EPSON
EXCEED YOUR VISION

CMOS 16-BIT SINGLE CHIP MICROCONTROLLER

S1C17554/564
Technical Manual

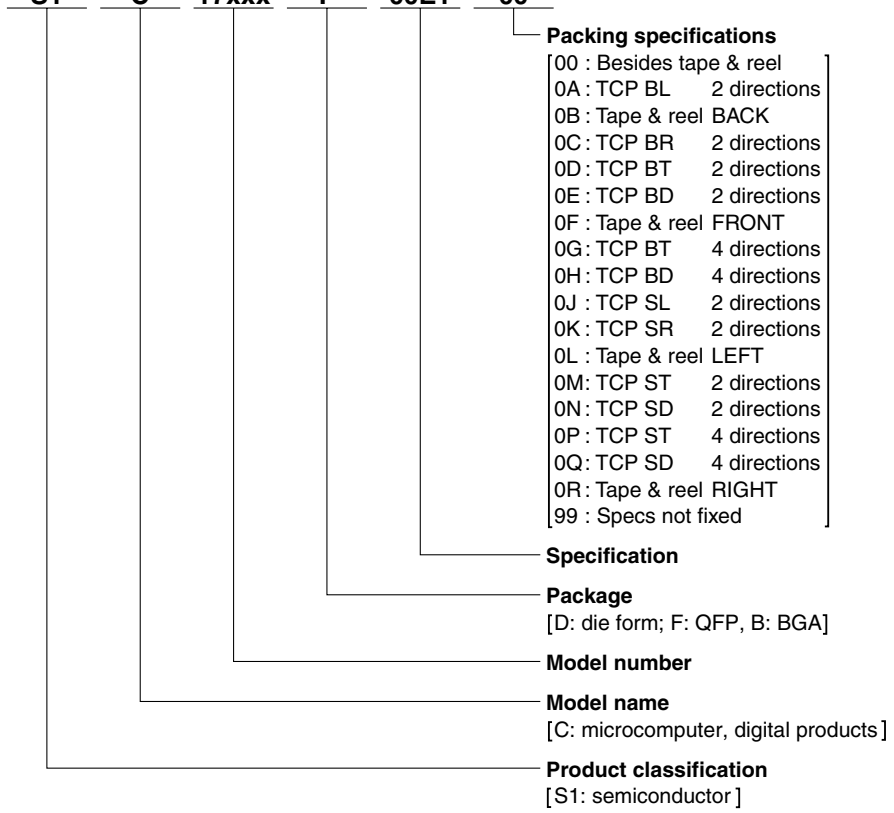
SEIKO EPSON CORPORATION

Rev.1.3

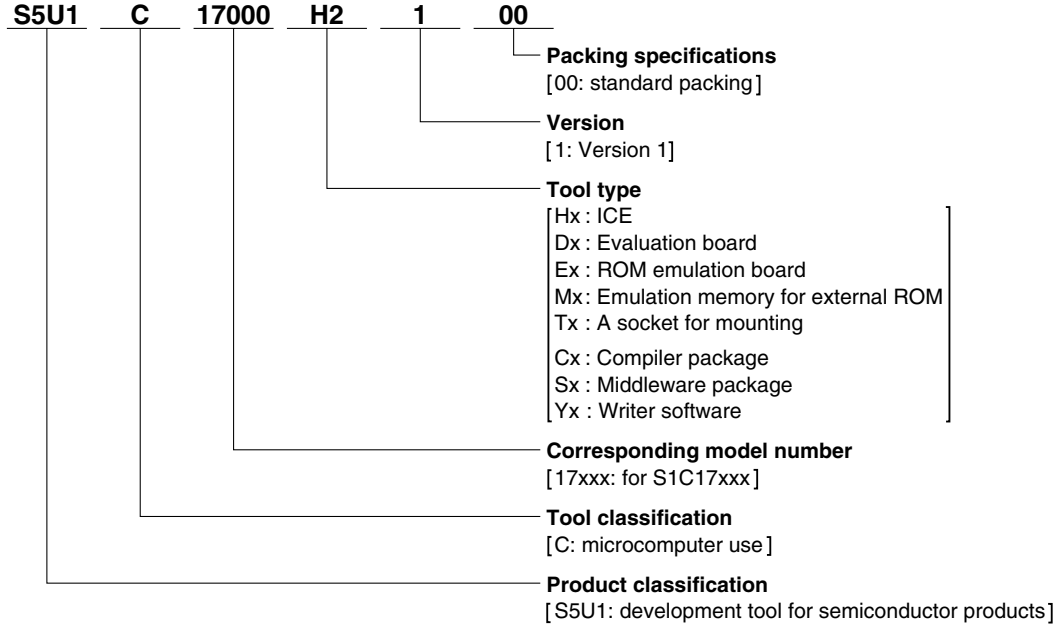
NOTICE

Configuration of product number

Devices



Development tools



– Contents –

1 Overview

1.1 Features

The main features of the S1C17554/564 are listed below.

Table 1.1.1 Features

Model	S1C17554	S1C17564
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	128K bytes (for both instructions and data)	
Erase/program count	Min. 10 times (Applied only when FLS V1.0 or later is used.)	
Other	• Read/program protection function • An erasing/programming power supply (V_{PP}) is required. • Allows on-board programming using a debugging tool such as ICDmini.	
Embedded RAM		
Capacity	16K bytes	
Clock generator		
System clock source	2 sources (OSC3/OSC1)	3 sources (IOSC/OSC3/OSC1)
IOSC oscillator circuit		2/4/8/12 MHz(typ.) internal oscillator circuit
OSC3 oscillator circuit	24 MHz (max.) crystal or ceramic oscillator circuit Supports an external clock input.	
OSC1 oscillator circuit	32.768 kHz (typ.) crystal oscillator circuit Supports an external clock input.	
Other	• Core clock frequency control • Peripheral module clock supply control	
I/O ports		
Number of general-purpose I/O ports	Max. 40 bits (TQFP13-64pin package) Max. 34 bits (WCSP-48 package) (Pins are shared with the peripheral I/O.)	Max. 40 bits (Pins are shared with the peripheral I/O.)
Serial interfaces		
SPI	3 channels	
I^2C master (I2CM)	1 channel	
I^2C slave (I2CS)	1 channel	
UART	2 channels (IrDA1.0 supported)	
IR remote controller (REMC)	1 channel	
Universal serial interface (USI)		2 channels (Usable as a UART, SPI, or I^2C)
Timers		
16-bit timer (T16)	3 channels	
Fine mode 16-bit timer (T16F)	2 channels	
16-bit PWM timer (T16A)	4 channels	
Clock timer (CT)	1 channel	
Stopwatch timer (SWT)	1 channel	
Watchdog timer (WDT)	1 channel	
A/D converter		
Conversion method	Successive approximation type	
Number of analog input channels	4 channels (max.)	
Resolution	10 bits	
Interrupts		
Reset interrupt	#RESET pin	
NMI	Watchdog timer	
Programmable interrupts	23 systems (8 levels)	
Power supply voltage		
Core voltage (LV_{DD})	1.65 V to 1.95 V	1.65 V to 1.95 V (Not required when the regulator is used.)
I/O voltage (HV_{DD})	1.65 V to 5.5 V	2.0 V to 5.5 V (Regulator used) 1.65 V to 5.5 V (Regulator not used)
Analog voltage (AV_{DD})	2.7 V to 5.5 V	
Flash programming/erasing voltage (V_{PP})	7 V/7.5 V	

Model	S1C17554	S1C17564
Regulator		
Input voltage		2.0 V to 5.5 V
Output voltage		1.8 V
Other		Enables the system to operate with a 3.3 V or 5.0 V single power supply.
Operating temperature		
Operating temperature range	-40°C to 85°C	
Current consumption (Typ value, LV_{DD} = HV_{DD} = 1.8 V)		
SLEEP state	0.8 µA (OSC1 = Off, OSC3 = Off)	1.2 µA (OSC1 = Off, IOSC = Off, OSC3 = Off)
HALT state	2.7 µA (OSC1 = 32 kHz, OSC3 = Off)	3.1 µA (OSC1 = 32 kHz, IOSC = Off, OSC3 = Off)
Run state	16 µA (OSC1 = 32 kHz, OSC3 = Off)	16 µA (OSC1 = 32 kHz, IOSC = Off, OSC3 = Off)
	3000 µA (OSC1 = Off, OSC3 = 8 MHz ceramic)	3000 µA (OSC1 = Off, IOSC = Off, OSC3 = 8 MHz ceramic)
		4500 µA (OSC1 = Off, IOSC = 12 MHz, OSC3 = Off)
A/D conversion	380 µA (AV_{DD} = 3.6 V, 100 kHz sampling, FSEL[1:0] = 0x0, XPD[1:0] = 0x3)	
Shipping form		
1	TQFP13-64pin (10 mm × 10 mm × 1.0 mm, lead pitch: 0.5 mm)	
2	Die form (3.137 mm × 3.137 mm, pad pitch: 140 µm)	
3	WCSP-48 (3.137 mm × 3.137 mm × 0.72 mm, ball pitch: 0.4 mm)	

1.2 Block Diagram

S1C17554

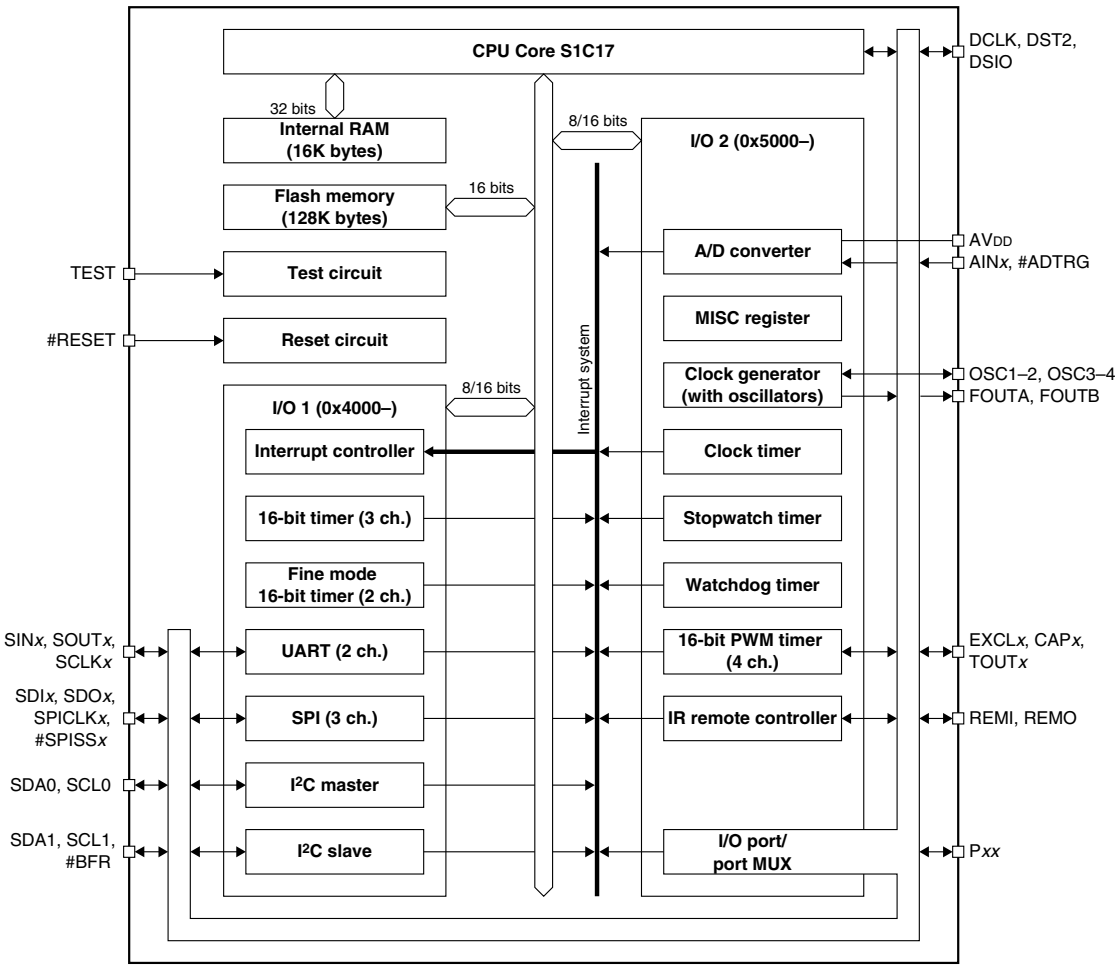


Figure 1.2.1 S1C17554 Block Diagram

S1C17564

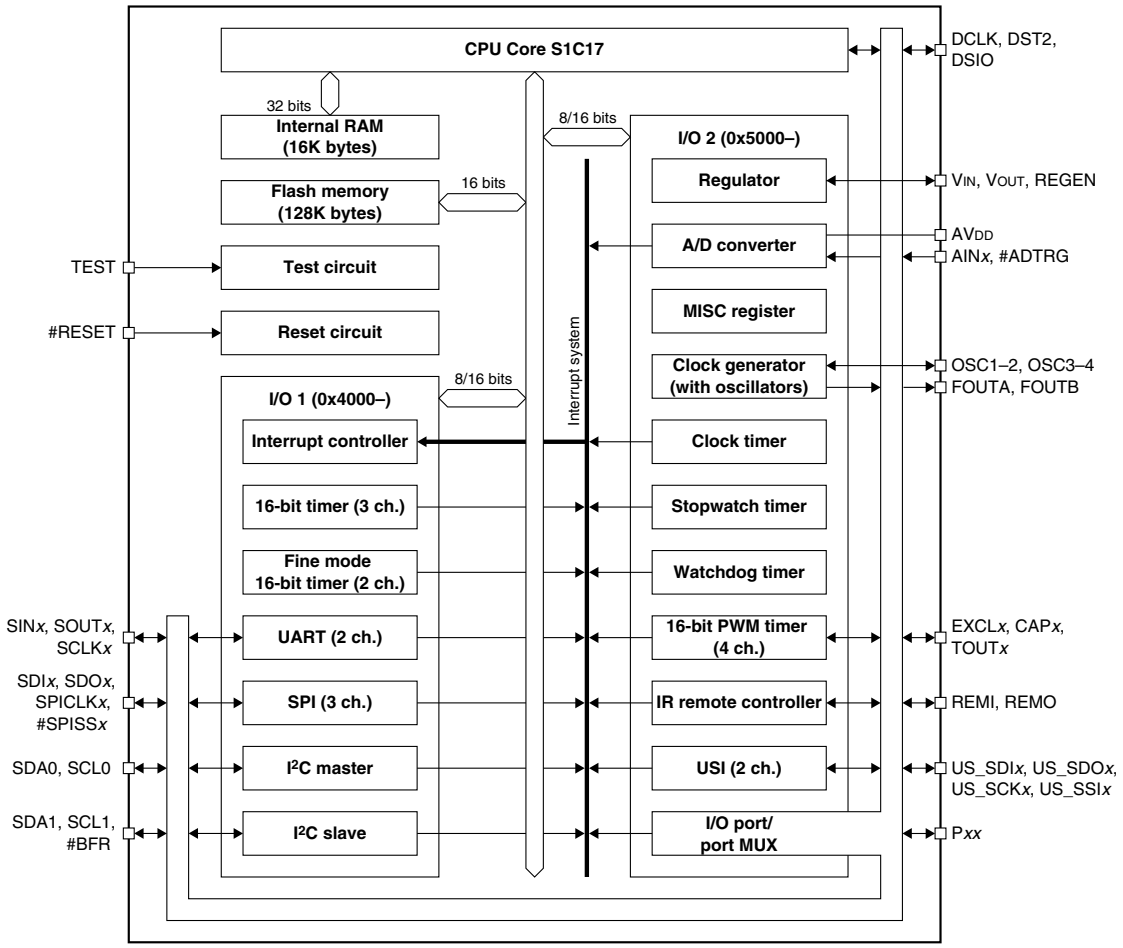
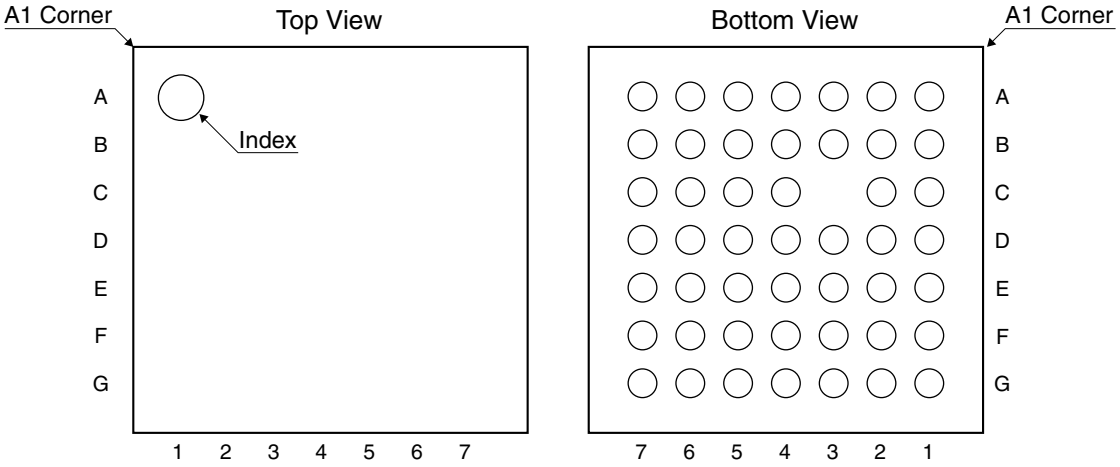


Figure 1.2.2 S1C17564 Block Diagram

1.3 Pins

1.3.1 S1C17554 Pin Configuration Diagrams

WCSP-48 (S1C17554)



Top View

	1	2	3	4	5	6	7
A	**P10** SDI0	**P11** SDO0	**P13** #SPISS0 TOUT5 CAP5	**P14** SIN1 SDI1	**P43** SDA1 REMI	**P24** **(EXCL3)** SDO2	**P26** SDA1
B	**P01** AIN1	**P00** AIN0	**P12** SPICLK0	**P15** SOUT1 SDO1	**P44** SCL1 REMO	**P25** #BFR #SPISS2	**P27** SCL1
C	**P03** AIN3	**P02** AIN2		**HVDD**	**P16** SCLK1 SPICLK1	**VSS**	**P30** TOUT0 CAP0
D	**P17** SCL0	**P32** TOUT4 CAP4 FOUTA	**AVDD**	**VSS**	**LVDD**	**P31** #BFR #ADTRG	**DST2** P37
E	**P45** **(EXCL0)** SDA0	**P42** SCLK0 TOUT1 CAP1	**P41** SOUT0 TOUT7 CAP7	**HVDD**	**DCLK** P35	**TEST**	**DSIO** P36
F	**P40** SIN0 TOUT6 CAP6	**LVDD**	**P22** **(EXCL1)** FOUTB	**P20** TOUT2 CAP2	**VPP**	**P34** REMO #SPISS1	**P33** REMI SPICLK2
G	**#RESET**	**P23** **(EXCL2)** SDI2	**P21** TOUT3 CAP3	**OSC4**	**OSC3**	**OSC2**	**OSC1**

Figure 1.3.1.1 S1C17554 Pin Configuration Diagram (WCSP-48)

TQFP13-64pin (S1C17554)

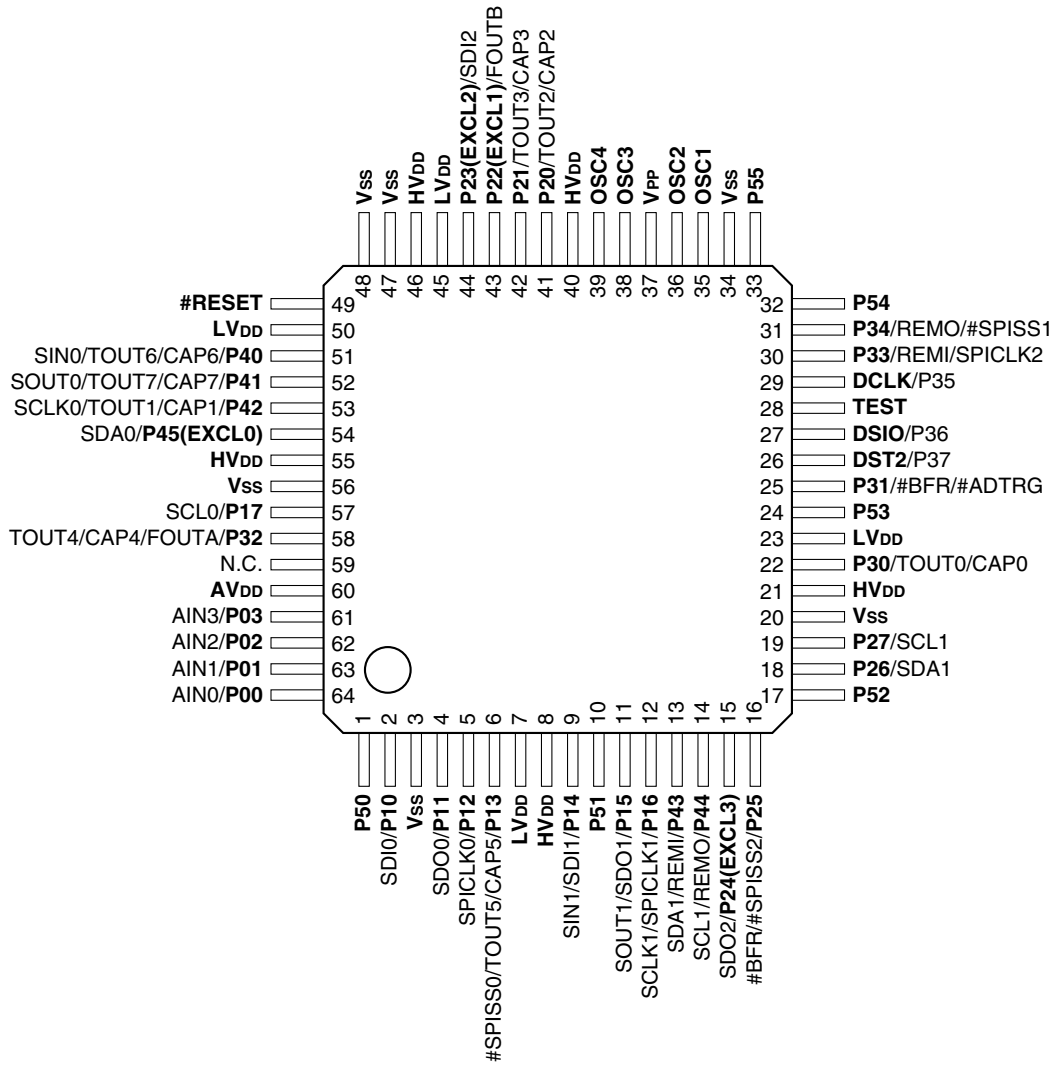


Figure 1.3.1.2 S1C17554 Pin Configuration Diagram (TQFP13-64pin)

Chip (S1C17554)

