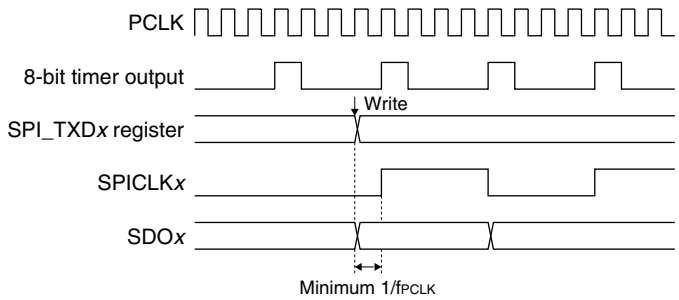


Figure 16.5.1 SDO*x* and SPICLK*x* Change Timings when CPHA = 0

The half SPICLK*x* cycle will be secured from change of data to change of the clock for the second and following transmit data bits and the second and following bytes during continuous transfer.

Data reception control

In master mode, write dummy data to SPTDB[7:0]/SPI_TXD*x* register. Writing to the SPI_TXD*x* register creates the trigger for reception as well as transmission start. Writing actual transmit data enables simultaneous transmission and reception.
This starts the SPI clock output from the SPICLK*x* pin.

Note: Make sure that SPEN is set to 1 before writing data to the SPI_TXD*x* register.

In slave mode, the module waits until the clock is input from the SPICLK*x* pin. There is no need to write to the SPI_TXD*x* register if no transmission is required. The receiving operation is started by the clock input from the master device. If data is transmitted simultaneously, write transmit data to the SPI_TXD*x* register before the clock is input.

The data is received in sequence in the shift register at the rising or falling edge of the clock determined by CPHA/SPI_CTL*x* register and CPOL/SPI_CTL*x* register. (See Figure 16.4.1.) The received data is loaded into the receive data buffer once the 8 bits of data are received in the shift register.
The received data in the buffer can be read from SPRDB[7:0]/SPI_RXD*x* register.

The SPI module includes SPRBF/SPI_ST*x* register for reception control.
The SPRBF flag indicates the receive data buffer status. This flag is set to 1 when the data received in the shift register is loaded into the receive data buffer, indicating that the received data can be read out. It reverts to 0 when the buffer data is read out from the SPI_RXD*x* register. An interrupt can be generated as soon as the flag is set to 1 (see Section 16.6). The received data should be read out either by using this interrupt or by inspecting the SPRBF flag to confirm that the receive data buffer contains valid received data. The receive data buffer is 1 byte in size, but a shift register is also provided, enabling received data to be retained in the buffer even while the subsequent data is being received. Note that the receive data buffer should be read out before receiving the subsequent data is complete. If receiving the subsequent data is complete before the receive data buffer contents are read out, the newly received data will overwrite the previous received data in the buffer.

In master mode, the SPBSY flag indicating the shift register status can be used in the same way while transferring data.

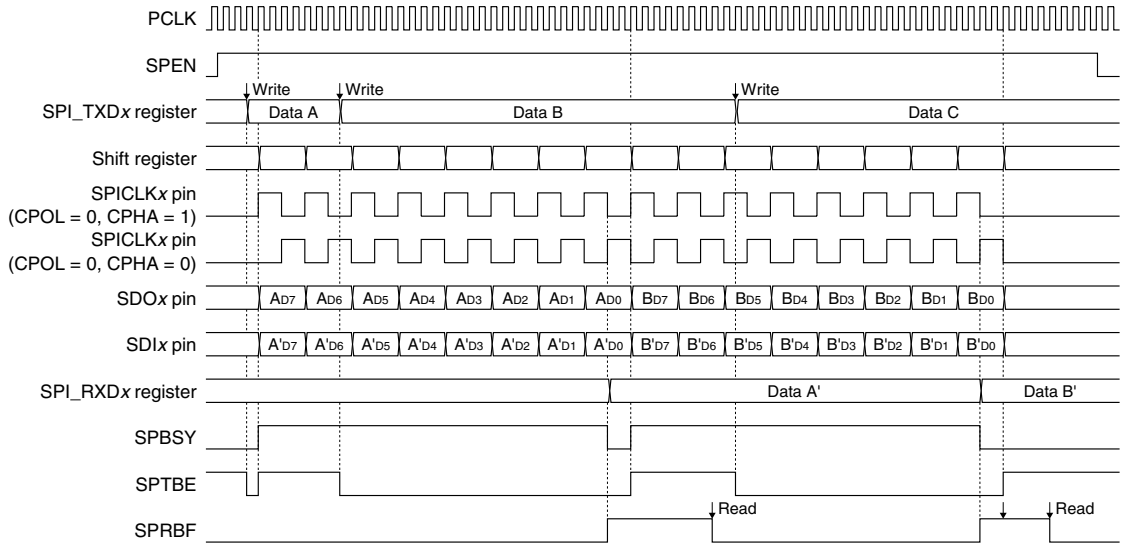


Figure 16.5.2 Data Transmission/Receiving Timing Chart (MSB first)

Disabling data transfers

After a data transfer is completed (both transmission and reception), write 0 to SPEN to disable data transfers. Confirm that the SPTBE flag is 1 and the SPBSY flag is 0 before disabling data transfer.

The data being transferred cannot be guaranteed if SPEN is set to 0 while data is being sent or received.

16.6 SPI Interrupts

Each channel of the SPI module includes a function for generating the following two different types of interrupts.

- Transmit buffer empty interrupt
- Receive buffer full interrupt

The SPI channel outputs one interrupt signal shared by the two above interrupt causes to the interrupt controller (ITC). Inspect the status flag to determine the interrupt cause occurred.

Transmit buffer empty interrupt

To use this interrupt, set SPTIE/SPI_CTL*x* register to 1. If SPTIE is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

When transmit data written to the transmit data buffer is transferred to the shift register, the SPI module sets SPTBE/SPI_ST*x* register to 1, indicating that the transmit data buffer is empty. If transmit buffer empty interrupts are enabled (SPTIE = 1), an interrupt request is sent simultaneously to the ITC.

An interrupt occurs if other interrupt conditions are met. You can inspect the SPTBE flag in the SPI interrupt handler routine to determine whether the SPI interrupt is attributable to a transmit buffer empty. If SPTBE is 1, the next transmit data can be written to the transmit data buffer by the interrupt handler routine.

Receive buffer full interrupt

To use this interrupt, set SPRIE/SPI_CTL*x* register to 1. If SPRIE is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

When data received in the shift register is loaded into the receive data buffer, the SPI module sets SPRBF/SPI_ST*x* register to 1, indicating that the receive data buffer contains readable received data. If receive buffer full interrupts are enabled (SPRIE = 1), an interrupt request is output to the ITC at the same time.

An interrupt occurs if other interrupt conditions are met. You can inspect the SPRBF flag in the SPI interrupt handler routine to determine whether the SPI interrupt is attributable to a receive buffer full. If SPRBF is 1, the received data can be read from the receive data buffer by the interrupt handler routine.

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

16.7 Control Register Details

Table 16.7.1 List of SPI Registers

Address	Register name		Function
0x4320	SPI_ST0	SPI Ch.0 Status Register	Indicates transfer and buffer statuses.
0x4322	SPI_TXD0	SPI Ch.0 Transmit Data Register	Transmit data
0x4324	SPI_RXD0	SPI Ch.0 Receive Data Register	Receive data
0x4326	SPI_CTL0	SPI Ch.0 Control Register	Sets the SPI mode and enables data transfer.

The SPI registers are described in detail below.

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

SPI Ch.*x* Status Register (SPI_ST*x*)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
SPI Ch.*x* Status Register (SPI_ST*x*)	**0x4320** (16 bits)	D15–3	**–**	reserved	–				–	–	0 when being read.
		D2	**SPBSY**	Transfer busy flag (master)	1	Busy	0	Idle	0	R	
				ss signal low flag (slave)	1	ss = L	0	ss = H			
		D1	**SPRBF**	Receive data buffer full flag	1	Full	0	Not full	0	R	
		D0	**SPTBE**	Transmit data buffer empty flag	1	Empty	0	Not empty	1	R	

D[15:3] Reserved

D2 SPBSY: Transfer Busy Flag Bit (Master Mode)/ss Signal Low Flag Bit (Slave Mode)

Master mode

Indicates the SPI transfer status.

1 (R): Operating

0 (R): Standby (default)

SPBSY is set to 1 when the SPI starts data transfer in master mode and is maintained at 1 while transfer is underway. It is cleared to 0 once the transfer is complete.

Slave mode

Indicates the slave selection (#SPISS*x*) signal status.

1 (R): Low level (this SPI is selected)

0 (R): High level (this SPI is not selected) (default)

SPBSY is set to 1 when the master device asserts the #SPISS*x* signal to select this SPI module (slave device). It is returned to 0 when the master device clears the SPI module selection by negating the #SPISS*x* signal.

D1 SPRBF: Receive Data Buffer Full Flag Bit

Indicates the receive data buffer status.

1 (R): Data full

0 (R): No data (default)

SPRBF is set to 1 when data received in the shift register is sent to the receive data buffer (when receiving is completed), indicating that the data can be read. It reverts to 0 once the buffer data is read from the SPI_RXD*x* register.

D0 SPTBE: Transmit Data Buffer Empty Flag Bit

Indicates the transmit data buffer status.

1 (R): Empty (default)

0 (R): Data exists

SPTBE is set to 0 when transmit data is written to the SPI_TXD*x* register (transmit data buffer), and is set to 1 when the data is transferred to the shift register (when transmission starts).
Transmission data must be written to the SPI_TXD*x* register when this bit is 1.

SPI Ch.*x* Transmit Data Register (SPI_TXD*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SPI Ch.*x* Transmit Data Register (SPI_TXD*x*)	0x4322 (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	SPTDB[7:0]	SPI transmit data buffer SPTDB7 = MSB SPTDB0 = LSB	0x0 to 0xff	0x0	R/W	

D[15:8] Reserved

D[7:0] SPTDB[7:0]: SPI Transmit Data Buffer Bits

Sets transmit data to be written to the transmit data buffer. (Default: 0x0)

In master mode, transmission is started by writing data to this register. In slave mode, the contents of this register are sent to the shift register and transmission begins when the clock is input from the master.

SPTBE/SPI_ST*x* register is set to 1 (empty) as soon as data written to this register has been transferred to the shift register. A transmit buffer empty interrupt is generated at the same time. The subsequent transmit data can then be written, even while data is being transmitted.

Serial converted data is output from the SDO*x* pin, with the bit set to 1 as High level and the bit set to 0 as Low level.

Note: Make sure that SPEN is set to 1 before writing data to the SPI_TXD*x* register to start data transmission/reception.

SPI Ch.*x* Receive Data Register (SPI_RXD*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SPI Ch.*x* Receive Data Register (SPI_RXD*x*)	0x4324 (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	SPRDB[7:0]	SPI receive data buffer SPRDB7 = MSB SPRDB0 = LSB	0x0 to 0xff	0x0	R	

D[15:8] Reserved

D[7:0] SPRDB[7:0]: SPI Receive Data Buffer Bits

Contains the received data. (Default: 0x0)

SPRBF/SPI_ST*x* register is set to 1 (data full) as soon as data is received and the shift register data has been transferred to the receive data buffer. A receive buffer full interrupt is generated at the same time. Data can then be read until subsequent data is received. If receiving the subsequent data is completed before the register has been read out, the new received data overwrites the contents.

Serial data input from the SDI*x* pin is converted to parallel, with the High level bit set to 1 and the Low level bit set to 0. The data is the loaded into this register.

This register is read-only.

SPI Ch.*x* Control Register (SPI_CTL*x*)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
SPI Ch.*x* Control Register (SPI_CTL*x*)	0x4326 (16 bits)	D15–10	–	reserved	–				–	–	0 when being read.
		D9	MCLK	SPI clock source select	1	T8 Ch.0	0	PCLK/4	0	R/W	
		D8	MLSB	LSB/MSB first mode select	1	LSB	0	MSB	0	R/W	
		D7–6	–	reserved	–				–	–	0 when being read.
		D5	SPRIE	Receive data buffer full int. enable	1	Enable	0	Disable	0	R/W	
		D4	SPTIE	Transmit data buffer empty int. enable	1	Enable	0	Disable	0	R/W	
		D3	CPHA	Clock phase select	1	Data out	0	Data in	0	R/W	These bits must be set before setting SPEN to 1.
		D2	CPOL	Clock polarity select	1	Active L	0	Active H	0	R/W	
		D1	MSSL	Master/slave mode select	1	Master	0	Slave	0	R/W	
		D0	SPEN	SPI enable	1	Enable	0	Disable	0	R/W	

Note: Do not access to the SPI_CTL*x* register while SPBSY/SPI_ST*x* register is set to 1 or SPRBF/SPI_ST*x* register is set to 1 (while data is being transmitted/received).

D[15:10] Reserved

D9 MCLK: SPI Clock Source Select Bit

Selects the SPI clock source.
1 (R/W): 8-bit timer Ch.0
0 (R/W): PCLK/4 (default)

D8 MLSB: LSB/MSB First Mode Select Bit

Selects whether data is transferred with MSB first or LSB first.
1 (R/W): LSB first
0 (R/W): MSB first (default)

D[7:6] Reserved

D5 SPRIE: Receive Data Buffer Full Interrupt Enable Bit

Enables or disables SPI receive data buffer full interrupts.
1 (R/W): Enabled
0 (R/W): Disabled (default)

Setting SPRIE to 1 enables the output of SPI interrupt requests to the ITC due to a receive data buffer full. These interrupt requests are generated when the data received in the shift register is transferred to the receive data buffer (when reception is completed).
SPI interrupts are not generated by receive data buffer full if SPRIE is set to 0.

D4 SPTIE: Transmit Data Buffer Empty Interrupt Enable Bit

Enables or disables SPI transmit data buffer empty interrupts.
1 (R/W): Enabled
0 (R/W): Disabled (default)

Setting SPTIE to 1 enables the output of SPI interrupt requests to the ITC due to a transmit data buffer empty. These interrupt requests are generated when the data written to the transmit data buffer is transferred to the shift register (when transmission starts).
SPI interrupts are not generated by transmit data buffer empty if SPTIE is set to 0.

D3 CPHA: Clock Phase Select Bit

Selects the SPI clock phase. (Default: 0)
Set the data transfer timing together with CPOL. (See Figure 16.7.1.)

D2 CPOL: Clock Polarity Select Bit

Selects the SPI clock polarity.
1 (R/W): Active Low
0 (R/W): Active High (default)

Set the data transfer timing together with CPHA. (See Figure 16.7.1.)

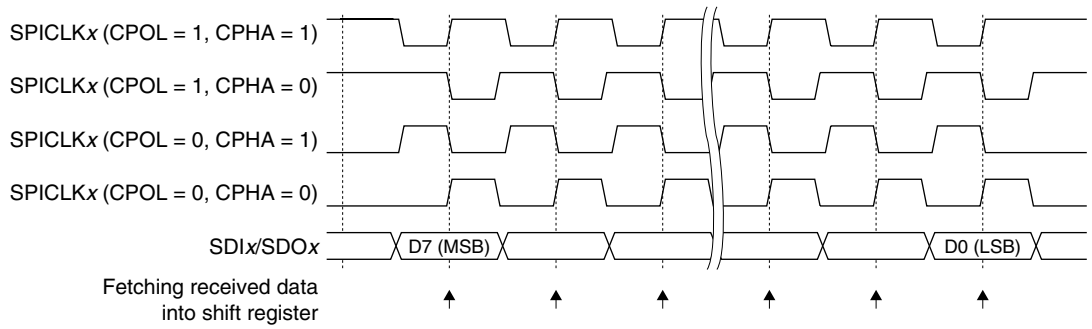


Figure 16.7.1 Clock and Data Transfer Timing

D1 MSSL: Master/Slave Mode Select Bit

Sets the SPI module to master or slave mode.
1 (R/W): Master mode
0 (R/W): Slave mode (default)

Setting MSSL to 1 selects master mode; setting it to 0 selects slave mode. Master mode performs data transfer with the internal clock. In slave mode, data is transferred by inputting the clock from the master device.

D0 SPEN: SPI Enable Bit

Enables or disables SPI module operation.
1 (R/W): Enabled
0 (R/W): Disabled (default)

Setting SPEN to 1 starts the SPI module operation, enabling data transfer.
Setting SPEN to 0 stops the SPI module operation.

Note: The SPEN bit should be set to 0 before setting the CPHA, CPOL, and MSSL bits.

17 LCD Driver (LCD)

17.1 LCD Module Overview

The S1C17651 includes an LCD driver capable of driving an LCD panel with up to 80 segments (20 segments × 4 commons).
The main features of the LCD driver are listed below.

• Number of SEG and COM outputs	20 SEG × 4/3/2/1 COM
• Drive bias	1/3 bias (fixed)
• Display data RAM	20 bytes
• Frame frequency configuration	Selectable from four different frequencies
• LCD display mode	Normal display mode All on mode All off mode Inverted display mode
• Other functions	LFRO signal output, frame interrupt

Figure 17.1.1 shows the LCD driver and drive power supply configuration.

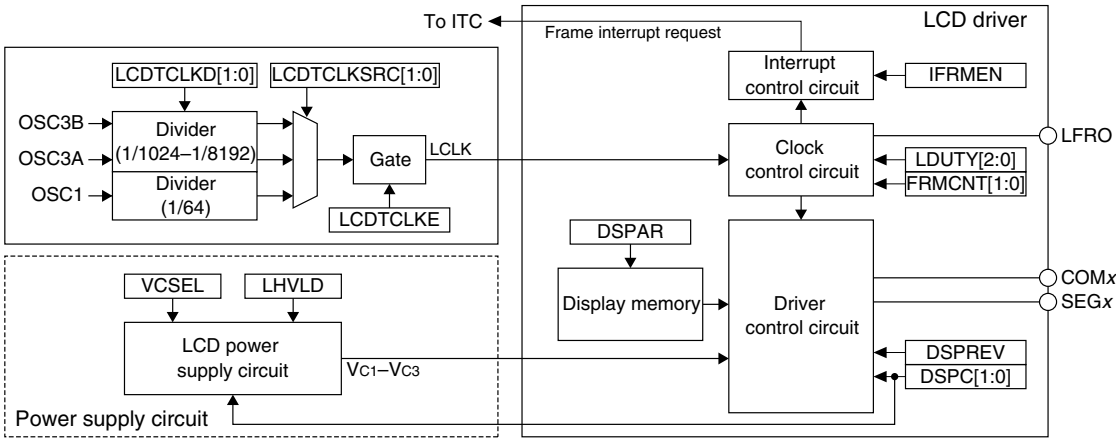


Figure 17.1.1 LCD Driver and Driver Power Supply Configuration

17.2 LCD Power Supply

The LCD drive voltages V_{C1} to V_{C3} are generated by the on-chip LCD power supply circuit. No external power supply is needed. For more information on the LCD power supply, see the "Power Supply" chapter.

17.3 LCD Clock

Figure 17.3.1 shows the LCD clock supply system.

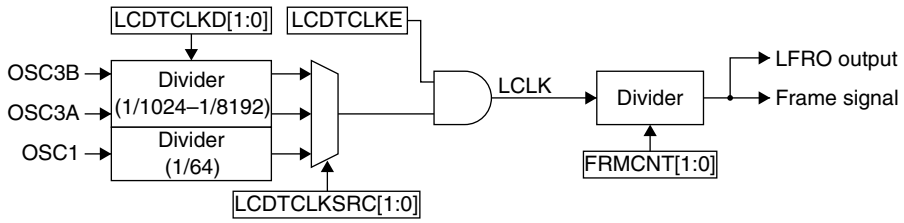


Figure 17.3.1 LCD Clock System

17.3.1 LCD Operating Clock (LCLK)

Clock source selection

Select the clock source from OSC3B, OSC3A, and OSC1 using LCDTCLKSRC[1:0]/LCD_TCLK register.

Table 17.3.1.1 Clock Source Selection

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

(Default: 0x0)

Clock division ratio selection

When the clock source is OSC1

No division ratio needs to be selected when OSC1 is selected for the clock source. The OSC1 clock is used as LCLK after dividing by 64 (typ. 512 Hz).

When the clock source is OSC3B/OSC3A

When OSC3B/OSC3A is selected for the clock source, use LCDTCLKD[1:0]/LCD_TCLK register to select the division ratio.

Table 17.3.1.2 OSC3B/OSC3A Division Ratio Selection

LCDTCLKD[1:0]	Division ratio
0x3	1/8192
0x2	1/4096
0x1	1/2048
0x0	1/1024

(Default: 0x0)

Select a division ratio so that it will generate the LCLK frequency nearest 512 Hz.

Clock enable

The LCLK supply is enabled with LCDTCLKE/LCD_TCLK register. The LCDTCLKE default setting is 0, which stops the clock. Setting LCDTCLKE to 1 feeds the clock generated as above to the LCD driver. If no LCD display is required, stop the clock to reduce current consumption.
If LCLK is not supplied, the LCD cannot display. However, the LCD driver control registers and display memory can be accessed even if LCLK is stopped.

Note: Be sure to set LCDTCLKE to 0 before selecting a clock division ratio.

17.3.2 Frame Signal

The LCD driver generates the frame signal by dividing LCLK. The clock division ratio can be set using FRMCNT[1:0]/LCD_CCTL register. Figures 17.4.2.1 to 17.4.2.4 show one cycle of the frame frequency as "1 frame." Tables 17.3.2.1 and 17.3.2.2 list the frame frequencies that can be programmed.

When the clock source is OSC1

Table 17.3.2.1 Frame Frequency Settings (when the clock source is OSC1 = 32.768 kHz (LCLK = 512 Hz))

Drive duty (LDUTY[2:0] setting)	FRMCNT[1:0] setting (LCLK division ratio)			
	0x0	0x1	0x2	0x3
1/4 duty (0x3)	128 Hz (1/4)	64 Hz (1/8) *	42.67 Hz (1/12)	32 Hz (1/16)
1/3 duty (0x2)	85.33 Hz (1/6)	56.89 Hz (1/9)	42.67 Hz (1/12)	34.13 Hz (1/15)
1/2 duty (0x1)	128 Hz (1/4)	64 Hz (1/8)	42.67 Hz (1/12)	32 Hz (1/16)
Static (0x0)	128 Hz (1/4)	64 Hz (1/8)	42.67 Hz (1/12)	32 Hz (1/16)

* Default setting

When the clock source is OSC3B/OSC3A

Table 17.3.2.2 Frame Frequency Settings (when the clock source is OSC3B/OSC3A)

Drive duty (LDUTY[2:0] setting)	FRMCNT[1:0] setting			
	0x0	0x1	0x2	0x3
1/4 duty (0x3)	$\frac{f_{OSC3} \times LCDTCLKD}{4}$	$\frac{f_{OSC3} \times LCDTCLKD}{8}$ *	$\frac{f_{OSC3} \times LCDTCLKD}{12}$	$\frac{f_{OSC3} \times LCDTCLKD}{16}$
1/3 duty (0x2)	$\frac{f_{OSC3} \times LCDTCLKD}{6}$	$\frac{f_{OSC3} \times LCDTCLKD}{9}$	$\frac{f_{OSC3} \times LCDTCLKD}{12}$	$\frac{f_{OSC3} \times LCDTCLKD}{15}$
1/2 duty (0x1)	$\frac{f_{OSC3} \times LCDTCLKD}{4}$	$\frac{f_{OSC3} \times LCDTCLKD}{8}$	$\frac{f_{OSC3} \times LCDTCLKD}{12}$	$\frac{f_{OSC3} \times LCDTCLKD}{16}$
Static (0x0)	$\frac{f_{OSC3} \times LCDTCLKD}{4}$	$\frac{f_{OSC3} \times LCDTCLKD}{8}$	$\frac{f_{OSC3} \times LCDTCLKD}{12}$	$\frac{f_{OSC3} \times LCDTCLKD}{16}$

* Default setting
f_{OSC3}: OSC3B or OSC3A clock frequency
LCDTCLKD: OSC3B/OSC3A division ratio (1/1024 to 1/8192)

The frame signal generated can be output to an external device via the LFRO pin. However, the output pin must be switched for LFRO output using the port function select bit, as the pin is configured for an I/O port by default. For detailed information on pin function switching, see the "I/O Ports (P)" chapter.

17.4 Drive Duty Control

17.4.1 Drive Duty Switching

Drive duty can be set to 1/4, 1/3, 1/2 or static drive using LDUTY[2:0]/LCD_CCTL register. Table 17.4.1.1 shows the correspondence between LDUTY[2:0] settings, drive duty, and maximum number of display segments.

Table 17.4.1.1 Drive Duty Settings (S1C17624/622)

LDUTY[2:0]	Duty	Valid COM pins	Valid SEG pins	Max. number of display segments
0x7–0x4	Reserved	–	–	–
0x3	1/4	COM0 to COM3	SEG0 to SEG19	80 segments
0x2	1/3	COM0 to COM2	SEG0 to SEG19	60 segments
0x1	1/2	COM0 to COM1	SEG0 to SEG19	40 segments
0x0	Static	COM0	SEG0 to SEG19	20 segments

(Default: 0x3)

The drive bias is fixed at 1/3 (three potentials V_{C1}, V_{C2}, V_{C3}) for all duty settings.

17.4.2 Drive Waveform

Figures 17.4.2.1 to 17.4.2.4 shows the drive waveforms according to the duty selections.

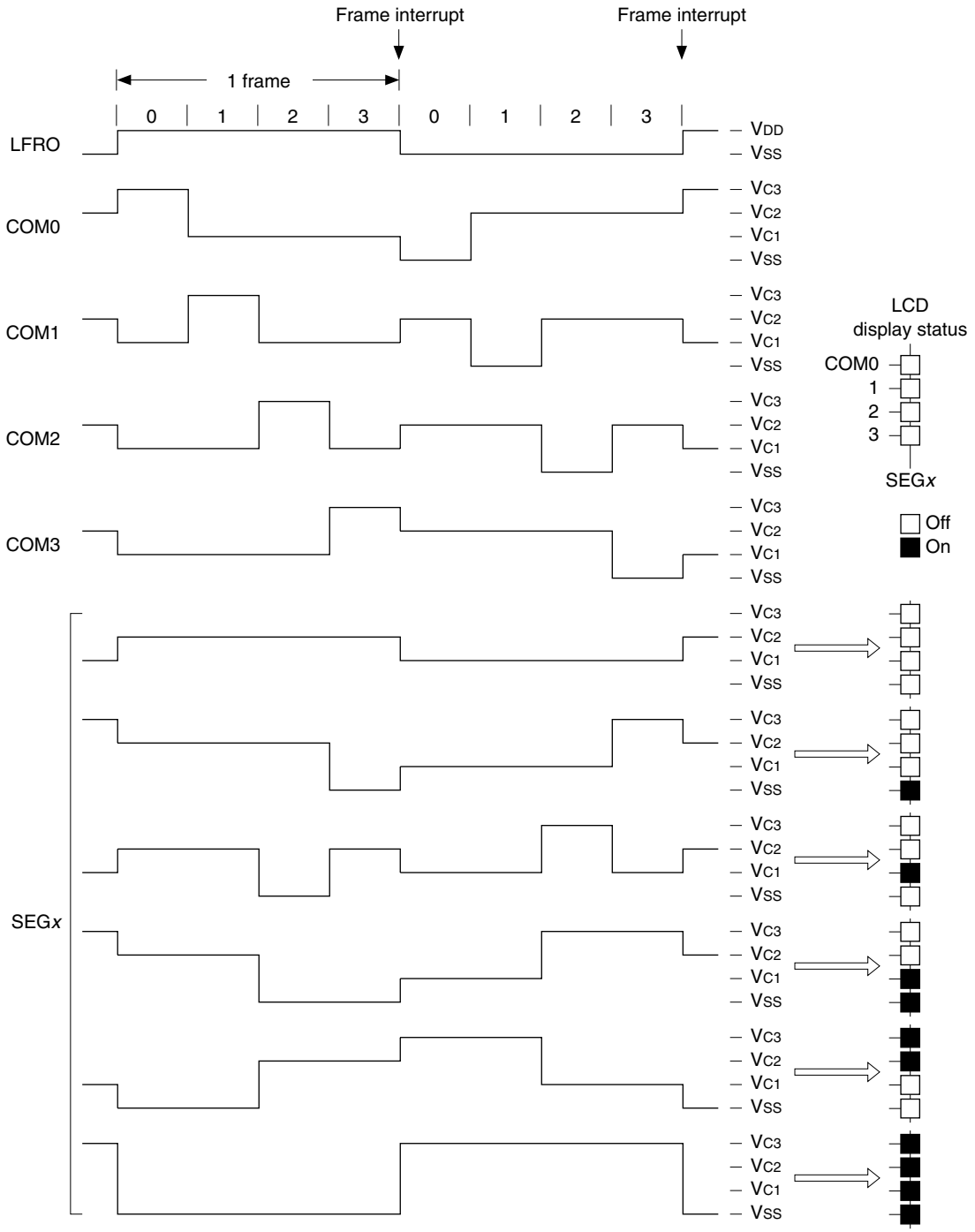


Figure 17.4.2.1 1/4 Duty Drive Waveform

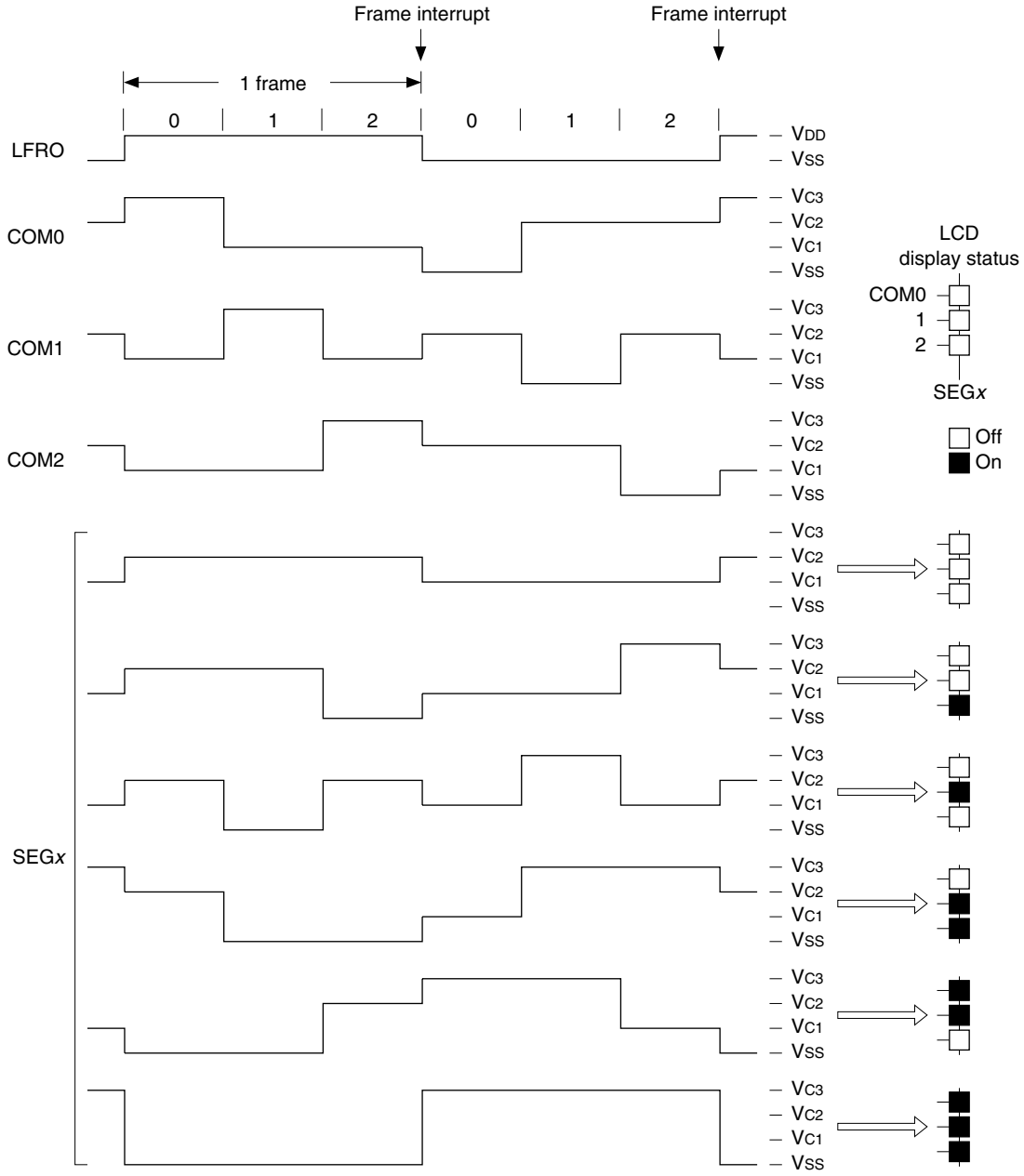


Figure 17.4.2.2 1/3 Duty Drive Waveform

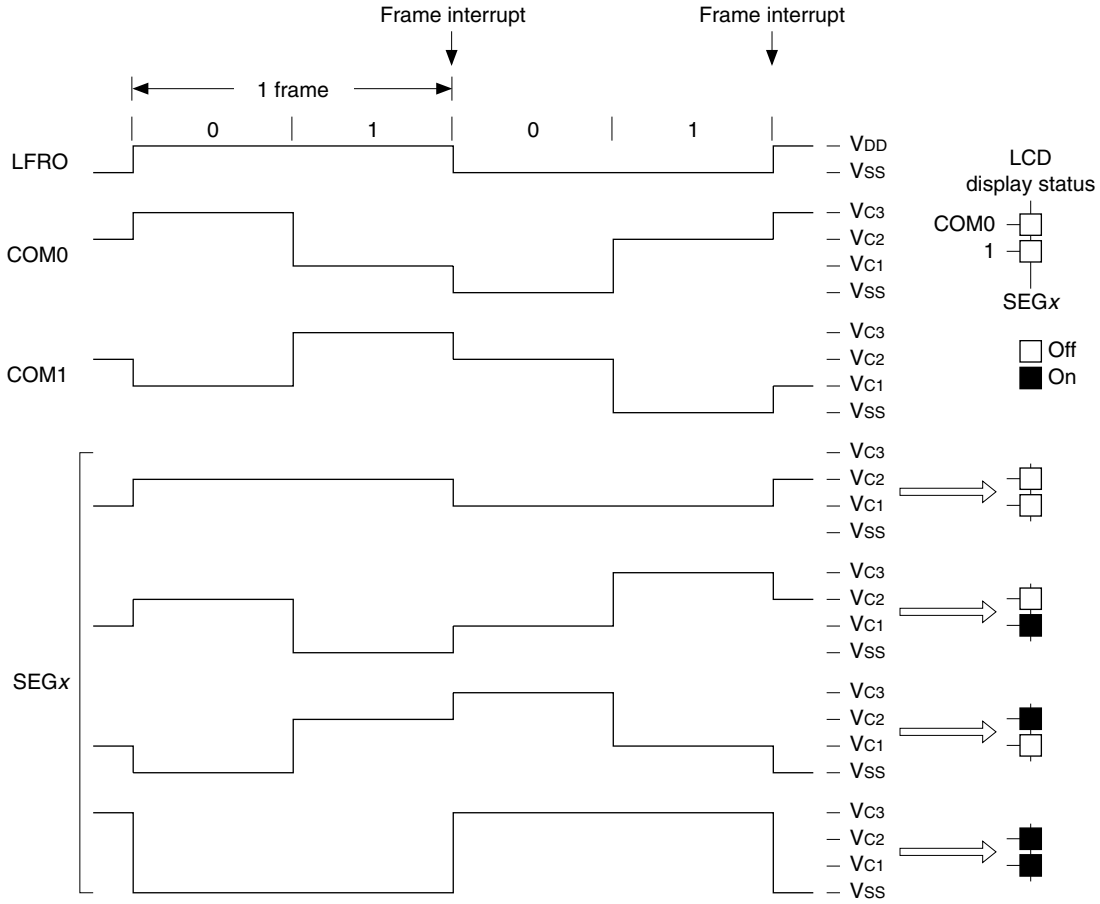


Figure 17.4.2.3 1/2 Duty Drive Waveform

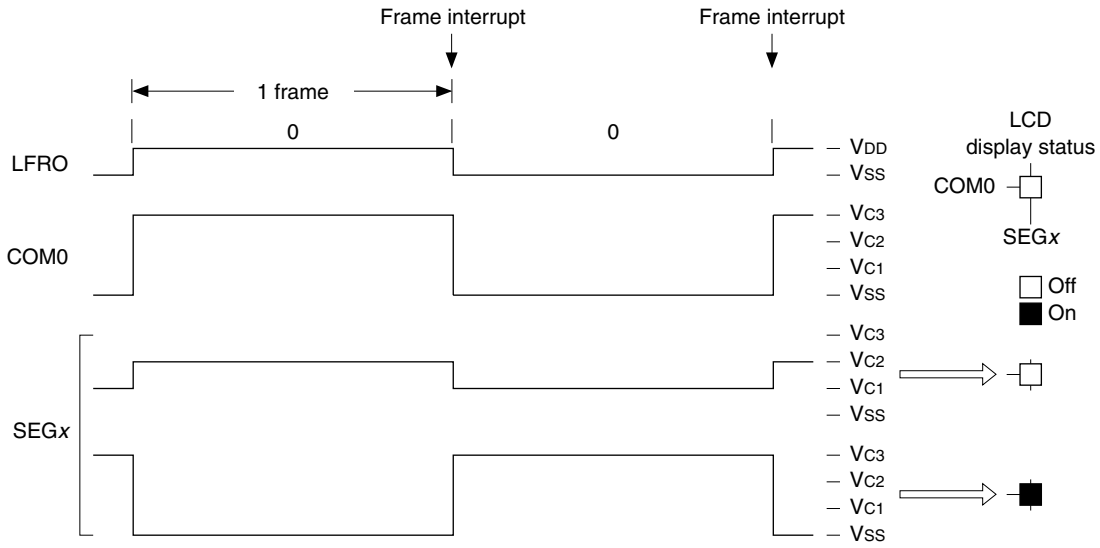


Figure 17.4.2.4 Static Drive Waveform

17.5 Display Memory

The S1C17651 includes a 20-byte display memory (address 0x53c0 to address 0x53d3).
Figures 17.5.1 to 17.5.4 show the correspondence between display memory and COM/SEG pins for each drive duty.
Writing 1 to a display memory bit corresponding to a segment on the LCD panel turns the segment on, while writing 0 turns the segment off. Since the display memory is a RAM allowing reading and writing, bits can be controlled individually using logic operation instructions (read-modify-write instructions).
Bits (D[3:0]) not assigned to the display area within the display memory can be used as general-purpose RAM that can be read and written to.

Bit	Address																			COM pin
	0x53c0	0x53c1	0x53c2	0x53c3	0x53c4	0x53c5	0x53c6	0x53c7	0x53c8	0x53c9	0x53ca	0x53cb	0x53cc	0x53cd	0x53ce	0x53cf	0x53d0	0x53d1	0x53d2	0x53d3	
D0																					COM0
D1									Display area												COM1
D2																					COM2
D3																					COM3
D4	Unavailable area (0 when being read)																				–
D5																					–
D6																					–
D7																					–
SEG pin	SEG0	SEG1	SEG2	SEG3	SEG4	SEG5	SEG6	SEG7	SEG8	SEG9	SEG10	SEG11	SEG12	SEG13	SEG14	SEG15	SEG16	SEG17	SEG18	SEG19	

Figure 17.5.1 Display Memory Map (1/4 duty)

Bit	Address																			COM pin
	0x53c0	0x53c1	0x53c2	0x53c3	0x53c4	0x53c5	0x53c6	0x53c7	0x53c8	0x53c9	0x53ca	0x53cb	0x53cc	0x53cd	0x53ce	0x53cf	0x53d0	0x53d1	0x53d2	0x53d3	
D0																					COM0
D1									Display area												COM1
D2																					COM2
D3	Unused area (general-purpose memory)																				–
D4	Unavailable area (0 when being read)																				–
D5																					–
D6																					–
D7																					–
SEG pin	SEG0	SEG1	SEG2	SEG3	SEG4	SEG5	SEG6	SEG7	SEG8	SEG9	SEG10	SEG11	SEG12	SEG13	SEG14	SEG15	SEG16	SEG17	SEG18	SEG19	

Figure 17.5.2 Display Memory Map (1/3 duty)

Bit	Address																			COM pin
	0x53c0	0x53c1	0x53c2	0x53c3	0x53c4	0x53c5	0x53c6	0x53c7	0x53c8	0x53c9	0x53ca	0x53cb	0x53cc	0x53cd	0x53ce	0x53cf	0x53d0	0x53d1	0x53d2	0x53d3	
D0									Display area												COM0
D1																					COM1
D2	Unused area (general-purpose memory)																				–
D3																					–
D4	Unavailable area (0 when being read)																				–
D5																					–
D6																					–
D7																					–
SEG pin	SEG0	SEG1	SEG2	SEG3	SEG4	SEG5	SEG6	SEG7	SEG8	SEG9	SEG10	SEG11	SEG12	SEG13	SEG14	SEG15	SEG16	SEG17	SEG18	SEG19	

Figure 17.5.3 Display Memory Map (1/2 duty)

Bit	Address																			COM pin
	0x53c0	0x53c1	0x53c2	0x53c3	0x53c4	0x53c5	0x53c6	0x53c7	0x53c8	0x53c9	0x53ca	0x53cb	0x53cc	0x53cd	0x53ce	0x53cf	0x53d0	0x53d1	0x53d2	0x53d3	
D0									Display area											COM0
D1	Unused area (general-purpose memory)																			–
D2																				–
D3																				–
D4	Unavailable area (0 when being read)																			–
D5																				–
D6																				–
D7																				–
SEG pin	SEG0	SEG1	SEG2	SEG3	SEG4	SEG5	SEG6	SEG7	SEG8	SEG9	SEG10	SEG11	SEG12	SEG13	SEG14	SEG15	SEG16	SEG17	SEG18	SEG19	

Figure 17.5.4 Display Memory Map (Static Drive)

17.6 Display Control

17.6.1 Display On/Off

The LCD display state is controlled using DSPC[1:0]/LCD_DCTL register.

Table 17.6.1.1 LCD Display Control

DSPC[1:0]	LCD display
0x3	All off (static)
0x2	All on (dynamic)
0x1	Normal display
0x0	Display off

(Default: 0x0)

For normal display, set DSPC[1:0] to 0x1. Note that the clock must be supplied. (See Section 17.3.)

If "Display off" is selected, the drive voltage supplied from the LCD power supply circuit stops, and the V_{C1} to V_{C3} pins are all set to V_{SS} level.

Since "All on" and "All off" directly control the driving waveform output by the LCD driver, display memory data is not altered. COM pins are set to dynamic drive for "All on" and to static drive for "All off." This function can be used to make the display flash on and off without altering the display memory.

DSPC[1:0] is reset to 0x0 (Display off) after an initial reset.
DSPC[1:0] is also reset to 0x0 when the `slp` instruction is executed and it reverts to the previous setting after SLEEP mode is canceled.

17.6.2 Inverted Display

The LCD display can be inverted (black/white inversion) using merely control bit manipulation, without changing the display memory. Setting DSPREV/LCD_DCTL register to 0 inverts the display; setting to 1 returns the display to normal status.
Note that the display will not be inverted if "All off" is selected using DSPC[1:0]. The display will be inverted when "All on" is selected.

17.7 LCD Interrupt

The LCD module includes a function for generating interrupts using the frame signal.

Frame interrupt

This cause of interrupt occurs every frame and sets the interrupt flag IFRMFLG/LCD_IFLG register in the LCD module to 1. See Figures 17.4.2.1 to 17.4.2.4 for interrupt timings.
To use this interrupt, set IFRMEN/LCD_IMSK register to 1. When IFRMEN is set to 0 (default), interrupt requests for this interrupt cause are not sent to the interrupt controller (ITC).
If IFRMFLG is set to 1 while IFRMEN is set to 1 (interrupt enabled), the LCD module outputs an interrupt request to the ITC. An interrupt is generated if the ITC and S1C17 Core interrupt conditions are satisfied.
For more information on interrupt processing, see the "Interrupt Controller (ITC)" chapter.

Notes: • To prevent interrupt recurrences, the LCD module interrupt flag IFRMFLG must be reset in the interrupt handler routine after an LCD interrupt has occurred.

• To prevent unwanted interrupts, IFRMFLG should be reset before enabling LCD interrupts with IFRMEN.

17.8 Control Register Details

Table 17.8.1 List of LCD Registers

Address	Register name		Function
0x5070	LCD_TCLK	LCD Clock Select Register	Selects the LCD clock.
0x50a0	LCD_DCTL	LCD Display Control Register	Controls the LCD display.
0x50a2	LCD_CCTL	LCD Clock Control Register	Controls the LCD drive duty.
0x50a3	LCD_VREG	LCD Voltage Regulator Control Register	Controls the LCD drive voltage regulator.
0x50a5	LCD_IMSK	LCD Interrupt Mask Register	Enables/disables interrupts.
0x50a6	LCD_IFLG	LCD Interrupt Flag Register	Indicates/resets interrupt occurrence status.

The LCD module registers are described in detail below.

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

LCD Timing Clock Select Register (LCD_TCLK)

Register name	Address	Bit	Name	Function	Setting			Init.	R/W	Remarks
LCD Timing Clock Select Register (LCD_TCLK)	**0x5070** (8 bits)	D7–6	**–**	reserved	–			–	–	0 when being read.
		D5–4	**LCDTCLKD [1:0]**	LCD clock division ratio select	LCDTCLKD [1:0]	Division ratio OSC3B/ OSC3A	Division ratio OSC1	0x0	R/W	
					0x3	1/8192	1/64			
					0x2	1/4096	1/64			
					0x1	1/2048	1/64			
					0x0	1/1024	1/64			
		D3–2	**LCDTCLK SRC[1:0]**	LCD clock source select	LCDTCLK SRC[1:0]	Clock source		0x0	R/W	
					0x3	reserved				
					0x2	OSC3A				
					0x1	OSC1				
					0x0	OSC3B				
		D1	**–**	reserved	–			–	–	0 when being read.
		D0	**LCDTCLKE**	LCD clock enable	1 Enable		0 Disable	0	R/W	

D[7:6] Reserved

D[5:4] LCDTCLKD[1:0]: LCD Clock Division Ratio Select Bits

Selects the division ratio when OSC3B or OSC3A is selected as the LCD clock source.

Table 17.8.2 Clock Division Ratio Selection

LCDTCLKD[1:0]	Division ratio
0x3	1/8192
0x2	1/4096
0x1	1/2048
0x0	1/1024

(Default: 0x0)

No division ratio needs to be selected if OSC1 is selected as the LCD clock source (fixed at 1/64).

D[3:2] LCDTCLKSRC[1:0]: LCD Clock Source Select Bits

Selects the LCD clock source.

Table 17.8.3 Clock Source Selection

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

(Default: 0x0)

D1 Reserved

D0 LCDTCLKE: LCD Clock Enable Bit

Enables or disables the LCD clock supply to the LCD driver.

1 (R/W): Enabled (on)

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

The LCDTCLKE default setting is 0, which stops the clock. Setting LCDTCLKE to 1 feeds the clock to the LCD driver. If no LCD display is required, stop the clock to reduce current consumption.

LCD Display Control Register (LCD_DCTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
LCD Display Control Register (LCD_DCTL)	0x50a0 (8 bits)	D7–5	–	reserved	–	–	–	0 when being read.
		D4	DSPREV	Reverse display control	1 Normal / 0 Reverse	1	R/W	
		D3–2	–	reserved	–	–	–	0 when being read.
		D1–0	DSPC[1:0]	LCD display control	DSPC[1:0]: 0x3 All off; 0x2 All on; 0x1 Normal display; 0x0 Display off	0x0	R/W	

D[7:5] Reserved

D4 DSPREV: Reverse Display Control Bit

Inverts (negative display) the LCD display.

1 (R/W): Normal display (default)

0 (R/W): Inverted display

Setting DSPREV to 0 inverts the LCD panel display; setting to 1 returns the display to normal status. This operation does not affect the contents of the display memory.

D[3:2] Reserved

D[1:0] DSPC[1:0]: LCD Display Control Bits

Controls the LCD display.

Table 17.8.4 LCD Display Control

DSPC[1:0]	LCD display
0x3	All off (static)
0x2	All on (dynamic)
0x1	Normal display
0x0	Display off

(Default: 0x0)

For normal display, set DSPC[1:0] to 0x1. Note that the clock must be supplied. (See Section 17.3.)

If "Display off" is selected, the drive voltage supplied from the LCD power supply circuit stops, and the V_{C1} to V_{C3} pins are all set to Vss level.

Since "All on" and "All off" directly control the driving waveform output by the LCD driver, display memory data is not altered. COM pins are set to dynamic drive for "All on" and to static drive for "All off." This function can be used to make the display flash on and off without altering the display memory.

DSPC[1:0] is reset to 0x0 (Display off) after an initial reset. DSPC[1:0] is also reset to 0x0 when the `slp` instruction is executed and it reverts to the previous setting after SLEEP mode is canceled.

LCD Clock Control Register (LCD_CCTL)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
LCD Clock Control Register (LCD_CCTL)	**0x50a2** (8 bits)	D7–6	**FRMCNT[1:0]**	Frame frequency control	FRMCNT[1:0]	Division ratio	0x1	R/W	Source clock: LCLK
					0x3	1/16			
					0x2	1/12			
					0x1	1/8			
					0x0	1/4			
		D5–3	–	reserved	–		–	–	0 when being read.
		D2–0	**LDUTY[2:0]**	LCD duty select	LDUTY[2:0]	Duty	0x3	R/W	
					0x7–0x4	reserved			
					0x3	1/4			
					0x2	1/3			
					0x1	1/2			
					0x0	Static			

D[7:6] FRMCNT[1:0]: Frame Frequency Control Bits

Sets the Frame frequency.

Table 17.8.5 Frame Frequency Settings (when the clock source is OSC1 = 32.768 kHz (LCLK = 512 Hz))

Drive duty (LDUTY[2:0] setting)	FRMCNT[1:0] setting (LCLK division ratio)			
	0x0	0x1	0x2	0x3
1/4 duty (0x3)	128 Hz (1/4)	64 Hz (1/8) *	42.67 Hz (1/12)	32 Hz (1/16)
1/3 duty (0x2)	85.33 Hz (1/6)	56.89 Hz (1/9)	42.67 Hz (1/12)	34.13 Hz (1/15)
1/2 duty (0x1)	128 Hz (1/4)	64 Hz (1/8)	42.67 Hz (1/12)	32 Hz (1/16)
Static (0x0)	128 Hz (1/4)	64 Hz (1/8)	42.67 Hz (1/12)	32 Hz (1/16)

* Default setting

Table 17.8.6 Frame Frequency Settings (when the clock source is OSC3B/OSC3A)

Drive duty (LDUTY[2:0] setting)	FRMCNT[1:0] setting			
	0x0	0x1	0x2	0x3
1/4 duty (0x3)	$\frac{f_{OSC3} \times LCDTCLKD}{4}$	$\frac{f_{OSC3} \times LCDTCLKD}{8}$ *	$\frac{f_{OSC3} \times LCDTCLKD}{12}$	$\frac{f_{OSC3} \times LCDTCLKD}{16}$
1/3 duty (0x2)	$\frac{f_{OSC3} \times LCDTCLKD}{6}$	$\frac{f_{OSC3} \times LCDTCLKD}{9}$	$\frac{f_{OSC3} \times LCDTCLKD}{12}$	$\frac{f_{OSC3} \times LCDTCLKD}{15}$
1/2 duty (0x1)	$\frac{f_{OSC3} \times LCDTCLKD}{4}$	$\frac{f_{OSC3} \times LCDTCLKD}{8}$	$\frac{f_{OSC3} \times LCDTCLKD}{12}$	$\frac{f_{OSC3} \times LCDTCLKD}{16}$
Static (0x0)	$\frac{f_{OSC3} \times LCDTCLKD}{4}$	$\frac{f_{OSC3} \times LCDTCLKD}{8}$	$\frac{f_{OSC3} \times LCDTCLKD}{12}$	$\frac{f_{OSC3} \times LCDTCLKD}{16}$

* Default setting
f_{OSC3}: OSC3B or OSC3A clock frequency
LCDTCLKD: OSC3B/OSC3A division ratio (1/1024 to 1/8192)

D[5:3] Reserved

D[2:0] LDUTY[2:0]: LCD Duty Select Bits

Selects the drive duty.

Table 17.8.7 Drive Duty Settings

LDUTY[2:0]	Duty	Valid COM pins	Valid SEG pins	Max. number of display segments
0x7–0x4	Reserved	–	–	–
0x3	1/4	COM0 to COM3	SEG0 to SEG19	80 segments
0x2	1/3	COM0 to COM2	SEG0 to SEG19	60 segments
0x1	1/2	COM0 to COM1	SEG0 to SEG19	40 segments
0x0	Static	COM0	SEG0 to SEG19	20 segments

(Default: 0x3)

LCD Voltage Regulator Control Register (LCD_VREG)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
LCD Voltage Regulator Control Register (LCD_VREG)	0x50a3 (8 bits)	D7–5	–	reserved	–	–	–	0 when being read.
		D4	LHVLD	V_C heavy load protection mode	1 On 0 Off	0	R/W	
		D3–1	–	reserved	–	–	–	0 when being read.
		D0	VCSEL	Reference voltage select	1 V_{C2} 0 V_{C1}	0	R/W	

For more information on the control bit, see "LCD Voltage Regulator Control Register (LCD_VREG)" in the "Power Supply" chapter.

LCD Interrupt Mask Register (LCD_IMSK)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
LCD Interrupt Mask Register (LCD_IMSK)	0x50a5 (8 bits)	D7–1	–	reserved	–	–	–	0 when being read.
		D0	IFRMEN	Frame signal interrupt enable	1 Enable 0 Disable	0	R/W	

D[7:1] Reserved

D0 IFRMEN: Frame Signal Interrupt Enable Bit

Enables or disables frame interrupts.

1 (R/W): Interrupt enabled

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

Setting IFRMEN to 1 enables LCD interrupt requests to the ITC. Setting to 0 disables interrupts.

LCD Interrupt Flag Register (LCD_IFLG)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
LCD Interrupt Flag Register (LCD_IFLG)	0x50a6 (8 bits)	D7–1	–	reserved	–	–	–	0 when being read.
		D0	IFRMFLG	Frame signal interrupt flag	1 Occurred 0 Not occurred	0	R/W	Reset by writing 1.

D[7:1] Reserved

D0 IFRMFLG: Frame Signal Interrupt Flag Bit

Indicates the frame interrupt cause occurrence status.

1 (R): Cause of interrupt has occurred

0 (R): No cause of interrupt has occurred (default)

1 (W): Flag is reset

0 (W): Ignored

IFRMFLG is set to 1 at the frame signal rising edge. IFRMFLG is reset to 0 by writing 1.

18 Sound Generator (SND)

18.1 SND Module Overview

The S1C17651 includes a sound generator (SND) for generating a buzzer signal.
The main features of the SND module are outlined below.

- Provides buzzer inverted and non-inverted output pins to directly drive a piezoelectric buzzer.
- Programmable buzzer signal frequency (eight frequencies) and volume level (eight levels)
- Duty ratio controlled digital envelope function (attenuation time is selectable from four types.)
- One-shot output function (output time is selectable from four types.)

Figure 18.1.1 shows the SND configuration.

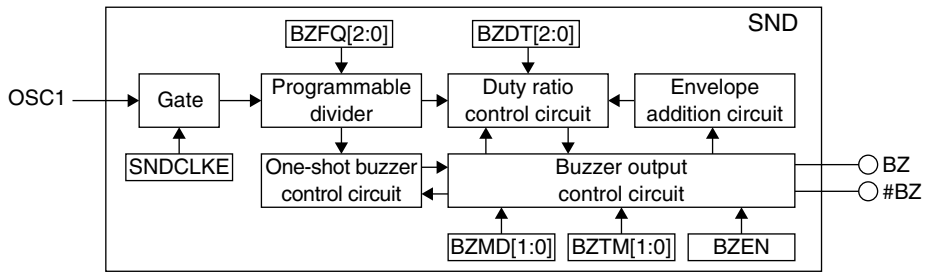


Figure 18.1.1 SND Module Configuration

18.2 SND Output Pins

Table 18.2.1 lists the SND pins.

Table 18.2.1 List of SND Pins

Pin name	I/O	Qty	Function
BZ	O	1	Buzzer non-inverted output pin Outputs the buzzer signal generated by the sound generator.
#BZ	O	1	Buzzer inverted output pin Outputs the inverted buzzer signal generated by the sound generator.

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

18.3 SND Operating Clock

The SND module uses the OSC1 clock (32.768 kHz Typ.) output from the CLG as its operating clock.
The OSC1 clock supply to the SND module is enabled with SNDCLKE/SND_CLK register. The SNDCLKE default setting is 0, which stops the clock. Setting SNDCLKE to 1 feeds the OSC1 clock to the SND module. Set SNDCLKE to 1 before performing buzzer output. If no buzzer output is required, stop the clock to reduce current consumption.
For more information on OSC1 oscillator control, see the "Clock Generator (CLG)" chapter.

Note: This chapter describes buzzer frequencies and one-shot output times assuming that the OSC1 clock frequency is 32.768 kHz. The frequencies and times vary depending on the OSC1 clock frequency.

18.4 Buzzer Frequency and Volume Settings

18.4.1 Buzzer Frequency

The SND module generates the buzzer signal by dividing the OSC1 clock (32.768 kHz). The buzzer frequency can be selected from among the eight types with different division ratios. BZFQ[2:0]/SND_BZFQ is used for this selection.

Table 18.4.1.1 Buzzer Frequency Selections

BZFQ[2:0]	Buzzer frequency (Hz)
0x7	1170.3
0x6	1365.3
0x5	1638.4
0x4	2048.0
0x3	2340.6
0x2	2730.7
0x1	3276.8
0x0	4096.0

(Default: 0x0)

18.4.2 Volume level

The buzzer volume level is controlled by changing the duty ratio of the buzzer signal.
The volume level can be selected from among eight types using BZDT[2:0]/SND_BZDT register.

Table 18.4.2.1 Volume Level Settings

Volume level	BZDT[2:0]	Duty ratio by buzzer frequency (Hz)			
		4096.0	3276.8	2730.7	2340.6
		2048.0	1638.4	1365.3	1170.3
Level 1 (Max.)	0x0	8/16	8/20	12/24	12/28
Level 2	0x1	7/16	7/20	11/24	11/28
Level 3	0x2	6/16	6/20	10/24	10/28
Level 4	0x3	5/16	5/20	9/24	9/28
Level 5	0x4	4/16	4/20	8/24	8/28
Level 6	0x5	3/16	3/20	7/24	7/28
Level 7	0x6	2/16	2/20	6/24	6/28
Level 8 (Min.)	0x7	1/16	1/20	5/24	5/28

(Default: 0x0)

Setting BZDT[2:0] to 0x0 turns the volume up to maximum level; setting it to 0x7 turns the volume down to minimum level.