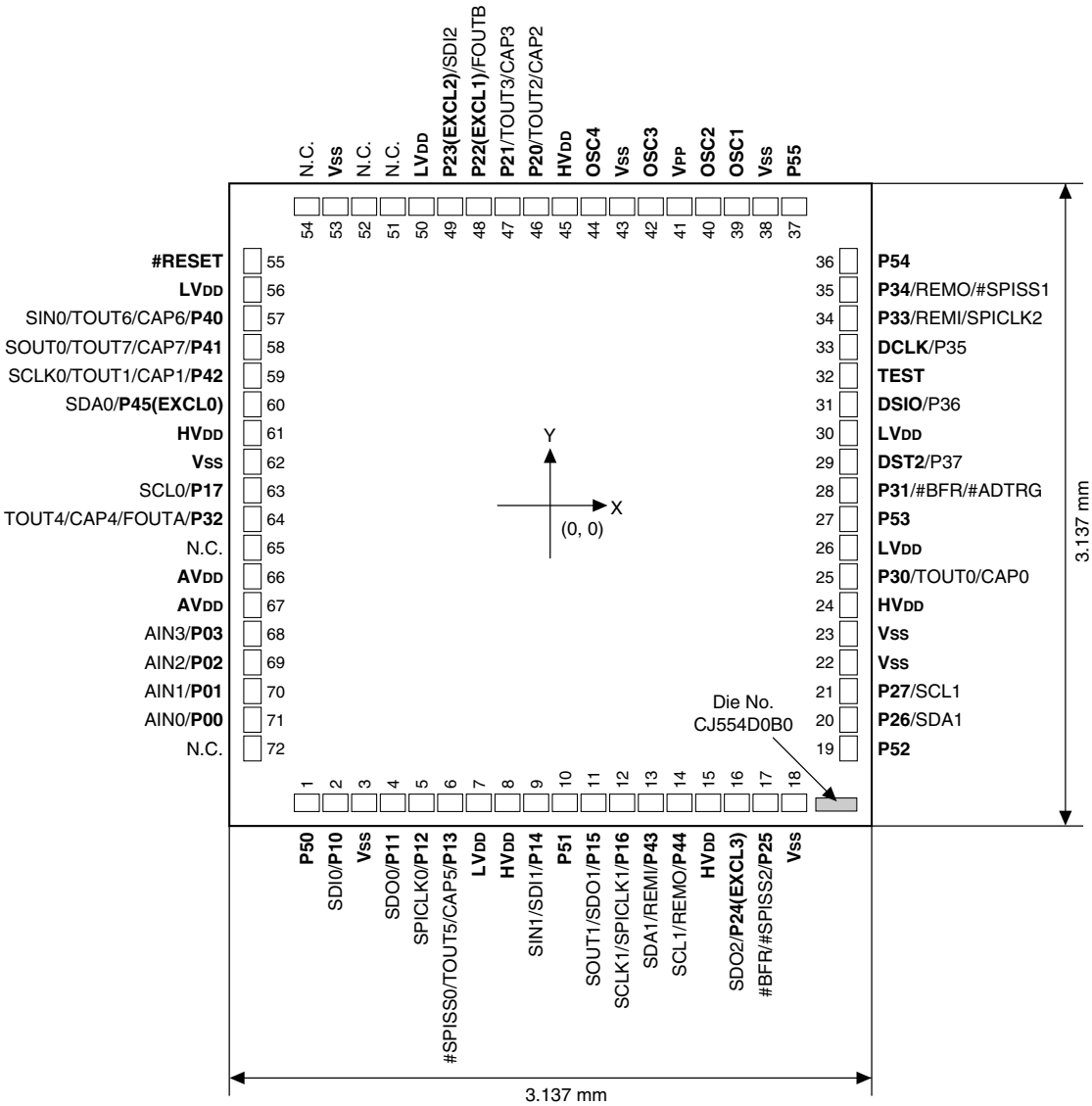


Figure 1.3.1.3 S1C17554 Pad Configuration Diagram

Chip size	X = 3.137 mm, Y = 3.137 mm
Pad opening	No. 1 to 18, 37 to 54: X = 122 µm, Y = 85 µm
	No. 19 to 36, 55 to 72: X = 85 µm, Y = 122 µm
Chip thickness	400 µm

Table 1.3.1.1 S1C17554 Pad Coordinates

No.	Name	X (mm)	Y (mm)	No.	Name	X (mm)	Y (mm)
1	**P50**	-1190	-1455.5	37	**P55**	1190	1455.5
2	**P10**/SDI0	-1050	-1455.5	38	**VSS**	1050	1455.5
3	**VSS**	-910	-1455.5	39	**OSC1**	910	1455.5
4	**P11**/SDO0	-770	-1455.5	40	**OSC2**	770	1455.5
5	**P12**/SPICLK0	-630	-1455.5	41	**VPP**	630	1455.5
6	**P13**/#SPISS0/TOUT5/CAP5	-490	-1455.5	42	**OSC3**	490	1455.5
7	**LVDD**	-350	-1455.5	43	**VSS**	350	1455.5
8	**HVDD**	-210	-1455.5	44	**OSC4**	210	1455.5
9	**P14**/SIN1/SDI1	-70	-1455.5	45	**HVDD**	70	1455.5
10	**P51**	70	-1455.5	46	**P20**/TOUT2/CAP2	-70	1455.5
11	**P15**/SOUT1/SDO1	210	-1455.5	47	**P21**/TOUT3/CAP3	-210	1455.5
12	**P16**/SCLK1/SPICLK1	350	-1455.5	48	**P22(EXCL1)**/FOUTB	-350	1455.5
13	**P43**/SDA1/REMI	490	-1455.5	49	**P23(EXCL2)**/SDI2	-490	1455.5
14	**P44**/SCL1/REMO	630	-1455.5	50	**LVDD**	-630	1455.5
15	**HVDD**	770	-1455.5	51	N.C.	-770	1455.5
16	**P24(EXCL3)**/SDO2	910	-1455.5	52	N.C.	-910	1455.5
17	**P25**/#BFR/SPISS2	1050	-1455.5	53	**VSS**	-1050	1455.5
18	**VSS**	1190	-1455.5	54	N.C.	-1190	1455.5
19	**P52**	1455.5	-1190	55	**#RESET**	-1455.5	1190
20	**P26**/SDA1	1455.5	-1050	56	**LVDD**	-1455.5	1050
21	**P27**/SCL1	1455.5	-910	57	**P40**/SIN0/TOUT6/CAP6	-1455.5	910
22	**VSS**	1455.5	-770	58	**P41**/SOUT0/TOUT7/CAP7	-1455.5	770
23	**VSS**	1455.5	-630	59	**P42**/SCLK0/TOUT1/CAP1	-1455.5	630
24	**HVDD**	1455.5	-490	60	**P45(EXCL0)**/SDA0	-1455.5	490
25	**P30**/TOUT0/CAP0	1455.5	-350	61	**HVDD**	-1455.5	350
26	**LVDD**	1455.5	-210	62	**VSS**	-1455.5	210
27	**P53**	1455.5	-70	63	**P17**/SCL0	-1455.5	70
28	**P31**/#BFR/ADTRG	1455.5	70	64	**P32**/TOUT4/CAP4/FOUTA	-1455.5	-70
29	**DST2**/P37	1455.5	210	65	N.C.	-1455.5	-210
30	**LVDD**	1455.5	350	66	**AVDD**	-1455.5	-350
31	**DSIO**/P36	1455.5	490	67	**AVDD**	-1455.5	-490
32	**TEST**	1455.5	630	68	**P03**/AIN3	-1455.5	-630
33	**DCLK**/P35	1455.5	770	69	**P02**/AIN2	-1455.5	-770
34	**P33**/REMI/SPICLK2	1455.5	910	70	**P01**/AIN1	-1455.5	-910
35	**P34**/REMO/#SPISS1	1455.5	1050	71	**P00**/AIN0	-1455.5	-1050
36	**P54**	1455.5	1190	72	N.C.	-1455.5	-1190

1.3.2 S1C17564 Pin Configuration Diagram

TQFP13-64pin (S1C17564)

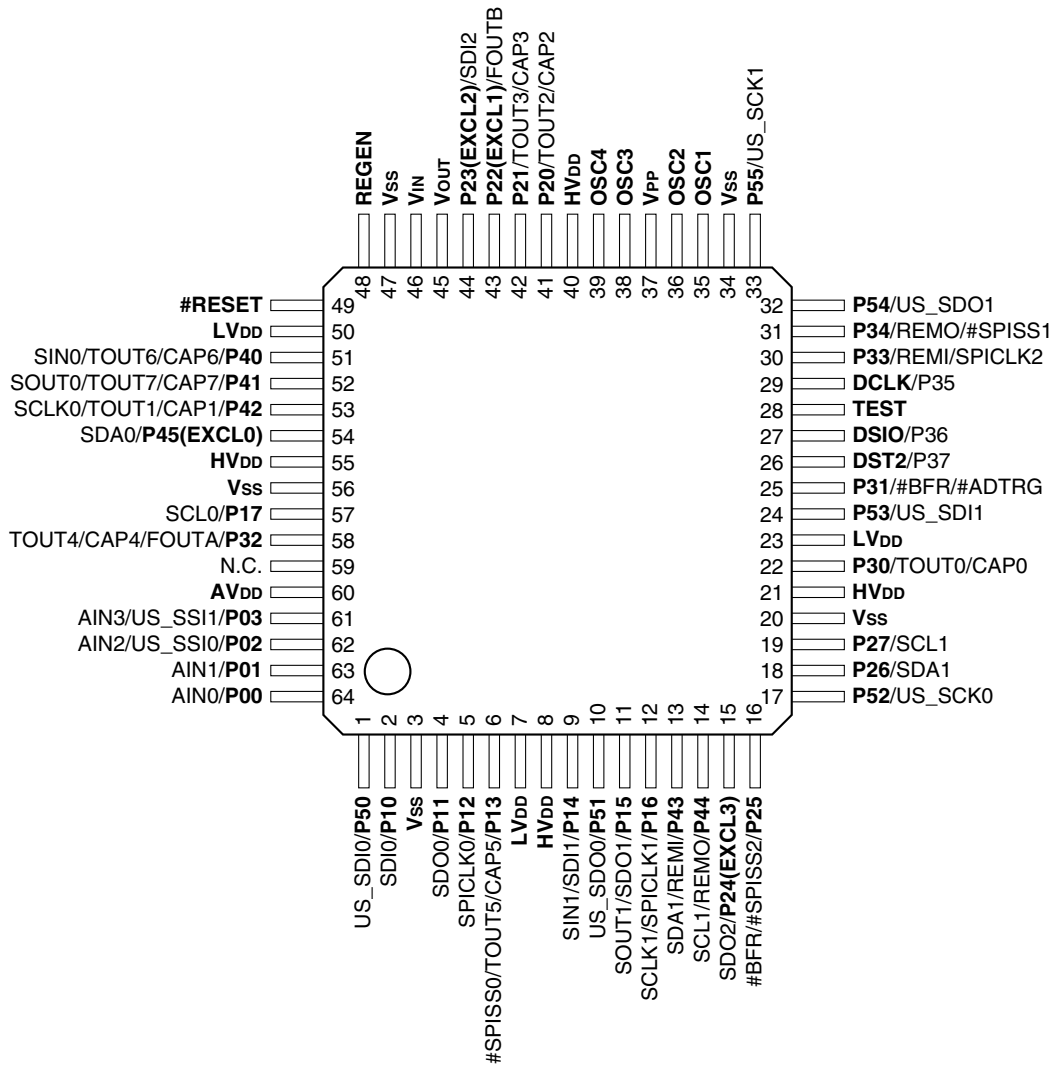


Figure 1.3.2.1 S1C17564 Pin Configuration Diagram (TQFP13-64pin)

Chip (S1C17564)

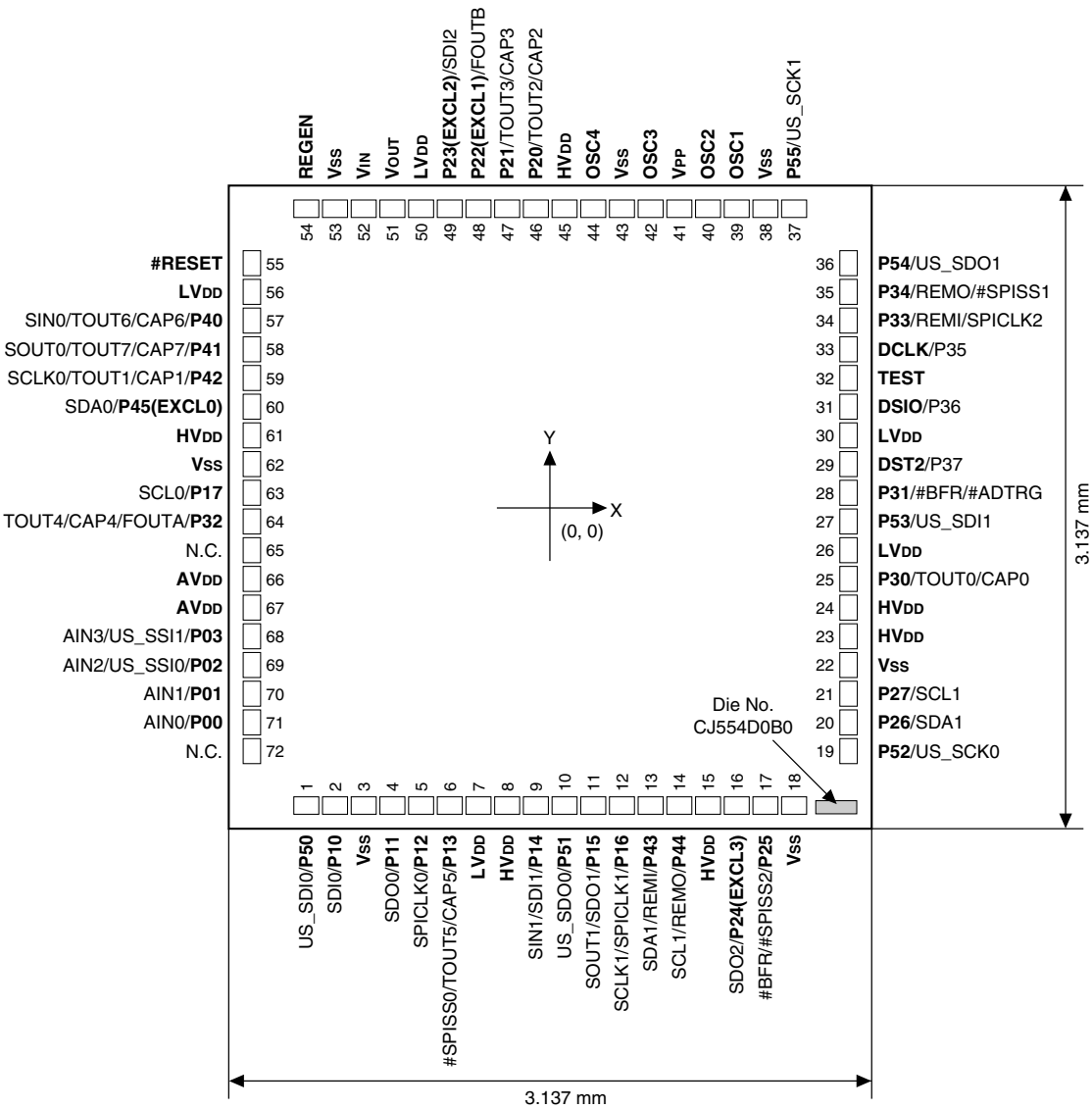


Figure 1.3.2.2 S1C17564 Pad Configuration Diagram

Chip size	X = 3.137 mm, Y = 3.137 mm
Pad opening	No. 1 to 18, 37 to 54: X = 122 µm, Y = 85 µm
	No. 19 to 36, 55 to 72: X = 85 µm, Y = 122 µm
Chip thickness	400 µm

Table 1.3.2.1 S1C17564 Pad Coordinates

No.	Name	X (mm)	Y (mm)	No.	Name	X (mm)	Y (mm)
1	**P50**/US_SDI0	-1190	-1455.5	37	**P55**/US_SCK1	1190	1455.5
2	**P10**/SDI0	-1050	-1455.5	38	**VSS**	1050	1455.5
3	**VSS**	-910	-1455.5	39	**OSC1**	910	1455.5
4	**P11**/SDO0	-770	-1455.5	40	**OSC2**	770	1455.5
5	**P12**/SPICLK0	-630	-1455.5	41	**VPP**	630	1455.5
6	**P13**/#SPISS0/TOUT5/CAP5	-490	-1455.5	42	**OSC3**	490	1455.5
7	**LVDD**	-350	-1455.5	43	**VSS**	350	1455.5
8	**HVDD**	-210	-1455.5	44	**OSC4**	210	1455.5
9	**P14**/SIN1/SDI1	-70	-1455.5	45	**HVDD**	70	1455.5
10	**P51**/US_SDO0	70	-1455.5	46	**P20**/TOUT2/CAP2	-70	1455.5
11	**P15**/SOUT1/SDO1	210	-1455.5	47	**P21**/TOUT3/CAP3	-210	1455.5
12	**P16**/SCLK1/SPICLK1	350	-1455.5	48	**P22(EXCL1)**/FOUTB	-350	1455.5
13	**P43**/SDA1/REMI	490	-1455.5	49	**P23(EXCL2)**/SDI2	-490	1455.5
14	**P44**/SCL1/REMO	630	-1455.5	50	**LVDD**	-630	1455.5
15	**HVDD**	770	-1455.5	51	**VOUT**	-770	1455.5
16	**P24(EXCL3)**/SDO2	910	-1455.5	52	**VIN**	-910	1455.5
17	**P25**/#BFR/SPISS2	1050	-1455.5	53	**VSS**	-1050	1455.5
18	**VSS**	1190	-1455.5	54	**REGEN**	-1190	1455.5
19	**P52**/US_SCK0	1455.5	-1190	55	**#RESET**	-1455.5	1190
20	**P26**/SDA1	1455.5	-1050	56	**LVDD**	-1455.5	1050
21	**P27**/SCL1	1455.5	-910	57	**P40**/SIN0/TOUT6/CAP6	-1455.5	910
22	**VSS**	1455.5	-770	58	**P41**/SOUT0/TOUT7/CAP7	-1455.5	770
23	**HVDD**	1455.5	-630	59	**P42**/SCLK0/TOUT1/CAP1	-1455.5	630
24	**HVDD**	1455.5	-490	60	**P45(EXCL0)**/SDA0	-1455.5	490
25	**P30**/TOUT0/CAP0	1455.5	-350	61	**HVDD**	-1455.5	350
26	**LVDD**	1455.5	-210	62	**VSS**	-1455.5	210
27	**P53**/US_SDI1	1455.5	-70	63	**P17**/SCL0	-1455.5	70
28	**P31**/#BFR/ADTRG	1455.5	70	64	**P32**/TOUT4/CAP4/FOUTA	-1455.5	-70
29	**DST2**/P37	1455.5	210	65	N.C.	-1455.5	-210
30	**LVDD**	1455.5	350	66	**AVDD**	-1455.5	-350
31	**DSIO**/P36	1455.5	490	67	**AVDD**	-1455.5	-490
32	**TEST**	1455.5	630	68	**P03**/AIN3/US_SSI1	-1455.5	-630
33	**DCLK**/P35	1455.5	770	69	**P02**/AIN2/US_SSI0	-1455.5	-770
34	**P33**/REMI/SPICLK2	1455.5	910	70	**P01**/AIN1	-1455.5	-910
35	**P34**/REMO/#SPISS1	1455.5	1050	71	**P00**/AIN0	-1455.5	-1050
36	**P54**/US_SDO1	1455.5	1190	72	N.C.	-1455.5	-1190

1.3.3 Pin Descriptions

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

Table 1.3.3.1 Pin Descriptions

Name	I/O	Default status	Function	Available (○)/unavailable (–)		
				S1C17554		S1C17564
				WCSP	TQFP/ Chip	TQFP/ Chip
HV_{DD}	–	–	I/O power supply pins (1.65 to 5.5 V)	○	○	○
LV_{DD}	–	–	Core power supply pins (1.65 to 1.95 V)	○	○	○
V_{SS}	–	–	GND pins	○	○	○
V_{PP}	–	–	Flash programming/erasing power supply pin (7/7.5 V) (Leave the pin open during normal operation.)	○	○	○
AV_{DD}	–	–	Analog power supply pin (2.7 to 5.5 V)	○	○	○
V_{IN}	–	–	Regulator input pin (2.0 to 5.5 V)	–	–	○
V_{OUT}	–	–	Regulator output pin (1.8 V)	–	–	○
REGEN	I	I	Regulator enable input pin	–	–	○
OSC3	I	I	OSC3 oscillator input or external clock (LV_{DD} level) input pin	○	○	○
OSC4	O	O	OSC3 oscillator output pin	○	○	○
OSC1	I	I	OSC1 oscillator input or external clock (LV_{DD} level) input pin	○	○	○
OSC2	O	O	OSC1 oscillator output pin	○	○	○
#RESET	I	I(Pull-up)	Initial reset input pin	○	○	○
TEST	I	I(Pull-down)	Test input pin (Connect to V_{SS} for normal operation.)	○	○	○
P00	I/O	I(Pull-up)	I/O port pin	○	○	○
AIN0	I		A/D converter Ch.0 analog signal input pin			
P01	I/O	I(Pull-up)	I/O port pin	○	○	○
AIN1	I		A/D converter Ch.1 analog signal input pin			
P02	I/O	I(Pull-up)	I/O port pin	○	○	○
AIN2	I		A/D converter Ch.2 analog signal input pin			
US_SSI0	I/O		USI Ch.0 data input/output pin (S1C17564)	–	–	
P03	I/O	I(Pull-up)	I/O port pin	○	○	○
AIN3	I		A/D converter Ch.3 analog signal input pin			
US_SSI1	I/O		USI Ch.1 data input/output pin (S1C17564)	–	–	
P10	I/O	I(Pull-up)	I/O port pin	○	○	○
SDI0	I		SPI Ch.0 data input pin			
P11	I/O	I(Pull-up)	I/O port pin	○	○	○
SDO0	O		SPI Ch.0 data output pin			
P12	I/O	I(Pull-up)	I/O port pin	○	○	○
SPICLK0	I/O		SPI Ch.0 clock input/output pin			
P13	I/O	I(Pull-up)	I/O port pin	○	○	○
#SPISS0	I		SPI Ch.0 slave select signal input pin			
TOUT5	O		T16A Ch.2 TOUT B signal output pin			
CAP5	I		T16A Ch.2 capture B trigger signal input pin			
P14	I/O	I(Pull-up)	I/O port pin	○	○	○
SIN1	I		UART Ch.1 data input pin			
SDI1	I		SPI Ch.1 data input pin			
P15	I/O	I(Pull-up)	I/O port pin	○	○	○
SOUT1	O		UART Ch.1 data output pin			
SDO1	O		SPI Ch.1 data output pin			
P16	I/O	I(Pull-up)	I/O port pin	○	○	○
SCLK1	I		UART Ch.1 external clock input pin			
SPICLK1	I/O		SPI Ch.1 clock input/output pin			
P17	I/O	I(Pull-up)	I/O port pin	○	○	○
SCL0	I/O		I^2C master SCL input/output pin			
P20	I/O	I(Pull-up)	I/O port pin	○	○	○
TOUT2	O		T16A Ch.1 TOUT A signal output pin			
CAP2	I		T16A Ch.1 capture A trigger signal input pin			
P21	I/O	I(Pull-up)	I/O port pin	○	○	○
TOUT3	O		T16A Ch.1 TOUT B signal output pin			
CAP3	I		T16A Ch.1 capture B trigger signal input pin			
P22 (EXCL1)	I/O	I(Pull-up)	I/O port pin (T16A Ch.1 external clock input pin)	○	○	○
FOUTB	O		Clock output pin			
P23 (EXCL2)	I/O	I(Pull-up)	I/O port pin (T16A Ch.2 external clock input pin)	○	○	○
SDI2	I		SPI Ch.2 data input pin			
P24 (EXCL3)	I/O	I(Pull-up)	I/O port pin (T16A Ch.3 external clock input pin)	○	○	○
SDO2	O		SPI Ch.2 data output pin			