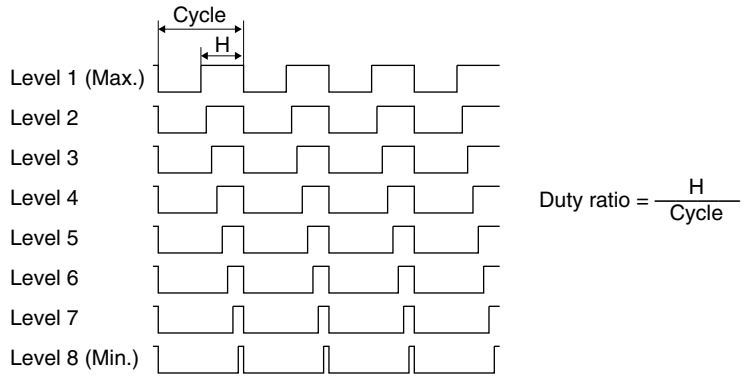


Figure 18.4.2.1 Buzzer Signal Waveforms by Different Duty Ratios

Note: BZDT[2:0] is ineffective in envelope mode, as the duty ratio is automatically controlled by the hardware.

18.5 Buzzer Mode and Output Control

18.5.1 Buzzer Mode Selection

The SND module supports three buzzer modes that allow different types of buzzer outputs.
BZMD[1:0]/SND_CTL register is used to select a buzzer mode.

Table 18.5.1.1 Buzzer Mode

BZMD[1:0]	Buzzer mode
0x3	Reserved
0x2	Envelope mode A software trigger starts buzzer output. The SND module automatically turns down the volume from Level 1 (maximum) and stops output when the volume reaches Level 8 (minimum).
0x1	One-shot mode This mode is provided for generating short buzzer sounds such as key operation sounds. The buzzer output starts by a software trigger and stops automatically after the specified time has elapsed.
0x0	Normal mode Buzzer output is turned on and off via software.

(Default: 0x0)

18.5.2 Output Control in Normal Mode

In normal mode, setting BZEN/SND_CTL register to 1 starts buzzer output and setting it to 0 stops the output. The buzzer frequency setting with BZFQ[2:0] and volume setting with BZDT[2:0] are both effective.

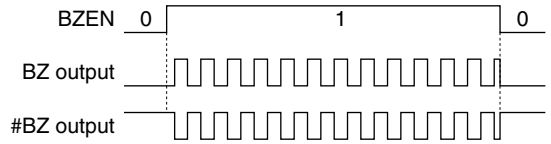


Figure 18.5.2.1 Buzzer Output in Normal Mode

Note: The buzzer signal is generated asynchronously to BZEN, so a hazard may occur when the signal is turned on or off by setting BZEN.

18.5.3 Output Control in One-shot Mode

The SND module has a one-shot output function for generating short buzzer sounds such as key operation sounds.

Output time selection

The one-shot buzzer output time can be selected from among four types shown below using BZTM[1:0]/SND_CTL register.

Table 18.5.3.1 One-shot Buzzer Output Time Selections

BZTM[1:0]	Output time
0x3	125 ms
0x2	62.5 ms
0x1	31.25 ms
0x0	15.63 ms

(Default: 0x0)

Output control

Writing 1 to BZEN/SND_CTL register starts one-shot buzzer output. When this trigger is issued, a buzzer signal is output from the buzzer output pin. When the set time has elapsed, the buzzer output stops.
BZEN functions as a status bit. It retains 1 while a one-shot buzzer signal is being output and reverts to 0 upon completion of the output.

Writing 0 to BZEN while a one-shot buzzer signal is being output stops the output immediately.
Writing 1 to BZEN again before a one-shot buzzer output is finished, a new one-shot output begins from that point.
The buzzer frequency setting with BZFQ[2:0] and volume setting with BZDT[2:0] are both effective in one-shot mode.
Figure 18.5.3.1 shows a timing chart in one-shot mode.

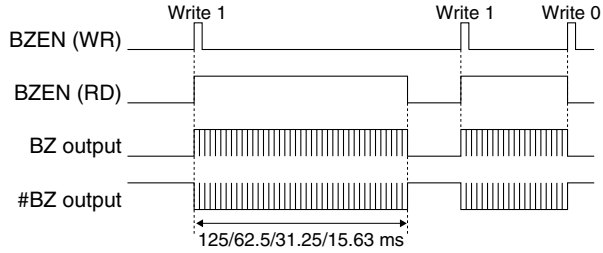


Figure 18.5.3.1 Buzzer Output in One-shot Mode

18.5.4 Output Control in Envelope Mode

In envelope mode, a digital envelope by duty control can be added to the buzzer signal. The SND module controls envelope by changing the duty ratio from Level 1 (maximum) to Level 8 (minimum) listed in Table 18.4.2.1.

Attenuation time selection

The envelope attenuation time (time to change the duty ratio) can be selected from among four types using BZTM[1:0]/SND_CTL register.

Table 18.5.4.1 Envelope Attenuation Time Selections

BZTM[1:0]	Attenuation time
0x3	125 ms
0x2	62.5 ms
0x1	31.25 ms
0x0	15.63 ms

(Default: 0x0)

Output control

Writing 1 to BZEN/SND_CTL register starts buzzer output in envelope mode. The duty ratio is set to Level 1 (maximum) at the beginning of the output and is stepped down every attenuation time selected. When attenuated down to Level 8 (minimum), the buzzer output stops.
BZEN functions as a status bit. It retains 1 while a buzzer signal is being output and reverts to 0 upon completion of the output.
Writing 0 to BZEN while a buzzer signal is being output stops the output immediately.
Writing 1 to BZEN again before a buzzer output is finished, the duty ratio returns to the maximum level and a new envelope output begins from that point.
Figure 18.5.4.1 shows a timing chart in envelope mode.

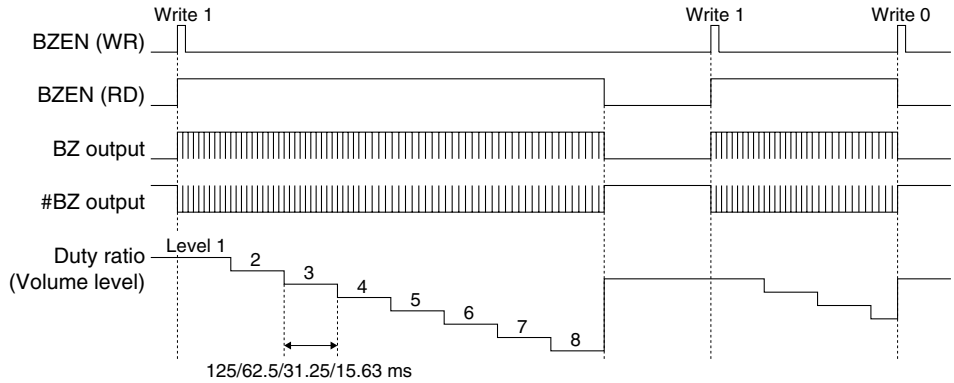


Figure 18.5.4.1 Buzzer Output in Envelope Mode

18.6 Control Register Details

Table 18.6.1 List of SND Registers

Address	Register name		Function
0x506e	SND_CLK	SND Clock Control Register	Controls the SND clock.
0x5180	SND_CTL	SND Control Register	Controls buzzer outputs.
0x5181	SND_BZFQ	Buzzer Frequency Control Register	Sets the buzzer frequency.
0x5182	SND_BZDT	Buzzer Duty Ratio Control Register	Sets the buzzer signal duty ratio.

The SND module registers are described in detail below.

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

SND Clock Control Register (SND_CLK)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SND Clock Control Register (SND_CLK)	**0x506e** (8 bits)	D7–1	–	reserved	–	–	–	0 when being read.
		D0	**SNDCLKE**	SND clock enable	1 Enable 0 Disable	0	R/W	

D[7:1] Reserved

D0 SNDCLKE: SND Clock Enable Bit

Enables or disables the OSC1 clock supply to the SND module.
1 (R/W): Enabled (on)
0 (R/W): Disabled (off) (default)

The SNDCLKE default setting is 0, which disables the clock supply. Setting SNDCLKE to 1 sends the OSC1 clock to the SND module to enable buzzer outputs. If no buzzer output is required, stop the clock to reduce current consumption.

SND Control Register (SND_CTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SND Control Register (SND_CTL)	**0x5180** (8 bits)	D7–6	–	reserved	–	–	–	0 when being read.
		D5–4	**BZTM[1:0]**	Buzzer envelope time/one-shot output time select	BZTM[1:0] Time 0x3 125 ms 0x2 62.5 ms 0x1 31.25 ms 0x0 15.63 ms	0x0	R/W	
		D3–2	**BZMD[1:0]**	Buzzer mode select	BZMD[1:0] Mode 0x3 reserved 0x2 Envelope 0x1 One-shot 0x0 Normal	0x0	R/W	
		D1	–	reserved	–	–	–	0 when being read.
		D0	**BZEN**	Buzzer output control	1 On/Trigger 0 Off	0	R/W	

D[7:6] Reserved

D[5:4] BZTM[1:0]: Buzzer Envelope Time/One-shot Output Time Select Bits

Selects an envelope attenuation time or a one-shot output time.

Table 18.6.2 Envelope Attenuation Time/One-shot Buzzer Output Time Selections

BZTM[1:0]	Attenuation time/One-shot output time
0x3	125 ms
0x2	62.5 ms
0x1	31.25 ms
0x0	15.63 ms

(Default: 0x0)

In envelope mode, an attenuation time (time to change the duty ratio) can be selected (see Figure 18.5.4.1).
In one-shot mode, a one-shot buzzer output time can be selected (see Figure 18.5.3.1).
BZTM[1:0] does not affect buzzer outputs in normal mode.

D[3:2] BZMD[1:0]: Buzzer Mode Select Bits

Selects a buzzer mode.

Table 18.6.3 Buzzer Mode

BZMD[1:0]	Buzzer mode
0x3	Reserved
0x2	Envelope mode A software trigger starts buzzer output. The SND module automatically turns down the volume from Level 1 (maximum) and stops output when the volume reaches Level 8 (minimum).
0x1	One-shot mode This mode is provided for generating short buzzer sounds such as key operation sounds. The buzzer output starts by a software trigger and stops automatically after the specified time has elapsed.
0x0	Normal mode Buzzer output is turned on and off via software.

(Default: 0x0)

D1 Reserved

D0 BZEN: Buzzer Output Control Bit

Controls buzzer output.
1 (R/W): On/Trigger
0 (R/W): Off (default)

Normal mode

Setting BZEN to 1 starts buzzer output and setting it to 0 stops the output.

One-shot mode

Writing 1 to BZEN starts one-shot buzzer output. When the time set with BZTM[1:0] has elapsed, the buzzer output stops. BZEN functions as a status bit. It retains 1 while a one-shot buzzer signal is being output and reverts to 0 upon completion of the output. Writing 0 to BZEN while a one-shot buzzer signal is being output stops the output immediately. Writing 1 to BZEN again before a one-shot buzzer output is finished, a new one-shot output begins from that point.

Envelope mode

Writing 1 to BZEN starts buzzer output in envelope mode. The duty ratio is set to Level 1 (maximum) at the beginning of the output and is stepped down every attenuation time selected. When attenuated down to Level 8 (minimum), the buzzer output stops. BZEN functions as a status bit. It retains 1 while a buzzer signal is being output and reverts to 0 upon completion of the output. Writing 0 to BZEN while a buzzer signal is being output stops the output immediately. Writing 1 to BZEN again before a buzzer output is finished, the duty ratio returns to the maximum level and a new envelope output begins from that point.

Buzzer Frequency Control Register (SND_BZFQ)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
Buzzer Frequency Control Register (SND_BZFQ)	0x5181 (8 bits)	D7–3	–	reserved	–		–	–	0 when being read.
		D2–0	BZFQ[2:0]	Buzzer frequency select	BZFQ[2:0]	Frequency	0x0	R/W	
					0x7	1170.3 Hz			
					0x6	1365.3 Hz			
					0x5	1638.4 Hz			
					0x4	2048.0 Hz			
					0x3	2340.6 Hz			
					0x2	2730.7 Hz			
					0x1	3276.8 Hz			
					0x0	4096.0 Hz			

D[7:3] Reserved

D[2:0] BZFQ[2:0]: Buzzer Frequency Select Bits

Selects a buzzer signal frequency.

Table 18.6.4 Buzzer Frequency Selections

BZFQ[2:0]	Buzzer frequency (Hz)
0x7	1170.3
0x6	1365.3
0x5	1638.4
0x4	2048.0
0x3	2340.6
0x2	2730.7
0x1	3276.8
0x0	4096.0

(Default: 0x0)

Buzzer Duty Ratio Control Register (SND_BZDT)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
Buzzer Duty Ratio Control Register (SND_BZDT)	0x5182 (8 bits)	D7–3	–	reserved	–		–	–	0 when being read.
		D2–0	BZDT[2:0]	Buzzer duty ratio select	BZDT[2:0]	Duty (volume)	0x0	R/W	
					0x7	Level 8 (Min.)			
					:	:			
					0x0	Level 1 (Max.)			

D[7:3] Reserved

D[2:0] BZDT[2:0]: Buzzer Duty Ratio Select Bits

Selects a duty ratio that determines the buzzer volume level.

Table 18.6.5 Volume Level Settings

Volume level	BZDT[2:0]	Duty ratio by buzzer frequency (Hz)			
		4096.0	3276.8	2730.7	2340.6
		2048.0	1638.4	1365.3	1170.3
Level 1 (Max.)	0x0	8/16	8/20	12/24	12/28
Level 2	0x1	7/16	7/20	11/24	11/28
Level 3	0x2	6/16	6/20	10/24	10/28
Level 4	0x3	5/16	5/20	9/24	9/28
Level 5	0x4	4/16	4/20	8/24	8/28
Level 6	0x5	3/16	3/20	7/24	7/28
Level 7	0x6	2/16	2/20	6/24	6/28
Level 8 (Min.)	0x7	1/16	1/20	5/24	5/28

(Default: 0x0)

Setting BZDT[2:0] to 0x0 turns the volume up to maximum level; setting it to 0x7 turns the volume down to minimum level.

Note: BZDT[2:0] is ineffective in envelope mode, as the duty ratio is automatically controlled by the hardware.

19 Supply Voltage Detection Circuit (SVD)

19.1 SVD Module Overview

The S1C17651 includes an SVD (supply voltage detection) circuit to monitor the power voltage supplied to the VDD pin. It can be used to check whether the power supply voltage drops below the detection level set with software or not. The following shows the features of the SVD module:

- Power supply voltage to be detected: VDD
- Detection voltage levels: 13 levels (2.0 V to 3.2 V)

Figure 19.1.1 shows the SVD configuration.

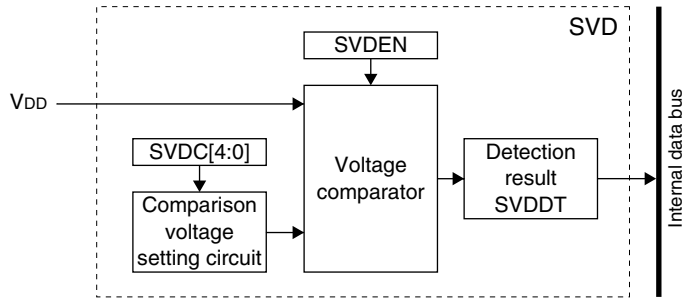


Figure 19.1.1 SVD Configuration

19.2 Comparison Voltage Setting

The SVD circuit compares the power supply voltage (VDD) against the comparison voltage set by software and outputs results indicating whether the power supply voltage exceeds this comparison voltage. The comparison voltage can be selected from among the 17 levels listed in Table 19.2.1 with the SVDC[4:0]/SVD_CMP register.

Table 19.2.1 Comparison Voltage Settings

SVDC[4:0]	Comparison Voltage	SVDC[4:0]	Comparison Voltage
0x1f	Reserved	0xf	2.10 V
0x1e		0xe	2.00 V
0x1d		0xd	Reserved
0x1c		0xc	
0x1b		0xb	
0x1a	3.20 V	0xa	
0x19	3.10 V	0x9	
0x18	3.00 V	0x8	
0x17	2.90 V	0x7	
0x16	2.80 V	0x6	
0x15	2.70 V	0x5	
0x14	2.60 V	0x4	
0x13	2.50 V	0x3	
0x12	2.40 V	0x2	
0x11	2.30 V	0x1	
0x10	2.20 V	0x0	

(Default: 0x0)

Note: The comparison voltage is effective only when it is set within the operating voltage range. If the comparison voltage set is out of the operating voltage range, no correct detection results will be obtained.

19.3 SVD Control

Power supply voltage detection using the SVD circuit is initiated by writing 1 to SVDEN/SVD_EN register. After that, the supply voltage detection results can be read out from SVDDT/SVD_RSLT register. By writing 0 to SVDEN, the SVD circuit sets the detection result at that point to SVDDT and stops detection.
The detection results and SVDDT readings are as follows.
• When power supply voltage (V_{DD}) ≥ comparison voltage: SVDDT = 0
• When power supply voltage (V_{DD}) < comparison voltage: SVDDT = 1

Notes: • An SVD circuit-enable response time is required to obtain stable detection results after SVDEN is altered from 0 to 1. Also when SVDC[4:0] is altered, an SVD circuit response time is required to obtain stable detection results. Wait until the response time has elapsed before reading SVDDT. Also when reading the detection results after stopping the SVD circuit, SVDEN should be set to 0 after the response time has elapsed. For these response times, see "Electrical Characteristics."

• Operating the SVD circuit increases current consumption. If power supply voltage detection is not required, stop SVD operations by setting SVDEN to 0.

19.4 Control Register Details

Table 19.4.1 List of SVD Registers

Address	Register name		Function
0x5100	SVD_EN	SVD Enable Register	Enables/disables the SVD operation.
0x5101	SVD_CMP	SVD Comparison Voltage Register	Sets the comparison voltage.
0x5102	SVD_RSLT	SVD Detection Result Register	Voltage detection results

The SVD module registers are described in detail below.

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

SVD Enable Register (SVD_EN)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SVD Enable Register (SVD_EN)	**0x5100** (8 bits)	D7–1	–	reserved	–	–	–	0 when being read.
		D0	**SVDEN**	SVD enable	1 Enable 0 Disable	0	R/W	

D[7:1] Reserved

D0 SVDEN: SVD Enable Bit

Enables or disables SVD operations.
1 (R/W): Enabled
0 (R/W): Disabled (default)

Setting SVDEN to 1 initiates power supply voltage detection; setting to 0 stops detection after loading the detection results to SVDDT/SVD_RSLT register.

Notes: • An SVD circuit-enable response time is required to obtain stable detection results after SVDEN is altered from 0 to 1. Also when SVDC[4:0] is altered, an SVD circuit response time is required to obtain stable detection results. Wait until the response time has elapsed before reading SVDDT. Also when reading the detection results after stopping the SVD circuit, SVDEN should be set to 0 after the response time has elapsed. For these response times, see "Electrical Characteristics."

• Operating the SVD circuit increases current consumption. If power supply voltage detection is not required, stop SVD operations by setting SVDEN to 0.

SVD Comparison Voltage Register (SVD_CMP)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
SVD Comparison Voltage Register (SVD_CMP)	0x5101 (8 bits)	D7–5	–	reserved	–		–	–	0 when being read.
		D4–0	SVDC[4:0]	SVD comparison voltage select	SVDC[4:0]	Voltage	0x0	R/W	
					0x1f–0x1b	reserved			
					0x1a	3.20 V			
					0x19	3.10 V			
					0x18	3.00 V			
					0x17	2.90 V			
					0x16	2.80 V			
					0x15	2.70 V			
					0x14	2.60 V			
					0x13	2.50 V			
					0x12	2.40 V			
					0x11	2.30 V			
					0x10	2.20 V			
					0xf	2.10 V			
					0xe	2.00 V			
					0xd–0x0	reserved			

D[7:5] Reserved

D[4:0] SVDC[4:0]: SVD Comparison Voltage Select Bits

Selects one of 13 comparison voltages for detecting voltage drops.

Table 19.4.2 Comparison Voltage Settings

SVDC[4:0]	Comparison Voltage	SVDC[4:0]	Comparison Voltage
0x1f	Reserved	0xf	2.10 V
0x1e		0xe	2.00 V
0x1d		0xd	Reserved
0x1c		0xc	
0x1b		0xb	
0x1a	3.20 V	0xa	
0x19	3.10 V	0x9	
0x18	3.00 V	0x8	
0x17	2.90 V	0x7	
0x16	2.80 V	0x6	
0x15	2.70 V	0x5	
0x14	2.60 V	0x4	
0x13	2.50 V	0x3	
0x12	2.40 V	0x2	
0x11	2.30 V	0x1	
0x10	2.20 V	0x0	

(Default: 0x0)

The SVD circuit compares the power supply voltage (V_{DD}) against the comparison voltage set by SVDC[4:0], and outputs results indicating whether the power supply voltage exceeds this comparison voltage.

Note: The comparison voltage is effective only when it is set within the operating voltage range. If the comparison voltage set is out of the operating voltage range, no correct detection results will be obtained.

SVD Detection Result Register (SVD_RSLT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SVD Detection Result Register (SVD_RSLT)	0x5102 (8 bits)	D7–1	–	reserved	–	–	–	0 when being read.
		D0	SVDDT	SVD detection result	1 Low 0 Normal	×	R	

D[7:1] Reserved

D0 SVDDT: SVD Detection Result Bit

Indicates the power supply voltage detection results.

1 (R): Power supply voltage (V_{DD}) < comparison voltage

0 (R): Power supply voltage (V_{DD}) ≥ comparison voltage

The SVD circuit compares the power supply voltage (V_{DD}) against the voltage set in SVDC[4:0]/SVD_CMP register while SVDEN/SVD_EN register = 1. The current power supply voltage status can be monitored by reading SVDDT. Also the detection result is set to SVDDT by writing 0 to SVDEN, so the power supply voltage status can be checked by reading SVDDT after that.

20 On-chip Debugger (DBG)

20.1 Resource Requirements and Debugging Tools

Debugging work area

Debugging requires a 64-byte debugging work area. For more information on the work area location, see the "Memory Map, Bus Control" chapter.
The start address for this debugging work area can be read from the DBRAM register (0xffff90).

Debugging tools

Debugging involves connecting ICDmini (S5U1C17001H) to the S1C17651 debug pins and inputting the debug instruction from the debugger on the personal computer.
The following tools are required:
• S1C17 Family In-Circuit Debugger ICDmini (S5U1C17001H)
• S1C17 Family C compiler package (e.g., S5U1C17001C)

Debug pins

The following debug pins are used to connect ICDmini (S5U1C17001H).

Table 20.1.1 List of Debug Pins

Pin name	I/O	Qty	Function
DCLK	O	1	On-chip debugger clock output pin Outputs a clock to the ICDmini (S5U1C17001H).
DSIO	I/O	1	On-chip debugger data input/output pin Used to input/output debugging data and input the break signal.
DST2	O	1	On-chip debugger status signal output pin Outputs the processor status during debugging.

The on-chip debugger input/output pins (DCLK, DST2, DSIO) are shared with I/O ports and are initially set as the debug pins. If the debugging function is not used, these pins can be switched using the port function select bits to enable use as general-purpose I/O port pins.
For detailed information on pin function switching, see the "I/O Ports (P)" chapter.

20.2 Debug Break Operation Status

The S1C17 Core enters debug mode when the brk instruction is executed or a debug interrupt is generated by a break signal (Low) input to the DSIO pin. This state persists until the retd instruction is executed. During this time, hardware interrupts and NMIs are disabled.
The default setting halts peripheral circuit operations. This setting can be modified even when debugging is underway.
The peripheral circuits that operate with PCLK will continue running in debug mode by setting DBRUN1/MISC_DMODE1 register to 1. Setting DBRUN1 to 0 (default) will stop these peripheral circuits in debug mode.
The peripheral circuits that operate with a clock other than PCLK will continue running in debug mode by setting DBRUN2/MISC_DMODE2 register to 1. Setting DBRUN2 to 0 (default) will stop these peripheral circuits in debug mode.
Some peripheral circuits, such as SPI and T16A2, that run with an external input clock will not stop operating even if the S1C17 Core enters debug mode.
The LCD driver and RTC continue the operating status at occurrence of the debug interrupt.

20.3 Additional Debugging Function

The S1C17651 expands the following on-chip debugging functions of the S1C17 Core.

Branching destination in debug mode

When a debug interrupt is generated, the S1C17 Core enters debug mode and branches to the debug processing routine. In this process, the S1C17 Core is designed to branch to address 0xfffc00. In addition to this branching destination, the S1C17651 also allows designation of address 0x0 (beginning address of the internal RAM) as the branching destination when debug mode is activated. The branching destination address is selected using DBADR/MISC_IRAMSZ register. When the DBADR is set to 0 (default), the branching destination is set to 0xfffc00. When it is set to 1, the branching destination is set to 0x0.

Adding instruction breaks

The S1C17 Core supports two instruction breaks (hardware PC breaks). The S1C17651 increased this number to five, adding the control bits and registers given below.

- IBE2/DCR register: Enables instruction breaks #2.
- IBE3/DCR register: Enables instruction breaks #3.
- IBE4/DCR register: Enables instruction breaks #4.
- IBAR2[23:0]/IBAR2 register: Set instruction break address #2.
- IBAR3[23:0]/IBAR3 register: Set instruction break address #3.
- IBAR4[23:0]/IBAR4 register: Set instruction break address #4.

Note that the debugger included in the S5U1C17001C (Ver. 1.2.1) or later is required to use five hardware PC breaks.

20.4 Control Register Details

Table 20.4.1 List of Debug Registers

Address	Register name		Function
0x4020	MISC_DMODE1	Debug Mode Control Register 1	Enables peripheral operations in debug mode (PCLK).
0x5322	MISC_DMODE2	Debug Mode Control Register 2	Enables peripheral operations in debug mode (except PCLK).
0x5326	MISC_IRAMSZ	IRAM Size Select Register	Selects the IRAM size.
0xffff90	DBRAM	Debug RAM Base Register	Indicates the debug RAM base address.
0xffffa0	DCR	Debug Control Register	Controls debugging.
0xffffb8	IBAR2	Instruction Break Address Register 2	Sets Instruction break address #2.
0xffffbc	IBAR3	Instruction Break Address Register 3	Sets Instruction break address #3.
0xffffd0	IBAR4	Instruction Break Address Register 4	Sets Instruction break address #4.

The debug registers are described in detail below.

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

• For debug registers not described here, refer to the S1C17 Core Manual.

Debug Mode Control Register 1 (MISC_DMODE1)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Debug Mode Control Register 1 (MISC_DMODE1)	**0x4020** (8 bits)	D7–2	–	reserved	–	–	–	0 when being read.
		D1	**DBRUN1**	Run/stop select in debug mode	1 Run 0 Stop	0	R/W	
		D0	–	reserved	–	–	–	0 when being read.

D[7:2] Reserved

D1 DBRUN1: Run/Stop Select Bit in Debug Mode

Selects the operating status of the peripheral circuits that operate with PCLK in debug mode.

1 (R/W): Run

0 (R/W): Stop (default)

Setting DBRUN1 to 1 enables the peripheral circuits that operate with PCLK to run even in debug mode. Setting it to 0 will stop them when the S1C17 Core enters debug mode. Set DBRUN1 to 1 to maintain running status for these peripheral circuits in debug mode.

D0 Reserved

Debug Mode Control Register 2 (MISC_DMODE2)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Debug Mode Control Register 2 (MISC_DMODE2)	0x5322 (16 bits)	D15–1	–	reserved	–	–	–	0 when being read.
		D0	DBRUN2	Run/stop select in debug mode (except PCLK peripheral circuits)	1 Run 0 Stop	0	R/W	

D[15:1] Reserved

D0 DBRUN2: Run/Stop Select Bit in Debug Mode (except PCLK peripheral circuits)

Selects the operating status of the peripheral circuits that operate with a clock other than PCLK in debug mode.

1 (R/W): Run

0 (R/W): Stop (default)

Setting DBRUN2 to 1 enables the peripheral circuits that operate with a clock other than PCLK to run even in debug mode. Setting it to 0 will stop them when the S1C17 Core enters debug mode. Set DBRUN2 to 1 to maintain running status for these peripheral circuits in debug mode.

Some peripheral circuits, such as SPI and T16A2, that run with an external input clock will not stop operating even if the S1C17 Core enters debug mode.

The LCD driver and RTC continue the operating status at occurrence of the debug interrupt.

IRAM Size Select Register (MISC_IRAMSZ)

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

D[15:9] Reserved

D8 DBADR: Debug Base Address Select Bit

Selects the branching destination address when a debug interrupt occurs.

1(R/W): 0x0

0(R/W): 0xfffc00 (default)

D7 Reserved

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

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

D3 Reserved

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

Selects the size of the internal RAM to be used.

Table 20.4.2 Internal RAM Size Selection

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

(Default: 0x3)

Note: The MISC_IRAMSZ register is write-protected. To alter this register settings, you must override this write-protection by writing 0x96 to the MISC_PROT register. Normally, the MISC_PROT register should be set to a value other than 0x96, except when altering the MISC_IRAMSZ register. Unnecessary rewriting of the MISC_IRAMSZ register may result in system malfunctions.

Debug RAM Base Register (DBRAM)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Debug RAM Base Register (DBRAM)	0xffff90 (32 bits)	D31–24	–	Unused (fixed at 0)	0x0	0x0	R	
		D23–0	DBRAM[23:0]	Debug RAM base address	0x7c0	0x7c0	R	

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

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

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

Debug Control Register (DCR)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Debug Control Register (DCR)	0xffffa0 (8 bits)	D7	IBE4	Instruction break #4 enable	1	Enable	0	Disable	0	R/W	
		D6	IBE3	Instruction break #3 enable	1	Enable	0	Disable	0	R/W	
		D5	IBE2	Instruction break #2 enable	1	Enable	0	Disable	0	R/W	
		D4	DR	Debug request flag	1	Occurred	0	Not occurred	0	R/W	Reset by writing 1.
		D3	IBE1	Instruction break #1 enable	1	Enable	0	Disable	0	R/W	
		D2	IBE0	Instruction break #0 enable	1	Enable	0	Disable	0	R/W	
		D1	SE	Single step enable	1	Enable	0	Disable	0	R/W	
		D0	DM	Debug mode	1	Debug mode	0	User mode	0	R	

D7 IBE4: Instruction Break #4 Enable Bit

Enables or disables instruction break #4.
1 (R/W): Enabled
0 (R/W): Disabled (default)

If this bit is set to 1, the instruction fetch address and the value set in the IBAR4 register are compared. If they match, an instruction break is generated. If this bit is set to 0, no comparison is performed.

D6 IBE3: Instruction Break #3 Enable Bit

Enables or disables instruction break #3.
1 (R/W): Enabled
0 (R/W): Disabled (default)

If this bit is set to 1, the instruction fetch address and the value set in the IBAR3 register are compared. If they match, an instruction break is generated. If this bit is set to 0, no comparison is performed.

D5 IBE2: Instruction Break #2 Enable Bit

Enables or disables instruction break #2.
1 (R/W): Enabled
0 (R/W): Disabled (default)

If this bit is set to 1, the instruction fetch address and the value set in the IBAR2 register are compared. If they match, an instruction break is generated. If this bit is set to 0, no comparison is performed.

D4 DR: Debug Request Flag Bit

Indicates the presence or absence of an external debug request.

1 (R): Request generated
0 (R): Request not generated (default)
1 (W): Flag is reset
0 (W): Ignored

This flag is cleared (reset to 0) when 1 is written. It must be cleared before the debug processing routine is terminated by the retd instruction.

D3 IBE1: Instruction Break #1 Enable Bit

Enables or disables instruction break #1.

1 (R/W): Enabled
0 (R/W): Disabled (default)

If this bit is set to 1, the instruction fetch address and the value set in the IBAR1 register are compared. If they match, an instruction break is generated. If this bit is set to 0, no comparison is performed.

D2 IBE0: Instruction Break #0 Enable Bit

Enables or disables instruction break #0.

1 (R/W): Enabled
0 (R/W): Disabled (default)

If this bit is set to 1, the instruction fetch address and the value set in the IBAR0 register are compared. If they match, an instruction break is generated. If this bit is set to 0, no comparison is performed.

D1 SE: Single Step Enable Bit

Enables or disables single-step operations.

1 (R/W): Enabled
0 (R/W): Disabled (default)

D0 DM: Debug Mode Bit

Indicates the processor operating mode (debug mode or user mode).

1 (R): Debug mode
0 (R): User mode (default)

Instruction Break Address Register 2 (IBAR2)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Instruction Break Address Register 2 (IBAR2)	0xffffb8 (32 bits)	D31–24	–	reserved	–	–	–	0 when being read.
		D23–0	IBAR2[23:0]	Instruction break address #2 IBAR223 = MSB IBAR20 = LSB	0x0 to 0xffffff	0x0	R/W	

D[31:24] Reserved

D[23:0] IBAR2[23:0]: Instruction Break Address #2 Bits

Sets instruction break address #2. (default: 0x000000)

Instruction Break Address Register 3 (IBAR3)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Instruction Break Address Register 3 (IBAR3)	0xffffbc (32 bits)	D31–24	–	reserved	–	–	–	0 when being read.
		D23–0	IBAR3[23:0]	Instruction break address #3 IBAR323 = MSB IBAR30 = LSB	0x0 to 0xffffff	0x0	R/W	

D[31:24] Reserved

D[23:0] IBAR3[23:0]: Instruction Break Address #3 Bits

Sets instruction break address #3. (default: 0x000000)

Instruction Break Address Register 4 (IBAR4)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Instruction Break Address Register 4 (IBAR4)	**0xffffd0** (32 bits)	D31–24	**–**	reserved	–	–	–	0 when being read.
		D23–0	**IBAR4[23:0]**	Instruction break address #4 IBAR423 = MSB IBAR40 = LSB	0x0 to 0xffffff	0x0	R/W	

D[31:24] Reserved

D[23:0] IBAR4[23:0]: Instruction Break Address #4 Bits

Sets instruction break address #4. (default: 0x000000)

21 Multiplier/Divider (COPRO)

21.1 Overview

The S1C17651 has an embedded coprocessor that provides multiplier/divider functions.
The following shows the features of the multiplier/divider:

- Multiplication: Supports signed/unsigned multiplications.
  (16 bits × 16 bits = 32 bits)
  Can be executed in 1 cycle.
- Multiplication and accumulation (MAC): Supports signed MAC operations with overflow detection function
  (16 bits × 16 bits + 32 bits = 32 bits)
  Can be executed in 1 cycle.
- Division: Supports signed/unsigned divisions.
  (16 bits ÷ 16 bits = 16 bits with 16-bit residue)
  Can be executed in 17 to 20 cycles.

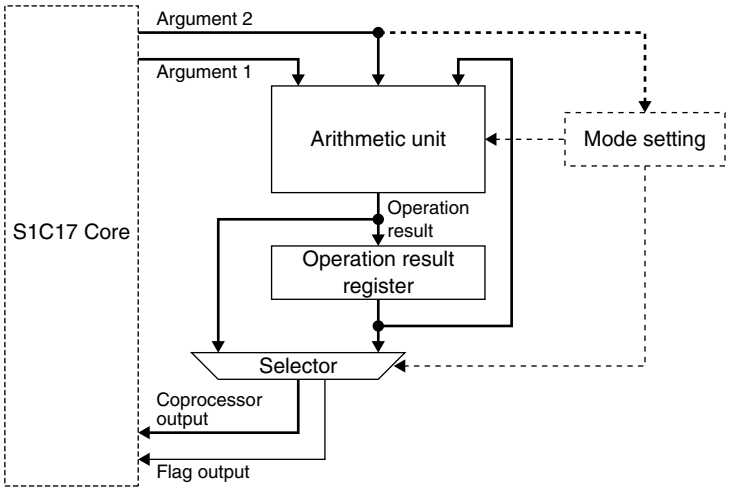


Figure 21.1.1 Multiplier/Divider Block Diagram

21.2 Operation Mode and Output Mode

The Multiplier/divider operates according to the operation mode specified by the application program. As listed in Table 21.2.1, the multiplier/divider supports nine operations.
The multiplication, division and MAC results are 32-bit data, therefore, the S1C17 Core cannot read them in one access cycle. The output mode is provided to specify the high-order 16 bits or low-order 16 bits of the operation results to be read from the multiplier/divider.
The operation and output modes can be specified with a 7-bit data by writing it to the mode setting register in the multiplier/divider. Use a "`ld.cw`" instruction for this writing.

`ld.cw %rd,%rs` %rs[6:0] is written to the mode setting register. (%rd: not used)
`ld.cw %rd,`*`imm7`* *imm7*[6:0] is written to the mode setting register. (%rd: not used)

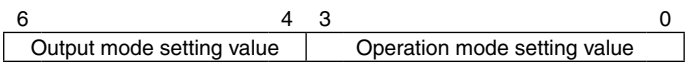


Figure 21.2.1 Mode Setting Register

Table 21.2.1 Mode Settings

Setting value (D[6:4])	Output mode	Setting value (D[3:0])	Operation mode
0x0	**16 low-order bits output mode** The low-order 16-bits of operation results can be read as the coprocessor output.	0x0	**Initialize mode 0** Clears the operation result register to 0x0.
0x1	**16 high-order bits output mode** The high-order 16-bits of operation results can be read as the coprocessor output.	0x1	**Initialize mode 1** Loads the 16-bit augend into the low-order 16 bits of the operation result register.
0x2–0x7	Reserved	0x2	**Initialize mode 2** Loads the 32-bit augend into the operation result register.
		0x3	**Operation result read mode** Outputs the data in the operation result register without computation.
		0x4	**Unsigned multiplication mode** Performs unsigned multiplication.
		0x5	**Signed multiplication mode** Performs signed multiplication.
		0x6	Reserved
		0x7	**Signed MAC mode** Performs signed MAC operation.
		0x8	**Unsigned division mode** Performs unsigned division.
		0x9	**Signed division mode** Performs signed division.
		0xa–0xf	Reserved

21.3 Multiplication

The multiplication function performs "A (32 bits) = B (16 bits) × C (16 bits)."
To perform a multiplication, set the operation mode to 0x4 (unsigned multiplication) or 0x5 (signed multiplication). Then send the 16-bit multiplicand (B) and 16-bit multiplier (C) to the multiplier/divider using a "`ld.ca`" instruction. The one-half (16 bits according to the output mode) result (A[15:0] or A[31:16]) and the flag status will be returned to the CPU registers. Another one-half should be read by setting the multiplier/divider into operation result read mode.

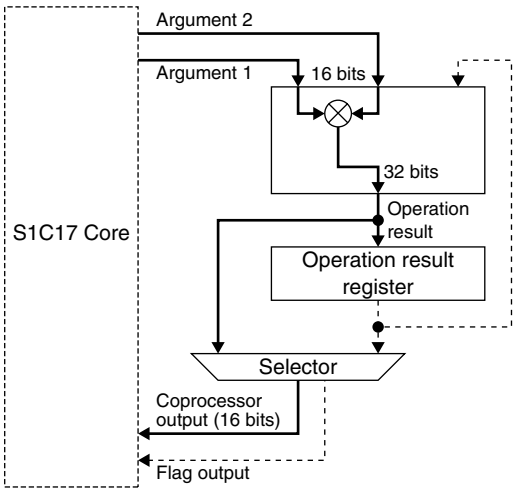


Figure 21.3.1 Data Path in Multiplication Mode

Table 21.3.1 Operation in Multiplication Mode

Mode setting value	Instruction	Operations	Flags	Remarks
0x04 or 0x05	`ld.ca %rd,%rs`	res[31:0] ← %rd × %rs %rd ← res[15:0]	psr (CVZN) ← 0b0000	The operation result register keeps the operation result until it is rewritten by other operation.
	(`ext` *imm9*) `ld.ca %rd,`*imm7*	res[31:0] ← %rd × *imm7/16* %rd ← res[15:0]		
0x14 or 0x15	`ld.ca %rd,%rs`	res[31:0] ← %rd × %rs %rd ← res[31:16]		
	(`ext` *imm9*) `ld.ca %rd,`*imm7*	res[31:0] ← %rd × *imm7/16* %rd ← res[31:16]		

res: operation result register

Example:

```
ld.cw %r0,0x4   ; Sets the modes (unsigned multiplication mode and 16 low-order bits output mode).
ld.ca %r0,%r1   ; Performs "res = %r0 × %r1" and loads the 16 low-order bits of the result to %r0.
ld.cw %r0,0x13  ; Sets the modes (operation result read mode and 16 high-order bits output mode).
ld.ca %r1,%r0   ; Loads the 16 high-order bits of the result to %r1.
```

21.4 Division

The division function performs "B (16 bits) ÷ C (16 bits) = A (16 bits), residue D (16 bits)."
To perform a division, set the operation mode to 0x8 (unsigned division) or 0x9 (signed division). Then send the 16-bit dividend (B) and 16-bit divisor (C) to the multiplier/divider using a "`ld.ca`" instruction. The quotient and the residue will be stored in the low-order 16 bits and the high-order 16 bits of the operation result register, respectively. The 16-bit quotient or residue according to the output mode specification and the flag status will be returned to the CPU registers. Another 16-bit result should be read by setting the multiplier/divider into operation result read mode.

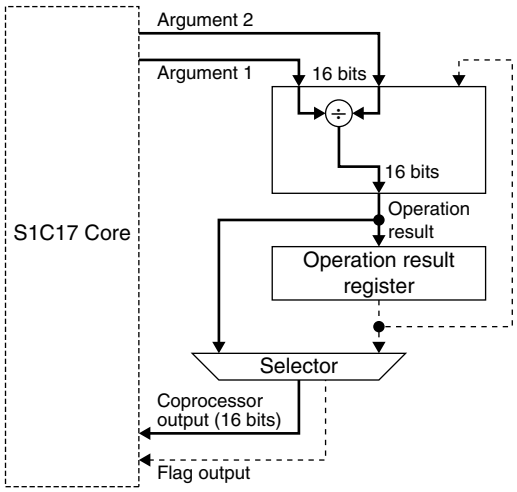


Figure 21.4.1 Data Path in Division Mode

Table 21.4.1 Operation in Division Mode

Mode setting value	Instruction	Operations	Flags	Remarks
0x08 or 0x09	`ld.ca %rd,%rs`	res[31:0] ← %rd ÷ %rs %rd ← res[15:0] (quotient)	psr (CVZN) ← 0b0000	The operation result register keeps the operation result until it is rewritten by other operation.
	(`ext` *imm9*) `ld.ca %rd,`*imm7*	res[31:0] ← %rd ÷ *imm7/16* %rd ← res[15:0] (quotient)		
0x018 or 0x19	`ld.ca %rd,%rs`	res[31:0] ← %rd ÷ %rs %rd ← res[31:16] (residue)		
	(`ext` *imm9*) `ld.ca %rd,`*imm7*	res[31:0] ← %rd ÷ *imm7/16* %rd ← res[31:16] (residue)		

res: operation result register

Example:

```
ld.cw %r0,0x8  ; Sets the modes (unsigned division mode and 16 low-order bits output mode).
ld.ca %r0,%r1  ; Performs "res = %r0 ÷ %r1" and loads the 16 low-order bits of the result (quotient) to %r0.
ld.cw %r0,0x13 ; Sets the modes (operation result read mode and 16 high-order bits output mode).
ld.ca %r1,%r0  ; Loads the 16 high-order bits of the result (residue) to %r1.
```

21.5 MAC

The MAC (multiplication and accumulation) function performs "A (32 bits) = B (16 bits) × C (16 bits) + A (32 bits)."
Before performing a MAC operation, the initial value (A) must be set to the operation result register.
To clear the operation result register (A = 0), just set the operation mode to 0x0. It is not necessary to send 0x0 to the multiplier/divider with another instruction.
To load a 16-bit value or a 32-bit value to the operation result register, set the operation mode to 0x1 (16 bits) or 0x2 (32 bits), respectively. Then send the initial value to the multiplier/divider using a "`ld.cf`" instruction.

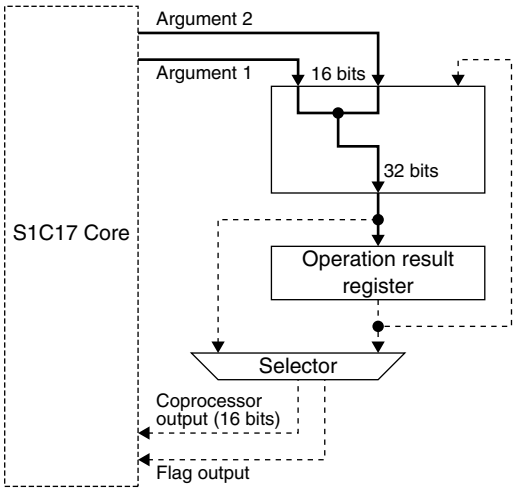


Figure 21.5.1 Data Path in Initialize Mode

Table 21.5.1 Initializing the Operation Result Register

Mode setting value	Instruction	Operations	Remarks
0x0	–	res[31:0] ← 0x0	Setting the operating mode executes the initialization without sending data.
0x1	`ld.cf %rd,%rs`	res[31:16] ← 0x0 res[15:0] ← %rs	
	(`ext` *imm9*) `ld.cf %rd,`*imm7*	res[31:16] ← 0x0 res[15:0] ← *imm7/16*	
0x2	`ld.cf %rd,%rs`	res[31:16] ← %rd res[15:0] ← %rs	
	(`ext` *imm9*) `ld.cf %rd,`*imm7*	res[31:16] ← %rd res[15:0] ← *imm7/16*	

res: operation result register

To perform a MAC operation, set the operation mode to 0x7 (signed MAC). Then send the 16-bit multiplicand (B) and 16-bit multiplier (C) to the multiplier/divider using a "`ld.ca`" instruction. The one-half (16 bits according to the output mode) result (A[15:0] or A[31:16]) and the flag status will be returned to the CPU registers. Another one-half should be read by setting the multiplier/divider into operation result read mode.
The overflow (V) flag in the PSR may be set to 1 according to the result. Other flags are set to 0.
When repeating the MAC operation without operation result read mode inserted, send multiplicand and multiplier data for number of required times. In this case it is not necessary to set the MAC mode every time.

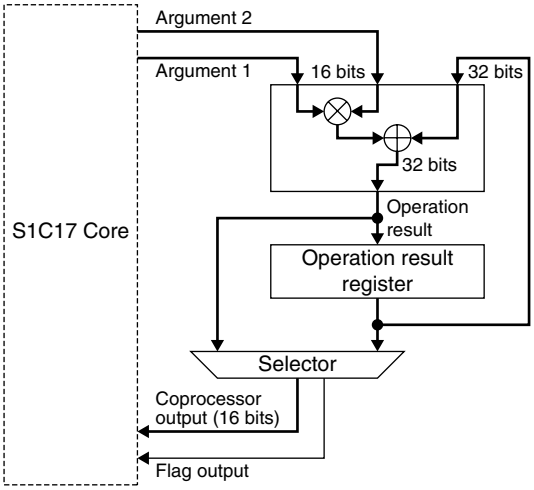


Figure 21.5.2 Data Path in MAC Mode

Table 21.5.2 Operation in MAC Mode

Mode setting value	Instruction	Operations	Flags	Remarks
0x07	`ld.ca %rd,%rs`	res[31:0] ← %rd × %rs + res[31:0] %rd ← res[15:0]	psr (CVZN) ← 0b0100 if an overflow has occurred Otherwise psr (CVZN) ← 0b0000	The operation result register keeps the operation result until it is rewritten by other operation.
	(`ext` *imm9*) `ld.ca %rd,`*imm7*	res[31:0] ← %rd × *imm7/16* + res[31:0] %rd ← res[15:0]		
0x17	`ld.ca %rd,%rs`	res[31:0] ← %rd × %rs + res[31:0] %rd ← res[31:16]		
	(`ext` *imm9*) `ld.ca %rd,`*imm7*	res[31:0] ← %rd × *imm7/16* + res[31:0] %rd ← res[31:16]		

res: operation result register

Example:

```
ld.cw %r0,0x7   ; Sets the modes (signed MAC mode and 16 low-order bits output mode).
ld.ca %r0,%r1   ; Performs "res = %r0 × %r1 + res" and loads the 16 low-order bits of the result to %r0.
ld.cw %r0,0x13  ; Sets the modes (operation result read mode and 16 high-order bits output mode).
ld.ca %r1,%r0   ; Loads the 16 high-order bits of the result to %r1.
```

Conditions to set the overflow (V) flag

An overflow occurs in a MAC operation and the overflow (V) flag is set to 1 when the signs of the multiplication result, operation result register value, and multiplication & accumulation result match the following conditions:

Table 21.5.3 Conditions to Set the Overflow (V) Flag

Mode setting value	Sign of multiplication result	Sign of operation result register value	Sign of multiplication & accumulation result
0x07	0 (positive)	0 (positive)	1 (negative)
0x07	1 (negative)	1 (negative)	0 (positive)

An overflow occurs when a MAC operation performs addition of positive values and a negative value results, or it performs addition of negative values and a positive value results. The coprocessor holds the operation result when the overflow (V) flag is cleared.

Conditions to clear the overflow (V) flag

The overflow (V) flag that has been set will be cleared when an overflow has not been occurred during execution of the "`ld.ca`" instruction for MAC operation or when the "`ld.ca`" or "`ld.cf`" instruction is executed in an operation mode other than operation result read mode.

21.6 Reading Results

The "ld.ca" instruction cannot load a 32-bit operation result to a CPU register, so a multiplication or MAC operation returns the one-half (16 bits according to the output mode) result (A[15:0] or A[31:16]) and the flag status to the CPU registers. Another one-half should be read by setting the multiplier/divider into operation result read mode. The operation result register keeps the loaded operation result until it is rewritten by other operation.

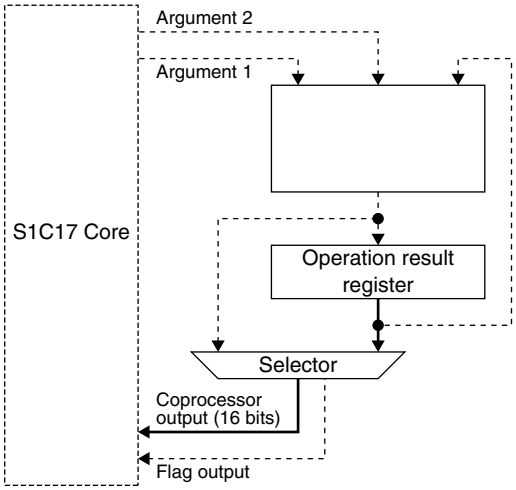


Figure 21.6.1 Data Path in Operation Result Read Mode

Table 21.6.1 Operation in Operation Result Read Mode

Mode setting value	Instruction	Operations	Flags	Remarks
0x03	ld.ca %rd,%rs	%rd ← res[15:0]	psr (CVZN) ← 0b0000	This operation mode does not affect the operation result register.
	ld.ca %rd,*imm7*	%rd ← res[15:0]		
0x13	ld.ca %rd,%rs	%rd ← res[31:16]		
	ld.ca %rd,*imm7*	%rd ← res[31:16]		

res: operation result register

22 Electrical Characteristics

22.1 Absolute Maximum Ratings

(V_{SS} = 0V)

Item	Symbol	Condition	Rated value	Unit
Power supply voltage	V_{DD}		-0.3 to 4.0	V
Flash programming voltage	V_{PP}		8	V
LCD power supply voltage	V_{C3}		-0.3 to 4.0	V
Input voltage	V_I		-0.3 to V_{DD} + 0.5	V
Output voltage	V_O		-0.3 to V_{DD} + 0.5	V
High level output current	I_{OH}	1 pin	-10	mA
		Total of all pins	-20	mA
Low level output current	I_{OL}	1 pin	10	mA
		Total of all pins	20	mA
Storage temperature	Tstg		-65 to 125	°C
Soldering temperature/time	Tsol		260°C, 10 seconds (lead section)	–

22.2 Recommended Operating Conditions

(V_{SS} = 0V)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Operating power supply voltage	V_{DD}	Normal operation mode	2.0		3.6	V
Flash programming voltage	$V_{PP}P$		6.8	7.0	7.2	V
Flash programming temperature	T_{PP}		10		40	°C
Flash erasing voltage	$V_{PP}E$		7.3	7.5	7.7	V
Operating frequency	f_{OSC3A}	Crystal/ceramic oscillation	0.2		8.2	MHz
	f_{OSC1A}	Crystal oscillation		32.768		kHz
Operating temperature	Ta	During normal operation (Flash read only)	-40		85	°C
		During Flash programming and erasing	10		40	°C
Capacitor between V_{SS} and V_{D1}	C_1			0.1		µF
Capacitor between V_{SS} and V_{OSC}	C_2			0.1		µF
Capacitor between V_{SS} and V_{C1} *1	C_3			0.1		µF
Capacitor between V_{SS} and V_{C2} *1	C_4			0.1		µF
Capacitor between V_{SS} and V_{C3} *1	C_5			0.1		µF
Capacitor between CA and CB *1	C_6			0.1		µF

*1 The capacitors are not required when LCD driver is not used. In this case, leave the V_{C1} to V_{C3}, CA, and CB pins open.

22.3 Current Consumption

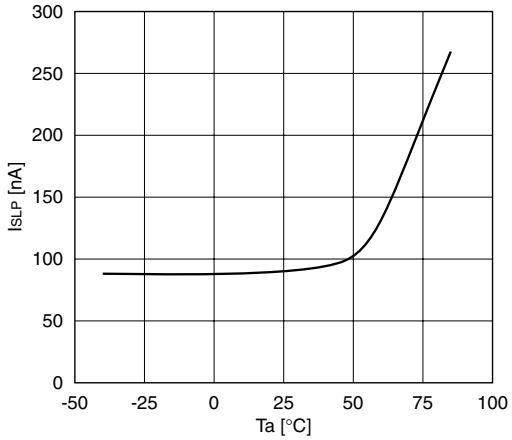
Unless otherwise specified: V_{DD} = 2.0 to 3.6V, V_{SS} = 0V, Ta = 25°C, PCKEN[1:0] = 0x3 (ON), RDWAIT[1:0] = 0x0 (no wait), OSC1A = no theoretical regulation correction, CCLKGR[1:0] = 0x0 (gear ratio 1/1), RTCRUN = 0 (OFF), HVLD = 0, LCD = OFF

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Current consumption in SLEEP mode	I_{SLP}	OSC1A = OFF, OSC1B = OFF, OSC3B = OFF		90	160	nA
		OSC1A = 32kHz, OSC3B = OFF, OSC3A = OFF, RTCRUN = 1 (ON)		170	220	nA
		OSC1B = 32kHz, OSC3B = OFF, OSC3A = OFF, RTCRUN = 1 (ON)		770	1000	nA
Current consumption in HALT mode	I_{HALT1}	OSC1A = 32kHz, OSC3B = OFF, OSC3A = OFF, PCKEN[1:0] = 0x0 (OFF)		0.42	0.55	µA
		OSC1A = 32kHz, OSC3B = OFF, OSC3A = OFF, PCKEN[1:0] = 0x0 (OFF), RTCRUN = 1 (ON)		0.42	0.55	µA
		OSC1A = 32kHz, OSC3B = OFF, OSC3A = OFF		0.88	1.10	µA
		OSC1B = 32kHz, OSC3B = OFF, OSC3A = OFF, PCKEN[1:0] = 0x0 (OFF)		0.95	1.20	µA
	I_{HALT2}	OSC1A = 32kHz, OSC3B = OFF, OSC3A = ON (1MHz ceramic)		67	100	µA
		OSC1A = 32kHz, OSC3B = OFF, OSC3A = ON (4MHz ceramic)		130	180	µA
	I_{HALT3}	OSC1A = 32kHz, OSC3B = ON (500kHz), OSC3A = OFF		68	120	µA
		OSC1A = 32kHz, OSC3B = ON (1MHz), OSC3A = OFF		87	150	µA
		OSC1A = 32kHz, OSC3B = ON (2MHz), OSC3A = OFF		130	200	µA
Current consumption during execution *1	I_{EXE1}	OSC1A = 32kHz, OSC3B = OFF, OSC3A = OFF, CPU = OSC1A		10	13	µA
		OSC1A = 32kHz, OSC3B = OFF, OSC3A = OFF, CPU = OSC1A, CCLKGR[1:0] = 0x2 (gear ratio 1/4)		4.0	5.5	µA
	I_{EXE2}	OSC1A = 32kHz, OSC3B = OFF, OSC3A = ON (1MHz ceramic), CPU = OSC3A		350	480	µA
		OSC1A = 32kHz, OSC3B = OFF, OSC3A = ON (1MHz ceramic), CPU = OSC3A, CCLKGR[1:0] = 0x2 (gear ratio 1/4)		200	280	µA
		OSC1A = 32kHz, OSC3B = OFF, OSC3A = ON (4MHz ceramic), CPU = OSC3A		1200	1800	µA
		OSC1A = 32kHz, OSC3B = OFF, OSC3A = ON (4MHz ceramic), CPU = OSC3A, CCLKGR[1:0] = 0x2 (gear ratio 1/4)		530	750	µA
	I_{EXE3}	OSC1A = 32kHz, OSC3B = ON (500kHz), OSC3A = OFF, CPU = OSC3B		230	310	µA
		OSC1A = 32kHz, OSC3B = ON (1MHz), OSC3A = OFF, CPU = OSC3B		370	510	µA
		OSC1A = 32kHz, OSC3B = ON (2MHz), OSC3A = OFF, CPU = OSC3B		650	900	µA
		OSC1A = 32kHz, OSC3B = ON (2MHz), OSC3A = OFF, CPU = OSC3B, CCLKGR[1:0] = 0x2 (gear ratio 1/4)		320	450	µA
Current consumption during execution in heavy load protection mode *1	I_{EXE1H}	OSC1A = 32kHz, OSC3B = OFF, OSC3A = OFF, CPU = OSC1A, HVLD = 1		21	27	µA

*1 The values of current consumption during execution were measured when a test program consisting of 60.5% ALU instructions, 17% branch instructions, 12% memory read instructions, and 10.5% memory write instructions was executed continuously in the Flash memory.