Name	I/O	Default status	Function	Available (○)/unavailable (–)		
				S1C17554		S1C17564
				WCSP	TQFP/ Chip	TQFP/ Chip
P25	I/O	I(Pull-up)	I/O port pin	○	○	○
#BFR	I		I²C slave bus free request input pin			
#SPISS2	I		SPI Ch.2 slave select signal input pin			
P26	I/O	I(Pull-up)	I/O port pin	○	○	○
SDA1	I/O		I²C slave data input/output pin			
P27	I/O	I(Pull-up)	I/O port pin	○	○	○
SCL1	I/O		I²C slave SCL input/output pin			
P30	I/O	I(Pull-up)	I/O port pin	○	○	○
TOUT0	O		T16A Ch.0 TOUT A signal output pin			
CAP0	I		T16A Ch.0 capture A trigger signal input pin			
P31	I/O	I(Pull-up)	I/O port pin	○	○	○
#BFR	I		I²C slave bus free request input pin			
#ADTRG	I		A/D converter external trigger input pin			
P32	I/O	I(Pull-up)	I/O port pin	○	○	○
TOUT4	O		T16A Ch.2 TOUT A signal output pin			
CAP4	I		T16A Ch.2 capture A trigger signal input pin			
FOUTA	O		Clock output pin			
P33	I/O	I(Pull-up)	I/O port pin	○	○	○
REMI	I		REMC input pin			
SPICLK2	I/O		SPI Ch.2 clock input/output pin			
P34	I/O	I(Pull-up)	I/O port pin	○	○	○
REMO	O		REMC output pin			
#SPISS1	I		SPI Ch.1 slave select signal input pin			
DCLK	O	O(H)	On-chip debugger clock output pin	○	○	○
P35	I/O		I/O port pin			
DSIO	I/O	I(Pull-up)	On-chip debugger data input/output pin	○	○	○
P36	I/O		I/O port pin			
DST2	O	O(L)	On-chip debugger status output pin	○	○	○
P37	I/O		I/O port pin			
P40	I/O	I(Pull-up)	I/O port pin	○	○	○
SIN0	I		UART Ch.0 data input pin			
TOUT6	O		T16A Ch.3 TOUT A signal output pin			
CAP6	I		T16A Ch.3 capture A trigger signal input pin			
P41	I/O	I(Pull-up)	I/O port pin	○	○	○
SOUT0	O		UART Ch.0 data output pin			
TOUT7	O		T16A Ch.3 TOUT B signal output pin			
CAP7	I		T16A Ch.3 capture B trigger signal input pin			
P42	I/O	I(Pull-up)	I/O port pin	○	○	○
SCLK0	I		UART Ch.0 external clock input pin			
TOUT1	O		T16A Ch.0 TOUT B signal output pin			
CAP1	I		T16A Ch.0 capture B trigger signal input pin			
P43	I/O	I(Pull-up)	I/O port pin	○	○	○
SDA1	I/O		I²C slave data input/output pin			
REMI	I		REMC input pin			
P44	I/O	I(Pull-up)	I/O port pin	○	○	○
SCL1	I/O		I²C slave SCL input/output pin			
REMO	O		REMC output pin			
P45 (EXCL0)	I/O	I(Pull-up)	I/O port pin (T16A Ch.0 external clock input pin)	○	○	○
SDA0	I/O		I²C master data input/output pin			
P50	I/O	I(Pull-up)	I/O port pin	–	○	○
US_SDI0	I/O		USI Ch.0 data input/output pin (S1C17564)		–	
P51	I/O	I(Pull-up)	I/O port pin	–	○	○
US_SDO0	O		USI Ch.0 data output pin (S1C17564)		–	
P52	I/O	I(Pull-up)	I/O port pin	–	○	○
US_SCK0	I/O		USI Ch.0 clock input/output pin (S1C17564)		–	
P53	I/O	I(Pull-up)	I/O port pin	–	○	○
US_SDI1	I/O		USI Ch.1 data input/output pin (S1C17564)		–	
P54	I/O	I(Pull-up)	I/O port pin	–	○	○
US_SDO1	O		USI Ch.1 data output pin (S1C17564)		–	
P55	I/O	I(Pull-up)	I/O port pin	–	○	○
US_SCK1	I/O		USI Ch.1 clock input/output pin (S1C17564)		–	

2 CPU

The S1C17554/564 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.

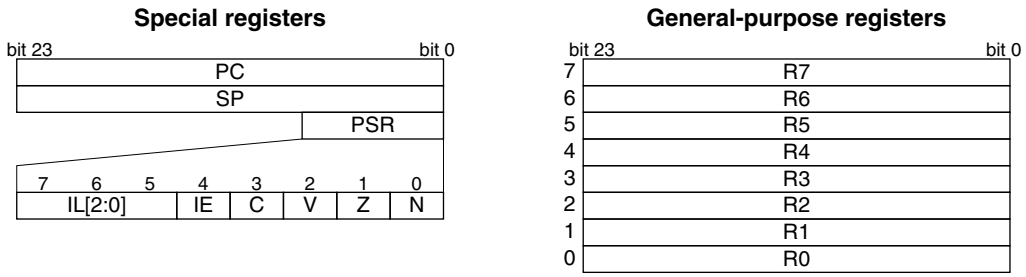


Figure 2.2.1 Registers

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 S1C17554/564 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 S1C17554/564 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 S1C17554/564 memory map.

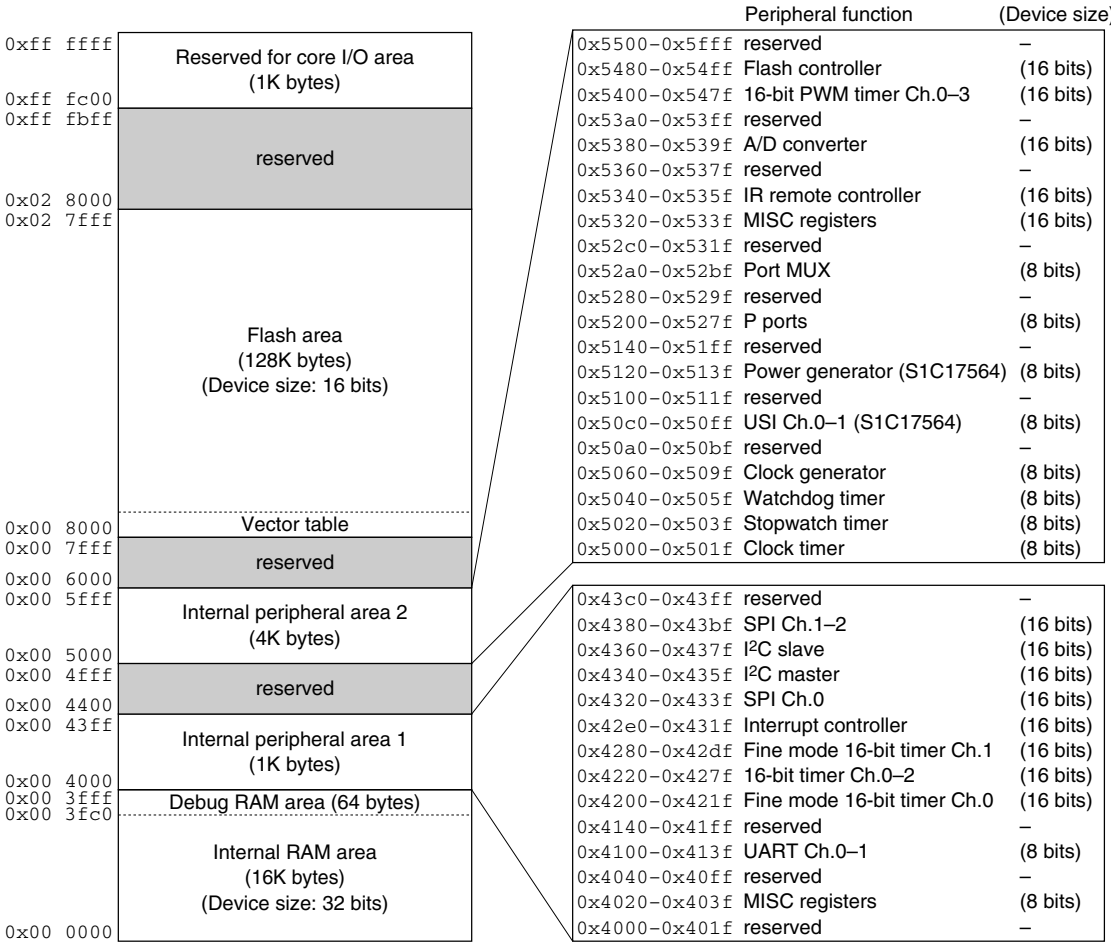


Figure 3.1 S1C17554/564 Memory Map

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.
Note that the Flash area and other areas require different number of system clocks for one bus cycle as follows:

Instruction/data read from areas other than the Flash area:
One system clock per one bus cycle
Instruction read from the Flash area:
One to three system clocks or equivalent per one bus cycle
Data read from the Flash area:
Two to four system clocks per one bus cycle

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 S1C17554/564 executes the instruction stored in the Flash area and accesses data in the Flash area
- When the S1C17554/564 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 128K-byte area from address 0x8000 to address 0x27fff contains a Flash memory (4K bytes × 32 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 S1C17554/564 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 16K-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
0x27ffc (16 bits)	D15–8	reserved	–				–	–	
	D7	reserved	1				1	R/W	Always set to 1.
	D6	Flash write-protect bit for 0x20000–0x23fff	1	Writable	0	Protected	1	R/W	
	D5	Flash write-protect bit for 0x1c000–0x1ffff	1	Writable	0	Protected	1	R/W	
	D4	Flash write-protect bit for 0x18000–0x1bfff	1	Writable	0	Protected	1	R/W	
	D3	Flash write-protect bit for 0x14000–0x17fff	1	Writable	0	Protected	1	R/W	
	D2	Flash write-protect bit for 0x10000–0x13fff	1	Writable	0	Protected	1	R/W	
	D1	Flash write-protect bit for 0xc000–0xffff	1	Writable	0	Protected	1	R/W	
	D0	Flash write-protect bit for 0x8000–0xbfff	1	Writable	0	Protected	1	R/W	
0x27ffe (16 bits)	D15–8	reserved	–				–	–	
	D7	Flash data-read-protect bit for 0x24000–0x27fff	1	Readable	0	Protected	1	R/W	
	D6	Flash data-read-protect bit for 0x20000–0x23fff	1	Readable	0	Protected	1	R/W	
	D5	Flash data-read-protect bit for 0x1c000–0x1ffff	1	Readable	0	Protected	1	R/W	
	D4	Flash data-read-protect bit for 0x18000–0x1bfff	1	Readable	0	Protected	1	R/W	
	D3	Flash data-read-protect bit for 0x14000–0x17fff	1	Readable	0	Protected	1	R/W	
	D2	Flash data-read-protect bit for 0x10000–0x13fff	1	Readable	0	Protected	1	R/W	
	D1	Flash data-read-protect bit for 0xc000–0xffff	1	Readable	0	Protected	1	R/W	
	D0	reserved	1				1	R/W	Always set to 1.

Notes: • The protection can be disabled by erasing the sector that includes the protect bits.
• Be sure not to locate the area with data-read protection into the .data and .rodata sections.
• Be sure to set D0 of address 0x27ffe 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)	D15–2	–	reserved	–		–	–	0 when being read.
		D1–0	**RDWAIT [1:0]**	Flash read wait cycle	RDWAIT[1:0]	Wait	0x3	R/W	
					0x3	2 wait			
					0x2	1 wait			
					0x1	No wait			
					0x0	reserved			

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. For the configurable wait cycles, see the "Electrical Characteristics" chapter.

Bus cycle when "no wait" is selected
Instruction read: 1 bus cycle = 1 system clock cycle or equivalent
Data read: 1 bus cycle = 2 system clock cycles

Note: Be sure to avoid setting a number of wait cycles that exceeds the maximum allowable system clock frequency, as it may cause a malfunction.

3.3 Internal RAM Area

3.3.1 Embedded RAM

The S1C17554/564 contains a RAM in the 16K-byte area from address 0x0 to address 0x3fff. 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 (0x3fc0–0x3fff) 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 S1C17554/564 enables the RAM size used to apply restrictions to 16KB, 12KB, 8KB, 4KB, 2KB, 1KB, or 512B. For example, when using the S1C17554/564 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	0x6 (= 16KB)	0x6	R	
		D3	–	reserved	–	–	–	0 when being read.
		D2–0	IRAMSZ[2:0]	IRAM size select	IRAMSZ[2:0] / Size: 0x7 reserved; 0x6 16KB; 0x5 512B; 0x4 1KB; 0x3 2KB; 0x2 4KB; 0x1 8KB; 0x0 12KB	0x6	R/W	

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

Indicates the actual internal RAM size embedded. (Default: 0x6)

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
0x7	Reserved
0x6	16KB
0x5	512B
0x4	1KB
0x3	2KB
0x2	4KB
0x1	8KB
0x0	12KB

(Default: 0x6)

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

3.4 Internal Peripheral Area

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

3.4.1 Internal Peripheral Area 1 (0x4000–)

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

- MISC register (MISC, 8-bit device)
- UART (UART, 8-bit device)
- Fine mode 16-bit timers (T16F, 16-bit device)
- 16-bit timers (T16, 16-bit device)
- Interrupt controller (ITC, 16-bit device)
- SPI (SPI, 16-bit device)
- I^2C master (I2CM, 16-bit device)
- I^2C slave (I2CS, 16-bit device)

3.4.2 Internal Peripheral Area 2 (0x5000–)

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

- Clock timer (CT, 8-bit device)
- Stopwatch timer (SWT, 8-bit device)
- Watchdog timer (WDT, 8-bit device)
- Clock generator (CLG, 8-bit device)
- Universal serial interface (USI, 8-bit device) Available only in S1C17564
- Power generator (VD1, 8-bit device) Available only in S1C17564
- I/O port & port MUX (P, 8-bit device)
- MISC register (MISC, 16-bit device)
- IR remote controller (REMC, 16-bit device)
- A/D converter (ADC10, 16-bit device)
- 16-bit PWM timers (T16A, 16-bit device)
- Flash controller (FLASHC, 16-bit device)

3.5 S1C17 Core I/O Area

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

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

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

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

4 Power Supply

4.1 Core Power Supply Voltage (LV_{DD})

The CPU core and internal logic circuits operate with a voltage supplied between the LV_{DD} and V_{SS} pins. Supply a voltage within the range shown below to the LV_{DD} pins with the V_{SS} pins as the GND level.

LV_{DD} = 1.65 V to 1.95 V (V_{SS} = GND)

The S1C17554/564 provides two or more LV_{DD} and V_{SS} pins. Do not leave any power supply pins open and be sure to connect them to + power source and GND.

The S1C17564 is able to operate with a 2.0 V or higher single power supply (e.g., 3.3 V or 5 V). In this case, the core power voltage can be generated using the embedded 1.8 V regulator and supplied as the LV_{DD} voltage.

4.2 I/O Power Supply Voltage (HV_{DD})

The HV_{DD} voltage is used for interfacing with external I/O signals. Supply a voltage within the range shown below to the HV_{DD} pins with the V_{SS} pins as the GND level.

HV_{DD} = 1.65 V to 5.50 V (V_{SS} = GND)

The S1C17554/564 provides two or more HV_{DD} pins. Do not leave any power supply pins open and be sure to connect them to the power source.

Notes: • When an external clock is input to the OSC3 or OSC1 pin, the clock signal level must be LV_{DD}.

• AV_{DD} is supplied to the P00 to P03 (AIN0 to AIN3) ports as their power source (see Section 4.3).

4.3 Analog Power Supply Voltage (AV_{DD})

The analog power supply pin (AV_{DD}) is provided separately from the HV_{DD} pin in order that the digital circuits do not affect the analog circuits (A/D converter). Supply a voltage within the range shown below to the AV_{DD} pins with the V_{SS} pins as the GND level.

AV_{DD} = 2.7 V to 5.5 V (V_{SS} = GND) when the A/D converter is used
or
AV_{DD} = 1.65 V to 5.5 V (V_{SS} = GND) when the A/D converter is not used

Notes: • The lower limit AV_{DD} voltage is 1.65 V the same as HV_{DD} when the A/D converter is not used. Supply the same voltage as HV_{DD} to the AV_{DD} pin.

• AV_{DD} is supplied to the P00 to P03 (AIN0 to AIN3) ports as their power source. When both the A/D converter and a port within P00 to P03 are used simultaneously, reduce the port switching operations minimum so that the digital circuit will not affect A/D conversion results.

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

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

The V_{PP} voltage is used for erasing/programming 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/erase 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.

4.5 Embedded Regulator (S1C17564)

The S1C17564 includes a voltage regulator that allows use of a 2 V or higher single power supply for IC operations.

Regulator input (V_{IN}): 2.0 V to 5.5 V
Regulator output (V_{OUT}): 1.8 V

When a single power supply is used, supply the voltage within the range shown above to the V_{IN}, HV_{DD}, and AV_{DD} pins. Also the same voltage should be supplied to the REGEN pin to enable the regulator to operate. The regulator generates a 1.8 V (typ.) voltage and outputs it to the V_{OUT} pin. By supplying the output voltage to the LV_{DD} pin, the core and internal circuits can be operated.
For power supply connection examples, see the "Basic External Connection Diagram" chapter.
The embedded regulator supports economy mode (power saving mode) to reduce current consumption in the regulator during low-speed (32 kHz) operation or standby mode (HALT or SLEEP).

4.6 Control Register Details (S1C17564)

Table 4.6.1 Power Control Register

Address	Register name		Function
0x5121	VD1_CTL	V_{D1} Control Register	Controls the regulator operation mode.

The power control register (S1C17564) is described in detail below.

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

V_{D1} Control Register (VD1_CTL)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
V_{D1} Control Register (VD1_CTL)	**0x5121** (8 bits)	D7–2	–	reserved	–		–	–	0 when being read.
		D1–0	**VD1ECO [1:0]**	Regulator operation mode select	VD1ECO[1:0]	Mode	0x0	R/W	
					0x3	reserved			
					0x2	Auto-control			
S1C17564					0x1	Economy			
					0x0	Normal			

D[7:2] Reserved

D[1:0] VD1ECO[1:0]: Regulator Operation Mode Select Bits

Sets the operation mode of the embedded regulator.

Table 4.6.2 Regulator Operation Mode

VD1ECO[1:0]	Operation mode
0x3	Reserved
0x2	Auto-control mode
0x1	Economy (power saving) mode
0x0	Normal mode

(Default: 0x0)

For normal operation (while the IC is operating with a clock other than OSC1), set VD1ECO[1:0] to 0x0 (default).
Setting VD1ECO[1:0] to 0x1 places the regulator into economy mode. The economy mode can reduce current consumption, note, however, that the regulator may become unstable due to changes of load that cannot be responded. The regulator should be set to economy mode before the slp instruction is executed.
Setting VD1ECO[1:0] to 0x2 places the regulator into auto-control mode. In this mode, the hardware automatically performs switching between normal mode and economy mode according to changes of load. The regulator enters economy mode when both IOSC and OSC3 are stopped or when the IC enters SLEEP mode, or enters normal mode otherwise.
While the IC is operating with a high-speed clock (other than OSC1), the regulator should not be set to economy mode, as the regulator output voltage may become unstable. Even if the IC is operating with OSC1, set the regulator to normal mode when driving a heavy load such as a lamp or buzzer.

4.7 Precautions on Power Supply

Power-on sequence

In order to operate the device normally, supply power in the following sequence:

Power-on: LV_{DD} → HV_{DD} (I/O), AV_{DD} (A/D) → Apply the input signal
or LV_{DD}, HV_{DD} (I/O), AV_{DD} (A/D) → Apply the input signal
(See Notes in "Power-off sequence" below.)

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

Power-off sequence

Shut off the power supply in the following sequence:

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

Notes:
- Applying only LV_{DD} with other power voltage turned off puts the HV_{DD} system circuits into unstable status and unstable current flows in the I/O cells. Be sure to avoid applying only LV_{DD} for a duration of one second or more.
- Be sure to avoid applying HV_{DD} or AV_{DD} for a duration of one second or more when the LV_{DD} power is off, as a breakdown may occur in the device or the characteristics may be degraded due to flow-through current of the HV_{DD} or AV_{DD}.

Latch-up

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

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

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

5 Initial Reset

5.1 Initial Reset Sources

The S1C17554/564 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.

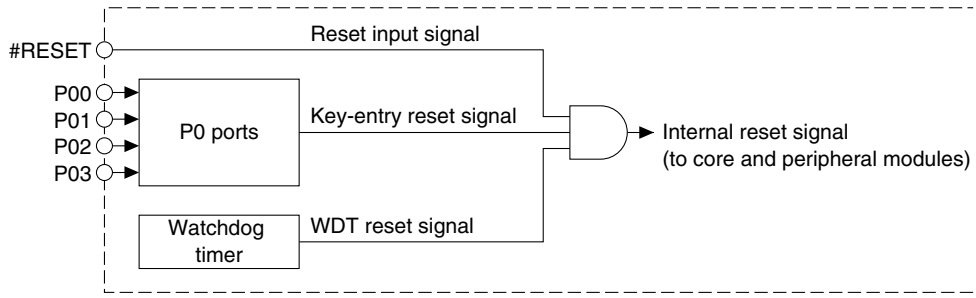


Figure 5.1.1 Configuration of Initial Reset Circuit

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 S1C17554/564 enters initial reset state. In order to initialize the S1C17554/564 for sure, the #RESET pin must be held at low for more than the prescribed time (see "AC Characteristics" in the "Electrical Characteristics" chapter) after the power supply voltage is supplied.
Initial reset state is canceled when the #RESET pin at low level is set to high level and the CPU starts executing the reset interrupt handler.
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 S1C17554/564 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(*), system clock internal supply start time (8 cycles), and Flash reset cancellation time (16 cycles) have elapsed.
Figure 5.2.1 shows the operating sequence following cancellation of initial reset.
The CPU starts operating in synchronization with the OSC3 or IOSC (S1C17564 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.

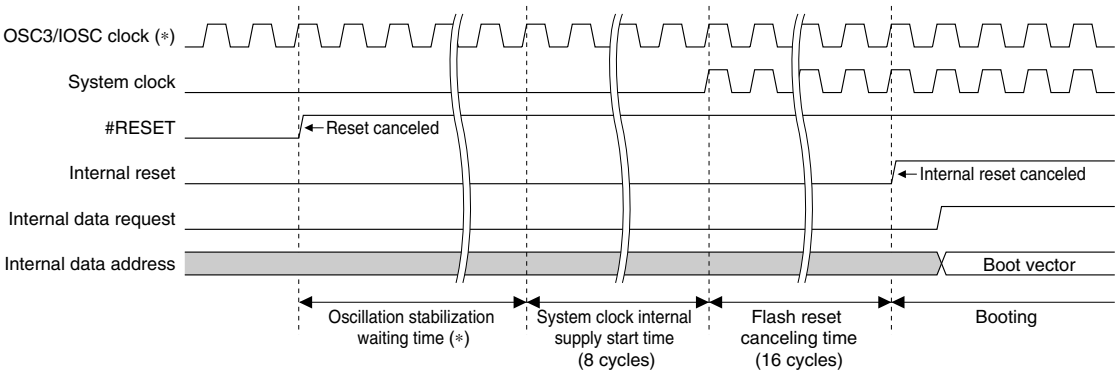


Figure 5.2.1 Operation Sequence Following Cancellation of Initial Reset

* Booting clock and oscillation stabilization waiting time
  S1C17554: Boots up with the OSC3 clock. The oscillation stabilization waiting time is configured to 1,024 cycles (OSC3 clock).
  S1C17564: Boots up with the IOSC clock. The oscillation stabilization waiting time is configured to 64 cycles (IOSC clock).

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 and display memory should be initialized with software as they are 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 23 maskable interrupt systems (for 26 interrupt sources listed below).
  1. P00–P03 input interrupt (4 types)
  2. P10–P17 input interrupt (8 types)
  3. P20–P27 input interrupt (8 types)
  4. P30–P37 input interrupt (8 types)
  5. P40–P45 input interrupt (6 types)
  6. P50–P55 input interrupt (6 types) ∗ Cannot be used in the S1C17554 WCSP-48 package.
  7. Stopwatch timer interrupt (3 types)
  8. Clock timer interrupt (4 types)
  9. 16-bit PWM timer Ch.0 interrupt (6 types)
  10. 16-bit PWM timer Ch.1 interrupt (6 types)
  11. 16-bit PWM timer Ch.2 interrupt (6 types)
  12. 16-bit PWM timer Ch.3 interrupt (6 types)
  13. Fine mode 16-bit timer Ch.0 & Ch.1 interrupt (2 types)
  14. 16-bit timer Ch.0 interrupt (1 type)
  15. 16-bit timer Ch.1 interrupt (1 type)
  16. 16-bit timer Ch.2 interrupt (1 type)
  17. USI Ch.0 & Ch.1 interrupt (6 types) ∗ Cannot be used in the S1C17554.
  18. UART Ch.0 interrupt (4 types)
  19. UART Ch.1 interrupt (4 types)
  20. IR remote controller interrupt (3 types)
  21. SPI Ch.0 interrupt (2 types)
  22. SPI Ch.1 interrupt (2 types)
  23. SPI Ch.2 interrupt (2 types)
  24. I^2C master interrupt (2 types)
  25. I^2C slave interrupt (3 types)
  26. A/D converter 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.

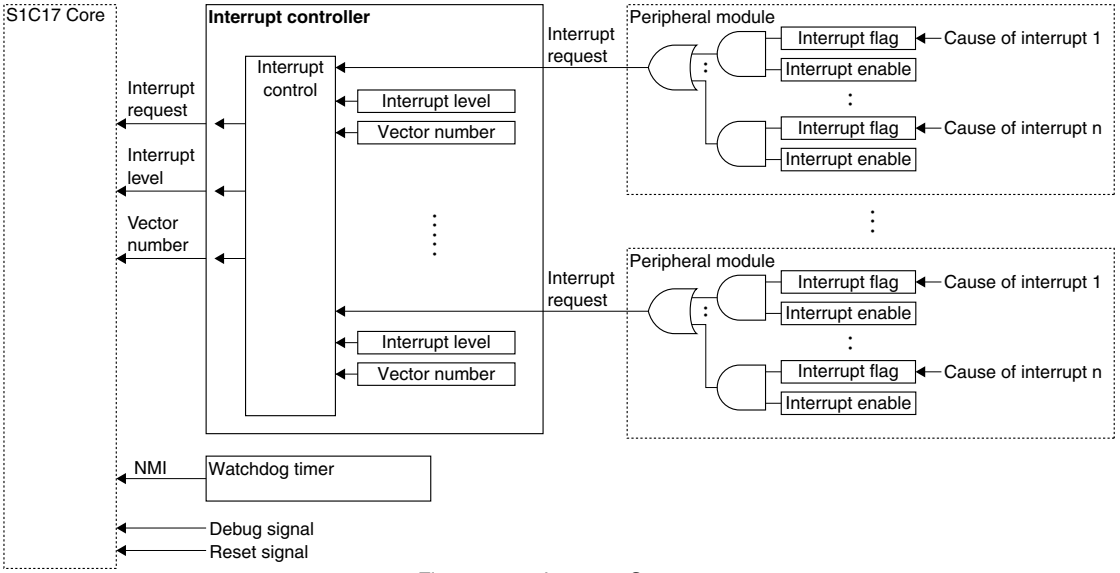


Figure 6.1.1 Interrupt System

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 S1C17554/564.

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–P03 port inputs	High *1
5 (0x05)	TTBR + 0x14	P1 port interrupt	P10–P17 port inputs	↑
6 (0x06)	TTBR + 0x18	Stopwatch timer interrupt	• 100 Hz timer signal • 10 Hz timer signal • 1 Hz timer signal	
7 (0x07)	TTBR + 0x1c	Clock timer interrupt	• 32 Hz timer signal • 8 Hz timer signal • 2 Hz timer signal • 1 Hz timer signal	
8 (0x08)	TTBR + 0x20	16-bit PWM timer Ch.2 interrupt	• Compare A/B • Capture A/B • Capture A/B overwrite	
9 (0x09)	TTBR + 0x24	P4 port interrupt	P40–P45 port inputs	
10 (0x0a)	TTBR + 0x28	SPI Ch.2 interrupt	• Transmit buffer empty • Receive buffer full	
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	Fine mode 16-bit timer Ch.0 and Ch.1 interrupt	• Ch.0 underflow • Ch.1 underflow	
		USI Ch.0 and Ch.1 interrupt	• Ch.0 transmit buffer empty • Ch.0 receive buffer full • Ch.0 receive error • Ch.1 transmit buffer empty • Ch.1 receive buffer full • Ch.1 receive error	
13 (0x0d)	TTBR + 0x34	16-bit timer Ch.0 interrupt	Timer underflow	

Vector No. Software interrupt No.	Vector address	Hardware interrupt name	Cause of hardware interrupt	Priority
14 (0x0e)	TTBR + 0x38	16-bit timer Ch.1 interrupt	Timer underflow	
15 (0x0f)	TTBR + 0x3c	16-bit timer Ch.2 interrupt	Timer underflow	
		16-bit PWM timer Ch.3 interrupt	• Compare A/B • Capture A/B • Capture A/B overwrite	
16 (0x10)	TTBR + 0x40	UART Ch.0 interrupt	• Transmit buffer empty • End of transmission • Receive buffer full • Receive error	
17 (0x11)	TTBR + 0x44	UART Ch.1 interrupt	• Transmit buffer empty • End of transmission • Receive buffer full • Receive error	
18 (0x12)	TTBR + 0x48	SPI Ch.0 interrupt	• Transmit buffer empty • Receive buffer full	
19 (0x13)	TTBR + 0x4c	I^2C Master interrupt	• Transmit buffer empty • Receive buffer full	
20 (0x14)	TTBR + 0x50	IR remote controller interrupt	• Data length counter underflow • Input rising edge detected • Input falling edge detected	
		SPI Ch.1 interrupt	• Transmit buffer empty • Receive buffer full	
21 (0x15)	TTBR + 0x54	16-bit PWM timer Ch.1 interrupt	• Compare A/B • Capture A/B • Capture A/B overwrite	
22 (0x16)	TTBR + 0x58	A/D converter interrupt	• Conversion completion • Conversion result overwrite	
23 (0x17)	TTBR + 0x5c	P5 port interrupt	P50–P55 port inputs	
24 (0x18)	TTBR + 0x60	P2 port interrupt	P20–P27 port inputs	
25 (0x19)	TTBR + 0x64	P3 port interrupt	P30–P37 port inputs	
26 (0x1a)	TTBR + 0x68	I^2C Slave interrupt	• Transmit buffer empty • Receive buffer full • Bus status	
27 (0x1b) : 31 (0x1f)	TTBR + 0x6c : TTBR + 0x7c	reserved : reserved	– : –	↓ Low *1

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

Vector numbers 4 to 26 are assigned to the maskable interrupts supported by the S1C17554/564.

Interrupts that share an interrupt vector

Interrupt vector numbers 12, 15, and 20 are shared with two different interrupt modules.
Interrupt vector 12: Fine mode 16-bit timer Ch.0/Ch.1 and USI Ch.0/Ch.1
Interrupt vector 15: 16-bit timer Ch.2 and 16-bit PWM timer Ch.3
Interrupt vector 20: IR remote controller and SPI Ch.1

The interrupt signals from the two modules are input to the ITC through an OR gate. When using the two interrupts, check if which interrupt has occurred by reading the interrupt flags in both modules.
The two modules cannot be set to different interrupt level, as they use the same interrupt vector.

Vector table base address

The S1C17554/564 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
P1 port interrupt	ILV1[2:0] (D[10:8]/ITC_LV0 register)	0x4306
Stopwatch timer interrupt	ILV2[2:0] (D[2:0]/ITC_LV1 register)	0x4308
Clock timer interrupt	ILV3[2:0] (D[10:8]/ITC_LV1 register)	0x4308
16-bit PWM timer Ch.2 interrupt	ILV4[2:0] (D[2:0]/ITC_LV2 register)	0x430a
P4 port interrupt	ILV5[2:0] (D[10:8]/ITC_LV2 register)	0x430a
SPI Ch.2 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
Fine mode 16-bit timer Ch.0 & Ch.1 interrupt / USI Ch.0 & Ch.1 interrupt	ILV8[2:0] (D[2:0]/ITC_LV4 register)	0x430e
16-bit timer Ch.0 interrupt	ILV9[2:0] (D[10:8]/ITC_LV4 register)	0x430e
16-bit timer Ch.1 interrupt	ILV10[2:0] (D[2:0]/ITC_LV5 register)	0x4310
16-bit timer Ch.2 interrupt / 16-bit PWM timer Ch.3 interrupt	ILV11[2:0] (D[10:8]/ITC_LV5 register)	0x4310
UART Ch.0 interrupt	ILV12[2:0] (D[2:0]/ITC_LV6 register)	0x4312
UART Ch.1 interrupt	ILV13[2:0] (D[10:8]/ITC_LV6 register)	0x4312
SPI Ch.0 interrupt	ILV14[2:0] (D[2:0]/ITC_LV7 register)	0x4314
I^2C master interrupt	ILV15[2:0] (D[10:8]/ITC_LV7 register)	0x4314
IR remote controller interrupt / SPI Ch.1 interrupt	ILV16[2:0] (D[2:0]/ITC_LV8 register)	0x4316
16-bit PWM timer Ch.1 interrupt	ILV17[2:0] (D[10:8]/ITC_LV8 register)	0x4316
A/D converter interrupt	ILV18[2:0] (D[2:0]/ITC_LV9 register)	0x4318
P5 port interrupt	ILV19[2:0] (D[10:8]/ITC_LV9 register)	0x4318
P2 port interrupt	ILV20[2:0] (D[2:0]/ITC_LV10 register)	0x431a
P3 port interrupt	ILV21[2:0] (D[10:8]/ITC_LV10 register)	0x431a
I^2C slave interrupt	ILV22[2:0] (D[2:0]/ITC_LV11 register)	0x431c

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 S1C17554/564, the watchdog timer can generate a non-maskable interrupt (NMI). The vector number for NMI is 2, with the vector address set to the vector table's starting address + 8 bytes.
This interrupt takes precedence over other interrupts and is unconditionally accepted by the S1C17 Core.

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

6.5 Software Interrupts

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

6.6 HALT and SLEEP Mode Cancellation

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

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

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

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

6.7 Control Register Details

Table 6.7.1 List of ITC Registers

Address	Register name		Function
0x4306	ITC_LV0	Interrupt Level Setup Register 0	Sets the P0 and P1 interrupt levels.
0x4308	ITC_LV1	Interrupt Level Setup Register 1	Sets the SWT and CT interrupt levels.
0x430a	ITC_LV2	Interrupt Level Setup Register 2	Sets the T16A Ch.2 and P4 interrupt levels.
0x430c	ITC_LV3	Interrupt Level Setup Register 3	Sets the SPI Ch.2 and T16A Ch.0 interrupt levels.
0x430e	ITC_LV4	Interrupt Level Setup Register 4	Sets the T16F Ch.0 & Ch.1/USI Ch.0 & Ch.1 and T16 Ch.0 interrupt levels.
0x4310	ITC_LV5	Interrupt Level Setup Register 5	Sets the T16 Ch.1 and T16 Ch.2/T16A Ch.3 interrupt levels.
0x4312	ITC_LV6	Interrupt Level Setup Register 6	Sets the UART Ch.0 and Ch.1 interrupt levels.
0x4314	ITC_LV7	Interrupt Level Setup Register 7	Sets the SPI Ch.0 and I2CM interrupt levels.
0x4316	ITC_LV8	Interrupt Level Setup Register 8	Sets the REMC/SPI Ch.1 and T16A Ch.1 interrupt levels.
0x4318	ITC_LV9	Interrupt Level Setup Register 9	Sets the ADC10 and P5 interrupt levels.
0x431a	ITC_LV10	Interrupt Level Setup Register 10	Sets the P2 and P3 interrupt levels.
0x431c	ITC_LV11	Interrupt Level Setup Register 11	Sets the I2CS interrupt level.

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

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

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 | 0x431c (16 bits)	D15–11	–	reserved	–	–	–	0 when being read.
		D10–8	ILV*n*[2:0]	*INTn* (1, 3, ... 21) interrupt level	0 to 7	0x0	R/W	
		D7–3	–	reserved	–	–	–	0 when being read.
		D2–0	ILV*n*[2:0]	*INTn* (0, 2, ... 22) 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* = 0–22)

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 0x431c) 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])	P1 port interrupt
ITC_LV1(0x4308)	ILV2[2:0] (D[2:0])	Stopwatch timer interrupt
	ILV3[2:0] (D[10:8])	Clock timer interrupt
ITC_LV2(0x430a)	ILV4[2:0] (D[2:0])	16-bit PWM timer Ch.2 interrupt
	ILV5[2:0] (D[10:8])	P4 port interrupt
ITC_LV3(0x430c)	ILV6[2:0] (D[2:0])	SPI Ch.2 interrupt
	ILV7[2:0] (D[10:8])	16-bit PWM timer Ch.0 interrupt
ITC_LV4(0x430e)	ILV8[2:0] (D[2:0])	Fine mode 16-bit timer Ch.0 & Ch.1 interrupt / USI Ch.0 & Ch.1 interrupt
	ILV9[2:0] (D[10:8])	16-bit timer Ch.0 interrupt
ITC_LV5(0x4310)	ILV10[2:0] (D[2:0])	16-bit timer Ch.1 interrupt
	ILV11[2:0] (D[10:8])	16-bit timer Ch.2 interrupt / 16-bit PWM timer Ch.3 interrupt
ITC_LV6(0x4312)	ILV12[2:0] (D[2:0])	UART Ch.0 interrupt
	ILV13[2:0] (D[10:8])	UART Ch.1 interrupt
ITC_LV7(0x4314)	ILV14[2:0] (D[2:0])	SPI Ch.0 interrupt
	ILV15[2:0] (D[10:8])	I^2C master interrupt
ITC_LV8(0x4316)	ILV16[2:0] (D[2:0])	IR remote controller interrupt / SPI Ch.1 interrupt
	ILV17[2:0] (D[10:8])	16-bit PWM timer Ch.1 interrupt
ITC_LV9(0x4318)	ILV18[2:0] (D[2:0])	A/D converter interrupt
	ILV19[2:0] (D[10:8])	P5 port interrupt
ITC_LV10(0x431a)	ILV20[2:0] (D[2:0])	P2 port interrupt
	ILV21[2:0] (D[10:8])	P3 port interrupt
ITC_LV11(0x431c)	ILV22[2:0] (D[2:0])	I^2C slave interrupt
	(ILV23[2:0] (D[10:8]))	Reserved

7 Clock Generator (CLG)

Note: The descriptions in this chapter regarding IOSC are applied to the S1C17564 only. The IOSC clock cannot be used in the S1C15554.

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.
  - IOSC oscillator circuit: 2/4/8/12 MHz MHz (typ.)*
  - OSC3 oscillator circuit: 24 MHz (max.) crystal or ceramic oscillator circuit, or an external clock input
  - OSC1 oscillator circuit: 32.768 kHz (typ.) crystal oscillator circuit or an external clock input
- Switches the system clock. The system clock source can be selected from IOSC*, OSC3, 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 IOSC on forcibly and switches the system clock to IOSC when SLEEP mode is canceled. *
- Controls two clock outputs to external devices.

* S1C17564 only

Figure 7.1.1 shows the clock system and CLG module configuration.

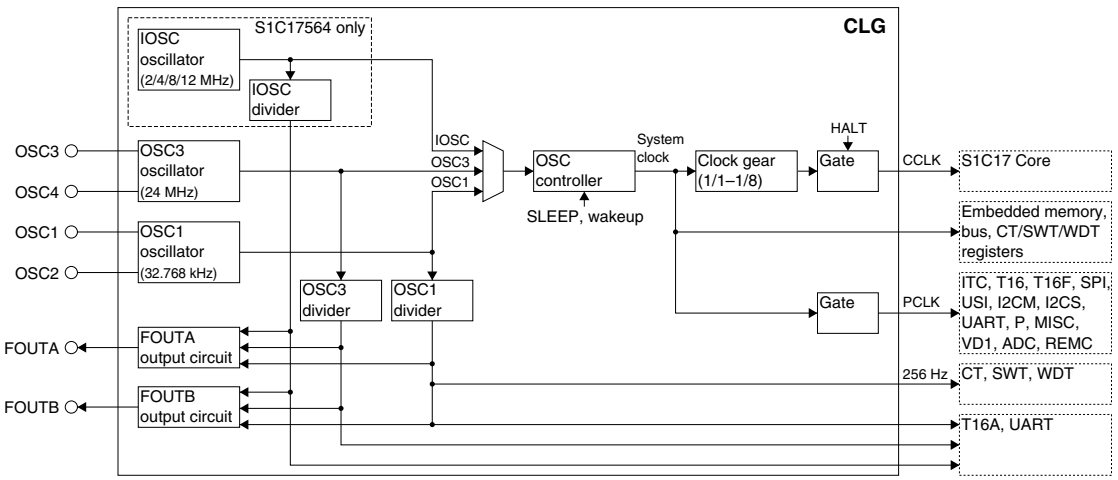


Figure 7.1.1 CLG Module Configuration

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	OSC1 oscillator input pin Connect a crystal resonator (32.768 kHz) and a gate capacitor. Or input an external clock used as the OSC1 clock.
OSC2	O	1	OSC1 oscillator output pin Connect a crystal resonator (32.768 kHz).
OSC3	I	1	OSC3 oscillator input pin Connect a crystal or ceramic resonator (max. 24 MHz), a feedback resistor, and a gate capacitor. Or input an external clock used as the OSC3 clock.
OSC4	O	1	OSC3 oscillator output pin Connect a crystal or ceramic resonator (max. 24 MHz), a feedback resistor, and a drain capacitor.
FOUTA	O	1	FOUTA clock output pin Outputs a divided IOSC/OSC3 clock or the OSC1 clock.
FOUTB	O	1	FOUTB clock output pin Outputs a divided IOSC/OSC3 clock or the 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 S1C17554 CLG module contains two internal oscillator circuits (OSC3, and OSC1). The S1C17564 CLG module contains three internal oscillator circuits (IOSC, OSC3, and OSC1). The OSC3 and IOSC oscillators generate the main clock for high-speed operation of the S1C17 Core and peripheral circuits. The OSC1 oscillator generates a sub-clock for timers and low-power operations. The OSC3 clock is selected as the system clock in the S1C17554 or the IOSC clock is selected in the S1C17564 after an initial reset. Oscillator on/off switching and system clock selection (from IOSC, OSC3 and OSC1) are controlled with software.

Table 7.3.1 Oscillator Configuration

Model	IOSC oscillator	OSC3 oscillator	OSC1 oscillator	Default system clock
S1C17554	Unavailable	Available (Default: On)	Available (Default: Off)	OSC3
S1C17564	Available (Default: On)	Available (Default: Off)	Available (Default: Off)	IOSC

7.3.1 OSC3 Oscillator

The OSC3 oscillator is a high-precision, high-speed oscillator circuit that uses either a crystal resonator or a ceramic resonator. Figure 7.3.1.1 shows the OSC3 oscillator configuration.

A crystal resonator (X'tal3) or a ceramic resonator (Ceramic) and a feedback resistor (R_{f3}) 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}.
To use an external clock, leave the OSC4 pin open and input an LV_{DD}-level clock (with a 50% duty cycle) to the OSC3 pin.
For the effective frequency range, oscillation characteristics, and external clock input characteristics, see the "Electrical Characteristics" chapter.