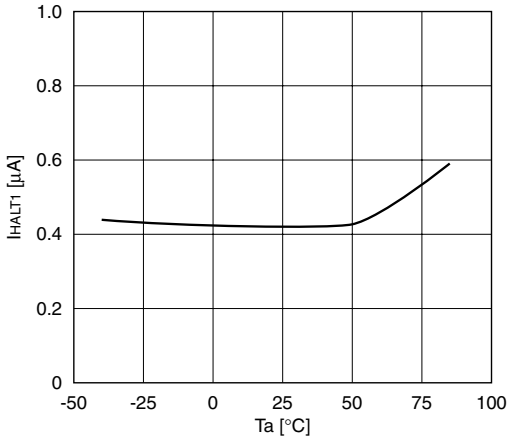
Current consumption-temperature characteristic in SLEEP mode

OSC1A = OFF, OSC1B = OFF, OSC3B = OFF, OSC3A = OFF, Typ. value



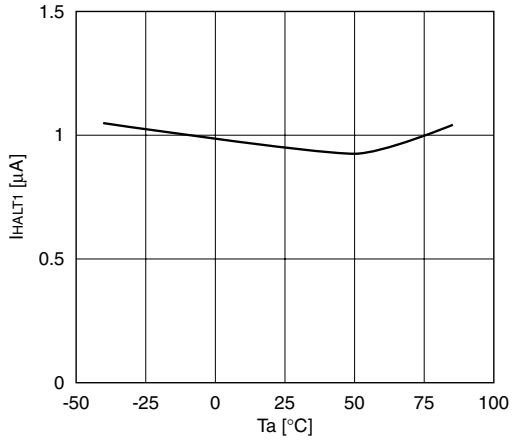
Current consumption-temperature characteristic in HALT mode (OSC1A operation)

OSC1A = 32.768kHz crystal, OSC3B = OFF, OSC3A = OFF, PCKEN[1:0] = 0x3, CCLKGR[1:0] = 0x0, Typ. value



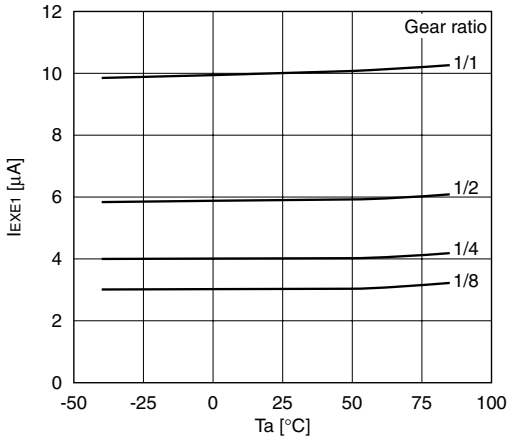
Current consumption-temperature characteristic in HALT mode (OSC1B operation)

OSC1B = 32kHz, OSC3B = OFF, OSC3A = OFF, PCKEN[1:0] = 0x0, CCLKGR[1:0] = 0x0, Typ. value



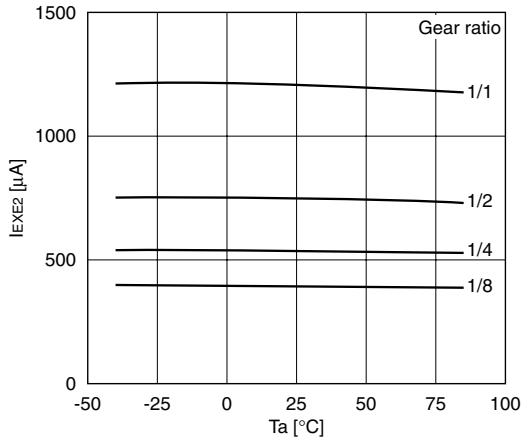
Current consumption-temperature characteristic during execution with OSC1A + clock gear

OSC1A = 32.768kHz crystal, OSC3B = OFF, OSC3A = OFF, PCKEN[1:0] = 0x3, Typ. value



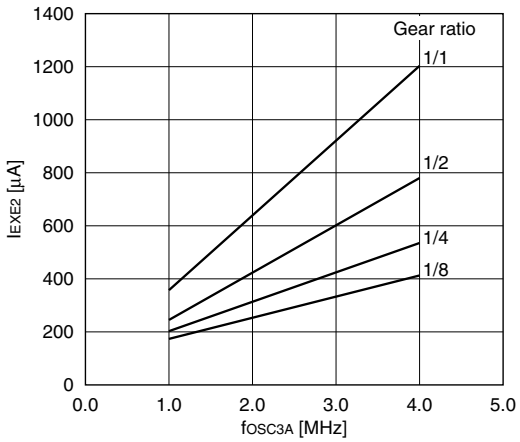
Current consumption-temperature characteristic during execution with OSC3A + clock gear

OSC3A = ON (4MHz ceramic), OSC3B = OFF, OSC1A = 32.768kHz crystal, PCKEN[1:0] = 0x3, Typ. value



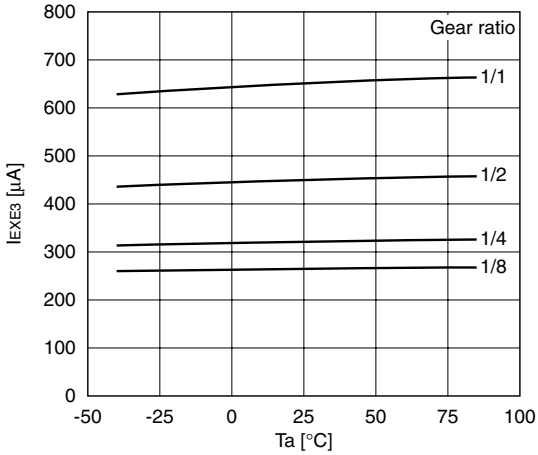
Current consumption-frequency characteristic during execution with OSC3A

OSC3A = ON, OSC3B = OFF, OSC1A = 32.768kHz crystal, PCKEN[1:0] = 0x3, Ta = 25°C, Typ. value



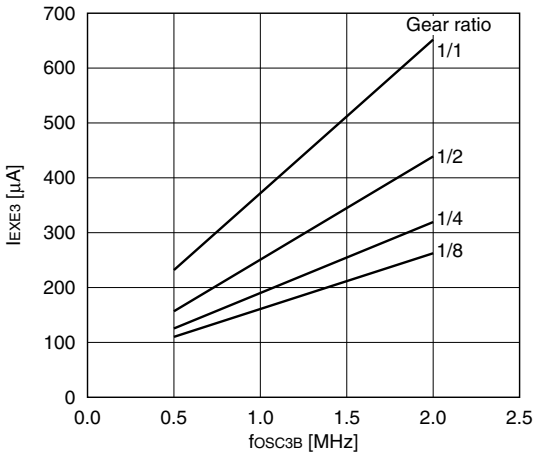
Current consumption-temperature characteristic during execution with OSC3B + clock gear

OSC3B = ON (2MHz), OSC3A = OFF,
OSC1A = 32.768kHz crystal, PCKEN[1:0] = 0x3, Typ. value



Current consumption-frequency characteristic during execution with OSC3B

OSC3B = ON, OSC3A = OFF, OSC1A = 32.768kHz crystal,
PCKEN[1:0] = 0x3, CCLKGR[1:0] = 0x2, Ta = 25°C, Typ. value



22.4 Oscillation Characteristics

Oscillation characteristics change depending on conditions (board pattern, components used, etc.). Use the following characteristics as reference values.

OSC1A crystal oscillation

Unless otherwise specified: V_{DD} = 2.0 to 3.6V, V_{SS} = 0V, Ta = 25°C, C_G = built-in, C_D = built-in, R_f = built-in, R_D = built-in, C_{G1} = 3pF, C_{D1} = 3pF

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Oscillation start time *1	t_{sta}				3	s
Built-in gate capacitance	C_G	In case of the chip		10		pF
Built-in drain capacitance	C_D	In case of the chip		5		pF

*1 Crystal resonator = C-002RX: manufactured by EPSON TOYOCOM (R_1 = 50kΩ Max., C_L = 7pF)
MC-146: manufactured by EPSON TOYOCOM (R_1 = 65kΩ Max., C_L = 7pF)

OSC1B oscillation

Unless otherwise specified: V_{DD} = 2.0 to 3.6V, V_{SS} = 0V, Ta = 25°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Oscillation start time	t_{sta}				200	µs
Oscillation frequency *1 *2	f_{OSC1B}		Typ. × 0.95	32.768	Typ. × 1.05	kHz
Dependence of oscillation frequency on temperature *2	T_{fOSC1B}	Frequency accuracy per ±1°C change in temperature (with reference to 25°C)		±0.12	±0.3	%/°C

*1 In chip mounting, the value may exceed the range shown above according to the mount condition on the board.
*2 Reference value

OSC3A crystal oscillation

Unless otherwise specified: V_{DD} = 2.0 to 3.6V, V_{SS} = 0V, Ta = 25°C, R_f = built-in, R_D = built-in, C_{G3} = 15pF, C_{D3} = 15pF

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Oscillation start time *1 *2	t_{sta}				20.0	ms

*1 Crystal resonator = MA-406: manufactured by EPSON TOYOCOM
*2 The oscillation start time varies according to the crystal resonator used and the C_{G3} and C_{D3} values.

OSC3A ceramic oscillation

Unless otherwise specified: V_{DD} = 2.0 to 3.6V, V_{SS} = 0V, Ta = 25°C, R_f = built-in, R_D = built-in

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Oscillation start time *1 *2	t_{sta}				1.0	ms

*1 Ceramic resonator = CSTCC2M00G56-R0: manufactured by Murata Manufacturing Co., Ltd. (SMD, C_{G3} = C_{D3} = 47pF built-in)
CSTCR4M00G53-R0: manufactured by Murata Manufacturing Co., Ltd. (SMD, C_{G3} = C_{D3} = 15pF built-in)
CSTLS4M00G53-B0: manufactured by Murata Manufacturing Co., Ltd. (leaded, C_{G3} = C_{D3} = 15pF built-in)
*2 The oscillation start time varies according to the ceramic resonator used and the C_{G3} and C_{D3} values.

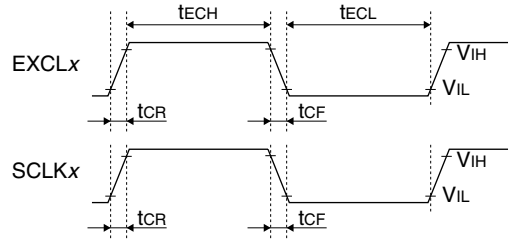
OSC3B oscillation

Unless otherwise specified: V_{DD} = 2.0 to 3.6V, V_{SS} = 0V, Ta = 25°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Oscillation start time	t_{sta}				5.0	µs
Oscillation frequency *1	f_{OSC3B}	OSC3BFSEL[1:0] = 0x0 (2MHz)	Typ. × 0.95	1.936	Typ. × 1.05	MHz
		OSC3BFSEL[1:0] = 0x1 (1MHz)		1.002		MHz
		OSC3BFSEL[1:0] = 0x2 (500kHz)		0.511		MHz
Dependence of oscillation frequency on temperature *1	Tf_{OSC3B}	OSC3BFSEL[1:0] = 0x0–0x2, Frequency accuracy per ±1°C change in temperature (with reference to 25°C)		0.05	0.07	%/°C

*1 Reference value

22.5 External Clock Input Characteristics



Unless otherwise specified: V_{DD} = 2.0 to 3.6V, V_{SS} = 0V, V_{IH} = 0.8V_{DD}, V_{IL} = 0.2V_{DD}, Ta = -40 to 85°C

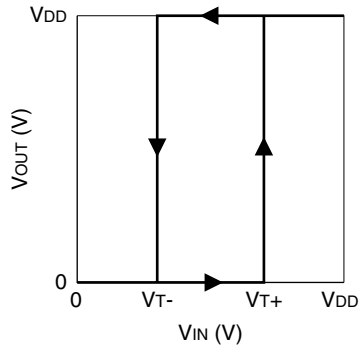
Item	Symbol	Min.	Typ.	Max.	Unit
EXCL*x* input High pulse width	t_{ECH}	60			ns
EXCL*x* input Low pulse width	t_{ECL}	60			ns
UART transfer rate	R_U			230400	bps
UART transfer rate (IrDA mode)	R_{UIrDA}			115200	bps
Input rise time	t_{CR}			80	ns
Input fall time	t_{CF}			80	ns

22.6 Input/Output Pin Characteristics

Unless otherwise specified: V_{DD} = 2.0 to 3.6V, V_{SS} = 0V, Ta = -40 to 85°C

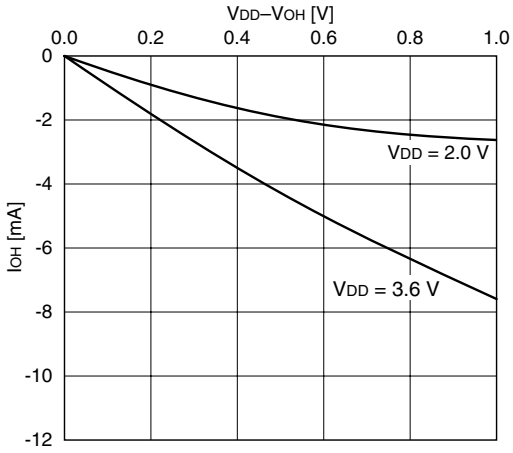
Item	Symbol	Condition	Min.	Typ.	Max.	Unit
High level Schmitt input threshold voltage	V_{T+}	P*xx*, #RESET	0.5V_{DD}		0.9V_{DD}	V
Low level Schmitt input threshold voltage	V_{T-}	P*xx*, #RESET	0.1V_{DD}		0.5V_{DD}	V
Hysteresis voltage	ΔV_T	P*xx*, #RESET	0.1V_{DD}			V
High level output current	I_{OH}	P*xx*, V_{OH} = 0.9V_{DD}			-0.5	mA
Low level output current	I_{OL}	P*xx*, V_{OL} = 0.1V_{DD}	0.5			mA
Leakage current	I_{LEAK}	P*xx*, #RESET	-100		100	nA
Input pull-up resistance	R_{IN}	P*xx*, #RESET	100		500	kΩ
Pin capacitance	C_{IN}	P*xx*, V_{IN} = 0V, f = 1MHz, Ta = 25°C			15	pF
Reset low pulse width	t_{SR}	V_{IH} = 0.8V_{DD}, V_{IL} = 0.2V_{DD}	100			µs
Operating power voltage	V_{SR}		2.0			V
#RESET power-on reset time	t_{PSR}		1.0			ms

Schmitt input threshold voltage



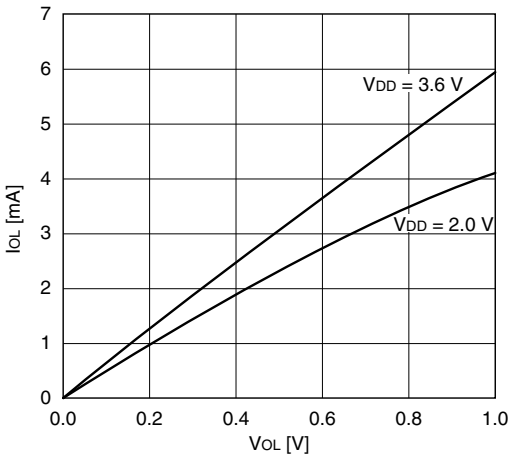
High-level output current characteristic

Ta = 85°C, Max. value

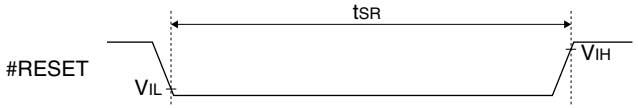


Low-level output current characteristic

Ta = 85°C, Min. value



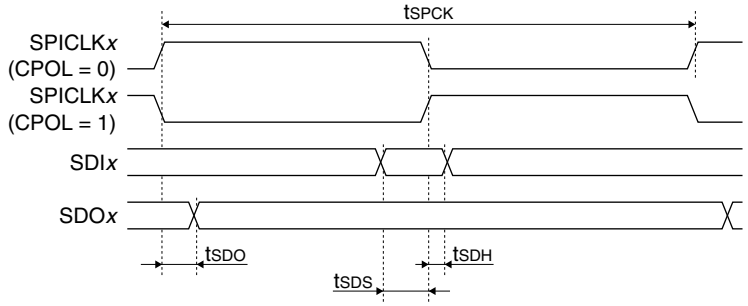
Reset pulse



#RESET power-on reset timing

Note: Be sure to set the #RESET pin to 0.1 VDD or less when performing a power-on reset after the power is turned off.

22.7 SPI Characteristics



Master mode

Unless otherwise specified: VDD = 2.0 to 3.6V, VSS = 0V, Ta = -40 to 85°C

Item	Symbol	Min.	Typ.	Max.	Unit
SPICLK*x* cycle time	tSPCK	500			ns
SDI*x* setup time	tSDS	70			ns
SDI*x* hold time	tSDH	10			ns
SDO*x* output delay time	tSDO			20	ns

Slave mode

Unless otherwise specified: V_{DD} = 2.0 to 3.6V, V_{SS} = 0V, Ta = -40 to 85°C

Item	Symbol	Min.	Typ.	Max.	Unit
SPICLK*x* cycle time	t_{SPCK}	500			ns
SDI*x* setup time	t_{SDS}	30			ns
SDI*x* hold time	t_{SDH}	50			ns
SDO*x* output delay time	t_{SDO}			80	ns

22.8 LCD Driver Characteristics

The typical values in the following LCD driver characteristics varies depending on the panel load (panel size, drive duty, number of display pixels and display contents), so evaluate them by connecting to the actually used LCD panel.

LCD drive voltage

Unless otherwise specified: V_{DD} = 2.2 to 3.6V, V_{SS} = 0V, Ta = 25°C, C_3–C_6 = 0.1µF, Checker pattern displayed, No panel load, VCSEL = 1 (V_{C2} reference voltage), LCD_BCLK[1:0] = 0x1 (2kHz/OSC1A = 32kHz source clock), FRMCNT[1:0] = 0x1 (64Hz/OSC1A = 32kHz (LCLK = 512Hz)), LDUTY[2:0] = 0x3 (1/4 duty), DSPC[1:0] = 0x1(Normal display)

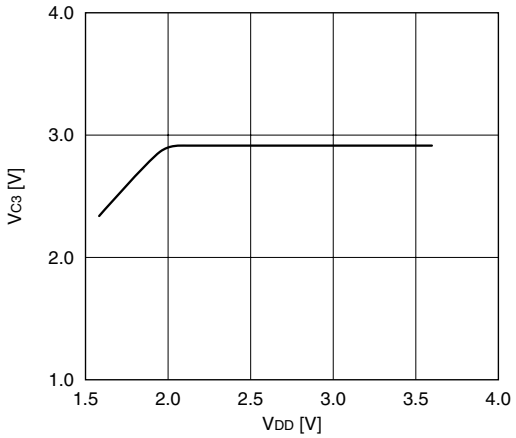
Item	Symbol	Condition	Min.	Typ.	Max.	Unit
LCD drive voltage (V_{C2} reference voltage)	V_{C1}	Connect 1MΩ load resistor between V_{SS} and V_{C1}	Typ. × 0.96	0.974	Typ. × 1.04	V
	V_{C2}	Connect 1MΩ load resistor between V_{SS} and V_{C2}		1.958		V
	V_{C3}	Connect 1MΩ load resistor between V_{SS} and V_{C3}		2.900		V

Unless otherwise specified: V_{DD} = 2.0 to 3.6V, V_{SS} = 0V, Ta = 25°C, C_3–C_6 = 0.1µF, Checker pattern displayed, No panel load, VCSEL = 0 (V_{C1} reference voltage), LCD_BCLK[1:0] = 0x1 (2kHz/OSC1A = 32kHz source clock), FRMCNT[1:0] = 0x1 (64Hz/OSC1A = 32kHz (LCLK = 512Hz)), LDUTY[2:0] = 0x3 (1/4 duty), DSPC[1:0] = 0x1(Normal display)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
LCD drive voltage (V_{C1} reference voltage)	V_{C1}	Connect 1MΩ load resistor between V_{SS} and V_{C1}	Typ. × 0.96	0.977	Typ. × 1.04	V
	V_{C2}	Connect 1MΩ load resistor between V_{SS} and V_{C2}		1.884		V
	V_{C3}	Connect 1MΩ load resistor between V_{SS} and V_{C3}		2.800		V

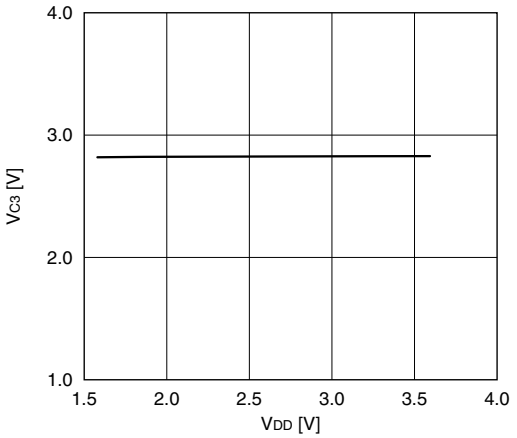
LCD drive voltage-supply voltage characteristic (when V_{C2} reference voltage is selected)

V_{DD} = 1.6 to 3.6V, LCD_BCLK[1:0] = 0x1 (2kHz/OSC1A = 32kHz source clock), FRMCNT[1:0] = 0x1 (64Hz/OSC1A = 32kHz (LCLK = 512Hz)), LDUTY[2:0] = 0x3 (1/4 duty), DSPC[1:0] = 0x1(Normal display), When a 1 MΩ load resistor is connected between V_{SS}–V_{C1}, V_{SS}–V_{C2}, and V_{SS}–V_{C3} (no panel load), Checker pattern displayed, Ta = 25°C, Typ. value



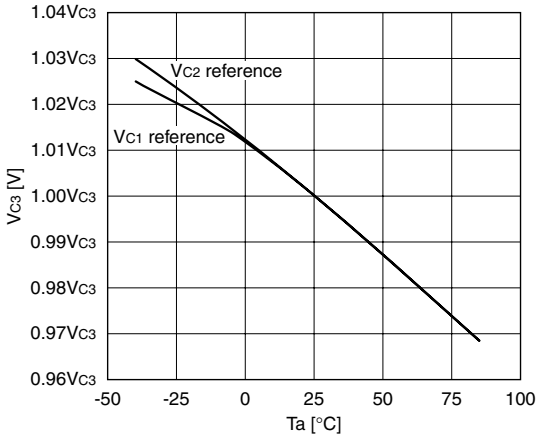
LCD drive voltage-supply voltage characteristic (when V_{C1} reference voltage is selected)

V_{DD} = 1.6 to 3.6V, LCD_BCLK[1:0] = 0x1 (2kHz/OSC1A = 32kHz source clock), FRMCNT[1:0] = 0x1 (64Hz/OSC1A = 32kHz (LCLK = 512Hz)), LDUTY[2:0] = 0x3 (1/4 duty), DSPC[1:0] = 0x1(Normal display), When a 1 MΩ load resistor is connected between V_{SS}–V_{C1}, V_{SS}–V_{C2}, and V_{SS}–V_{C3} (no panel load), Checker pattern displayed, Ta = 25°C, Typ. value



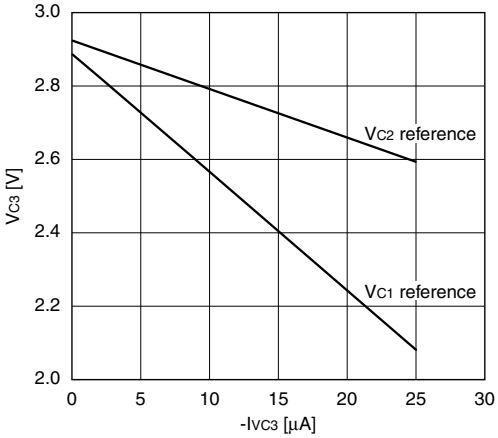
LCD drive voltage-temperature characteristic

V_{DD} = 3.0V, LCD_BCLK[1:0] = 0x1 (2kHz/OSC1A = 32kHz source clock), FRMCNT[1:0] = 0x1 (64Hz/OSC1A = 32kHz (LCLK = 512Hz)), LDUTY[2:0] = 0x3 (1/4 duty), DSPC[1:0] = 0x1(Normal display), When a 1 MΩ load resistor is connected between V_{SS}–V_{C1}, V_{SS}–V_{C2}, and V_{SS}–V_{C3} (no panel load), Checker pattern displayed, Typ. value



LCD drive voltage-load characteristic

V_{DD} = 2.2V (V_{C2} reference)/V_{DD} = 1.8V (V_{C1} reference), LCD_BCLK[1:0] = 0x1 (2kHz/OSC1A = 32kHz source clock), FRMCNT[1:0] = 0x1 (64Hz/OSC1A = 32kHz (LCLK = 512Hz)), LDUTY[2:0] = 0x3 (1/4 duty), DSPC[1:0] = 0x1(Normal display), When a 1 MΩ load resistor is connected between V_{SS}–V_{C1}, V_{SS}–V_{C2}, and V_{SS}–V_{C3} (no panel load), Checker pattern displayed, Ta = 25°C, Typ. value



SEG/COM output characteristics

Unless otherwise specified: V_{DD} = 2.0 to 3.6V, V_{SS} = 0V, Ta = -40 to 85°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Segment/Common output current	I_{SEGH}	SEG*xx*, COM*xx*, V_{SEGH} = V_{C3} - 0.1V			-10	µA
	I_{SEGL}	SEG*xx*, COM*xx*, V_{SEGL} = 0.1V	10			µA

LCD driver circuit current consumption

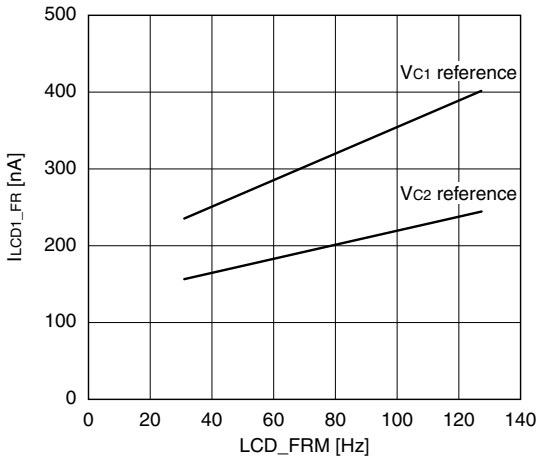
Unless otherwise specified: V_{DD} = 2.2 to 3.6V (V_{C2} reference)/2.0 to 3.6V (V_{C1} reference), V_{SS} = 0V, Ta = 25°C, C_3–C_6 = 0.1µF, Checker pattern displayed, No panel load, PCKEN[1:0] = 0x0 (OFF), LCD_BCLK[1:0] = 0x1 (2kHz/OSC1A = 32kHz source clock), FRMCNT[1:0] = 0x1 (64Hz/OSC1A = 32kHz (LCLK = 512Hz)), LDUTY[2:0] = 0x3 (1/4 duty), DSPC[1:0] = 0x1(Normal display)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
LCD circuit current with V_{C2} reference voltage *1	I_{LCD2}	VCSEL = 1		200	215	nA
LCD circuit current with V_{C1} reference voltage *1	I_{LCD1}	VCSEL = 0		300	330	nA
Heavy load protection mode LCD circuit current with V_{C2} reference voltage *1	I_{LCD2H}	LHVLD = 1, VCSEL = 1		2.2	3.5	µA
Heavy load protection mode LCD circuit current with V_{C1} reference voltage *1	I_{LCD1H}	LHVLD = 1, VCSEL = 0		1.3	2.0	µA

*1 This value is added to the current consumption in HALT mode or current consumption during execution when the LCD circuit is active. Current consumption increases according to the display contents and panel load.