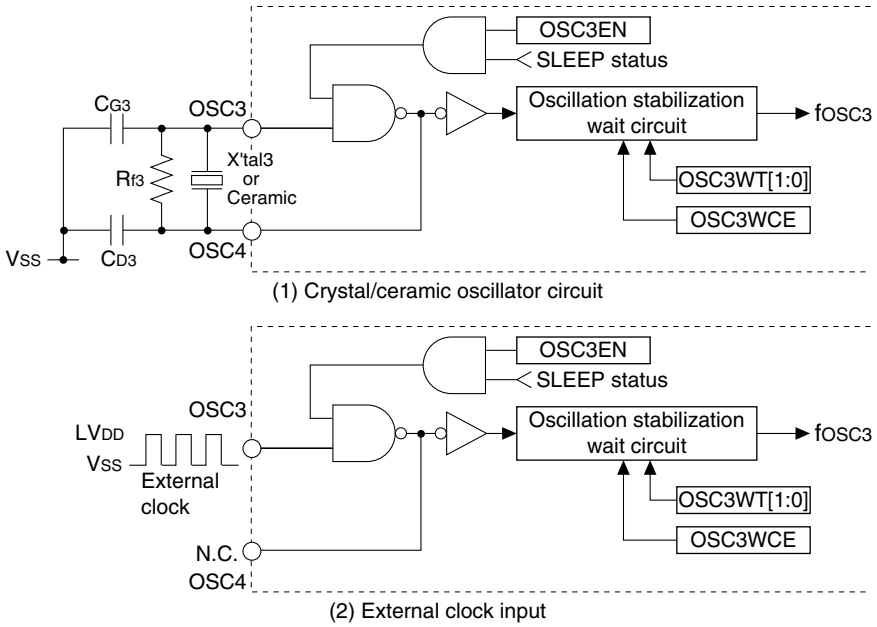


Figure 7.3.1.1 OSC3 Oscillator Circuit

OSC3 oscillation on/off

The OSC3 oscillator circuit starts oscillating when OSC3EN/CLG_CTL register is set to 1 and stops oscillating when it is set to 0. The OSC3 oscillator circuit also stops oscillating in SLEEP mode.
After an initial reset, OSC3EN is initialized as follows:

Table 7.3.1.1 OSC3 Oscillator Initial Status

Model	OSC3EN initial value	OSC3 oscillator initial status
S1C17554	1	On
S1C17564	0	Off

Stabilization wait time at start of OSC3 oscillation

The OSC3 oscillator circuit includes an oscillation stabilization wait circuit to prevent malfunctions due to unstable clock operations at the start of OSC3 oscillation—e.g., when the OSC3 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.

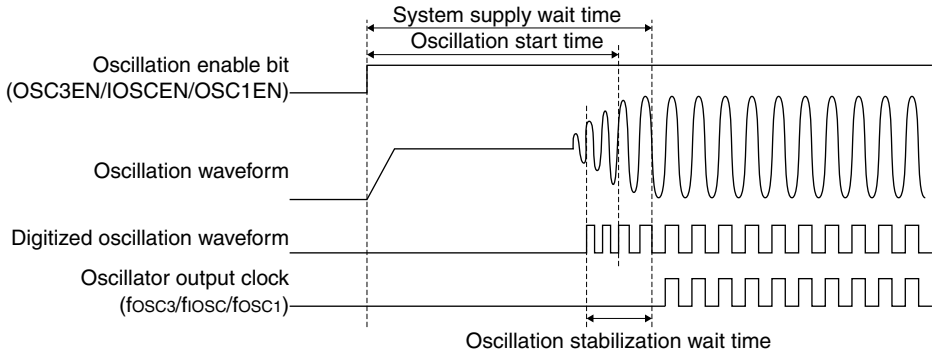


Figure 7.3.1.2 Oscillation Start Time and Oscillation Stabilization Wait Time

The OSC3 clock is not supplied to the system until the time set for this circuit has elapsed. Use OSC3WT[1:0]/ CLG_CTL register to select one of four oscillation stabilization wait times.

Table 7.3.1.2 OSC3 Oscillation Stabilization Wait Time Settings

OSC3WT[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 (OSC3 clock) after an initial reset.
When the system clock is switched to OSC3 immediately after the OSC3 oscillator circuit is turned on, the OSC3 clock is supplied to the system after the OSC3 clock system supply wait time indicated below (at a maximum) has elapsed. For the oscillation start time, see the "Electrical Characteristics" chapter.

OSC3 clock system supply wait time ≤ OSC3 oscillation start time (max.) + OSC3 oscillation stabilization wait time

Note: Oscillation stability will vary, depending on the resonator and other external components. Carefully consider the OSC3 oscillation stabilization wait time before reducing the time.

The OSC3 oscillation stabilization wait circuit can be enabled or disabled using OSC3WCE/CLG_NFEN register. After an initial reset, the OSC3 oscillation stabilization wait circuit is enabled (OSC3WCE = 1) and it controls the clock supply to the system. When a stabilized external clock is input to the OSC3 pin, setting OSC3WCE to 0 enables the system to start operating without a stabilization wait time.

7.3.2 OSC1 Oscillator

The OSC1 oscillator is a high-precision, low-speed oscillator circuit that uses a 32.768 kHz crystal resonator.
The OSC1 clock is generally used as the timer operation clock (for the clock timer, stopwatch timer, watchdog timer, and 16-bit PWM timer). It can be used as the system clock instead of the OSC3 or IOSC clock to reduce power consumption when no high-speed processing is required.
Figure 7.3.2.1 shows the OSC1 oscillator configuration.

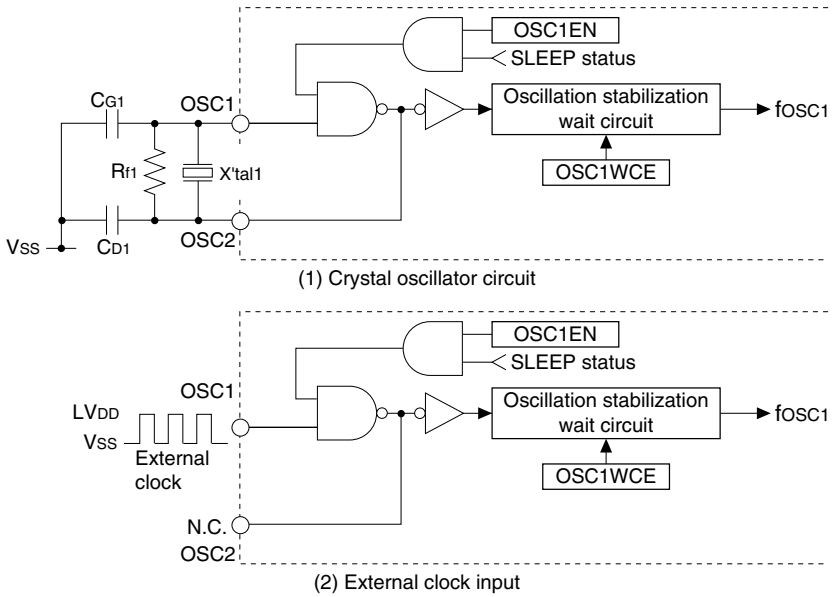


Figure 7.3.2.1 OSC1 Oscillator Circuit

A crystal resonator (X'tal1) and a feedback resistor (R_{f1}) 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}.
To use an external clock, leave the OSC2 pin open and input an LV_{DD}-level clock (with a 50% duty cycle) to the OSC1 pin.
For oscillation characteristics and external clock input characteristics, see the "Electrical Characteristics" chapter.

OSC1 oscillation on/off

The OSC1 oscillator stops oscillating when OSC1EN/CLG_CTL register is set to 0 and starts oscillating when set to 1. The OSC1 oscillator circuit stops oscillating in SLEEP mode.
After an initial reset, OSC1EN is set to 0, and the OSC1 oscillator circuit is halted.

Stabilization wait time at start of OSC1 oscillation

The OSC1 oscillator includes an oscillation stabilization wait circuit (fixed at 256 cycles) to prevent malfunctions caused by unstable clock operations at the start of OSC1 oscillation—e.g., when the OSC1 oscillator is turned on with software. 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 (256 cycles)

The OSC1 oscillation stabilization wait circuit can be enabled or disabled using OSC1WCE/CLG_NFEN register. After an initial reset, the OSC1 oscillation stabilization wait circuit is enabled (OSC1WCE = 1) and it controls the clock supply to the system. When a stabilized external clock is input to the OSC1 pin, setting OSC1WCE to 0 enables the system to start operating without a stabilization wait time.

7.3.3 IOSC Oscillator (S1C17564)

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

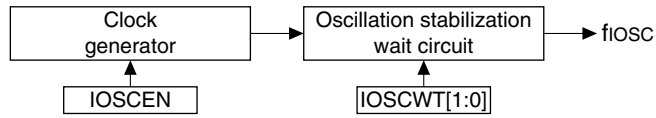


Figure 7.3.3.1 IOSC Oscillator Circuit

IOSC oscillation frequency

The IOSC oscillation frequency can be selected from four types shown below using IOSCSEL[1:0]/CLG_IOSC register.

Table 7.3.3.1 IOSC Oscillation Frequency Setting

IOSCSEL[1:0]	IOSC oscillation frequency (typ.)
0x3	2 MHz
0x2	4 MHz
0x1	12 MHz
0x0	8 MHz

(Default: 0x1)

IOSC oscillation on/off

The IOSC oscillator stops oscillating when IOSCEN/CLG_CTL register is set to 0 and starts oscillating when set to 1. The IOSC oscillator stops oscillating in SLEEP mode.
After an initial reset, IOSCEN is set to 1, and the IOSC oscillator goes on. Since the IOSC clock is used as the system clock, the S1C17 Core starts operating using the IOSC clock.
When SLEEP mode is canceled, the IOSC oscillator circuit is turned on and is used as the system clock source regardless of the system clock configured before the chip entered SLEEP mode.

Stabilization wait time at start of IOSC oscillation

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

Table 7.3.3.2 IOSC Oscillation Stabilization Wait Time Settings

IOSCWT[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 (IOSC 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 ≤ IOSC oscillation start time (max.) + IOSC oscillation stabilization wait time (64 cycles)

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

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

7.4 System Clock Switching

The figure below shows the system clock selector.

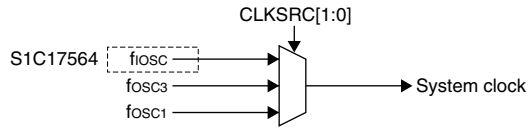


Figure 7.4.1 System Clock Selector

The S1C17554 has two system clock sources (OSC3 and OSC1) and the S1C17564 has three system clock sources (IOSC, OSC3, and OSC1). The system clock can be switched using CLKSRC[1:0]/CLG_SRC register. After an initial reset, the S1C17554 starts operating using OSC3 as the system clock and the S1C17564 starts operating using IOSC. When no high-speed processing is required, switch the system clock to OSC1 and stop the high-speed oscillator circuit to reduce current consumption.

Table 7.4.1 System Clock Selection

CLKSRC[1:0]	System clock source	
	S1C17554	S1C17564
0x3	Reserved	
0x2	OSC3 (default)	OSC3
0x1	OSC1	OSC1
0x0	Reserved	IOSC (default)

The following shows system clock switching procedures:

Switching the system clock to OSC3 from IOSC or OSC1

1. Set the OSC3 oscillation stabilization wait time if necessary. (OSC3WT[1:0])
2. Disable the OSC3 oscillation stabilization wait circuit when a stabilized external clock is input to the OSC3 pin. (OSC3WCE = 0)
3. Turn the OSC3 oscillator on if it is off. (OSC3EN = 1)
4. Select the OSC3 clock as the system clock. (CLKSRC[1:0] = 0x2)
5. Turn the IOSC or OSC1 oscillator off if peripheral modules and FOUTA/B output circuits have not used the IOSC or OSC1 clock.

Switching the system clock to OSC1 from IOSC or OSC3

1. Disable the OSC1 oscillation stabilization wait circuit when a stabilized external clock is input to the OSC1 pin. (OSC1WCE = 0)
2. Turn the OSC1 oscillator on. (OSC1EN = 1)
3. Select the OSC1 clock as the system clock. (CLKSRC[1:0] = 0x1)
4. Turn the IOSC or OSC3 oscillator off if peripheral modules and FOUTA/B output circuits have not used the IOSC or OSC3 clock.

Switching the system clock to IOSC from OSC3 or OSC1 (S1C17564)

1. Set the IOSC oscillation stabilization wait time if necessary. (IOSCWT[1:0])
2. Turn the IOSC oscillator on if it is off. (IOSCEN = 1)
3. Select the IOSC clock as the system clock. (CLKSRC[1:0] = 0x0)
4. Turn the OSC3 or OSC1 oscillator off if peripheral modules and FOUTA/B output circuits have not used the OSC3 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 tables below list the combinations of clock operating status and register settings enabling system clock selection.

Table 7.4.2 System Clock Switching Conditions (S1C17554)

OSC3EN	OSC1EN	System clock
1	1	OSC3 or OSC1

Table 7.4.3 System Clock Switching Conditions (S1C17564)

IOSCEN	OSC3EN	OSC1EN	System clock
1	1	1	IOSC, OSC3, or OSC1
1	1	0	IOSC or OSC3
1	0	1	IOSC or OSC1
0	1	1	OSC3 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 in the S1C17564, the IOSC oscillator circuit is turned on (IOSCEN = 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.

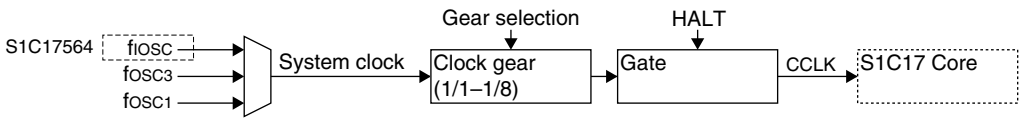


Figure 7.5.1 CCLK Supply System

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).

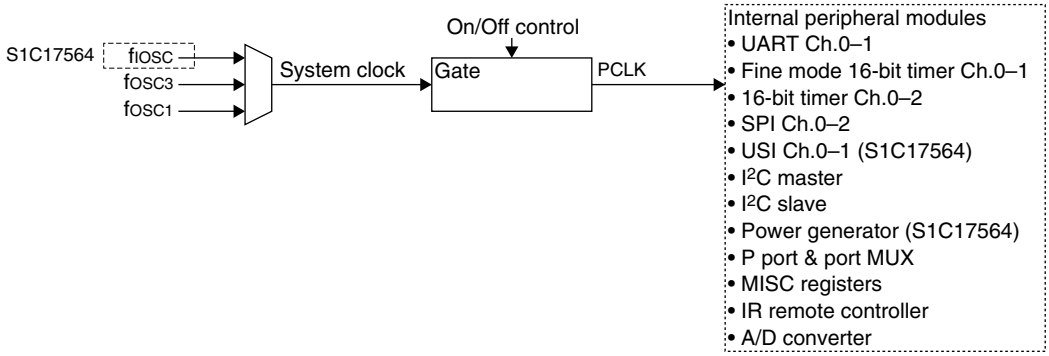


Figure 7.6.1 Peripheral Module Clock Control Circuit

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
UART Ch.0 and 1	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.
Fine mode 16-bit timer Ch.0 and 1		
16-bit timer Ch.0 to 2		
SPI Ch.0 to 2		
USI Ch.0 and 1 (S1C17564)		
I²C master		
I²C slave		
Power generator (S1C17564)		
P port & port MUX		
MISC registers		
IR remote controller		
A/D converter		
Clock timer	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.
Stopwatch timer		
Watchdog timer		
16-bit PWM timer Ch.0 to 3	Clock selected by software (divided IOSC/OSC3/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.
FOUTA/FOUTB outputs		

7.7 Clock External Output (FOUTA, FOUTB)

A divided IOSC/OSC3 clock or the OSC1 clock can be output to external devices.

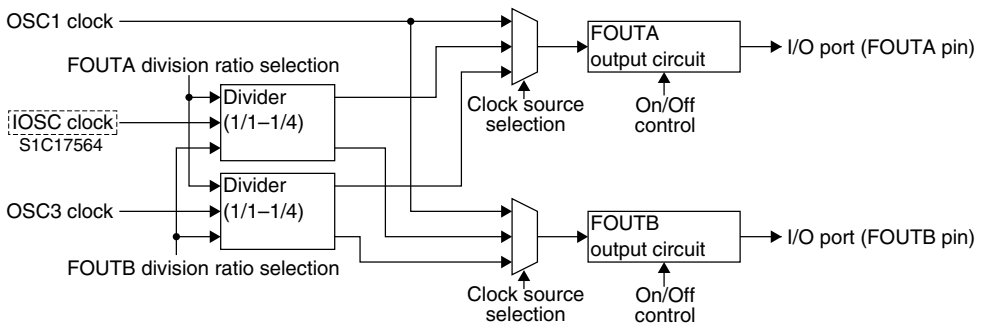


Figure 7.7.1 Clock Output Circuit

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 IOSC (S1C17564), OSC3, 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	
	S1C17554	S1C17564
0x3	Reserved	
0x2	OSC3	OSC3
0x1	OSC1	OSC1
0x0	Reserved	IOSC

(Default: 0x0)

Clock frequency selection

Three different clock output frequencies can be selected when OSC3 or IOSC is used as the clock source. Select the division ratio for the source clock using FOUTAD[1:0]/CLG_FOUTA register or FOUTBD[1:0]/CLG_FOUTB register.

Table 7.7.2 IOSC/OSC3 Division Ratio Selection

FOUTAD[1:0]/FOUTBD[1:0]	Division ratio
0x3	Reserved
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.

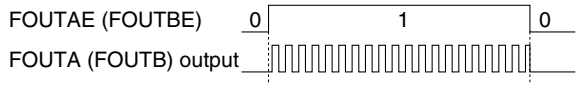


Figure 7.7.2 FOUTA/FOUTB Output

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

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.
0x5062	CLG_NFEN	Noise Filter Enable Register	Turns oscillation stabilization wait circuit/noise filter on/off.
0x5064	CLG_FOUTA	FOUTA Control Register	Controls FOUTA clock output.
0x5065	CLG_FOUTB	FOUTB Control Register	Controls FOUTB clock output.
0x506e	CLG_IOSC	IOSC Control Register	Configures IOSC oscillation frequency.
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. These are 8-bit registers.

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

Clock Source Select Register (CLG_SRC)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
Clock Source Select Register (CLG_SRC) **S1C17554**	**0x5060** (8 bits)	D7–2	–	reserved	–		–	–	0 when being read.
		D1–0	**CLKSRC[1:0]**	System clock source select	CLKSRC[1:0]	Clock source	0x2	R/W	
					0x3	reserved			
					0x2	OSC3			
					0x1	OSC1			
					0x0	reserved			
Clock Source Select Register (CLG_SRC) **S1C17564**	**0x5060** (8 bits)	D7–2	–	reserved	–		–	–	0 when being read.
		D1–0	**CLKSRC[1:0]**	System clock source select	CLKSRC[1:0]	Clock source	0x0	R/W	
					0x3	reserved			
					0x2	OSC3			
					0x1	OSC1			
					0x0	IOSC			

D[7:2] Reserved

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

Selects the system clock source.

Table 7.8.2 System Clock Selection

CLKSRC[1:0]	System clock source	
	S1C17554	S1C17564
0x3	Reserved	
0x2	OSC3 (default)	OSC3
0x1	OSC1	OSC1
0x0	Reserved	IOSC (default)

Select IOSC or OSC3 for normal (high-speed) operations. If no high-speed clock is required, OSC1 can be set as the system clock and IOSC and OSC3 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 tables below list the combinations of clock operating status and register settings enabling system clock selection.

Table 7.8.3 System Clock Switching Conditions (S1C17554)

OSC3EN	OSC1EN	System clock
1	1	OSC3 or OSC1

Table 7.8.4 System Clock Switching Conditions (S1C17564)

IOSCEN	OSC3EN	OSC1EN	System clock
1	1	1	IOSC, OSC3, or OSC1
1	1	0	IOSC or OSC3
1	0	1	IOSC or OSC1
0	1	1	OSC3 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 in the S1C17564, the IOSC oscillator circuit is turned on (IOSCEN = 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) **S1C17554**	**0x5061** (8 bits)	D7–6	–	reserved	–	–	–	0 when being read.
		D5–4	**OSC3WT[1:0]**	OSC3 wait cycle select	OSC3WT[1:0] / Wait cycle 0x3 / 128 cycles 0x2 / 256 cycles 0x1 / 512 cycles 0x0 / 1024 cycles	0x0	R/W	
		D3–2	–	reserved	–	–	–	0 when being read.
		D1	**OSC1EN**	OSC1 enable	1 Enable 0 Disable	0	R/W	
		D0	**OSC3EN**	OSC3 enable	1 Enable 0 Disable	1	R/W	
Oscillation Control Register (CLG_CTL) **S1C17564**	**0x5061** (8 bits)	D7–6	**IOSCWT[1:0]**	IOSC wait cycle select	IOSCWT[1:0] / Wait cycle 0x3 / 8 cycles 0x2 / 16 cycles 0x1 / 32 cycles 0x0 / 64 cycles	0x0	R/W	
		D5–4	**OSC3WT[1:0]**	OSC3 wait cycle select	OSC3WT[1:0] / Wait cycle 0x3 / 128 cycles 0x2 / 256 cycles 0x1 / 512 cycles 0x0 / 1024 cycles	0x0	R/W	
		D3	–	reserved	–	–	–	0 when being read.
		D2	**IOSCEN**	IOSC enable	1 Enable 0 Disable	1	R/W	
		D1	**OSC1EN**	OSC1 enable	1 Enable 0 Disable	0	R/W	
		D0	**OSC3EN**	OSC3 enable	1 Enable 0 Disable	0	R/W	

D[7:6] Reserved (S1C17554)

IOSCWT[1:0]: IOSC Wait Cycle Select Bits (S1C17564)

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

Table 7.8.5 IOSC Oscillation Stabilization Wait Time Settings

IOSCWT[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 (IOSC 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 ≤ IOSC oscillation start time (max.) + IOSC oscillation stabilization wait time (64 cycles)

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

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

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

An oscillation stabilization wait time is set to prevent malfunctions due to unstable clock operation at the start of OSC3 oscillation.
The OSC3 clock is not supplied to the system immediately after OSC3 oscillation starts—e.g., when the OSC3 oscillator is turned on with software—until the time set here has elapsed.

Table 7.8.6 OSC3 Oscillation Stabilization Wait Time Settings

OSC3WT[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 (OSC3 clock) after an initial reset.
When the system clock is switched to OSC3 immediately after the OSC3 oscillator circuit is turned on, the OSC3 clock is supplied to the system after the OSC3 clock system supply wait time indicated below (at a maximum) has elapsed.

OSC3 clock system supply wait time ≤ OSC3 oscillation start time (max.) + OSC3 oscillation stabilization wait time

Note: Oscillation stability will vary, depending on the resonator and other external components. Carefully consider the OSC3 oscillation stabilization wait time before reducing the time.

The OSC3 oscillation stabilization wait circuit is enabled only when OSC3WCE/CLG_NFEN register is set to 1 (default).

D3 Reserved

D2 **Reserved (S1C17554)**
IOSCEN: IOSC Enable Bit (S1C17564)
Enables or disables IOSC oscillator operations.
1 (R/W): Enabled (on) (default)
0 (R/W): Disabled (off)

Note: The IOSC oscillator cannot be stopped if the IOSC 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)

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.

OSC1 clock system supply wait time ≤ OSC1 oscillation start time (max.) + OSC1 oscillation stabilization wait time (256 cycles)

Note: The OSC1 oscillator cannot be stopped if the OSC1 clock is being used as the system clock.

D0 **OSC3EN: OSC3 Enable Bit**
Enables or disables OSC3 oscillator operations.
1 (R/W): Enabled (on) (default in S1C17554)
0 (R/W): Disabled (off) (default in S1C17564)

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

Noise Filter Enable Register (CLG_NFEN)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Noise Filter Enable Register (CLG_NFEN)	**0x5062** (8 bits)	D7–6	–	reserved	–	–	–	0 when being read.
		D5	**OSC1WCE**	OSC1 wait cycle enable	1 Enable 0 Disable	1	R/W	
		D4	**OSC3WCE**	OSC3 wait cycle enable	1 Enable 0 Disable	1	R/W	
		D3–0	–	reserved	–	–	–	0 when being read.

D[7:6] **Reserved**

D5 **OSC1WCE: OSC1 Wait Cycle Enable Bit**
Enables or disables the OSC1 oscillation stabilization wait circuit.
1 (R/W): Enabled (default)
0 (R/W): Disabled

When using the internal OSC1 oscillator circuit, enable the OSC1 oscillation stabilization wait circuit (OSC1WCE = 1). When the OSC1 oscillator circuit is turned on, the OSC1 clock is supplied to the system after 256 cycles of oscillation stabilization wait time has elapsed.
When a stabilized external clock is input to the OSC1 pin, setting OSC1WCE to 0 enables the system to start operating without a stabilization wait time.

D4 **OSC3WCE: OSC3 Wait Cycle Enable Bit**
Enables or disables the OSC3 oscillation stabilization wait circuit.
1 (R/W): Enabled (default)
0 (R/W): Disabled

When using the internal OSC3 oscillator circuit, enable the OSC3 oscillation stabilization wait circuit (OSC3WCE = 1). When the OSC3 oscillator circuit is turned on, the OSC3 clock is supplied to the system after the oscillation stabilization wait time set using OSC3WT[1:0]/CLG_CTL register has elapsed.
When a stabilized external clock is input to the OSC3 pin, setting OSC3WCE to 0 enables the system to start operating without a stabilization wait time.

D[3:0] **Reserved**

FOUTA Control Register (CLG_FOUTA)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
FOUTA Control Register (CLG_FOUTA) S1C17554	0x5064 (8 bits)	D7–6	–	reserved	–	–	–	0 when being read.
		D5–4	FOUTAD [1:0]	FOUTA clock division ratio select	FOUTAD[1:0] / Division ratio 0x3 reserved 0x2 1/4 0x1 1/2 0x0 1/1	0x0	R/W	When the clock source is OSC3
		D3–2	FOUTASRC [1:0]	FOUTA clock source select	FOUTASRC[1:0] / Clock source 0x3 reserved 0x2 OSC3 0x1 OSC1 0x0 reserved	0x0	R/W	
		D1	–	reserved	–	–	–	0 when being read.
		D0	FOUTAE	FOUTA output enable	1 Enable 0 Disable	0	R/W	
FOUTA Control Register (CLG_FOUTA) S1C17564	0x5064 (8 bits)	D7–6	–	reserved	–	–	–	0 when being read.
		D5–4	FOUTAD [1:0]	FOUTA clock division ratio select	FOUTAD[1:0] / Division ratio 0x3 reserved 0x2 1/4 0x1 1/2 0x0 1/1	0x0	R/W	When the clock source is IOSC or OSC3
		D3–2	FOUTASRC [1:0]	FOUTA clock source select	FOUTASRC[1:0] / Clock source 0x3 reserved 0x2 OSC3 0x1 OSC1 0x0 IOSC	0x0	R/W	
		D1	–	reserved	–	–	–	0 when being read.
		D0	FOUTAE	FOUTA output enable	1 Enable 0 Disable	0	R/W	

D[7:6] Reserved

D[5:4] FOUTAD[1:0]: FOUTA Clock Division Ratio Select Bits

Selects the clock division ratio to set the FOUTA clock frequency when OSC3 or IOSC is used as the clock source.

Table 7.8.7 OSC3/IOSC Division Ratio Selection

FOUTAD[1:0]	Division ratio
0x3	Reserved
0x2	1/4
0x1	1/2
0x0	1/1

(Default: 0x0)

When OSC1 is used as the clock source, FOUTAD[1:0] is ineffective and the OSC1 clock is output without frequency division.

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

Selects the FOUTA clock source.

Table 7.8.8 FOUTA Clock Source Selection

FOUTASRC[1:0]	Clock source	
	S1C17554	S1C17564
0x3	Reserved	
0x2	OSC3	OSC3
0x1	OSC1	OSC1
0x0	Reserved	IOSC

(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) **S1C17554**	**0x5065** (8 bits)	D7–6	**–**	reserved	–	–	–	0 when being read.
		D5–4	**FOUTBD [1:0]**	FOUTB clock division ratio select	FOUTBD[1:0] Division ratio 0x3 reserved 0x2 1/4 0x1 1/2 0x0 1/1	0x0	R/W	When the clock source is OSC3
		D3–2	**FOUTBSRC [1:0]**	FOUTB clock source select	FOUTBSRC[1:0] Clock source 0x3 reserved 0x2 OSC3 0x1 OSC1 0x0 reserved	0x0	R/W	
		D1	**–**	reserved	–	–	–	0 when being read.
		D0	**FOUTBE**	FOUTB output enable	1 Enable 0 Disable	0	R/W	
FOUTB Control Register (CLG_FOUTB) **S1C17564**	**0x5065** (8 bits)	D7–6	**–**	reserved	–	–	–	0 when being read.
		D5–4	**FOUTBD [1:0]**	FOUTB clock division ratio select	FOUTBD[1:0] Division ratio 0x3 reserved 0x2 1/4 0x1 1/2 0x0 1/1	0x0	R/W	When the clock source is IOSC or OSC3
		D3–2	**FOUTBSRC [1:0]**	FOUTB clock source select	FOUTBSRC[1:0] Clock source 0x3 reserved 0x2 OSC3 0x1 OSC1 0x0 IOSC	0x0	R/W	
		D1	**–**	reserved	–	–	–	0 when being read.
		D0	**FOUTBE**	FOUTB output enable	1 Enable 0 Disable	0	R/W	

D[7:6] Reserved

D[5:4] FOUTBD[1:0]: FOUTB Clock Division Ratio Select Bits

Selects the clock division ratio to set the FOUTB clock frequency when OSC3 or IOSC is used as the clock source.

Table 7.8.9 OSC3/IOSC Division Ratio Selection

FOUTBD[1:0]	Division ratio
0x3	Reserved
0x2	1/4
0x1	1/2
0x0	1/1

(Default: 0x0)

When OSC1 is used as the clock source, FOUTBD[1:0] is ineffective and the OSC1 clock is output without frequency division.

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

Selects the FOUTB clock source.

Table 7.8.10 FOUTB Clock Source Selection

FOUTBSRC[1:0]	Clock source	
	S1C17554	S1C17564
0x3	Reserved	
0x2	OSC3	OSC3
0x1	OSC1	OSC1
0x0	Reserved	IOSC

(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.

IOSC Control Register (CLG_IOSC)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
IOSC Control Register (CLG_IOSC) S1C17564	0x506e (8 bits)	D7–2	–	reserved	–		–	–	0 when being read.
		D1–0	IOSCSEL [1:0]	IOSC frequency select	IOSCSEL[1:0]	Frequency	0x1	R/W	
					0x3	2 MHz			
					0x2	4 MHz			
					0x1	12 MHz			
					0x0	8 MHz			

D[7:2] Reserved

D[1:0] Reserved (S1C17554)

IOSCSEL[1:0]: IOSC Frequency Select Bits (S1C17564)

Selects the IOSC oscillation frequency.

Table 7.8.11 IOSC Oscillation Frequency Setting

IOSCSEL[1:0]	IOSC oscillation frequency (typ.)
0x3	2 MHz
0x2	4 MHz
0x1	12 MHz
0x0	8 MHz

(Default: 0x1)

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 7.8.12 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

- UART Ch.0 and 1
- Fine mode 16-bit timer Ch.0 and 1
- 16-bit timer Ch.0 to 2
- SPI Ch.0 to 2
- USI Ch.0 and 1 (S1C17564)
- I^2C master
- I^2C slave
- Power generator (S1C17564)
- P port & port MUX
- MISC registers
- IR remote controller
- A/D converter

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

- Clock timer
- Stopwatch timer
- Watchdog timer
- 16-bit PWM timer Ch.0 to 3
- 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 7.8.13 CCLK Gear Ratio Selection

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

(Default: 0x0)

8 I/O Ports (P)

8.1 P Module Overview

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

- S1C17564, S1C17554 (TQFP package)
  Maximum 40 I/O ports (P0[3:0], P1[7:0], P2[7:0], P3[7:0], P4[5:0], P5[5:0]) are available.

  S1C17554 (WCSP package)
  Maximum 34 I/O ports (P0[3:0], P1[7:0], P2[7:0], P3[7:0], P4[5:0]) 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.
- Each port can generate input interrupts at the signal edge selected with software.
- Each port includes 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 8.1.1 shows the I/O port configuration.

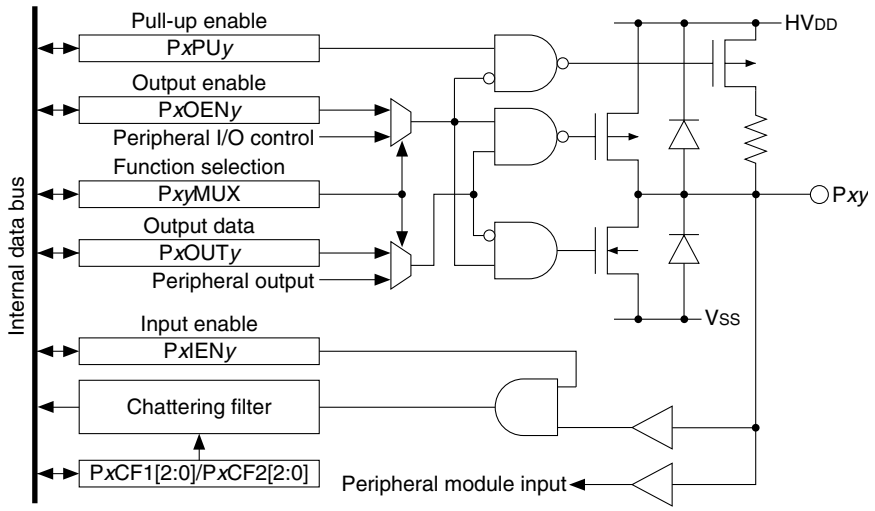


Figure 8.1.1 I/O Port Configuration

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 chattering filters.

• The "*xy*" in the register and bit names refers to the port number (P*xy*, *x* = 0 to 5, *y* = 0 to 7).
Example: P*x*IN*y*/P*x*_IN register
P00: P0IN0/P0_IN register
P17: P1IN7/P1_IN register

8.2 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 8.2.1 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	AIN0 (ADC10)	–	–	P00MUX[1:0]/P00_03PMUX register
P01	AIN1 (ADC10)	–	–	P01MUX[1:0]/P00_03PMUX register
P02	AIN2 (ADC10)	US_SSI0 (USI)*	–	P02MUX[1:0]/P00_03PMUX register
P03	AIN3 (ADC10)	US_SSI1 (USI)*	–	P03MUX[1:0]/P00_03PMUX register
P10	SDI0 (SPI)	–	–	P10MUX[1:0]/P10_13PMUX register
P11	SDO0 (SPI)	–	–	P11MUX[1:0]/P10_13PMUX register
P12	SPICLK0 (SPI)	–	–	P12MUX[1:0]/P10_13PMUX register
P13	#SPISS0 (SPI)	TOUT5/CAP5 (T16A)	–	P13MUX[1:0]/P10_13PMUX register
P14	SIN1 (UART)	SDI1 (SPI)	–	P14MUX[1:0]/P14_17PMUX register
P15	SOUT1 (UART)	SDO1 (SPI)	–	P15MUX[1:0]/P14_17PMUX register
P16	SCLK1 (UART)	SPICLK1 (SPI)	–	P16MUX[1:0]/P14_17PMUX register
P17	SCL0 (I2CM)	–	–	P17MUX[1:0]/P14_17PMUX register
P20	TOUT2/CAP2 (T16A)	–	–	P20MUX[1:0]/P20_23PMUX register
P21	TOUT3/CAP3 (T16A)	–	–	P21MUX[1:0]/P20_23PMUX register
P22/EXCL1 (T16A)	FOUTB (CLG)	–	–	P22MUX[1:0]/P20_23PMUX register
P23/EXCL2 (T16A)	SDI2 (SPI)	–	–	P23MUX[1:0]/P20_23PMUX register
P24/EXCL3 (T16A)	SDO2 (SPI)	–	–	P24MUX[1:0]/P24_27PMUX register
P25	#BFR (I2CS)	#SPISS2 (SPI)	–	P25MUX[1:0]/P24_27PMUX register
P26	SDA1 (I2CS)	–	–	P26MUX[1:0]/P24_27PMUX register
P27	SCL1 (I2CS)	–	–	P27MUX[1:0]/P24_27PMUX register
P30	TOUT0/CAP0 (T16A)	–	–	P30MUX[1:0]/P30_33PMUX register
P31	#BFR (I2CS)	#ADTRG (ADC10)	–	P31MUX[1:0]/P30_33PMUX register
P32	TOUT4/CAP4 (T16A)	FOUTA (CLG)	–	P32MUX[1:0]/P30_33PMUX register
P33	REMI (REMC)	SPICLK2 (SPI)	–	P33MUX[1:0]/P30_33PMUX register
P34	REMO (REMC)	#SPISS1 (SPI)	–	P34MUX[1:0]/P34_37PMUX register
DCLK (DBG)	P35	–	–	P35MUX[1:0]/P34_37PMUX register
DSIO (DBG)	P36	–	–	P36MUX[1:0]/P34_37PMUX register
DST2 (DBG)	P37	–	–	P37MUX[1:0]/P34_37PMUX register
P40	SIN0 (UART)	TOUT6/CAP6 (T16A)	–	P40MUX[1:0]/P40_43PMUX register
P41	SOUT0 (UART)	TOUT7/CAP7 (T16A)	–	P41MUX[1:0]/P40_43PMUX register
P42	SCLK0 (UART)	TOUT1/CAP1 (T16A)	–	P42MUX[1:0]/P40_43PMUX register
P43	SDA1 (I2CS)	REMI (REMC)	–	P43MUX[1:0]/P40_33PMUX register
P44	SCL1 (I2CS)	REMO (REMC)	–	P44MUX[1:0]/P44_45PMUX register
P45/EXCL0 (T16A)	SDA0 (I2CM)	–	–	P45MUX[1:0]/P44_45PMUX register
P50	US_SDI0 (USI)*	–	–	P50MUX[1:0]/P50_53PMUX register
P51	US_SDO0 (USI)*	–	–	P51MUX[1:0]/P50_53PMUX register
P52	US_SCK0 (USI)*	–	–	P52MUX[1:0]/P50_53PMUX register
P53	US_SDI1 (USI)*	–	–	P53MUX[1:0]/P50_53PMUX register
P54	US_SDO1 (USI)*	–	–	P54MUX[1:0]/P54_55PMUX register
P55	US_SCK1 (USI)*	–	–	P55MUX[1:0]/P54_55PMUX register

* Available only in S1C17564

At initial reset, each I/O port pin (P*xy*) is initialized for the default function ("Pin function 1" in Table 8.2.1).

Pins P22, P23, P24, and P45 can also be used as 16-bit PWM timer external clock input pins by setting them to input mode. However, general-purpose input port function is also effective in this case.

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

8.3 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 8.3.1 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 (HV_{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 (HV_{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.

8.4 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.
This pull-up setting is also enabled for ports for which the peripheral module function has been selected.

A delay will occur in the waveform 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

8.5 P0–P3 Port Chattering Filter Function

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

Table 8.5.1 Chattering Filter Function Settings

P*x*CF1[2:0]/P*x*CF2[2:0]	Check time *
0x7	16384/f_{PCLK} (8 ms)
0x6	8192/f_{PCLK} (4 ms)
0x5	4096/f_{PCLK} (2 ms)
0x4	2048/f_{PCLK} (1 ms)
0x3	1024/f_{PCLK} (512 µs)
0x2	512/f_{PCLK} (256 µs)
0x1	256/f_{PCLK} (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 a minimum input time of the check time and a maximum input time of twice the check time.
- The P*x* port interrupt must be disabled before setting the P*x*_CHAT register. Setting the register while the interrupt is enabled may generate inadvertent P*x* port interrupt. Also the chattering filter circuit requires a maximum of twice the check time for stabilizing the operation status. Before enabling the interrupt, make sure that the stabilization time has elapsed.

8.6 Port Input Interrupt

The I/O ports include input interrupt functions.
Select which of the 40 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 8.6.1 shows the port input interrupt circuit configuration.

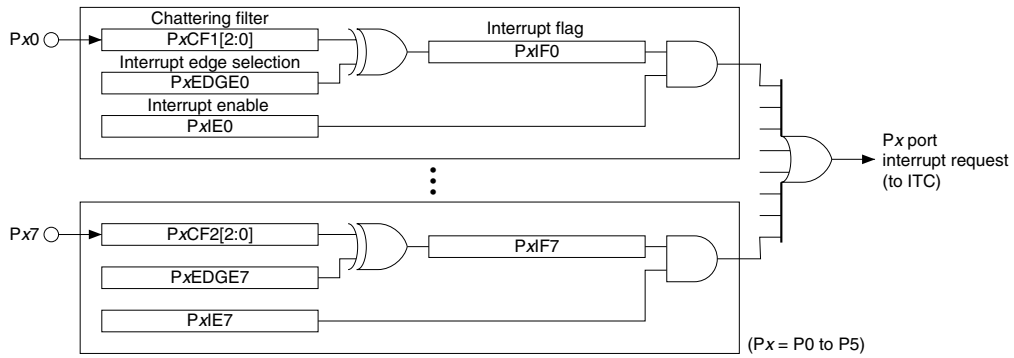


Figure 8.6.1 Port Input Interrupt Circuit Configuration

Interrupt port selection

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

Interrupt edge selection

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

Interrupt flags

The ITC is able to accept six interrupt requests from the P0–P5 ports, and the P port module contains interrupt flags P*x*IF*y*/P*x*_IFLG register corresponding to the individual 40 ports to enable individual control of the 40 P*xy* port interrupts. P*x*IF*y* is set to 1 at the specified edge (rising or falling edge) of the input signal. If the corresponding P*x*IE*y* has been set to 1, an interrupt request signal is also output to the ITC at the same time. An interrupt is generated if the ITC and S1C17 Core interrupt conditions are satisfied.
P*x*IF*y* is reset by writing 1.
For specific information on interrupt processing, see the "Interrupt Controller (ITC)" chapter.

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

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

8.7 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 8.7.1 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.

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

8.8 Control Register Details

Table 8.8.1 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.

Address	Register name		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.
0x5215	P1_IMSK	P1 Port Interrupt Mask Register	Enables P1 port interrupts.
0x5216	P1_EDGE	P1 Port Interrupt Edge Select Register	Selects the signal edge for generating P1 port interrupts.
0x5217	P1_IFLG	P1 Port Interrupt Flag Register	Indicates/resets the P1 port interrupt occurrence status.
0x5218	P1_CHAT	P1 Port Chattering Filter Control Register	Controls the P1 port chattering filter.
0x521a	P1_IEN	P1 Port Input Enable Register	Enables P1 port inputs.
0x5220	P2_IN	P2 Port Input Data Register	P2 port input data
0x5221	P2_OUT	P2 Port Output Data Register	P2 port output data
0x5222	P2_OEN	P2 Output Enable Register	Enables P2 port outputs.
0x5223	P2_PU	P2 Port Pull-up Control Register	Controls the P2 port pull-up resistor.
0x5225	P2_IMSK	P2 Port Interrupt Mask Register	Enables P2 port interrupts.
0x5226	P2_EDGE	P2 Port Interrupt Edge Select Register	Selects the signal edge for generating P2 port interrupts.
0x5227	P2_IFLG	P2 Port Interrupt Flag Register	Indicates/resets the P2 port interrupt occurrence status.
0x5228	P2_CHAT	P2 Port Chattering Filter Control Register	Controls the P2 port chattering filter.
0x522a	P2_IEN	P2 Port Input Enable Register	Enables P2 port inputs.
0x5230	P3_IN	P3 Port Input Data Register	P3 port input data
0x5231	P3_OUT	P3 Port Output Data Register	P3 port output data
0x5232	P3_OEN	P3 Port Output Enable Register	Enables P3 port outputs.
0x5233	P3_PU	P3 Port Pull-up Control Register	Controls the P3 port pull-up resistor.
0x5235	P3_IMSK	P3 Port Interrupt Mask Register	Enables P3 port interrupts.
0x5236	P3_EDGE	P3 Port Interrupt Edge Select Register	Selects the signal edge for generating P3 port interrupts.
0x5237	P3_IFLG	P3 Port Interrupt Flag Register	Indicates/resets the P3 port interrupt occurrence status.
0x5238	P3_CHAT	P3 Port Chattering Filter Control Register	Controls the P3 port chattering filter.
0x523a	P3_IEN	P3 Port Input Enable Register	Enables P3 port inputs.
0x5240	P4_IN	P4 Port Input Data Register	P4 port input data
0x5241	P4_OUT	P4 Port Output Data Register	P4 port output data
0x5242	P4_OEN	P4 Port Output Enable Register	Enables P4 port outputs.
0x5243	P4_PU	P4 Port Pull-up Control Register	Controls the P4 port pull-up resistor.
0x5245	P4_IMSK	P4 Port Interrupt Mask Register	Enables P4 port interrupts.
0x5246	P4_EDGE	P4 Port Interrupt Edge Select Register	Selects the signal edge for generating P4 port interrupts.
0x5247	P4_IFLG	P4 Port Interrupt Flag Register	Indicates/resets the P4 port interrupt occurrence status.
0x5248	P4_CHAT	P4 Port Chattering Filter Control Register	Controls the P4 port chattering filter.
0x524a	P4_IEN	P4 Port Input Enable Register	Enables P4 port inputs.
0x5250	P5_IN	P5 Port Input Data Register	P5 port input data
0x5251	P5_OUT	P5 Port Output Data Register	P5 port output data
0x5252	P5_OEN	P5 Port Output Enable Register	Enables P5 port outputs.
0x5253	P5_PU	P5 Port Pull-up Control Register	Controls the P5 port pull-up resistor.
0x5255	P5_IMSK	P5 Port Interrupt Mask Register	Enables P5 port interrupts.
0x5256	P5_EDGE	P5 Port Interrupt Edge Select Register	Selects the signal edge for generating P5 port interrupts.
0x5257	P5_IFLG	P5 Port Interrupt Flag Register	Indicates/resets the P5 port interrupt occurrence status.
0x5258	P5_CHAT	P5 Port Chattering Filter Control Register	Controls the P5 port chattering filter.
0x525a	P5_IEN	P5 Port Input Enable Register	Enables P5 port inputs.
0x52a0	P00_03PMUX	P0[3:0] Port Function Select Register	Selects the P0[3:0] port functions.
0x52a2	P10_13PMUX	P1[3:0] Port Function Select Register	Selects the P1[3:0] port functions.
0x52a3	P14_17PMUX	P1[7:4] Port Function Select Register	Selects the P1[7:4] port functions.
0x52a4	P20_23PMUX	P2[3:0] Port Function Select Register	Selects the P2[3:0] port functions.
0x52a5	P24_27PMUX	P2[7:4] Port Function Select Register	Selects the P2[7:4] port functions.
0x52a6	P30_33PMUX	P3[3:0] Port Function Select Register	Selects the P3[3:0] port functions.
0x52a7	P34_37PMUX	P3[7:4] Port Function Select Register	Selects the P3[7:4] port functions.
0x52a8	P40_43PMUX	P4[3:0] Port Function Select Register	Selects the P4[3:0] port functions.
0x52a9	P44_45PMUX	P4[5:4] Port Function Select Register	Selects the P4[5:4] port functions.
0x52aa	P50_53PMUX	P5[3:0] Port Function Select Register	Selects the P5[3:0] port functions.
0x52ab	P54_55PMUX	P5[5:4] Port Function Select Register	Selects the P5[5:4] port functions.