LCD frame frequency dependence current consumption characteristic

V_{DD} = 3.0V, Ta = 25°C, LCD_BCLK[1:0] = 0x1 (2kHz/OSC1A = 32kHz source clock), FRMCNT[1:0] = 0x0 to 0x3 (128 to 32Hz/OSC1A = 32kHz (LCLK = 512Hz)), LDUTY[2:0] = 0x3 (1/4 duty), DSPC[1:0] = 0x1(Normal display), Checker pattern displayed, No panel load, PCKEN[1:0] = 0x0 (OFF), Typ. value



LCD boost clock frequency dependence current consumption characteristic/ drive voltage characteristic

Common conditions:

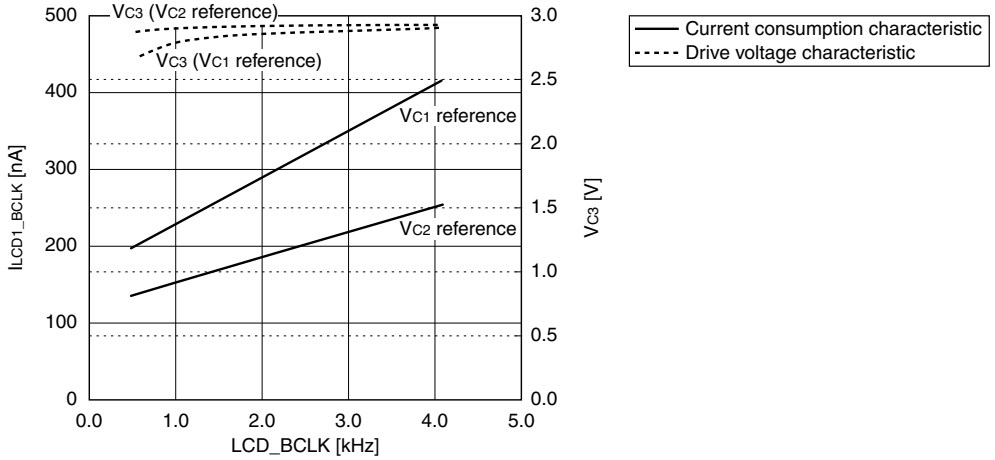
Ta = 25°C, LCD_BCLK[1:0] = 0x0 to 0x3 (4kHz to 512Hz/OSC1A = 32kHz source clock), FRMCNT[1:0] = 0x1 (64Hz/OSC1A = 32kHz (LCLK = 512Hz)), LDUTY[2:0] = 0x3 (1/4 duty), DSPC[1:0] = 0x1(Normal display), Checker pattern displayed, No panel load, PCKEN[1:0] = 0x0 (OFF), Typ. value

Current consumption characteristic conditions:

V_{DD} = 3.0V

Drive voltage characteristic conditions:

V_{DD} = 1.8V (V_{C1} reference)/V_{DD} = 2.2V (V_{C2} reference), When a 1 MΩ load resistor is connected between V_{SS}–V_{C1} and V_{SS}–V_{C2}, and a 1µA load is connected to V_{C3}



22.9 SVD Circuit Characteristics

Analog characteristics

Unless otherwise specified: V_{DD} = 2.0 to 3.6V, V_{SS} = 0V, Ta = -40 to 85°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
SVD voltage	V_{SVD}	SVDC[4:0] = 0x0		–		V
		SVDC[4:0] = 0x1		–		V
		SVDC[4:0] = 0x2		–		V
		SVDC[4:0] = 0x3		–		V
		SVDC[4:0] = 0x4		–		V
		SVDC[4:0] = 0x5		–		V
		SVDC[4:0] = 0x6		–		V
		SVDC[4:0] = 0x7		–		V
		SVDC[4:0] = 0x8		–		V
		SVDC[4:0] = 0x9		–		V
		SVDC[4:0] = 0xa		–		V
		SVDC[4:0] = 0xb		–		V
		SVDC[4:0] = 0xc		–		V
		SVDC[4:0] = 0xd		–		V
		SVDC[4:0] = 0xe	Typ. ×	2.00	Typ. ×	V
		SVDC[4:0] = 0xf	0.96	2.10	1.04	V
		SVDC[4:0] = 0x10		2.20		V
		SVDC[4:0] = 0x11		2.30		V
		SVDC[4:0] = 0x12		2.40		V
		SVDC[4:0] = 0x13		2.50		V
		SVDC[4:0] = 0x14		2.60		V
		SVDC[4:0] = 0x15		2.70		V
		SVDC[4:0] = 0x16		2.80		V
		SVDC[4:0] = 0x17		2.90		V
		SVDC[4:0] = 0x18		3.00		V
		SVDC[4:0] = 0x19		3.10		V
		SVDC[4:0] = 0x1a		3.20		V
		SVDC[4:0] = 0x1b		–		V
		SVDC[4:0] = 0x1c		–		V
		SVDC[4:0] = 0x1d		–		V

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
SVD voltage	V_{SVD}	SVDC[4:0] = 0x1e	Typ. × 0.96	–	Typ. × 1.04	V
		SVDC[4:0] = 0x1f		–		V
SVD circuit-enable response time *1	t_{SVDEN}				500	µs
SVD circuit response time *2	t_{SVD}				60	µs

*1 This time is required to obtain stable detection results after SVDEN is altered from 0 to 1.
*2 This time is required to obtain stable detection results after SVDC[4:0] is altered.

SVD circuit current consumption

Unless otherwise specified: V_{DD} = 2.0 to 3.6V, V_{SS} = 0V, Ta = 25°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
SVD circuit current *1	I_{SVD}	V_{DD} = 3.6V, SVDC[4:0] = 0xe (2.0V)		12	17	µA

*1 This value is added to the current consumption during SLEEP/HALT/execution (with or without heavy load protection mode) when the SVD circuit is active.

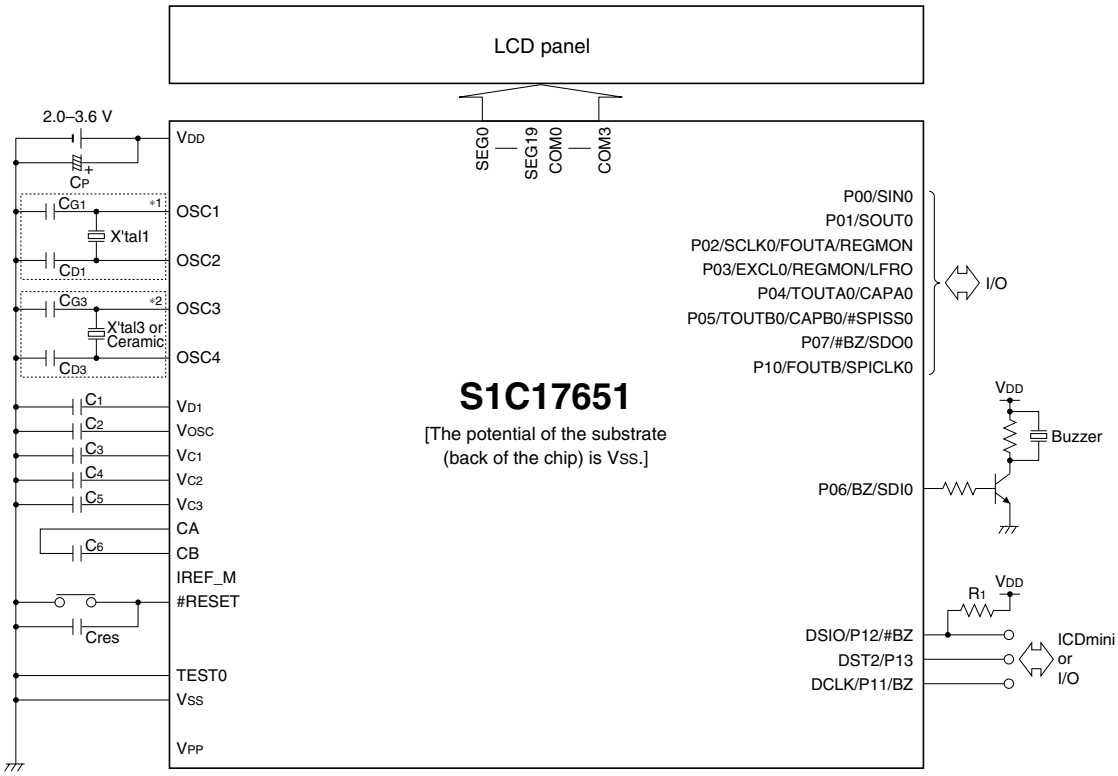
22.10 Flash Memory Characteristics

Unless otherwise specified: V_{DD} = 2.0 to 3.6V, V_{PP} = 7.0V (for programming)/7.5V (for erasing), V_{SS} = 0V, Ta = 10 to 40°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Programming count *1	C_{FEP}	Programmed data is guaranteed to be retained for 10 years.	3			times

*1 Assumed that Erasing + Programming as count of 1. The count includes programming in the factory.

23 Basic External Connection Diagram



*1: This external circuit is required only when the OSC1A oscillator is used.
*2: This external circuit is required only when the OSC3A oscillator is used.

Recommended values for external parts

External parts for the OSC1A oscillator circuit

Symbol	Resonator	Recommended manufacturer	Frequency [Hz]	Product number	Recommended values		Recommended operating condition
					C_{D1} [pF]	C_{G1} [pF]	Temperature range [°C]
X'tal1	Crystal	Epson Toyocom Corporation	32.768k	C-002RX (R_1 = 50 kΩ (Max.), C_L = 7 pF)	3	3	-10 to 60°C
				MC-146 (R_1 = 65 kΩ (Max.), C_L = 7 pF)	3	3	-40 to 85°C

External parts for the OSC3A oscillator circuit

Symbol	Resonator	Recommended manufacturer	Frequency [Hz]	Product number	Recommended values *		Recommended operating condition
					C_{D3} [pF]	C_{G3} [pF]	Temperature range [°C]
X'tal3	Crystal	Epson Toyocom Corporation	4M	MA-406	15	15	-20 to 70°C
Ceramic	Ceramic	Murata Manufacturing Co., Ltd.	2M	CSTCC2M00G56-R0 (SMD)	(47)	(47)	-40 to 85°C
			4M	CSTCR4M00G53-R0 (SMD)	(15)	(15)	-40 to 85°C
			4M	CSTLS4M00G53-B0 (lead)	(15)	(15)	-40 to 85°C

* The C_{D3} and C_{G3} values enclosed with ( ) are the built-in capacitances of the resonator.

Other

Symbol	Name	Recommended value
CP	Capacitor for power supply	3.3 µF
C_{G1}	Gate capacitor	3 pF
C_{D1}	Drain capacitor	3 pF
C_{G3}	Gate capacitor	15 pF
C_{D3}	Drain capacitor	15 pF
Cres	Capacitor for #RESET pin	0.47 µF
C_1	Capacitor between V_{D1} and V_{SS}	0.1 µF
C_2	Capacitor between V_{OSC} and V_{SS}	0.1 µF
C_3	Capacitor between V_{C1} and V_{SS}	0.1 µF
C_4	Capacitor between V_{C2} and V_{SS}	0.1 µF
C_5	Capacitor between V_{C3} and V_{SS}	0.1 µF
C_6	Capacitor between CA and CB	0.1 µF
R_1	DSIO pull-up resistor	10 kΩ

Notes: • The values in the above table are shown only for reference and not guaranteed.

• Crystal and ceramic resonators are extremely sensitive to influence of external components and printed-circuit boards. Before using a resonator, please contact the manufacturer for further information on conditions of use.

24 Package/Chip

24.1 TQFP Package

TQFP13-64pin package

(Unit: mm)

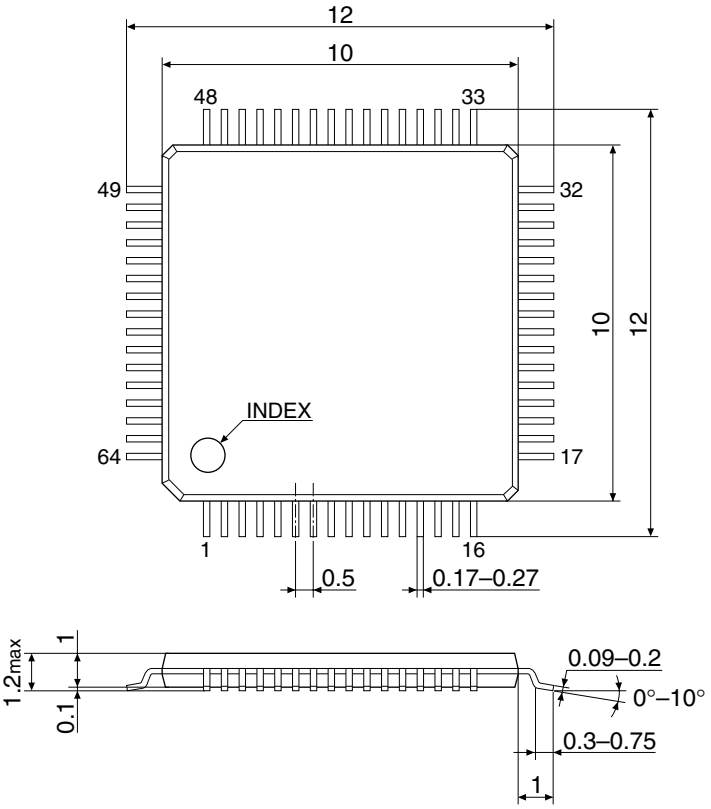


Figure 24.1.1 TQFP13-64pin Package Dimensions

24.2 Chip

24.2.1 Pad Configuration

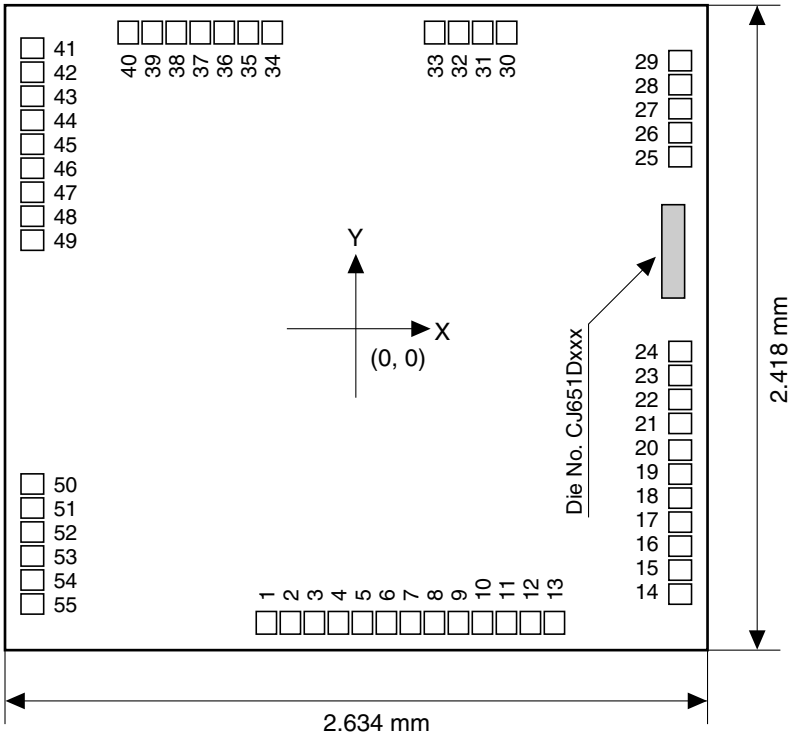


Figure 24.2.1.1 S1C17651 Pad Configuration Diagram

Chip size	X = 2.634 mm, Y = 2.418 mm
Pad opening	No. 1 to 13, 30 to 40: X = 76 µm, Y = 85 µm
	No. 14 to 29, 41 to 55: X = 85 µm, Y = 76 µm
Chip thickness	400 µm

Table 24.2.1.1 S1C17651 Pad Coordinates

No.	Name	X (µm)	Y (µm)
1	**SEG19**	-337.0	-1106.5
2	**SEG18**	-247.0	-1106.5
3	**SEG17**	-157.0	-1106.5
4	**SEG16**	-67.0	-1106.5
5	**SEG15**	23.0	-1106.5
6	**SEG14**	113.0	-1106.5
7	**SEG13**	203.0	-1106.5
8	**SEG12**	293.0	-1106.5
9	**SEG11**	383.0	-1106.5
10	**SEG10**	473.0	-1106.5
11	**SEG9**	563.0	-1106.5
12	**SEG8**	653.0	-1106.5
13	**SEG7**	743.0	-1106.5
14	**SEG6**	1214.5	-999.0
15	**SEG5**	1214.5	-909.0
16	**SEG4**	1214.5	-819.0
17	**SEG3**	1214.5	-729.0
18	**SEG2**	1214.5	-639.0
19	**SEG1**	1214.5	-549.0
20	**SEG0**	1214.5	-459.0
21	**COM3**	1214.5	-357.0
22	**COM2**	1214.5	-267.0
23	**COM1**	1214.5	-177.0
24	**COM0**	1214.5	-87.0
25	**V_{C3}**	1214.5	640.5
26	**V_{C2}**	1214.5	730.5
27	**V_{C1}**	1214.5	820.5
28	**CB**	1214.5	910.5
29	**CA**	1214.5	1000.5
30	**IREF_M**	564.0	1106.5
31	**V_{OSC}**	474.0	1106.5
32	**OSC1**	384.0	1106.5
33	**OSC2**	294.0	1106.5
34	**V_{D1}**	-315.0	1106.5
35	**OSC3**	-405.0	1106.5
36	**OSC4**	-495.0	1106.5
37	**TEST0**	-585.0	1106.5
38	**V_{SS}**	-675.0	1106.5
39	**V_{DD}**	-765.0	1106.5
40	**#RESET**	-855.0	1106.5
41	**P00**/SIN0	-1214.5	1048.0
42	**P01**/SOUT0	-1214.5	958.0
43	**P02**/SCLK0/FOUTA/REGMON	-1214.5	868.0
44	**P03**/EXCL0/REGMON/LFRO	-1214.5	778.0
45	**P04**/TOUTA0/CAPA0	-1214.5	688.0
46	**P05**/TOUTB0/CAPB0/#SPISS0	-1214.5	598.0
47	**P06**/BZ/SDI0	-1214.5	508.0
48	**P07**/#BZ/SDO0	-1214.5	418.0
49	**P10**/FOUTB/SPICLK0	-1214.5	328.0
50	**DCLK**/P11/BZ	-1214.5	-586.0
51	**DSIO**/P12/#BZ	-1214.5	-676.0
52	**DST2**/P13	-1214.5	-766.0
53	**V_{DD}**	-1214.5	-856.0
54	**V_{SS}**	-1214.5	-946.0
55	**V_{PP}**	-1214.5	-1036.0

Appendix A List of I/O Registers

Internal peripheral circuit area 1 (0x4000–0x43ff)

Peripheral	Address	Register name		Function
MISC register (8-bit device)	0x4020	MISC_ DMODE1	Debug Mode Control Register 1	Enables peripheral operations in debug mode (PCLK).
UART (with IrDA) Ch.0 (8-bit device)	0x4100	UART_ST0	UART Ch.0 Status Register	Indicates transfer, buffer and error statuses.
	0x4101	UART_TXD0	UART Ch.0 Transmit Data Register	Transmit data
	0x4102	UART_RXD0	UART Ch.0 Receive Data Register	Receive data
	0x4103	UART_MOD0	UART Ch.0 Mode Register	Sets transfer data format.
	0x4104	UART_CTL0	UART Ch.0 Control Register	Controls data transfer.
	0x4105	UART_EXP0	UART Ch.0 Expansion Register	Sets IrDA mode.
	0x4106	UART_BR0	UART Ch.0 Baud Rate Register	Sets baud rate.
	0x4107	UART_FMD0	UART Ch.0 Fine Mode Register	Sets fine mode.
8-bit timer Ch. 0 (16-bit device)	0x4240	T8_CLK0	T8 Ch.0 Count Clock Select Register	Selects a count clock.
	0x4242	T8_TR0	T8 Ch.0 Reload Data Register	Sets reload data.
	0x4244	T8_TC0	T8 Ch.0 Counter Data Register	Counter data
	0x4246	T8_CTL0	T8 Ch.0 Control Register	Sets the timer mode and starts/stops the timer.
	0x4248	T8_INT0	T8 Ch.0 Interrupt Control Register	Controls the interrupt.
Interrupt controller (16-bit device)	0x4306	ITC_LV0	Interrupt Level Setup Register 0	Sets the P0 interrupt level.
	0x4308	ITC_LV1	Interrupt Level Setup Register 1	Sets the CT interrupt level.
	0x430a	ITC_LV2	Interrupt Level Setup Register 2	Sets the RTC interrupt level.
	0x430c	ITC_LV3	Interrupt Level Setup Register 3	Sets the LCD and T16A2 Ch.0 interrupt levels.
	0x4310	ITC_LV5	Interrupt Level Setup Register 5	Sets the T8 Ch.0 interrupt level.
	0x4312	ITC_LV6	Interrupt Level Setup Register 6	Sets the UART Ch.0 interrupt level.
	0x4314	ITC_LV7	Interrupt Level Setup Register 7	Sets the SPI Ch.0 interrupt level.
SPI Ch.0 (16-bit device)	0x4320	SPI_ST0	SPI Ch.0 Status Register	Indicates transfer and buffer statuses.
	0x4322	SPI_TXD0	SPI Ch.0 Transmit Data Register	Transmit data
	0x4324	SPI_RXD0	SPI Ch.0 Receive Data Register	Receive data
	0x4326	SPI_CTL0	SPI Ch.0 Control Register	Sets the SPI mode and enables data transfer.

Internal Peripheral Circuit Area 2 (0x5000–0x5fff)

Peripheral	Address	Register name		Function
Clock timer (8-bit device)	0x5000	CT_CTL	Clock Timer Control Register	Resets and starts/stops the timer.
	0x5001	CT_CNT	Clock Timer Counter Register	Counter data
	0x5002	CT_IMSK	Clock Timer Interrupt Mask Register	Enables/disables interrupt.
	0x5003	CT_IFLG	Clock Timer Interrupt Flag Register	Indicates/resets interrupt occurrence status.
Watchdog timer (8-bit device)	0x5040	WDT_CTL	Watchdog Timer Control Register	Resets and starts/stops the timer.
	0x5041	WDT_ST	Watchdog Timer Status Register	Sets the timer mode and indicates NMI status.
Clock generator /Theoretical regulation (8-bit device) (T16A2, UART, SND, LCD, TR)	0x5060	CLG_SRC	Clock Source Select Register	Selects the clock source.
	0x5061	CLG_CTL	Oscillation Control Register	Controls oscillation.
	0x5064	CLG_FOUTA	FOUTA Control Register	Controls FOUTA clock output.
	0x5065	CLG_FOUTB	FOUTB Control Register	Controls FOUTB clock output.
	0x5068	T16A_CLK0	T16A2 Clock Control Register Ch.0	Controls the T16A2 Ch.0 clock.
	0x506c	UART_CLK0	UART Ch.0 Clock Control Register	Selects the baud rate generator clock.
	0x506e	SND_CLK	SND Clock Control Register	Controls the SND clock.
	0x5070	LCD_TCLK	LCD Timing Clock Control Register	Controls the LCD clock.
	0x5071	LCD_BCLK	LCD Booster Clock Control Register	Controls the LCD booster clock.
	0x5078	TR_CTL	TR Control Register	Controls theoretical regulation.
	0x5079	TR_VAL	TR Value Register	Sets a regulation value.
	0x507d	CLG_WAIT	Oscillation Stabilization Wait Control Register	Controls oscillation stabilization waiting time.
	0x5080	CLG_PCLK	PCLK Control Register	Controls the PCLK supply.
	0x5081	CLG_CCLK	CCLK Control Register	Configures the CCLK division ratio.
LCD driver (8-bit device)	0x5063	LCD_TCLK	LCD Clock Select Register	Selects the LCD clock.
	0x50a0	LCD_DCTL	LCD Display Control Register	Controls the LCD display.
	0x50a2	LCD_CCTL	LCD Clock Control Register	Controls the LCD drive duty.
	0x50a3	LCD_VREG	LCD Voltage Regulator Control Register	Controls the LCD drive voltage regulator.
	0x50a5	LCD_IMSK	LCD Interrupt Mask Register	Enables/disables interrupts.
	0x50a6	LCD_IFLG	LCD Interrupt Flag Register	Indicates/resets interrupt occurrence status.
SVD circuit (8-bit device)	0x5100	SVD_EN	SVD Enable Register	Enables/disables the SVD operation.
	0x5101	SVD_CMP	SVD Comparison Voltage Register	Sets the comparison voltage.
	0x5102	SVD_RSLT	SVD Detection Result Register	Voltage detection results
Power generator (8-bit device)	0x5120	VD1_CTL	V_{D1} Control Register	Controls the V_{D1} regulator heavy load protection mode.
Sound generator (8-bit device)	0x5180	SND_CTL	SND Control Register	Controls buzzer outputs.
	0x5181	SND_BZFQ	Buzzer Frequency Control Register	Sets the buzzer frequency.
	0x5182	SND_BZDT	Buzzer Duty Ratio Control Register	Sets the buzzer signal duty ratio.

Peripheral	Address	Register name		Function
P port & port MUX (8-bit device)	0x5200	P0_IN	P0 Port Input Data Register	P0 port input data
	0x5201	P0_OUT	P0 Port Output Data Register	P0 port output data
	0x5202	P0_OEN	P0 Port Output Enable Register	Enables P0 port outputs.
	0x5203	P0_PU	P0 Port Pull-up Control Register	Controls the P0 port pull-up resistor.
	0x5205	P0_IMSK	P0 Port Interrupt Mask Register	Enables P0 port interrupts.
	0x5206	P0_EDGE	P0 Port Interrupt Edge Select Register	Selects the signal edge for generating P0 port interrupts.
	0x5207	P0_IFLG	P0 Port Interrupt Flag Register	Indicates/resets the P0 port interrupt occurrence status.
	0x5208	P0_CHAT	P0 Port Chattering Filter Control Register	Controls the P0 port chattering filter.
	0x5209	P0_KRST	P0 Port Key-Entry Reset Configuration Register	Configures the P0 port key-entry reset function.
	0x520a	P0_IEN	P0 Port Input Enable Register	Enables P0 port inputs.
	0x5210	P1_IN	P1 Port Input Data Register	P1 port input data
	0x5211	P1_OUT	P1 Port Output Data Register	P1 port output data
	0x5212	P1_OEN	P1 Port Output Enable Register	Enables P1 port outputs.
	0x5213	P1_PU	P1 Port Pull-up Control Register	Controls the P1 port pull-up resistor.
	0x521a	P1_IEN	P1 Port Input Enable Register	Enables P1 port inputs.
	0x52a0	P00_03PMUX	P0[3:0] Port Function Select Register	Selects the P0[3:0] port functions.
	0x52a1	P04_07PMUX	P0[7:4] Port Function Select Register	Selects the P0[7:4] port functions.
	0x52a2	P10_13PMUX	P1[3:0] Port Function Select Register	Selects the P1[3:0] port functions.
MISC registers (16-bit device)	0x5322	MISC_ DMODE2	Debug Mode Control Register 2	Enables peripheral operations in debug mode (except PCLK).
	0x5324	MISC_PROT	MISC Protect Register	Enables writing to the MISC registers.
	0x5326	MISC_IRAMSZ	IRAM Size Select Register	Selects the IRAM size.
	0x5328	MISC_TTBRL	Vector Table Address Low Register	Sets vector table address.
	0x532a	MISC_TTBRH	Vector Table Address High Register	
	0x532c	MISC_PSR	PSR Register	Indicates the S1C17 Core PSR values.
16-bit PWM timer Ch.0 (16-bit device)	0x5400	T16A_CTL0	T16A Counter Ch.0 Control Register	Controls the counter.
	0x5402	T16A_TC0	T16A Counter Ch.0 Data Register	Counter data
	0x5404	T16A_CCCTL0	T16A Comparator/Capture Ch.0 Control Register	Controls the comparator/capture block and TOUT.
	0x5406	T16A_CCA0	T16A Compare/Capture Ch.0 A Data Register	Compare A/capture A data
	0x5408	T16A_CCB0	T16A Compare/Capture Ch.0 B Data Register	Compare B/capture B data
	0x540a	T16A_IEN0	T16A Compare/Capture Ch.0 Interrupt Enable Register	Enables/disables interrupts.
	0x540c	T16A_IFLG0	T16A Compare/Capture Ch.0 Interrupt Flag Register	Displays/sets interrupt occurrence status.
Flash controller (16-bit device)	0x54b0	FLASHC_ WAIT	FLASHC Read Wait Control Register	Sets Flash read wait cycle.
Real-time clock (16-bit device)	0x56c0	RTC_CTL	RTC Control Register	Controls the RTC.
	0x56c2	RTC_IEN	RTC Interrupt Enable Register	Enables/disables interrupts.
	0x56c4	RTC_IFLG	RTC Interrupt Flag Register	Displays/sets interrupt occurrence status.
	0x56c6	RTC_MS	RTC Minute/Second Counter Register	Minute/second counter data
	0x56c8	RTC_H	RTC Hour Counter Register	Hour counter data

Core I/O Reserved Area (0xffff84–0xffffd0)

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

Note: Addresses marked as “Reserved” or unused peripheral circuit areas not marked in the table must not be accessed by application programs.

0x4100–0x4107, 0x506c UART (with IrDA) Ch.0

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
UART Ch.0 Status Register (UART_ST0)	**0x4100** (8 bits)	D7	**TRED**	End of transmission flag	1 Completed / 0 Not completed	0	R/W	Reset by writing 1.
		D6	**FER**	Framing error flag	1 Error / 0 Normal	0	R/W	
		D5	**PER**	Parity error flag	1 Error / 0 Normal	0	R/W	
		D4	**OER**	Overrun error flag	1 Error / 0 Normal	0	R/W	
		D3	**RD2B**	Second byte receive flag	1 Ready / 0 Empty	0	R	
		D2	**TRBS**	Transmit busy flag	1 Busy / 0 Idle	0	R	Shift register status
		D1	**RDRY**	Receive data ready flag	1 Ready / 0 Empty	0	R	
		D0	**TDBE**	Transmit data buffer empty flag	1 Empty / 0 Not empty	1	R	
UART Ch.0 Transmit Data Register (UART_TXD0)	**0x4101** (8 bits)	D7–0	**TXD[7:0]**	Transmit data TXD7(6) = MSB TXD0 = LSB	0x0 to 0xff (0x7f)	0x0	R/W	
UART Ch.0 Receive Data Register (UART_RXD0)	**0x4102** (8 bits)	D7–0	**RXD[7:0]**	Receive data in the receive data buffer RXD7(6) = MSB RXD0 = LSB	0x0 to 0xff (0x7f)	0x0	R	Older data in the buffer is read out first.
UART Ch.0 Mode Register (UART_MOD0)	**0x4103** (8 bits)	D7–5	**–**	reserved	–	–	–	0 when being read.
		D4	**CHLN**	Character length select	1 8 bits / 0 7 bits	0	R/W	
		D3	**PREN**	Parity enable	1 With parity / 0 No parity	0	R/W	
		D2	**PMD**	Parity mode select	1 Odd / 0 Even	0	R/W	
		D1	**STPB**	Stop bit select	1 2 bits / 0 1 bit	0	R/W	
		D0	**–**	reserved	–	–	–	0 when being read.
UART Ch.0 Control Register (UART_CTL0)	**0x4104** (8 bits)	D7	**TEIEN**	End of transmission int. enable	1 Enable / 0 Disable	0	R/W	
		D6	**REIEN**	Receive error int. enable	1 Enable / 0 Disable	0	R/W	
		D5	**RIEN**	Receive buffer full int. enable	1 Enable / 0 Disable	0	R/W	
		D4	**TIEN**	Transmit buffer empty int. enable	1 Enable / 0 Disable	0	R/W	
		D3–2	**–**	reserved	–	–	–	0 when being read.
		D1	**RBFI**	Receive buffer full int. condition setup	1 2 bytes / 0 1 byte	0	R/W	
		D0	**RXEN**	UART enable	1 Enable / 0 Disable	0	R/W	
UART Ch.0 Expansion Register (UART_EXP0)	**0x4105** (8 bits)	D7–1	**–**	reserved	–	–	–	0 when being read.
		D0	**IRMD**	IrDA mode select	1 On / 0 Off	0	R/W	
UART Ch.0 Baud Rate Register (UART_BR0)	**0x4106** (8 bits)	D7–0	**BR[7:0]**	Baud rate setting	0x0 to 0xff	0x0	R/W	
UART Ch.0 Fine Mode Register (UART_FMD0)	**0x4107** (8 bits)	D7–4	**–**	reserved	–	–	–	0 when being read.
		D3–0	**FMD[3:0]**	Fine mode setup	0x0 to 0xf	0x0	R/W	Set a number of times to insert delay into a 16-underflow period.
UART Ch.0 Clock Control Register (UART_CLK0)	**0x506c** (8 bits)	D7–6	**–**	reserved	–	–	–	0 when being read.
		D5–4	**UTCLKD [1:0]**	Clock division ratio select	UTCLKD[1:0] / Division ratio 0x3 / 1/8 0x2 / 1/4 0x1 / 1/2 0x0 / 1/1	0x0	R/W	When the clock source is OSC3B or OSC3A
		D3–2	**UTCLKSRC [1:0]**	Clock source select	UTCLKSRC [1:0] / Clock source 0x3 / External clock 0x2 / OSC3A 0x1 / OSC1 0x0 / OSC3B	0x0	R/W	
		D1	**–**	reserved	–	–	–	0 when being read.
		D0	**UTCLKE**	UART clock enable	1 Enable / 0 Disable	0	R/W	

0x4240–0x4248 8-bit Timer Ch.0

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T8 Ch.0 Count Clock Select Register (T8_CLK0)	0x4240 (16 bits)	D15–4	–	reserved	–	–	–	0 when being read.
		D3–0	DF[3:0]	Count clock division ratio select	DF[3:0] / Division ratio: 0xf reserved 0xe 1/16384 0xd 1/8192 0xc 1/4096 0xb 1/2048 0xa 1/1024 0x9 1/512 0x8 1/256 0x7 1/128 0x6 1/64 0x5 1/32 0x4 1/16 0x3 1/8 0x2 1/4 0x1 1/2 0x0 1/1	0x0	R/W	Source clock = PCLK
T8 Ch.0 Reload Data Register (T8_TR0)	0x4242 (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	TR[7:0]	Reload data TR7 = MSB TR0 = LSB	0x0 to 0xff	0x0	R/W	
T8 Ch.0 Counter Data Register (T8_TC0)	0x4244 (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	TC[7:0]	Counter data TC7 = MSB TC0 = LSB	0x0 to 0xff	0xff	R	
T8 Ch.0 Control Register (T8_CTL0)	0x4246 (16 bits)	D15–5	–	reserved	–	–	–	Do not write 1.
		D4	TRMD	Count mode select	1 One shot / 0 Repeat	0	R/W	
		D3–2	–	reserved	–	–	–	0 when being read.
		D1	PRESER	Timer reset	1 Reset / 0 Ignored	0	W	
		D0	PRUN	Timer run/stop control	1 Run / 0 Stop	0	R/W	
T8 Ch.0 Interrupt Control Register (T8_INT0)	0x4248 (16 bits)	D15–9	–	reserved	–	–	–	0 when being read.
		D8	T8IE	T8 interrupt enable	1 Enable / 0 Disable	0	R/W	
		D7–1	–	reserved	–	–	–	0 when being read.
		D0	T8IF	T8 interrupt flag	1 Cause of interrupt occurred / 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.

0x4306–0x4314 Interrupt Controller

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Interrupt Level Setup Register 0 (ITC_LV0)	0x4306 (16 bits)	D15–3	–	reserved	–	–	–	0 when being read.
		D2–0	ILV0[2:0]	P0 interrupt level	0 to 7	0x0	R/W	
Interrupt Level Setup Register 1 (ITC_LV1)	0x4308 (16 bits)	D15–11	–	reserved	–	–	–	0 when being read.
		D10–8	ILV3[2:0]	CT interrupt level	0 to 7	0x0	R/W	
		D7–0	–	reserved	–	–	–	0 when being read.
Interrupt Level Setup Register 2 (ITC_LV2)	0x430a (16 bits)	D15–3	–	reserved	–	–	–	0 when being read.
		D2–0	ILV4[2:0]	RTC interrupt level	0 to 7	0x0	R/W	
Interrupt Level Setup Register 3 (ITC_LV3)	0x430c (16 bits)	D15–11	–	reserved	–	–	–	0 when being read.
		D10–8	ILV7[2:0]	T16A2 Ch.0 interrupt level	0 to 7	0x0	R/W	
		D7–3	–	reserved	–	–	–	0 when being read.
		D2–0	ILV6[2:0]	LCD interrupt level	0 to 7	0x0	R/W	
Interrupt Level Setup Register 5 (ITC_LV5)	0x4310 (16 bits)	D15–3	–	reserved	–	–	–	0 when being read.
		D2–0	ILV10[2:0]	T8 Ch.0 interrupt level	0 to 7	0x0	R/W	
Interrupt Level Setup Register 6 (ITC_LV6)	0x4312 (16 bits)	D15–3	–	reserved	–	–	–	0 when being read.
		D2–0	ILV12[2:0]	UART Ch.0 interrupt level	0 to 7	0x0	R/W	
Interrupt Level Setup Register 7 (ITC_LV7)	0x4314 (16 bits)	D15–3	–	reserved	–	–	–	0 when being read.
		D2–0	ILV14[2:0]	SPI Ch.0 interrupt level	0 to 7	0x0	R/W	

0x4320–0x4326 SPI Ch.0

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SPI Ch.0 Status Register (SPI_ST0)	0x4320 (16 bits)	D15–3	–	reserved	–	–	–	0 when being read.
		D2	SPBSY	Transfer busy flag (master)	1 Busy / 0 Idle	0	R	
				ss signal low flag (slave)	1 ss = L / 0 ss = H			
		D1	SPRBF	Receive data buffer full flag	1 Full / 0 Not full	0	R	
		D0	SPTBE	Transmit data buffer empty flag	1 Empty / 0 Not empty	1	R	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SPI Ch.0 Transmit Data Register (SPI_TXD0)	**0x4322** (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	**SPTDB[7:0]**	SPI transmit data buffer SPTDB7 = MSB SPTDB0 = LSB	0x0 to 0xff	0x0	R/W	
SPI Ch.0 Receive Data Register (SPI_RXD0)	**0x4324** (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	**SPRDB[7:0]**	SPI receive data buffer SPRDB7 = MSB SPRDB0 = LSB	0x0 to 0xff	0x0	R	
SPI Ch.0 Control Register (SPI_CTL0)	**0x4326** (16 bits)	D15–10	–	reserved	–	–	–	0 when being read.
		D9	**MCLK**	SPI clock source select	1 T8 Ch.0 0 PCLK/4	0	R/W	
		D8	**MLSB**	LSB/MSB first mode select	1 LSB 0 MSB	0	R/W	
		D7–6	–	reserved	–	–	–	0 when being read.
		D5	**SPRIE**	Receive data buffer full int. enable	1 Enable 0 Disable	0	R/W	
		D4	**SPTIE**	Transmit data buffer empty int. enable	1 Enable 0 Disable	0	R/W	
		D3	**CPHA**	Clock phase select	1 Data out 0 Data in	0	R/W	These bits must be
		D2	**CPOL**	Clock polarity select	1 Active L 0 Active H	0	R/W	set before setting
		D1	**MSSL**	Master/slave mode select	1 Master 0 Slave	0	R/W	SPEN to 1.
		D0	**SPEN**	SPI enable	1 Enable 0 Disable	0	R/W	

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	

0x5040–0x5041 Watchdog Timer

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Watchdog Timer Control Register (WDT_CTL)	**0x5040** (8 bits)	D7–5	–	reserved	–	–	–	0 when being read.
		D4	**WDTRST**	Watchdog timer reset	1 Reset 0 Ignored	0	W	
		D3–0	**WDTRUN[3:0]**	Watchdog timer run/stop control	Other than 1010 Run / 1010 Stop	1010	R/W	
Watchdog Timer Status Register (WDT_ST)	**0x5041** (8 bits)	D7–2	–	reserved	–	–	–	0 when being read.
		D1	**WDTMD**	NMI/Reset mode select	1 Reset 0 NMI	0	R/W	
		D0	**WDTST**	NMI status	1 NMI occurred 0 Not occurred	0	R	

0x5060–0x5081 Clock Generator

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Clock Source Select Register (CLG_SRC)	**0x5060** (8 bits)	D7–6	**OSC3B FSEL[1:0]**	OSC3B frequency select	OSC3BFSEL[1:0] / Frequency 0x3 reserved 0x2 500 kHz 0x1 1 MHz 0x0 2 MHz	0x0	R/W	
		D5	–	reserved	–	–	–	0 when being read.
		D4	**OSC1SEL**	OSC1 source select	1 OSC1B 0 OSC1A	1	R/W	
		D3–2	–	reserved	–	–	–	0 when being read.
		D1–0	**CLKSRC[1:0]**	System clock source select	CLKSRC[1:0] / Clock source 0x3 reserved 0x2 OSC3A 0x1 OSC1 0x0 OSC3B	0x0	R/W	
Oscillation Control Register (CLG_CTL)	**0x5061** (8 bits)	D7–3	–	reserved	–	–	–	0 when being read.
		D2	**OSC3BEN**	OSC3B enable	1 Enable 0 Disable	1	R/W	
		D1	**OSC1EN**	OSC1 enable	1 Enable 0 Disable	0	R/W	
		D0	**OSC3AEN**	OSC3A enable	1 Enable 0 Disable	0	R/W	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
FOUTA Control Register (CLG_FOUTA)	**0x5064** (8 bits)	D7	**–**	reserved	–	–	–	0 when being read.
		D6–4	**FOUTAD [2:0]**	FOUTA clock division ratio select	FOUTAD[2:0] / Division ratio 0x7 1/128 0x6 1/64 0x5 1/32 0x4 1/16 0x3 1/8 0x2 1/4 0x1 1/2 0x0 1/1	0x0	R/W	
		D3–2	**FOUTASRC [1:0]**	FOUTA clock source select	FOUTASRC[1:0] / Clock source 0x3 reserved 0x2 OSC3A 0x1 OSC1 0x0 OSC3B	0x0	R/W	
		D1	**–**	reserved	–	–	–	0 when being read.
		D0	**FOUTAE**	FOUTA output enable	1 Enable 0 Disable	0	R/W	
FOUTB Control Register (CLG_FOUTB)	**0x5065** (8 bits)	D7	**–**	reserved	–	–	–	0 when being read.
		D6–4	**FOUTBD [2:0]**	FOUTB clock division ratio select	FOUTBD[2:0] / Division ratio 0x7 1/128 0x6 1/64 0x5 1/32 0x4 1/16 0x3 1/8 0x2 1/4 0x1 1/2 0x0 1/1	0x0	R/W	
		D3–2	**FOUTBSRC [1:0]**	FOUTB clock source select	FOUTBSRC[1:0] / Clock source 0x3 reserved 0x2 OSC3A 0x1 OSC1 0x0 OSC3B	0x0	R/W	
		D1	**–**	reserved	–	–	–	0 when being read.
		D0	**FOUTBE**	FOUTB output enable	1 Enable 0 Disable	0	R/W	
Oscillation Stabilization Wait Control Register (CLG_WAIT)	**0x507d** (8 bits)	D7–6	**OSC3BWT [1:0]**	OSC3B stabilization wait cycle select	OSC3BWT[1:0] / Wait cycle 0x3 8 cycles 0x2 16 cycles 0x1 32 cycles 0x0 64 cycles	0x0	R/W	
		D5–4	**OSC3AWT [1:0]**	OSC3A stabilization wait cycle select	OSC3AWT[1:0] / Wait cycle 0x3 128 cycles 0x2 256 cycles 0x1 512 cycles 0x0 1024 cycles	0x0	R/W	
		D3–2	**OSC1BWT [1:0]**	OSC1B stabilization wait cycle select	OSC1BWT[1:0] / Wait cycle 0x3 8 cycles 0x2 16 cycles 0x1 32 cycles 0x0 64 cycles	0x0	R/W	
		D1–0	**OSC1AWT [1:0]**	OSC1A stabilization wait cycle select	OSC1AWT[1:0] / Wait cycle 0x3 2048 cycles 0x2 4096 cycles 0x1 8192 cycles 0x0 16384 cycles	0x0	R/W	
PCLK Control Register (CLG_PCLK)	**0x5080** (8 bits)	D7–2	**–**	reserved	–	–	–	0 when being read.
		D1–0	**PCKEN[1:0]**	PCLK enable	PCKEN[1:0] / PCLK supply 0x3 Enable 0x2 Not allowed 0x1 Not allowed 0x0 Disable	0x3	R/W	
CCLK Control Register (CLG_CCLK)	**0x5081** (8 bits)	D7–2	**–**	reserved	–	–	–	0 when being read.
		D1–0	**CCLKGR[1:0]**	CCLK clock gear ratio select	CCLKGR[1:0] / Gear ratio 0x3 1/8 0x2 1/4 0x1 1/2 0x0 1/1	0x0	R/W	

0x5078–0x5079 Theoretical Regulation Circuit

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

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

0x5070–0x5071, 0x50a0–0x50a6 LCD Driver

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
LCD Timing Clock Select Register (LCD_TCLK)	**0x5070** (8 bits)	D7–6	**–**	reserved	–	–	–	0 when being read.
		D5–4	**LCDTCLKD [1:0]**	LCD clock division ratio select	LCDTCLKD [1:0] / Division ratio: OSC3B/OSC3A / OSC1 0x3 / 1/8192 / 1/64 0x2 / 1/4096 / 1/64 0x1 / 1/2048 / 1/64 0x0 / 1/1024 / 1/64	0x0	R/W	
		D3–2	**LCDTCLK SRC[1:0]**	LCD clock source select	LCDTCLK SRC[1:0] / Clock source 0x3 / reserved 0x2 / OSC3A 0x1 / OSC1 0x0 / OSC3B	0x0	R/W	
		D1	**–**	reserved	–	–	–	0 when being read.
		D0	**LCDTCLKE**	LCD clock enable	1 Enable 0 Disable	0	R/W	
LCD Booster Clock Control Register (LCD_BCLK)	**0x5071** (8 bits)	D7	**–**	reserved	–	–	–	0 when being read.
		D6–4	**LCDBCLKD [2:0]**	LCD booster clock division ratio select	LCDB CLKD [2:0] / Division ratio: OSC3B / OSC3A / OSC1 0x7 / – / – / – 0x6 / 1/4096 / 1/8192 / – 0x5 / 1/2048 / 1/4096 / – 0x4 / 1/1024 / 1/2048 / – 0x3 / 1/512 / 1/1024 / 1/64 0x2 / 1/256 / 1/512 / 1/32 0x1 / 1/128 / 1/256 / 1/16 0x0 / 1/64 / 1/128 / 1/8	0x0	R/W	
		D3–2	**LCDBCLK SRC[1:0]**	LCD Booster clock source select	LCDBCLK SRC[1:0] / Clock source 0x3 / reserved 0x2 / OSC3A 0x1 / OSC1 0x0 / OSC3B	0x0	R/W	
		D1	**–**	reserved	–	–	–	0 when being read.
		D0	**LCDBCLKE**	LCD Booster clock enable	1 Enable 0 Disable	0	R/W	
LCD Display Control Register (LCD_DCTL)	**0x50a0** (8 bits)	D7–5	**–**	reserved	–	–	–	0 when being read.
		D4	**DSPREV**	Reverse display control	1 Normal 0 Reverse	1	R/W	
		D3–2	**–**	reserved	–	–	–	0 when being read.
		D1–0	**DSPC[1:0]**	LCD display control	DSPC[1:0] / Display 0x3 / All off 0x2 / All on 0x1 / Normal display 0x0 / Display off	0x0	R/W	
LCD Clock Control Register (LCD_CCTL)	**0x50a2** (8 bits)	D7–6	**FRMCNT[1:0]**	Frame frequency control	FRMCNT[1:0] / Division ratio 0x3 / 1/16 0x2 / 1/12 0x1 / 1/8 0x0 / 1/4	0x1	R/W	Source clock: LCLK
		D5–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**LDUTY[2:0]**	LCD duty select	LDUTY[2:0] / Duty 0x7–0x4 / reserved 0x3 / 1/4 0x2 / 1/3 0x1 / 1/2 0x0 / Static	0x3	R/W	
LCD Voltage Regulator Control Register (LCD_VREG)	**0x50a3** (8 bits)	D7–5	**–**	reserved	–	–	–	0 when being read.
		D4	**LHVLD**	Vc heavy load protection mode	1 On 0 Off	0	R/W	
		D3–1	**–**	reserved	–	–	–	0 when being read.
		D0	**VCSEL**	Reference voltage select	1 V_{C2} 0 V_{C1}	0	R/W	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
LCD Interrupt Mask Register (LCD_IMSK)	0x50a5 (8 bits)	D7–1	–	reserved	–	–	–	0 when being read.
		D0	IFRMEN	Frame signal interrupt enable	1 Enable 0 Disable	0	R/W	
LCD Interrupt Flag Register (LCD_IFLG)	0x50a6 (8 bits)	D7–1	–	reserved	–	–	–	0 when being read.
		D0	IFRMFLG	Frame signal interrupt flag	1 Occurred 0 Not occurred	0	R/W	Reset by writing 1.

0x5100–0x5102 SVD Circuit

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SVD Enable Register (SVD_EN)	0x5100 (8 bits)	D7–1	–	reserved	–	–	–	0 when being read.
		D0	SVDEN	SVD enable	1 Enable 0 Disable	0	R/W	
SVD Comparison Voltage Register (SVD_CMP)	0x5101 (8 bits)	D7–5	–	reserved	–	–	–	0 when being read.
		D4–0	SVDC[4:0]	SVD comparison voltage select	SVDC[4:0] / Voltage: 0x1f–0x1b reserved; 0x1a 3.20 V; 0x19 3.10 V; 0x18 3.00 V; 0x17 2.90 V; 0x16 2.80 V; 0x15 2.70 V; 0x14 2.60 V; 0x13 2.50 V; 0x12 2.40 V; 0x11 2.30 V; 0x10 2.20 V; 0xf 2.10 V; 0xe 2.00 V; 0xd–0x0 reserved	0x0	R/W	
SVD Detection Result Register (SVD_RSLT)	0x5102 (8 bits)	D7–1	–	reserved	–	–	–	0 when being read.
		D0	SVDDT	SVD detection result	1 Low 0 Normal	×	R	

0x5120 Power Generator

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

0x506e, 0x5180–0x5182 Sound Generator

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SND Clock Control Register (SND_CLK)	0x506e (8 bits)	D7–1	–	reserved	–	–	–	0 when being read.
		D0	SNDCLKE	SND clock enable	1 Enable 0 Disable	0	R/W	
SND Control Register (SND_CTL)	0x5180 (8 bits)	D7–6	–	reserved	–	–	–	0 when being read.
		D5–4	BZTM[1:0]	Buzzer envelope time/one-shot output time select	BZTM[1:0] / Time: 0x3 125 ms; 0x2 62.5 ms; 0x1 31.25 ms; 0x0 15.63 ms	0x0	R/W	
		D3–2	BZMD[1:0]	Buzzer mode select	BZMD[1:0] / Mode: 0x3 reserved; 0x2 Envelope; 0x1 One-shot; 0x0 Normal	0x0	R/W	
		D1	–	reserved	–	–	–	0 when being read.
		D0	BZEN	Buzzer output control	1 On/Trigger 0 Off	0	R/W	
Buzzer Frequency Control Register (SND_BZFQ)	0x5181 (8 bits)	D7–3	–	reserved	–	–	–	0 when being read.
		D2–0	BZFQ[2:0]	Buzzer frequency select	BZFQ[2:0] / Frequency: 0x7 1170.3 Hz; 0x6 1365.3 Hz; 0x5 1638.4 Hz; 0x4 2048.0 Hz; 0x3 2340.6 Hz; 0x2 2730.7 Hz; 0x1 3276.8 Hz; 0x0 4096.0 Hz	0x0	R/W	

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
Buzzer Duty Ratio Control Register (SND_BZDT)	**0x5182** (8 bits)	D7–3	**–**	reserved	–		–	–	0 when being read.
		D2–0	**BZDT[2:0]**	Buzzer duty ratio select	BZDT[2:0]	Duty (volume)	0x0	R/W	
					0x7	Level 8 (Min.)			
					:	:			
					0x0	Level 1 (Max.)			

0x5200–0x52a2 P Port & Port MUX

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
P0 Port Input Data Register (P0_IN)	**0x5200** (8 bits)	D7–0	**P0IN[7:0]**	P0[7:0] port input data	1 1 (H)	0 0 (L)	×	R	
P0 Port Output Data Register (P0_OUT)	**0x5201** (8 bits)	D7–0	**P0OUT[7:0]**	P0[7:0] port output data	1 1 (H)	0 0 (L)	0	R/W	
P0 Port Output Enable Register (P0_OEN)	**0x5202** (8 bits)	D7–0	**P0OEN[7:0]**	P0[7:0] port output enable	1 Enable	0 Disable	0	R/W	
P0 Port Pull-up Control Register (P0_PU)	**0x5203** (8 bits)	D7–0	**P0PU[7:0]**	P0[7:0] port pull-up enable	1 Enable	0 Disable	1 (0xff)	R/W	
P0 Port Interrupt Mask Register (P0_IMSK)	**0x5205** (8 bits)	D7–0	**P0IE[7:0]**	P0[7:0] port interrupt enable	1 Enable	0 Disable	0	R/W	
P0 Port Interrupt Edge Select Register (P0_EDGE)	**0x5206** (8 bits)	D7–0	**P0EDGE[7:0]**	P0[7:0] port interrupt edge select	1 Falling edge	0 Rising edge	0	R/W	
P0 Port Interrupt Flag Register (P0_IFLG)	**0x5207** (8 bits)	D7–0	**P0IF[7:0]**	P0[7:0] port interrupt flag	1 Cause of interrupt occurred	0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
P0 Port Chattering Filter Control Register (P0_CHAT)	**0x5208** (8 bits)	D7	**–**	reserved	–		–	–	0 when being read.
		D6–4	**P0CF2[2:0]**	P0[7:4] chattering filter time	P0CF2[2:0]	Filter time	0	R/W	
					0x7	16384/f_{PCLK}	0x0	R/W	
					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			
		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/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			
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			
P0 Port Input Enable Register (P0_IEN)	**0x520a** (8 bits)	D7–0	**P0IEN[7:0]**	P0[7:0] port input enable	1 Enable	0 Disable	1 (0xff)	R/W	
P1 Port Input Data Register (P1_IN)	**0x5210** (8 bits)	D7–4	**–**	reserved	–		–	–	0 when being read.
		D3–0	**P1IN[3:0]**	P1[3:0] port input data	1 1 (H)	0 0 (L)	×	R	
P1 Port Output Data Register (P1_OUT)	**0x5211** (8 bits)	D7–4	**–**	reserved	–		–	–	0 when being read.
		D3–0	**P1OUT[3:0]**	P1[3:0] port output data	1 1 (H)	0 0 (L)	0	R/W	
P1 Port Output Enable Register (P1_OEN)	**0x5212** (8 bits)	D7–4	**–**	reserved	–		–	–	0 when being read.
		D3–0	**P1OEN[3:0]**	P1[3:0] port output enable	1 Enable	0 Disable	0	R/W	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
P1 Port Pull-up Control Register (P1_PU)	**0x5213** (8 bits)	D7–4	**–**	reserved	–	–	–	0 when being read.
		D3–0	**P1PU[3:0]**	P1[3:0] port pull-up enable	1 Enable 0 Disable	1 (0xf)	R/W	
P1 Port Input Enable Register (P1_IEN)	**0x521a** (8 bits)	D7–4	**–**	reserved	–	–	–	0 when being read.
		D3–0	**P1IEN[3:0]**	P1[3:0] port input enable	1 Enable 0 Disable	1 (0xf)	R/W	
P0[3:0] Port Function Select Register (P00_03PMUX)	**0x52a0** (8 bits)	D7–6	**P03MUX[1:0]**	P03 port function select	P03MUX[1:0] Function 0x3 LFRO 0x2 REGMON 0x1 EXCL0 0x0 P03	0x0	R/W	
		D5–4	**P02MUX[1:0]**	P02 port function select	P02MUX[1:0] Function 0x3 REGMON 0x2 FOUTA 0x1 SCLK0 0x0 P02	0x0	R/W	
		D3–2	**P01MUX[1:0]**	P01 port function select	P01MUX[1:0] Function 0x3 reserved 0x2 reserved 0x1 SOUT0 0x0 P01	0x0	R/W	
		D1–0	**P00MUX[1:0]**	P00 port function select	P00MUX[1:0] Function 0x3 reserved 0x2 reserved 0x1 SIN0 0x0 P00	0x0	R/W	
P0[7:4] Port Function Select Register (P04_07PMUX)	**0x52a1** (8 bits)	D7–6	**P07MUX[1:0]**	P07 port function select	P07MUX[1:0] Function 0x3 reserved 0x2 SDO0 0x1 #BZ 0x0 P07	0x0	R/W	
		D5–4	**P06MUX[1:0]**	P06 port function select	P06MUX[1:0] Function 0x3 reserved 0x2 SDI0 0x1 BZ 0x0 P06	0x0	R/W	
		D3–2	**P05MUX[1:0]**	P05 port function select	P05MUX[1:0] Function 0x3 reserved 0x2 #SPISS0 0x1 TOUTB0/CAPB0 0x0 P05	0x0	R/W	
		D1–0	**P04MUX[1:0]**	P04 port function select	P04MUX[1:0] Function 0x3 reserved 0x2 reserved 0x1 TOUTA0/CAPA0 0x0 P04	0x0	R/W	
P1[3:0] Port Function Select Register (P10_13PMUX)	**0x52a2** (8 bits)	D7–6	**P13MUX[1:0]**	P13 port function select	P13MUX[1:0] Function 0x3 reserved 0x2 reserved 0x1 P13 0x0 DST2	0x0	R/W	
		D5–4	**P12MUX[1:0]**	P12 port function select	P12MUX[1:0] Function 0x3 reserved 0x2 #BZ 0x1 P12 0x0 DSIO	0x0	R/W	
		D3–2	**P11MUX[1:0]**	P11 port function select	P11MUX[1:0] Function 0x3 reserved 0x2 BZ 0x1 P11 0x0 DCLK	0x0	R/W	
		D1–0	**P10MUX[1:0]**	P10 port function select	P10MUX[1:0] Function 0x3 reserved 0x2 SPICLK0 0x1 FOUTB 0x0 P10	0x0	R/W	

0x4020, 0x5322–0x532c MISC Registers

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Debug Mode Control Register 1 (MISC_DMODE1)	**0x4020** (8 bits)	D7–2	**–**	reserved	–	–	–	0 when being read.
		D1	**DBRUN1**	Run/stop select in debug mode	1 Run 0 Stop	0	R/W	
		D0	**–**	reserved	–	–	–	0 when being read.