The I/O port registers are described in detail below. These are 8-bit registers.

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

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

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
P*x* Port Input Data Register (P*x*_IN)	**0x5200** **0x5210** **0x5220** **0x5230** **0x5240** **0x5250** (8 bits)	D7–0	**P*x*IN[7:0]**	P*x*[7:0] port input data	1	1 (H)	0	0 (L)	×	R	

Note: P0IN[3:0] only are available for the P0 ports. P*x*IN[5:0] only are available for the P4 and P5 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** **0x5221** **0x5231** **0x5241** **0x5251** (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: P0OUT[3:0] only are available for the P0 ports. P*x*OUT[5:0] only are available for the P4 and P5 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** **0x5222** **0x5232** **0x5242** **0x5252** (8 bits)	D7–0	**P*x*OEN[7:0]**	P*x*[7:0] port output enable	1	Enable	0	Disable	0	R/W	

Note: P0OEN[3:0] only are available for the P0 ports. P*x*OEN[5:0] only are available for the P4 and P5 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 8.3.1 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** **0x5223** **0x5233** **0x5243** **0x5253** (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: P0PU[3:0] only are available for the P0 ports. P*x*PU[5:0] only are available for the P4 and P5 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.
This pull-up setting is also enabled for ports for which the peripheral module input function is selected.

P*x* Port Interrupt Mask Registers (P*x*_IMSK)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
P*x* Port Interrupt Mask Register (P*x*_IMSK)	**0x5205** **0x5215** **0x5225** **0x5235** **0x5245** **0x5255** (8 bits)	D7–0	**P*x*IE[7:0]**	P*x*[7:0] port interrupt enable	1	Enable	0	Disable	0	R/W	

Note: P0IE[3:0] only are available for the P0 ports. P*x*IE[5:0] only are available for the P4 and P5 ports. Other bits are reserved and always read as 0.

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

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

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

P*x* Port Interrupt Edge Select Registers (P*x*_EDGE)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
P*x* Port Interrupt Edge Select Register (P*x*_EDGE)	**0x5206** **0x5216** **0x5226** **0x5236** **0x5246** **0x5256** (8 bits)	D7–0	**P*x*EDGE[7:0]**	P*x*[7:0] port interrupt edge select	1	Falling edge	0	Rising edge	0	R/W	

Note: P0EDGE[3:0] only are available for the P0 ports. P*x*EDGE[5:0] only are available for the P4 and P5 ports. Other bits are reserved and always read as 0.

D[7:0] P*x*EDGE[7:0]: P*x*[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 P*x*EDGE*y* is set to 1 and at the rising edge when set to 0.

P*x* Port Interrupt Flag Registers (P*x*_IFLG)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
P*x* Port Interrupt Flag Register (P*x*_IFLG)	**0x5207** **0x5217** **0x5227** **0x5237** **0x5247** **0x5257** (8 bits)	D7–0	**P*x*IF[7:0]**	P*x*[7:0] port interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.

Note: P0IF[3:0] only are available for the P0 ports. P*x*IF[5:0] only are available for the P4 and P5 ports. Other bits are reserved and always read as 0.

D[7:0] P*x*IF[7:0]: P*x*[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

P*x*IF*y* is the interrupt flag that corresponds directly to the P*xy* port and is set to 1 at the specified edge (rising or falling edge) of the input signal. When the corresponding P*x*IE*y*/P*x*_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.
P*x*IF*y* is reset by writing 1.

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

Px Port Chattering Filter Control Registers (Px_CHAT)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
Px Port Chattering Filter Control Register (Px_CHAT)	0x5208 0x5218 0x5228 0x5238 0x5248 0x5258 (8 bits)	D7	–	reserved	–		–	–	0 when being read.
		D6–4	PxCF2[2:0]	Px[7:4] chattering filter time select	PxCF2[2:0]	Filter time	0x0	R/W	
					0x7	16384/fPCLK			
					0x6	8192/fPCLK			
					0x5	4096/fPCLK			
					0x4	2048/fPCLK			
					0x3	1024/fPCLK			
					0x2	512/fPCLK			
					0x1	256/fPCLK			
					0x0	None			
		D3	–	reserved	–		–	–	0 when being read.
		D2–0	PxCF1[2:0]	Px[3:0] chattering filter time select	PxCF1[2:0]	Filter time	0x0	R/W	
					0x7	16384/fPCLK			
					0x6	8192/fPCLK			
					0x5	4096/fPCLK			
					0x4	2048/fPCLK			
					0x3	1024/fPCLK			
					0x2	512/fPCLK			
					0x1	256/fPCLK			
					0x0	None			

Note: P0CF1[2:0] only are available for the P0 ports. Other bits are reserved and always read as 0.

D7 Reserved

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

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

D3 Reserved

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

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

The I/O ports include a chattering filter circuit for key entry that can be disabled or enabled with a check time specified individually for the four Px[3:0] and Px[7:4] ports using PxCF1[2:0] and PxCF2[2:0], respectively.

Table 8.8.2 Chattering Filter Function Settings

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

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

Notes:
- An unexpected interrupt may occur after SLEEP status is canceled if the slp instruction is executed while the chattering filter function is enabled. The chattering filter must be disabled before placing the CPU into SLEEP status.
- The chattering filter check time refers to the maximum pulse width that can be filtered. Generating an input interrupt requires a minimum input time of the check time and a maximum input time of twice the check time.
- The Px port interrupt must be disabled before setting the Px_CHAT register. Setting the register while the interrupt is enabled may generate inadvertent Px 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 8.8.3 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.

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

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

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
P*x* Port Input Enable Register (P*x*_IEN)	0x520a 0x521a 0x522a 0x523a 0x524a (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: P0IEN[3:0] only are available for the P0 ports. P*x*IEN[5:0] only are available for the P4 and P5 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 8.3.1 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	* S1C17564 only
					0x3	reserved			
					0x2	US_SSI1*			
					0x1	AIN3			
					0x0	P03			
		D5–4	P02MUX[1:0]	P02 port function select	P02MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	US_SSI0*			
					0x1	AIN2			
					0x0	P02			
		D3–2	P01MUX[1:0]	P01 port function select	P01MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	AIN1			
					0x0	P01			
		D1–0	P00MUX[1:0]	P00 port function select	P00MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	AIN0			
					0x0	P00			

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

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

0x3 (R/W): Reserved
0x2 (R/W): USI_SSI1 (S1C17564 USI Ch.1)
0x1 (R/W): AIN3 (ADC10)
0x0 (R/W): P03 port (default)

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

0x3 (R/W): Reserved
0x2 (R/W): USI_SSI0 (S1C17564 USI Ch.0)
0x1 (R/W): AIN2 (ADC10)
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): AIN1 (ADC10)
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): AIN0 (ADC10)
0x0 (R/W): P00 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	TOUT5/CAP5			
					0x1	#SPISS0			
					0x0	P13			
		D5–4	P12MUX[1:0]	P12 port function select	P12MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	SPICLK0			
					0x0	P12			
		D3–2	P11MUX[1:0]	P11 port function select	P11MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	SDO0			
					0x0	P11			
		D1–0	P10MUX[1:0]	P10 port function select	P10MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	SDI0			
					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): TOUT5 (T16A Ch.2 comparator mode) or CAP5 (T16A Ch.2 capture mode)
0x1 (R/W): #SPISS0 (SPI Ch.0)
0x0 (R/W): P13 port (default)

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

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): SPICLK0 (SPI Ch.0)
0x0 (R/W): P12 port (default)

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

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

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

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): SDI0 (SPI Ch.0)
0x0 (R/W): P10 port (default)

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

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
P1[7:4] Port Function Select Register (P14_17PMUX)	0x52a3 (8 bits)	D7–6	P17MUX[1:0]	P17 port function select	P17MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	SCL0			
					0x0	P17			
		D5–4	P16MUX[1:0]	P16 port function select	P16MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	SPICLK1			
					0x1	SCLK1			
					0x0	P16			
		D3–2	P15MUX[1:0]	P15 port function select	P15MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	SDO1			
					0x1	SOUT1			
					0x0	P15			
		D1–0	P14MUX[1:0]	P14 port function select	P14MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	SDI1			
					0x1	SIN1			
					0x0	P14			

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

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

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): SCL0 (I2CM)
0x0 (R/W): P17 port (default)

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

0x3 (R/W): Reserved
0x2 (R/W): SPICLK1 (SPI Ch.1)
0x1 (R/W): SCLK1 (UART Ch.1)
0x0 (R/W): P16 port (default)

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

0x3 (R/W): Reserved
0x2 (R/W): SDO1 (SPI Ch.1)
0x1 (R/W): SOUT1 (UART Ch.1)
0x0 (R/W): P15 port (default)

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

0x3 (R/W): Reserved
0x2 (R/W): SDI1 (SPI Ch.1)
0x1 (R/W): SIN1 (UART Ch.1)
0x0 (R/W): P14 port (default)

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

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
P2[3:0] Port Function Select Register (P20_23PMUX)	0x52a4 (8 bits)	D7–6	P23MUX[1:0]	P23 port function select	P23MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	SDI2			
					0x0	P23/EXCL2			
		D5–4	P22MUX[1:0]	P22 port function select	P22MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	FOUTB			
					0x0	P22/EXCL1			
		D3–2	P21MUX[1:0]	P21 port function select	P21MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	TOUT3/CAP3			
					0x0	P21			
		D1–0	P20MUX[1:0]	P20 port function select	P20MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	TOUT2/CAP2			
					0x0	P20			

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

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

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): SDI2 (SPI Ch.2)
0x0 (R/W): P23 port and EXCL2 (T16A Ch.2) (default)

To use the P23 pin for EXCL2 input, P2OEN3/P2_OEN register must be set to 0 and P2IEN3/P2_IEN register must be set to 1.

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

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): FOUTB (CLG)
0x0 (R/W): P22 port and EXCL1 (T16A Ch.1) (default)

To use the P22 pin for EXCL1 input, P2OEN2/P2_OEN register must be set to 0 and P2IEN2/P2_IEN register must be set to 1.

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

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): TOUT3 (T16A Ch.1 comparator mode) or CAP3 (T16A Ch.1 capture mode)
0x0 (R/W): P21 port (default)

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

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): TOUT2 (T16A Ch.1 comparator mode) or CAP2 (T16A Ch.1 capture mode)
0x0 (R/W): P20 port (default)

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

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
P2[7:4] Port Function Select Register (P24_27PMUX)	0x52a5 (8 bits)	D7–6	P27MUX[1:0]	P27 port function select	P27MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	SCL1			
					0x0	P27			
		D5–4	P26MUX[1:0]	P26 port function select	P26MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	SDA1			
					0x0	P26			
		D3–2	P25MUX[1:0]	P25 port function select	P25MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	#SPISS2			
					0x1	#BFR			
					0x0	P25			
		D1–0	P24MUX[1:0]	P24 port function select	P24MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	SDO2			
					0x0	P24/EXCL3			

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

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

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): SCL1 (I2S)
0x0 (R/W): P27 port (default)

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

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): SDA1 (I2S)
0x0 (R/W): P26 port (default)

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

0x3 (R/W): Reserved
0x2 (R/W): #SPISS2 (SPI Ch.2)
0x1 (R/W):#BFR (I2S)
0x0 (R/W): P25 port (default)

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

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): SDO2 (SPI Ch.2)
0x0 (R/W): P24 port and EXCL3 (T16A Ch.3) (default)

To use the P24 pin for EXCL3 input, P2OEN4/P2_OEN register must be set to 0 and P2IEN4/P2_IEN register must be set to 1.

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

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
P3[3:0] Port Function Select Register (P30_33PMUX)	0x52a6 (8 bits)	D7–6	P33MUX[1:0]	P33 port function select	P33MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	SPICLK2			
					0x1	REMI			
					0x0	P33			
		D5–4	P32MUX[1:0]	P32 port function select	P32MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	FOUTA			
					0x1	TOUT4/CAP4			
					0x0	P32			
		D3–2	P31MUX[1:0]	P31 port function select	P31MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	#ADTRG			
					0x1	#BFR			
					0x0	P31			
		D1–0	P30MUX[1:0]	P30 port function select	P30MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	TOUT0/CAP0			
					0x0	P30			

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

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

0x3 (R/W): Reserved
0x2 (R/W): SPICLK2 (SPI Ch.2)
0x1 (R/W): REMI (REMC)
0x0 (R/W): P33 port (default)

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

0x3 (R/W): Reserved
0x2 (R/W): FOUTA (CLG)
0x1 (R/W): TOUT4 (T16A Ch.2 comparator mode) or CAP4 (T16A Ch.2 capture mode)
0x0 (R/W): P32 port (default)

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

0x3 (R/W): Reserved
0x2 (R/W): #ADTRG (ADC10)
0x1 (R/W): #BFR (I2CS)
0x0 (R/W): P31 port (default)

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

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): TOUT0 (T16A Ch.0 comparator mode) or CAP0 (T16A Ch.0 capture mode)
0x0 (R/W): P30 port (default)

P3[7:4] Port Function Select Register (P34_37PMUX)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
P3[7:4] Port Function Select Register (P34_37PMUX)	0x52a7 (8 bits)	D7–6	P37MUX[1:0]	P37 port function select	P37MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	P37			
					0x0	DST2			
		D5–4	P36MUX[1:0]	P36 port function select	P36MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	P36			
					0x0	DSIO			
		D3–2	P35MUX[1:0]	P35 port function select	P35MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	P35			
					0x0	DCLK			
		D1–0	P34MUX[1:0]	P34 port function select	P34MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	#SPISS1			
					0x1	REMO			
					0x0	P34			

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

D[7:6] P37MUX[1:0]: P37 Port Function Select Bits

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): P37 port
0x0 (R/W): DST2 (DBG) (default)

D[5:4] P36MUX[1:0]: P36 Port Function Select Bits

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): P36 port
0x0 (R/W): DSIO (DBG) (default)

D[3:2] P35MUX[1:0]: P35 Port Function Select Bits

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): P35 port
0x0 (R/W): DCLK (DBG) (default)

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

0x3 (R/W): Reserved
0x2 (R/W): #SPISS1 (SPI Ch.1)
0x1 (R/W): REMO (REMC)
0x0 (R/W): P34 port (default)

P4[3:0] Port Function Select Register (P40_43PMUX)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
P4[3:0] Port Function Select Register (P40_43PMUX)	0x52a8 (8 bits)	D7–6	P43MUX[1:0]	P43 port function select	P43MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	REMI			
					0x1	SDA1			
					0x0	P43			
		D5–4	P42MUX[1:0]	P42 port function select	P42MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	TOUT1/CAP1			
					0x1	SCLK0			
					0x0	P42			
		D3–2	P41MUX[1:0]	P41 port function select	P41MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	TOUT7/CAP7			
					0x1	SOUT0			
					0x0	P41			
		D1–0	P40MUX[1:0]	P40 port function select	P40MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	TOUT6/CAP6			
					0x1	SIN0			
					0x0	P40			

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

D[7:6] P43MUX[1:0]: P43 Port Function Select Bits

0x3 (R/W): Reserved
0x2 (R/W): REMI (REMC)
0x1 (R/W): SDA1 (I2CS)
0x0 (R/W): P43 port (default)

D[5:4] P42MUX[1:0]: P42 Port Function Select Bits

0x3 (R/W): Reserved
0x2 (R/W): TOUT1 (T16A Ch.0 comparator mode) or CAP1 (T16A Ch.0 capture mode)
0x1 (R/W): SCLK0 (UART Ch.0)
0x0 (R/W): P42 port (default)

D[3:2] P41MUX[1:0]: P41 Port Function Select Bits

0x3 (R/W): Reserved
0x2 (R/W): TOUT7 (T16A Ch.3 comparator mode) or CAP7 (T16A Ch.3 capture mode)
0x1 (R/W): SOUT0 (UART Ch.0)
0x0 (R/W): P41 port (default)

D[1:0] P40MUX[1:0]: P40 Port Function Select Bits

0x3 (R/W): Reserved
0x2 (R/W): TOUT7 (T16A Ch.3 comparator mode) or CAP7 (T16A Ch.3 capture mode)
0x1 (R/W): SIN0 (UART Ch.0)
0x0 (R/W): P40 port (default)

P4[5:4] Port Function Select Register (P44_45PMUX)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
P4[5:4] Port Function Select Register (P44_45PMUX)	0x52a9 (8 bits)	D7–4	–	reserved	–		–	–	0 when being read.
		D3–2	P45MUX[1:0]	P45 port function select	P45MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	SDA0			
					0x0	P45/EXCL0			
		D1–0	P44MUX[1:0]	P44 port function select	P44MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	REMO			
					0x1	SCL1			
					0x0	P44			

The P44 and P45 port pins are shared with the peripheral module pins. This register is used to select how the pins are used.

D[7:4] Reserved

D[3:2] P45MUX[1:0]: P45 Port Function Select Bits

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): SDA0 (I2CM)
0x0 (R/W): P45 port and EXCL0 (T16A Ch.0) (default)

To use the P45 pin for EXCL0 input, P4OEN5/P4_OEN register must be set to 0 and P4IEN5/P4_IEN register must be set to 1.

D[1:0] P44MUX[1:0]: P44 Port Function Select Bits

0x3 (R/W): Reserved
0x2 (R/W): REMO (REMC)
0x1 (R/W): SCL1 (I2CS)
0x0 (R/W): P44 port (default)

P5[3:0] Port Function Select Register (P50_53PMUX)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
P5[3:0] Port Function Select Register (P50_53PMUX)	**0x52aa** (8 bits)	D7–6	**P53MUX[1:0]**	P53 port function select	P53MUX[1:0]	Function	0x0	R/W	* S1C17564 only
					0x3	reserved			
					0x2	reserved			
					0x1	US_SDI1*			
					0x0	P53			
		D5–4	**P52MUX[1:0]**	P52 port function select	P52MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	US_SCK0*			
					0x0	P52			
		D3–2	**P51MUX[1:0]**	P51 port function select	P51MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	US_SDO0*			
					0x0	P51			
		D1–0	**P50MUX[1:0]**	P50 port function select	P50MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	US_SDI0*			
					0x0	P50			

Note: This register can only be used in the S1C17564.

The P50 to P53 port pins of the S1C17564 are shared with the peripheral module pins. This register is used to select how the pins are used.

D[7:6] P53MUX[1:0]: P53 Port Function Select Bits

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): US_SDI1 (S1C17564 USI Ch.1)
0x0 (R/W): P53 port (default)

D[5:4] P52MUX[1:0]: P52 Port Function Select Bits

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): US_SCK0 (S1C17564 USI Ch.0)
0x0 (R/W): P52 port (default)

D[3:2] P51MUX[1:0]: P51 Port Function Select Bits

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): US_SDO0 (S1C17564 USI Ch.0)
0x0 (R/W): P51 port (default)

D[1:0] P50MUX[1:0]: P50 Port Function Select Bits

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): US_SDI0 (S1C17564 USI Ch.0)
0x0 (R/W): P50 port (default)

P5[5:4] Port Function Select Register (P54_55PMUX)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
P5[5:4] Port Function Select Register (P54_55PMUX)	0x52ab (8 bits)	D7–4	–	reserved	–		–	–	0 when being read.
		D3–2	P55MUX[1:0]	P55 port function select	P55MUX[1:0]	Function	0x0	R/W	* S1C17564 only
					0x3	reserved			
					0x2	reserved			
					0x1	US_SCK1*			
					0x0	P55			
		D1–0	P54MUX[1:0]	P54 port function select	P54MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	US_SDO1*			
					0x0	P54			

Note: This register can only be used in the S1C17564.

The P54 and P55 port pins of the S1C17564 are shared with the peripheral module pins. This register is used to select how the pins are used.

D[7:4] Reserved

D[3:2] P55MUX[1:0]: P55 Port Function Select Bits

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): US_SCK1 (S1C17564 USI Ch.1)
0x0 (R/W): P55 port (default)

D[1:0] P54MUX[1:0]: P54 Port Function Select Bits

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): US_SDO1 (S1C17564 USI Ch.1)
0x0 (R/W): P54 port (default)

9 16-bit Timers (T16)

9.1 T16 Module Overview

The S1C17554/564 includes three-channel 16-bit timer module (T16).
The features of the T16 module are listed below.

- 16-bit presettable down counter with a 16-bit reload data register for setting the preset value
- Generates the SPI, I²C master operating clocks and A/D conversion trigger signal 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 9.1.1 shows the T16 configuration.

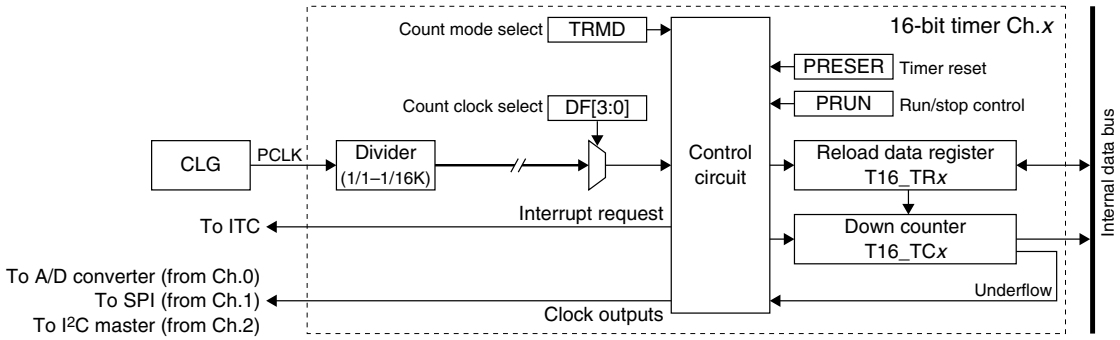


Figure 9.1.1 T16 Configuration (1 Channel)

Each channel of the T16 module consists of a 16-bit presettable down counter and a 16-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 or an A/D converter trigger signal. 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: All three T16 channels have the same functions except for the control register addresses. The description in this chapter applies to all channels. The '*x*' in the register name refers to the channel number (0 to 2).
Example: T16_CTL*x* register
Ch.0: T16_CTL0 register
Ch.1: T16_CTL1 register
Ch.2: T16_CTL2 register

9.2 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]/T16_CLK*x* register.

Table 9.2.1 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.

9.3 Count Mode

The T16 module features two count modes: repeat mode and one-shot mode. These modes are selected using TRMD/T16_CTL*x* register.