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Debug Mode Control Register 2 (MISC_DMODE2)	**0x5322** (16 bits)	D15–1	**–**	reserved	–	–	–	0 when being read.
		D0	**DBRUN2**	Run/stop select in debug mode (except PCLK peripheral circuits)	1 Run / 0 Stop	0	R/W	
MISC Protect Register (MISC_PROT)	**0x5324** (16 bits)	D15–0	**PROT[15:0]**	MISC register write protect	Writing 0x96 removes the write protection of the MISC registers (0x5326–0x532a). Writing another value set the write protection.	0x0	R/W	
IRAM Size Select Register (MISC_IRAMSZ)	**0x5326** (16 bits)	D15–9	**–**	reserved	–	–	–	0 when being read.
		D8	**DBADR**	Debug base address select	1 0x0 / 0 0xfffc00	0	R/W	
		D7	**–**	reserved	–	–	–	0 when being read.
		D6–4	**IRAMACTSZ [2:0]**	IRAM actual size	0x3 (= 2KB)	0x3	R	
		D3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**IRAMSZ[2:0]**	IRAM size select	IRAMSZ[2:0]: 0x5 = 512B; 0x4 = 1KB; 0x3 = 2KB; Other = reserved	0x3	R/W	
Vector Table Address Low Register (MISC_TTBRL)	**0x5328** (16 bits)	D15–8	**TTBR[15:8]**	Vector table base address A[15:8]	0x0–0xff	0x80	R/W	
		D7–0	**TTBR[7:0]**	Vector table base address A[7:0] (fixed at 0)	0x0	0x0	R	
Vector Table Address High Register (MISC_TTBRH)	**0x532a** (16 bits)	D15–8	**–**	reserved	–	–	–	0 when being read.
		D7–0	**TTBR[23:16]**	Vector table base address A[23:16]	0x0–0xff	0x0	R/W	
PSR Register (MISC_PSR)	**0x532c** (16 bits)	D15–8	**–**	reserved	–	–	–	0 when being read.
		D7–5	**PSRIL[2:0]**	PSR interrupt level (IL) bits	0x0 to 0x7	0x0	R	
		D4	**PSRIE**	PSR interrupt enable (IE) bit	1 1 (enable) / 0 0 (disable)	0	R	
		D3	**PSRC**	PSR carry (C) flag	1 1 (set) / 0 0 (cleared)	0	R	
		D2	**PSRV**	PSR overflow (V) flag	1 1 (set) / 0 0 (cleared)	0	R	
		D1	**PSRZ**	PSR zero (Z) flag	1 1 (set) / 0 0 (cleared)	0	R	
		D0	**PSRN**	PSR negative (N) flag	1 1 (set) / 0 0 (cleared)	0	R	

0x5068, 0x5400–0x540c 16-bit PWM Timer Ch.0

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16A Clock Control Register Ch.0 (T16A_CLK0)	**0x5068** (8 bits)	D7–4	**T16ACLKD [3:0]**	Clock division ratio select	T16ACLKD[3:0]: 0xf – / –; 0xe 1/16384 / –; 0xd 1/8192 / –; 0xc 1/4096 / –; 0xb 1/2048 / –; 0xa 1/1024 / –; 0x9 1/512 / F256; 0x8 1/256 / 1/256; 0x7 1/128 / 1/128; 0x6 1/64 / 1/64; 0x5 1/32 / 1/32; 0x4 1/16 / 1/16; 0x3 1/8 / 1/8; 0x2 1/4 / 1/4; 0x1 1/2 / 1/2; 0x0 1/1 / 1/1 (Division ratio: OSC3A or OSC3B / OSC1)	0x0	R/W	F256: Regulated 256 Hz clock
		D3–2	**T16ACLK SRC[1:0]**	Clock source select	T16ACLKSRC[1:0]: 0x3 External clock; 0x2 OSC3A; 0x1 OSC1; 0x0 OSC3B	0x0	R/W	
		D1	**–**	reserved	–	–	–	0 when being read.
		D0	**T16ACLKE**	Count clock enable	1 Enable / 0 Disable	0	R/W	
T16A Counter Ch.0 Control Register (T16A_CTL0)	**0x5400** (16 bits)	D15–7	**–**	reserved	–	–	–	0 when being read.
		D6	**HCM**	Half clock mode enable	1 Enable / 0 Disable	0	R/W	
		D5–4	**–**	reserved	–	–	–	0 when being read.
		D3	**CBUFEN**	Compare buffer enable	1 Enable / 0 Disable	0	R/W	
		D2	**TRMD**	Count mode select	1 One-shot / 0 Repeat	0	R/W	
		D1	**PRESET**	Counter reset	1 Reset / 0 Ignored	0	W	0 when being read.
		D0	**PRUN**	Counter run/stop control	1 Run / 0 Stop	0	R/W	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16A Counter Ch.0 Data Register (T16A_TC0)	**0x5402** (16 bits)	D15–0	**T16ATC [15:0]**	Counter data T16ATC15 = MSB T16ATC0 = LSB	0x0 to 0xffff	0x0	R	
T16A Comparator/ Capture Ch.0 Control Register (T16A_CCCTL0)	**0x5404** (16 bits)	D15–14	**CAPBTRG [1:0]**	Capture B trigger select	CAPBTRG[1:0] / Trigger edge 0x3 / ↑ and ↓ 0x2 / ↓ 0x1 / ↑ 0x0 / None	0x0	R/W	
		D13–12	**TOUTBMD [1:0]**	TOUT B mode select	TOUTBMD[1:0] / Mode 0x3 / cmp B: ↑ or ↓ 0x2 / cmp A: ↑ or ↓ 0x1 / cmp A: ↑, B: ↓ 0x0 / Off	0x0	R/W	
		D11–10	–	reserved	–	–	–	0 when being read.
		D9	**TOUTBINV**	TOUT B invert	1 Invert 0 Normal	0	R/W	
		D8	**CCBMD**	T16A_CCB register mode select	1 Capture 0 Comparator	0	R/W	
		D7–6	**CAPATRG [1:0]**	Capture A trigger select	CAPATRG[1:0] / Trigger edge 0x3 / ↑ and ↓ 0x2 / ↓ 0x1 / ↑ 0x0 / None	0x0	R/W	
		D5–4	**TOUTAMD [1:0]**	TOUT A mode select	TOUTAMD[1:0] / Mode 0x3 / cmp B: ↑ or ↓ 0x2 / cmp A: ↑ or ↓ 0x1 / cmp A: ↑, B: ↓ 0x0 / Off	0x0	R/W	
		D3–2	–	reserved	–	–	–	0 when being read.
		D1	**TOUTAINV**	TOUT A invert	1 Invert 0 Normal	0	R/W	
		D0	**CCAMD**	T16A_CCA register mode select	1 Capture 0 Comparator	0	R/W	
T16A Comparator/ Capture Ch.0 A Data Register (T16A_CCA0)	**0x5406** (16 bits)	D15–0	**CCA[15:0]**	Compare/capture A data CCA15 = MSB CCA0 = LSB	0x0 to 0xffff	0x0	R/W	
T16A Comparator/ Capture Ch.0 B Data Register (T16A_CCB0)	**0x5408** (16 bits)	D15–0	**CCB[15:0]**	Compare/capture B data CCB15 = MSB CCB0 = LSB	0x0 to 0xffff	0x0	R/W	
T16A Comparator/ Capture Ch.0 Interrupt Enable Register (T16A_IEN0)	**0x540a** (16 bits)	D15–6	–	reserved	–	–	–	0 when being read.
		D5	**CAPBOWIE**	Capture B overwrite interrupt enable	1 Enable 0 Disable	0	R/W	
		D4	**CAPAOWIE**	Capture A overwrite interrupt enable	1 Enable 0 Disable	0	R/W	
		D3	**CAPBIE**	Capture B interrupt enable	1 Enable 0 Disable	0	R/W	
		D2	**CAPAIE**	Capture A interrupt enable	1 Enable 0 Disable	0	R/W	
		D1	**CBIE**	Compare B interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	**CAIE**	Compare A interrupt enable	1 Enable 0 Disable	0	R/W	
T16A Comparator/ Capture Ch.0 Interrupt Flag Register (T16A_IFLG0)	**0x540c** (16 bits)	D15–6	–	reserved	–	–	–	0 when being read.
		D5	**CAPBOWIF**	Capture B overwrite interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D4	**CAPAOWIF**	Capture A overwrite interrupt flag		0	R/W	
		D3	**CAPBIF**	Capture B interrupt flag		0	R/W	
		D2	**CAPAIF**	Capture A interrupt flag		0	R/W	
		D1	**CBIF**	Compare B interrupt flag		0	R/W	
		D0	**CAIF**	Compare A interrupt flag		0	R/W	

0x54b0 Flash Controller

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

0x56c0–0x56c8 Real-time Clock

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

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RTC Interrupt Enable Register (RTC_IEN)	**0x56c2** (16 bits)	D15–10	**–**	reserved	–	–	–	0 when being read.
		D9	**INT1DEN**	1-day interrupt enable	1 Enable 0 Disable	0	R/W	
		D8	**INTHDEN**	Half-day interrupt enable	1 Enable 0 Disable	0	R/W	
		D7	**INT1HEN**	1-hour interrupt enable	1 Enable 0 Disable	0	R/W	
		D6	**INT10MEN**	10-minute interrupt enable	1 Enable 0 Disable	0	R/W	
		D5	**INT1MEN**	1-minute interrupt enable	1 Enable 0 Disable	0	R/W	
		D4	**INT10SEN**	10-second interrupt enable	1 Enable 0 Disable	0	R/W	
		D3	**INT1HZEN**	1 Hz interrupt enable	1 Enable 0 Disable	0	R/W	
		D2	**INT4HZEN**	4 Hz interrupt enable	1 Enable 0 Disable	0	R/W	
		D1	**INT8HZEN**	8 Hz interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	**INT32HZEN**	32 Hz interrupt enable	1 Enable 0 Disable	0	R/W	
RTC Interrupt Flag Register (RTC_IFLG)	**0x56c4** (16 bits)	D15–10	**–**	reserved	–	–	–	0 when being read.
		D9	**INT1D**	1-day interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D8	**INTHD**	Half-day interrupt flag		0	R/W	
		D7	**INT1H**	1-hour interrupt flag		0	R/W	
		D6	**INT10M**	10-minute interrupt flag		0	R/W	
		D5	**INT1M**	1-minute interrupt flag		0	R/W	
		D4	**INT10S**	10-second interrupt flag		0	R/W	
		D3	**INT1HZ**	1 Hz interrupt flag		0	R/W	
		D2	**INT4HZ**	4 Hz interrupt flag		0	R/W	
		D1	**INT8HZ**	8 Hz interrupt flag		0	R/W	
		D0	**INT32HZ**	32 Hz interrupt flag		0	R/W	
RTC Minute/Second Counter Register (RTC_MS)	**0x56c6** (16 bits)	D15	**–**	reserved	–	–	–	0 when being read.
		D14–8	**RTCMIN [6:0]**	Minute counter	0x0 to 0x3b (binary mode) 0x00 to 0x59 (BCD mode)	X	R/W	
		D7	**–**	reserved	–	–	–	0 when being read.
		D6–0	**RTCSEC [6:0]**	Second counter	0x0 to 0x3b (binary mode) 0x00 to 0x59 (BCD mode)	X	R/W	
RTC Hour Counter Register (RTC_H)	**0x56c8** (16 bits)	D15–8	**–**	reserved	–	–	–	0 when being read.
		D7	**AMPM**	AM/PM	1 PM 0 AM	X	R/W	
		D6	**–**	reserved	–	–	–	0 when being read.
		D5–0	**RTCHOUR [5:0]**	Hour counter	0x0 to 0x17 (binary mode) 0x00 to 0x23 (BCD mode)	X	R/W	

0xffff84–0xffffd0 S1C17 Core I/O

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Processor ID Register (IDIR)	**0xffff84** (8 bits)	D7–0	**IDIR[7:0]**	Processor ID 0x10: S1C17 Core	0x10	0x10	R	
Debug RAM Base Register (DBRAM)	**0xffff90** (32 bits)	D31–24	**–**	Unused (fixed at 0)	0x0	0x0	R	
		D23–0	**DBRAM[23:0]**	Debug RAM base address	0x7c0	0x7c0	R	
Debug Control Register (DCR)	**0xffffa0** (8 bits)	D7	**IBE4**	Instruction break #4 enable	1 Enable 0 Disable	0	R/W	
		D6	**IBE3**	Instruction break #3 enable	1 Enable 0 Disable	0	R/W	
		D5	**IBE2**	Instruction break #2 enable	1 Enable 0 Disable	0	R/W	
		D4	**DR**	Debug request flag	1 Occurred 0 Not occurred	0	R/W	Reset by writing 1.
		D3	**IBE1**	Instruction break #1 enable	1 Enable 0 Disable	0	R/W	
		D2	**IBE0**	Instruction break #0 enable	1 Enable 0 Disable	0	R/W	
		D1	**SE**	Single step enable	1 Enable 0 Disable	0	R/W	
		D0	**DM**	Debug mode	1 Debug mode 0 User mode	0	R	
Instruction Break Address Register 1 (IBAR1)	**0xffffb4** (32 bits)	D31–24	**–**	reserved	–	–	–	0 when being read.
		D23–0	**IBAR1[23:0]**	Instruction break address #1 IBAR123 = MSB IBAR10 = LSB	0x0 to 0xffffff	0x0	R/W	
Instruction Break Address Register 2 (IBAR2)	**0xffffb8** (32 bits)	D31–24	**–**	reserved	–	–	–	0 when being read.
		D23–0	**IBAR2[23:0]**	Instruction break address #2 IBAR223 = MSB IBAR20 = LSB	0x0 to 0xffffff	0x0	R/W	
Instruction Break Address Register 3 (IBAR3)	**0xffffbc** (32 bits)	D31–24	**–**	reserved	–	–	–	0 when being read.
		D23–0	**IBAR3[23:0]**	Instruction break address #3 IBAR323 = MSB IBAR30 = LSB	0x0 to 0xffffff	0x0	R/W	
Instruction Break Address Register 4 (IBAR4)	**0xffffd0** (32 bits)	D31–24	**–**	reserved	–	–	–	0 when being read.
		D23–0	**IBAR4[23:0]**	Instruction break address #4 IBAR423 = MSB IBAR40 = LSB	0x0 to 0xffffff	0x0	R/W	

Appendix B Power Saving

Current consumption will vary dramatically, depending on CPU operating mode, operation clock frequency, and the peripheral circuits being operated. Listed below are the control methods for saving power.

B.1 Clock Control Power Saving

This section describes clock systems that can be controlled via software and power-saving control details. For more information on control registers and control methods, refer to the respective module sections.

System SLEEP

- Execute the `slp` instruction (when RTC is stopped)
  When the entire system can be stopped, stop the RTC and execute the `slp` instruction. The CPU enters SLEEP mode and the OSC1/OSC3A/OSC3B clocks stop. This also stops all peripheral circuits using the OSC1/OSC3A/OSC3B clocks. Starting up the CPU from SLEEP mode is therefore limited to startup using a port (described later).
- Execute the `slp` instruction (when RTC is running)
  When the system except the RTC for time keeping can be stopped, maintain the RTC in running state and execute the `slp` instruction. The CPU enters SLEEP mode and the OSC3A/OSC3B clocks stop. This also stops all peripheral circuits using the OSC3A/OSC3B clocks. Starting up the CPU from SLEEP mode is therefore limited to startup using a port or RTC (described later).

System clocks

- Select a low-speed clock source (CLG module)
  Select a low-speed oscillator for the system clock source. You can reduce current consumption by selecting the OSC1 clock when low-speed processing is possible.
- Disable unnecessary oscillator circuits (CLG module)
  Operate the oscillator comprising the system clock source. Where possible, stop the other oscillators. You can reduce current consumption by using OSC1 as the system clock and disable the OSC3B and OSC3A oscillators.

CPU clock (CCLK)

- Execute the `halt` instruction
  Execute the `halt` instruction when program execution by the CPU is not required, for example, when the system is waiting for an interrupt. The CPU enters HALT mode and suspends operations, but the peripheral circuits maintain the status in place at the time of the `halt` instruction, enabling use of peripheral circuits for generating interrupts and the LCD driver. You can reduce power consumption even further by suspending unnecessary oscillator and peripheral circuits before executing the `halt` instruction. The CPU is started from HALT mode by an interrupt from a port or the peripheral circuit operating in HALT mode.
- Select a low-speed clock gear (CLG module)
  The CLG module can reduce CPU clock speeds to between 1/1 and 1/8 of the system clock via the clock gear settings. You can reduce current consumption by operating the CPU at the minimum speed required for the application.

Regulated clock (F256)

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

Peripheral clock (PCLK)

- Stop PCLK (CLG module)
  Stop the PCLK clock supplied from the CLG to peripheral circuits if none of the following peripheral circuits is required.

Peripheral circuits that use PCLK
- Interrupt controller
- 8-bit timer Ch.0
- SPI Ch.0
- Power generator
- P ports and port MUX (control registers, chattering filters)
- MISC registers

PCLK is not required for the peripheral modules/functions shown below.

Peripheral circuits/functions that do not use PCLK
- Real-time clock
- Clock timer
- Watchdog timer
- LCD driver
- Sound generator
- SVD circuit
- 16-bit PWM timer Ch.0
- UART Ch.0
- FOUTA/FOUTB outputs

Table B.1.1 shows a list of methods for clock control and starting/stopping the CPU.

Table B.1.1 Clock Control List

Current consumption	OSC1	OSC3B/ OSC3A	CPU (CCLK)	PCLK peripheral	RTC	OSC1 peripheral	CPU stop method	CPU startup method
↑ Low	Stop	Stop	Stop	Stop	Stop	Stop	Execute `slp` instruction	1
	Oscillation (for RTC)	Stop	Stop	Stop	Run	Stop	Execute `slp` instruction	1, 2
	Oscillation (for RTC)	Stop	Stop	Stop	Run	Stop	Execute `halt` instruction	1, 2
	Oscillation (system CLK)	Stop	Stop	Stop	Run	Run	Execute `halt` instruction	1, 2, 3
	Oscillation (system CLK)	Stop	Stop	Run	Run	Run	Execute `halt` instruction	1, 2, 3, 4
	Oscillation (system CLK)	Stop	Run (1/1)	Run	Run	Run		
	Oscillation	Oscillation (system CLK)	Stop	Run	Run	Run	Execute `halt` instruction	1, 2, 3, 4
	Oscillation	Oscillation (system CLK)	Run (low gear)	Run	Run	Run		
High ↓	Oscillation	Oscillation (system CLK)	Run (1/1)	Run	Run	Run		

HALT and SLEEP mode cancelation methods (CPU startup method)

1. Startup by port
   Started up by an I/O port interrupt or a debug interrupt (ICD forced break).
2. Startup by RTC
   Started up by an RTC interrupt.
3. Startup by OSC1 peripheral circuit
   Started up by a clock timer or watchdog timer interrupt.
4. Startup by PCLK peripheral circuit
   Started up by a PCLK peripheral circuit interrupt.

B.2 Reducing Power Consumption via Power Supply Control

The available power supply controls are listed below.

V_{D1}/V_{OSC} regulators

- Note that turning on internal voltage regulator heavy load protection will increase current consumption. Turn off heavy load protection for normal operations. Turn on only if operations are unstable.

LCD power supply circuit

- Setting VCSEL to 0 (V_{C1} reference voltage) will increase current consumption.
  Set VCSEL to 1 (V_{C2} reference voltage) if the power supply voltage V_{DD} is 2.2 V or higher.
- Turning on the LCD power supply heavy load protection will increase current consumption.
  Turn off heavy load protection for normal operations. Turn on only if the display is unstable.

Power supply voltage detection (SVD) circuit

- Operating the SVD circuit will increase current consumption.
  Turn off power supply voltage detection unless it is required.

B.3 Other Power Saving Methods

Theoretical regulation

- When data input from the I/O port is used to set the theoretical regulation value register (TR_VAL), place the port into output mode and set the read data as the output data after data is read. This reduces the pull-up resistor current that constantly flows.

Appendix C Mounting Precautions

This section describes various precautions for circuit board design and IC mounting.

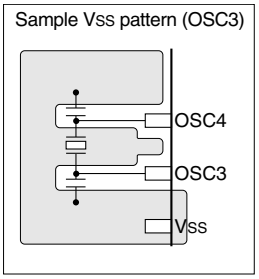
Oscillator circuit

- Oscillation characteristics depend on factors such as components used (resonator, CG, CD) and circuit board patterns. In particular, with ceramic or crystal resonators, select the appropriate capacitors (CG, CD) only after fully evaluating components actually mounted on the circuit board.
- Oscillator clock disturbances caused by noise may cause malfunctions. To prevent such disturbances, consider the following points. The latest devices, in particular, are manufactured by microscopic processes, making them especially susceptible to noise.
  Areas in which noise countermeasures are especially important include the OSC2 pin and related circuit components and wiring. OSC1 pin handling is equally important. The noise precautions required for the OSC1 and OSC2 pins are described below. We also recommend applying similar noise countermeasures to the high-speed oscillator circuit, such as the OSC3 and OSC4 pins and wiring.

(1) Components such as a resonator, resistors, and capacitors connected to the OSC1 (OSC3) and OSC2 (OSC4) pins should have the shortest connections possible.

(2) Wherever possible, avoid locating digital signal lines within 3 mm of the OSC1 (OSC3) and OSC2 (OSC4) pins or related circuit components and wiring. Rapidly-switching signals, in particular, should be kept at a distance from these components. Since the spacing between layers of multi-layer printed circuit boards is a mere 0.1 mm to 0.2 mm, the above precautions also apply when positioning digital signal lines on other layers.
  Never place digital signal lines alongside such components or wiring, even if more than 3 mm distance or located on other layers. Avoid crossing wires.

(3) Use VSS to shield OSC1 (OSC3) and OSC2 (OSC4) pins and related wiring (including wiring for adjacent circuit board layers). Layers wired should be adequately shielded as shown to the right. Fully ground adjacent layers, where possible. At minimum, shield the area at least 5 mm around the above pins and wiring.
  Even after implementing these precautions, avoid configuring digital signal lines in parallel, as described in (2) above. Avoid crossing even on discrete layers, except for lines carrying signals with low switching frequencies.



(4) After implementing these precautions, check the output clock waveform by running the actual application program within the product. Use an oscilloscope to check the FOUTA or FOUTB pin output.
  You can check the quality of the OSC3 output waveform via the FOUTA/B output. Confirm that the frequency is as designed, is free of noise, and has minimal jitter.
  You can check the quality of the OSC1 waveform via the FOUTA/B output. In particular, enlarge the areas before and after the clock rising and falling edges and take special care to confirm that the regions approximately 100 ns to either side are free of clock or spiking noise.

  Failure to observe precautions (1) to (3) adequately may lead to jitter in the OSC3 output and noise in the OSC1 output. Jitter in the OSC3 output will reduce operating frequencies, while noise in the OSC1 output will destabilize timers operated by the OSC1 clock as well as CPU Core operations when the system clock switches to OSC1.

Reset circuit

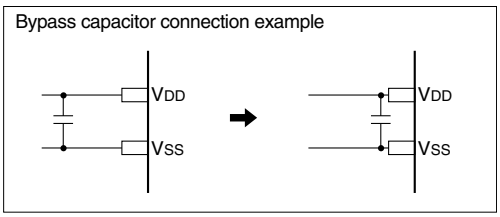
- The reset signal input to the #RESET pin when power is turned on will vary, depending on various factors, such as power supply start-up time, components used, and circuit board patterns. Constants such as capacitance and resistance should be determined through testing with real-world products.
- Components such as capacitors and resistors connected to the #RESET pin should have the shortest connections possible to prevent noise-induced resets.

Power supply circuit

Sudden power supply fluctuations due to noise will cause malfunctions. Consider the following issues.

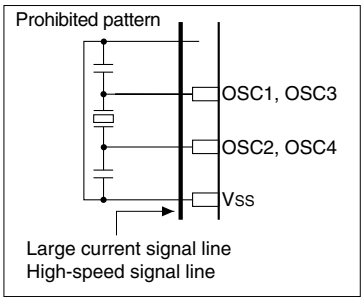
(1) Connections from the power supply to the VDD and VSS pins should be implemented via the shortest, thickest patterns possible.

(2) If a bypass capacitor is connected between VDD and VSS, connections between the VDD and VSS pins should be as short as possible.



Signal line location

- To prevent electromagnetically-induced noise arising from mutual induction, large-current signal lines should not be positioned close to circuits susceptible to noise, such as oscillators.
- Locating signal lines in parallel over significant distances or crossing signal lines operating at high speed will cause malfunctions due to noise generated by mutual interference. Specifically, avoid positioning crossing signal lines operating at high speed close to circuits susceptible to noise, such as oscillators.



Handling of light (for bare chip mounting)

The characteristics of semiconductor components can vary when exposed to light. ICs may malfunction or nonvolatile memory data may be corrupted if ICs are exposed to light.
Consider the following precautions for circuit boards and products in which this IC is mounted to prevent IC malfunctions attributable to light exposure.

(1) Design and mount the product so that the IC is shielded from light during use.

(2) Shield the IC from light during inspection processes.

(3) Shield the IC on the upper, underside, and side faces of the IC chip.

(4) Mount the IC chip within one week of opening the package. If the IC chip must be stored before mounting, take measures to ensure light shielding.

(5) Adequate evaluations are required to assess nonvolatile memory data retention characteristics before product delivery if the product is subjected to heat stress exceeding regular reflow conditions during mounting processes.

Unused pins

(1) I/O port (P) pins
Unused pins should be left open. The control registers should be fixed at the initial status (input with pull-up enabled).

(2) OSC1, OSC2, OSC3, and OSC4 pins
If the OSC1A or OSC3A oscillator circuit is not used, the OSC1 and OSC2 pins or the OSC3 and OSC4 pins should be left open. The control registers should be fixed at the initial status (oscillation disabled).

(3) V_{C1-3}, CA, CB, SEG*x*, and COM*x* pins
If the LCD driver is not used, these pins should be left open. The control registers should be fixed at the initial status (display off). The unused SEG*x* pins that are not required to connect should be left open even if the LCD driver is used.

Miscellaneous

This product series is manufactured using microscopic processes.
Although it is designed to ensure basic IC reliability meeting EIAJ and MIL standards, minor variations over time may result in electrical damage arising from disturbances in the form of voltages exceeding the absolute maximum rating when mounting the product in addition to physical damage. The following factors can give rise to these variations:

(1) Electromagnetically-induced noise from industrial power supplies used in mounting reflow, reworking after mounting, and individual characteristic evaluation (testing) processes
(2) Electromagnetically-induced noise from a solder iron when soldering

In particular, during soldering, take care to ensure that the soldering iron GND (tip potential) has the same potential as the IC GND.

Appendix D Measures Against Noise

To improve noise immunity, take measures against noise as follows:

Noise Measures for V_{DD} and V_{SS} Power Supply Pins

The IC will malfunction at the instant when noise falling below the rated voltage is input. Take measures on the circuit board, including close patterns for circuit board power supply circuits, noise-filtering decoupling capacitors, and surge/noise prevention components on the power supply line.
For the recommended patterns on the circuit board, see "Mounting Precautions" in Appendix.

Noise Measures for #RESET Pin

The pull-up resistor for the #RESET pin included in this product has a relatively high impedance of 100 kΩ to 500 kΩ and is not noise-resistant. Extraneous noise may pull the #RESET pin down to a low level and this will cause the IC to reset. Therefore, the circuit board must be designed properly taking noise measures into consideration.
For the recommended patterns on the circuit board, see "Mounting Precautions" in Appendix.

Noise Measures for Oscillator Pins

The oscillator input pins must pass a signal of small amplitude, so they are hypersensitive to noise. Therefore, the circuit board must be designed properly taking noise measures into consideration.
For the recommended patterns on the circuit board, see "Mounting Precautions" in Appendix.

Noise Measures for Debug Pins

This product provides the input/output pins (DCLK, DST2, and DSIO) to connect ICDmini (S5U1C17001H) for debugging. If noise is input to these pins, the S1C17 Core may enter debug mode. To prevent unexpected transitions to debug mode caused by extraneous noise, switch the DCLK, DST2, and DSIO pins to general-purpose I/O port pins within the initialization routine when the debug functions are not used.
For details of the pin functions and the function switch control, see the "I/O Ports (P)" chapter.

Note: Do not perform the function switching shown above when the application is under development, as the debug functions must be used. The debugging cannot be performed after the pin function is switched. The above processing must be added after the application development has completed and debugging is no longer necessary.

The DSIO pin should be pulled up with a 10 kΩ resistor when using the debug pin functions. The pull-up resistor for the DSIO pin included in this product has a relatively high impedance of 100 kΩ to 500 kΩ and is not noise-resistant.

Noise Measures for Interrupt Input Pins

This product is able to generate a port input interrupt when the input signal changes. The interrupt is generated when an input signal edge is detected, therefore, an interrupt may occur if the signal changes due to extraneous noise.
To prevent occurrence of unexpected interrupts due to extraneous noise, enable the chattering filter circuit when using the port input interrupt.
For details of the port input interrupt and chattering filter circuit, see the "I/O Ports (P)" chapter.

Noise Measures for UART Pins

This product includes a UART module for asynchronous communications. The UART starts receive operation when it detects a low level input from the SIN*x* pin. Therefore, a receive operation may be started if the SIN*x* pin is set to low due to extraneous noise. In this case, a receive error will occur or invalid data will be received. To prevent UART from malfunction caused by extraneous noise, take the following measures:

- Stop the UART operations (RXEN/UART_CTL*x* register = 0) while asynchronous communication is not performed.
- Execute the resending process via software after executing the receive error handler with a parity check.

For details of the pin functions and the function switch control, see the "I/O Ports (P)" chapter. For the UART control and details of receive errors, see the "UART" chapter.

Appendix E Initialization Routine

The following lists typical vector tables and initialization routines:

boot.s

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

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

.long BOOT                      ; 0x00       0x00    reset                ...(2)
.long unalign_handler           ; 0x01       0x04    unalign
.long nmi_handler               ; 0x02       0x08    NMI
.long int03_handler             ; 0x03       0x0c    -
.long p0_handler                ; 0x04       0x10    P0 port
.long int05_handler             ; 0x05       0x14    -
.long int06_handler             ; 0x06       0x18    -
.long ct_handler                ; 0x07       0x1c    CT
.long rtc_handler               ; 0x08       0x20    RTC
.long int09_handler             ; 0x09       0x24    -
.long lcd_handler               ; 0x0a       0x28    LCD
.long t16a2_0_handler           ; 0x0b       0x2c    T16A2 ch0
.long int0c_handler             ; 0x0c       0x30    -
.long int0d_handler             ; 0x0d       0x34    -
.long t8_0_handler              ; 0x0e       0x38    T8 ch0
.long int0f_handler             ; 0x0f       0x3c    -
.long uart_0_handler            ; 0x10       0x40    UART ch0
.long int11_handler             ; 0x11       0x44    -
.long spi_0_handler             ; 0x12       0x48    SPI ch0
.long int13_handler             ; 0x13       0x4c    -
.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, 0x07c0                                               ...(4)

        ; ----- Memory controller ----------------
        Xld.a   %r1, 0x54b0     ; FLASHC register address

        ; Flash read wait cycle
        Xld.a   %r0, 0x00       ; No wait
        ld.b    [%r1], %r0      ; [0x54b0] <= 0x00                        ...(5)

        ; ===== Main routine ==========================================
         ...
```

```
; ========================================================================
;       Interrupt handler
; ========================================================================
; ----- Address unalign --------------------------
unalign_handler:
        ...

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

---

(1) A "`.rodata`" section is declared to locate the vector table in the "`.vector`" section.

(2) Interrupt handler routine addresses are defined as vectors.
"`int`*`XX`*`_handler`" can be used for software interrupts.

(3) The program code is written in the "`.text`" section.

(4) Sets the stack pointer.

(5) Sets the number of Flash memory wait cycles.
Can be set to no wait in the S1C17651.
(See the "Memory Map, Bus Control" chapter.)

Revision History

Code No.	Page	Contents
412120600	All	New establishment

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: 412120600
Issue April 2011 in JAPAN Ⓛ