Repeat mode (TRMD = 0, default)

Setting TRMD to 0 sets T16 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. T16 should be set to this mode to generate periodic interrupts or A/D conversion triggers at desired intervals or to generate a serial transfer clock.

One-shot mode (TRMD = 1)

Setting TRMD to 1 sets T16 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. T16 should be set to this mode to set a specific wait time.

9.4 Reload Data Register and Underflow Cycle

The reload data register T16_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 or A/D conversion triggers, and the programmable serial interface transfer clock.

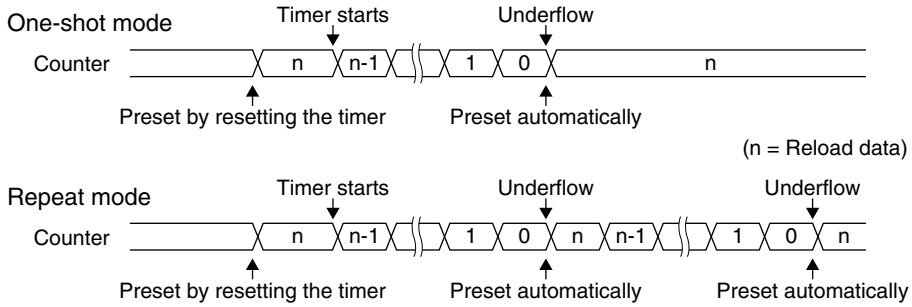


Figure 9.4.1 Preset Timing

The underflow cycle can be calculated as follows:

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

ct_clk: Count clock frequency [Hz]
TR: Reload data (0–65535)

9.5 Timer Reset

The timer is reset by writing 1 to PRESER/T16_CTL*x* register. The reload data is preset and the counter is initialized.

9.6 Timer RUN/STOP Control

Make the following settings before starting the timer.
(1) Select the count clock. See Section 9.2.
(2) Set the count mode (one-shot or repeat). See Section 9.3.
(3) Calculate the initial counter value and set it to the reload data register. See Section 9.4.
(4) Reset the timer to preset the counter to the initial value. See Section 9.5.
(5) When using timer interrupts, set the interrupt level and enable interrupts for the relevant timer channel. See Section 9.8.

To start the timer, write 1 to PRUN/T16_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.

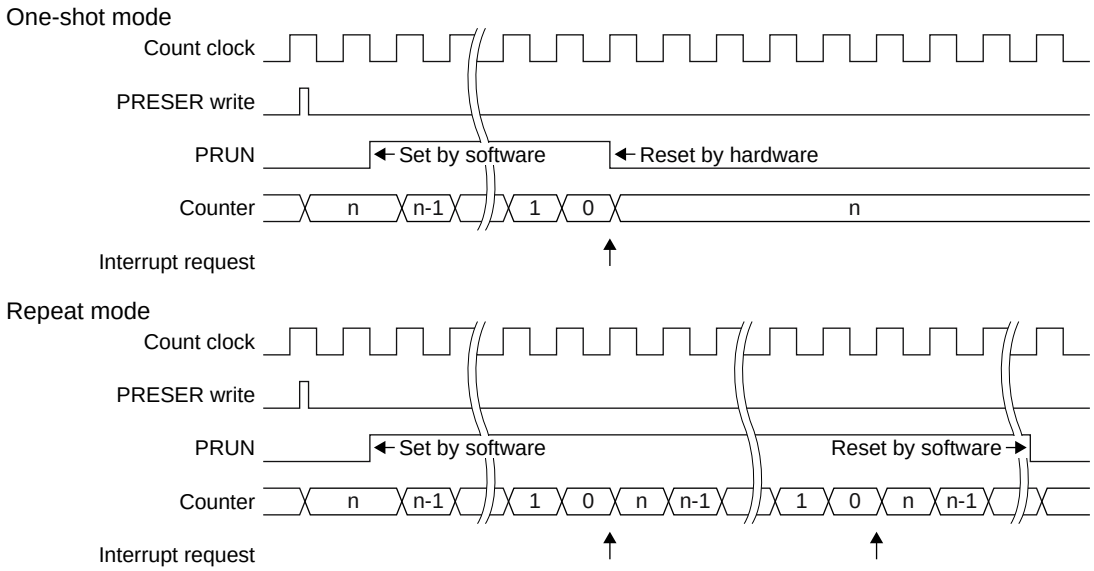


Figure 9.6.1 Count Operation

9.7 T16 Output Signals

The T16 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 or the A/D conversion trigger signal.
The clock generated is sent to the internal peripheral module, as shown below.
T16 Ch.0 output clock → A/D converter
T16 Ch.1 output clock → SPI
T16 Ch.2 output clock → I²C master

Use the following equations to calculate the reload data register value for obtaining the desired transfer rate or A/D conversion interval:

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

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

A/D converter $TR = (ct\_clk \times adi) - 1$

ct_clk: Count clock frequency (Hz)
TR: Reload data (0–65535)
bps: Transfer rate (bits/s)
adi: A/D conversion interval (s)

9.8 T16 Interrupts

Each channel of the T16 module outputs an interrupt request to the interrupt controller (ITC) when the counter underflows.

Underflow interrupt

When the counter underflows, the interrupt flag T16IF/T16_INT*x* register, which is provided for each channel in the T16 module, is set to 1. At the same time, an interrupt request is sent to the ITC if T16IE/T16_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 T16IE 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 T16 module interrupt flag T16IF must be reset in the interrupt handler routine after a T16 interrupt has occurred to prevent recurring interrupts.

• Reset T16IF before enabling T16 interrupts with T16IE to prevent occurrence of unwanted interrupt. T16IF is reset by writing 1.

9.9 Control Register Details

Table 9.9.1 List of T16 Registers

Address	Register name		Function
0x4220	T16_CLK0	T16 Ch.0 Count Clock Select Register	Selects a count clock.
0x4222	T16_TR0	T16 Ch.0 Reload Data Register	Sets reload data.
0x4224	T16_TC0	T16 Ch.0 Counter Data Register	Counter data
0x4226	T16_CTL0	T16 Ch.0 Control Register	Sets the timer mode and starts/stops the timer.
0x4228	T16_INT0	T16 Ch.0 Interrupt Control Register	Controls the interrupt.
0x4240	T16_CLK1	T16 Ch.1 Count Clock Select Register	Selects a count clock.
0x4242	T16_TR1	T16 Ch.1 Reload Data Register	Sets reload data.
0x4244	T16_TC1	T16 Ch.1 Counter Data Register	Counter data
0x4246	T16_CTL1	T16 Ch.1 Control Register	Sets the timer mode and starts/stops the timer.
0x4248	T16_INT1	T16 Ch.1 Interrupt Control Register	Controls the interrupt.
0x4260	T16_CLK2	T16 Ch.2 Count Clock Select Register	Selects a count clock.
0x4262	T16_TR2	T16 Ch.2 Reload Data Register	Sets reload data.
0x4264	T16_TC2	T16 Ch.2 Counter Data Register	Counter data
0x4266	T16_CTL2	T16 Ch.2 Control Register	Sets the timer mode and starts/stops the timer.
0x4268	T16_INT2	T16 Ch.2 Interrupt Control Register	Controls the interrupt.

The T16 registers are described in detail below. These are 16-bit registers.

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

T16 Ch.*x* Count Clock Select Registers (T16_CLK*x*)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
T16 Ch.*x* Count Clock Select Register (T16_CLK*x*)	**0x4220** **0x4240** **0x4260** (16 bits)	D15–4	–	reserved	–		–	–	0 when being read.
		D3–0	**DF[3:0]**	Count clock division ratio select	DF[3:0]	Division ratio	0x0	R/W	Source clock = PCLK
					0xf	reserved			
					0xe	1/16384			
					0xd	1/8192			
					0xc	1/4096			
					0xb	1/2048			
					0xa	1/1024			
					0x9	1/512			
					0x8	1/256			
					0x7	1/128			
					0x6	1/64			
					0x5	1/32			
					0x4	1/16			
					0x3	1/8			
					0x2	1/4			
					0x1	1/2			
					0x0	1/1			

D[15:4] Reserved

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

Selects a PCLK division ratio to generate the count clock.

Table 9.9.2 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.

T16 Ch.*x* Reload Data Registers (T16_TR*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16 Ch.*x*	**0x4222**	D15–0	**TR[15:0]**	Reload data	0x0 to 0xffff	0x0	R/W	
Reload Data	**0x4242**			TR15 = MSB				
Register	**0x4262**			TR0 = LSB				
(T16_TR*x*)	(16 bits)							

D[15:0] TR[15:0]: 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 or A/D conversion trigger, and the programmable serial interface transfer clock.

T16 Ch.*x* Counter Data Registers (T16_TC*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16 Ch.*x*	**0x4224**	D15–0	**TC[15:0]**	Counter data	0x0 to 0xffff	0xffff	R	
Counter Data	**0x4244**			TC15 = MSB				
Register	**0x4264**			TC0 = LSB				
(T16_TC*x*)	(16 bits)							

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

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

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

T16 Ch.*x* Control Registers (T16_CTL*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16 Ch.*x*	**0x4226**	D15–5	**–**	reserved	–	–	–	Do not write 1.
Control Register	**0x4246**							
(T16_CTL*x*)	**0x4266**	D4	**TRMD**	Count mode select	1 One shot 0 Repeat	0	R/W	
	(16 bits)	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 A/D conversion triggers at desired intervals or to generate a serial transfer clock.

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

T16 Ch.*x* Interrupt Control Registers (T16_INT*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16 Ch.*x* Interrupt Control Register (T16_INT*x*)	0x4228 0x4248 0x4268 (16 bits)	D15–9	–	reserved	–	–	–	0 when being read.
		D8	**T16IE**	T16 interrupt enable	1 Enable 0 Disable	0	R/W	
		D7–1	–	reserved	–	–	–	0 when being read.
		D0	**T16IF**	T16 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 T16IE: T16 Interrupt Enable Bit

Enables or disables interrupts caused by counter underflows for each channel.

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

Setting T16IE to 1 enables T16 interrupt requests to the ITC; setting to 0 disables interrupts.

D[7:1] Reserved

D0 T16IF: T16 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

T16IF is the T16 module interrupt flag that is set to 1 when the counter underflows.
T16IF is reset by writing 1.

10 Fine Mode 16-bit Timers (T16F)

10.1 T16F Module Overview

The S1C17554/564 includes two-channel fine mode 16-bit timer module (T16F).
The features of the T16F module are listed below.

- 16-bit presettable down counter with a 16-bit reload data register for setting the preset value
- Generates the USI operating clock from the counter underflow signals. (S1C17564)
- 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.
- Fine mode is provided to minimize transfer rate errors.

Figure 10.1.1 shows the T16F configuration.

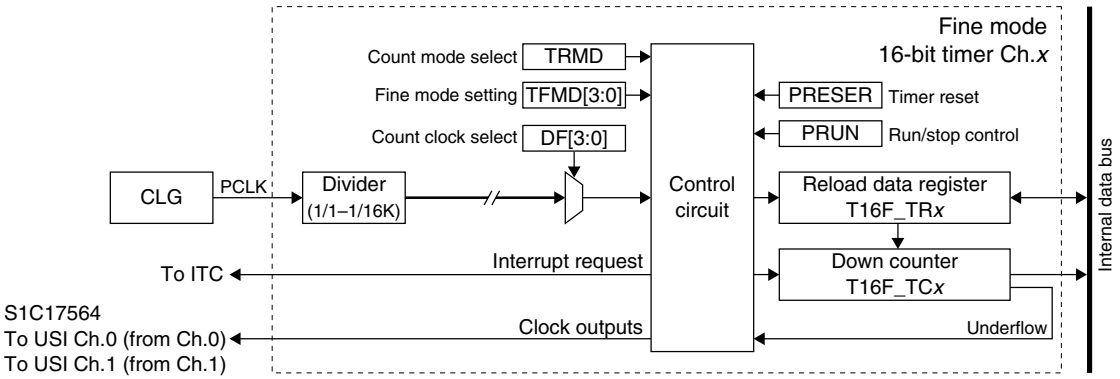


Figure 10.1.1 T16F Configuration (1 Channel)

Each channel of the T16F module consists of a 16-bit presettable down counter and a 16-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: Both T16F channels have the same functions except for the control register addresses. The description in this chapter applies to both channels. The '*x*' in the register name refers to the channel number (0 or 1).
Example: T16F_CTL*x* register
Ch.0: T16F_CTL0 register
Ch.1: T16F_CTL1 register

10.2 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]/T16F_CLK*x* register.

Table 10.2.1 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.

10.3 Count Mode

The T16F module features two count modes: repeat mode and one-shot mode. These modes are selected using TRMD/T16F_CTL*x* register.

Repeat mode (TRMD = 0, default)

Setting TRMD to 0 sets T16F 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. T16F should be set to this mode to generate periodic interrupts or to generate a serial transfer clock.

One-shot mode (TRMD = 1)

Setting TRMD to 1 sets T16F 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. T16F should be set to this mode to set a specific wait time.

10.4 Reload Data Register and Underflow Cycle

The reload data register T16F_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.

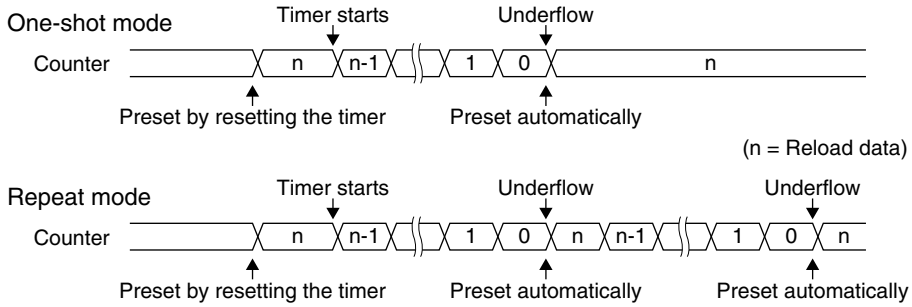


Figure 10.4.1 Preset Timing

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 (0–65535)

10.5 Timer Reset

The timer is reset by writing 1 to PRESER/T16F_CTL*x* register. The reload data is preset and the counter is initialized.

10.6 Timer RUN/STOP Control

Make the following settings before starting the timer.
(1) Select the count clock. See Section 10.2.
(2) Set the count mode (one-shot or repeat). See Section 10.3.
(3) Calculate the initial counter value and set it to the reload data register. See Section 10.4.
(4) Reset the timer to preset the counter to the initial value. See Section 10.5.
(5) When using timer interrupts, set the interrupt level and enable interrupts for the relevant timer channel. See Section 10.9.

To start the timer, write 1 to PRUN/T16F_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.

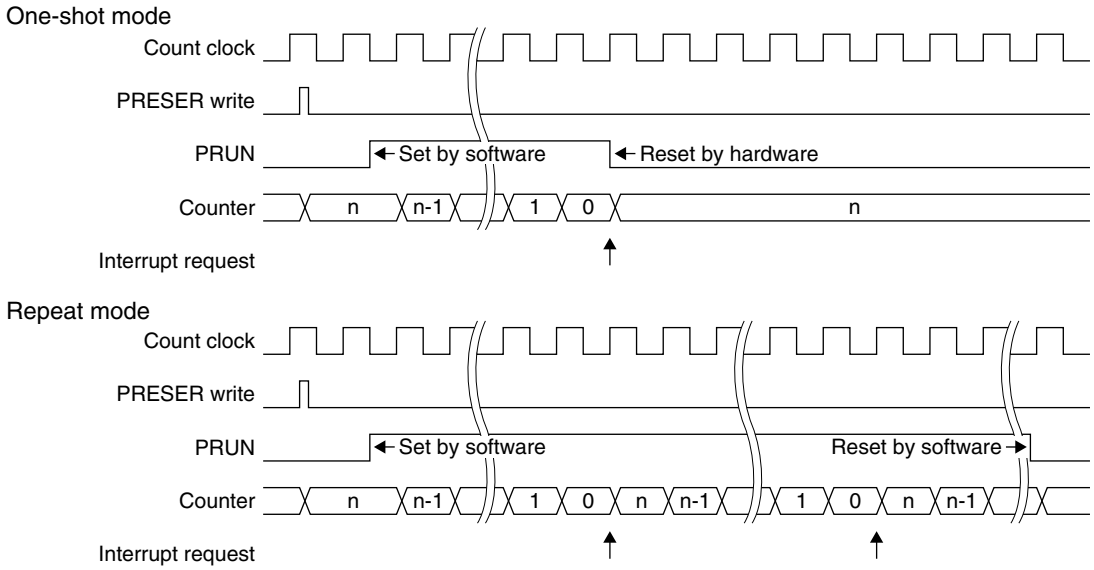


Figure 10.6.1 Count Operation

10.7 T16F Output Signals

The T16F module outputs underflow pulses when the counter underflows.
These pulses are used for timer interrupt requests.
In the S1C17564, these pulses are also used to generate a USI clock. The clock generated is sent to the internal peripheral module, as shown below.
T16F Ch.0 output clock → USI Ch.0
T16F Ch.1 output clock → USI Ch.1

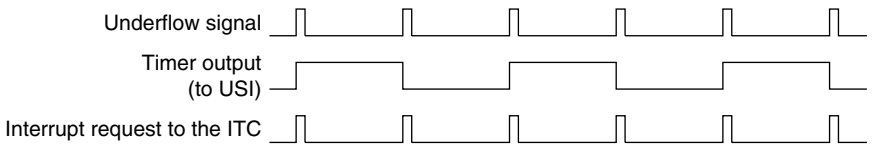


Figure 10.7.1 T16F Output Clock

10.8 Fine Mode

Fine mode provides a function that minimizes transfer rate errors.
T16F can output a programmable clock signal for use as the USI serial transfer clock. The timer output clock can be set to the required frequency by selecting the appropriate count clock and reload data. Note that errors may occur, depending on the transfer rate. Fine mode extends the output clock cycle by delaying the underflow pulse from the counter. This delay can be specified with the TFMD[3:0]/T16F_CTL*x* register.
TFMD[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. This setting delays the interrupt timing in the same way.

Table 10.8.1 Delay Patterns Specified by TFMD[3:0]

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

D: Indicates the insertion of a delay cycle.

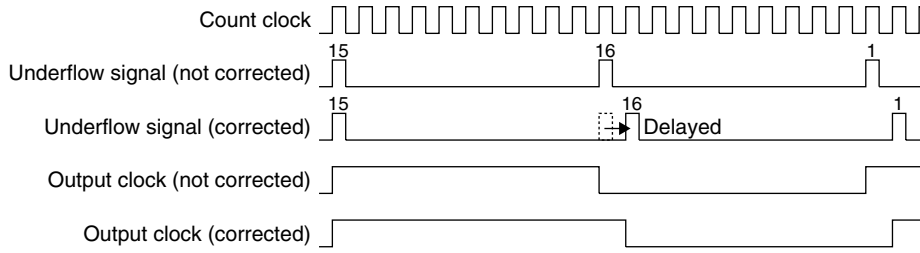


Figure 10.8.1 Delay Cycle Insertion in Fine Mode

At initial reset, TFMD[3:0] is set to 0x0, preventing insertion of delay cycles.

10.9 T16F Interrupts

Each channel of the T16F module outputs an interrupt request to the interrupt controller (ITC) when the counter underflows.

Underflow interrupt

When the counter underflows, the interrupt flag T16FIF/T16F_INT*x* register, which is provided for each channel in the T16F module, is set to 1. At the same time, an interrupt request is sent to the ITC if T16FIE/T16F_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 T16FIE 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 T16F module interrupt flag T16FIF must be reset in the interrupt handler routine after a T16F interrupt has occurred to prevent recurring interrupts.

• Reset T16FIF before enabling T16F interrupts with T16FIE to prevent occurrence of unwanted interrupt. T16FIF is reset by writing 1.

10.10 Control Register Details

Table 10.10.1 List of T16F Registers

Address	Register name		Function
0x4200	T16F_CLK0	T16F Ch.0 Count Clock Select Register	Selects a count clock.
0x4202	T16F_TR0	T16F Ch.0 Reload Data Register	Sets reload data.
0x4204	T16F_TC0	T16F Ch.0 Counter Data Register	Counter data
0x4206	T16F_CTL0	T16F Ch.0 Control Register	Sets the timer mode and starts/stops the timer.
0x4208	T16F_INT0	T16F Ch.0 Interrupt Control Register	Controls the interrupt.
0x4280	T16F_CLK1	T16F Ch.1 Count Clock Select Register	Selects a count clock.
0x4282	T16F_TR1	T16F Ch.1 Reload Data Register	Sets reload data.
0x4284	T16F_TC1	T16F Ch.1 Counter Data Register	Counter data
0x4286	T16F_CTL1	T16F Ch.1 Control Register	Sets the timer mode and starts/stops the timer.
0x4288	T16F_INT1	T16F Ch.1 Interrupt Control Register	Controls the interrupt.

The T16F registers are described in detail below. These are 16-bit registers.

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

T16F Ch.*x* Count Clock Select Registers (T16F_CLK*x*)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
T16F Ch.*x*	**0x4200**	D15–4	**–**	reserved	–		–	–	0 when being read.
Count Clock	**0x4280**	D3–0	**DF[3:0]**	Count clock division ratio select	DF[3:0]	Division ratio	0x0	R/W	Source clock = PCLK
Select Register	(16 bits)				0xf	reserved			
(T16F_CLK*x*)					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 10.10.2 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.

T16F Ch.*x* Reload Data Registers (T16F_TR*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16F Ch.*x*	**0x4202**	D15–0	**TR[15:0]**	Reload data	0x0 to 0xffff	0x0	R/W	
Reload Data	**0x4282**			TR15 = MSB				
Register	(16 bits)			TR0 = LSB				
(T16F_TR*x*)								

D[15:0] TR[15:0]: 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.

T16F Ch.*x* Counter Data Registers (T16F_TC*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16F Ch.*x* Counter Data Register (T16F_TC*x*)	**0x4204** **0x4284** (16 bits)	D15–0	**TC[15:0]**	Counter data TC15 = MSB TC0 = LSB	0x0 to 0xffff	0xffff	R	

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

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

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

T16F Ch.*x* Control Registers (T16F_CTL*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16F Ch.*x* Control Register (T16F_CTL*x*)	**0x4206** **0x4286** (16 bits)	D15–12	**–**	reserved	–	–	–	0 when being read.
		D11–8	**TFMD[3:0]**	Fine mode setup	0x0 to 0xf	0x0	R/W	Set a number of times to insert delay into a 16-underflow period.
		D7–5	**–**	reserved	–	–	–	0 when being read.
		D4	**TRMD**	Count mode select	1 One shot 0 Repeat	0	R/W	
		D3–2	**–**	reserved	–	–	–	0 when being read.
		D1	**PRESER**	Timer reset	1 Reset 0 Ignored	0	W	
		D0	**PRUN**	Timer run/stop control	1 Run 0 Stop	0	R/W	

D[15:12] Reserved

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

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

TFMD[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. This setting delays the interrupt timing in the same way.

Table 10.10.3 Delay Patterns Specified by TFMD[3:0]

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

D: Indicates the insertion of a delay cycle.

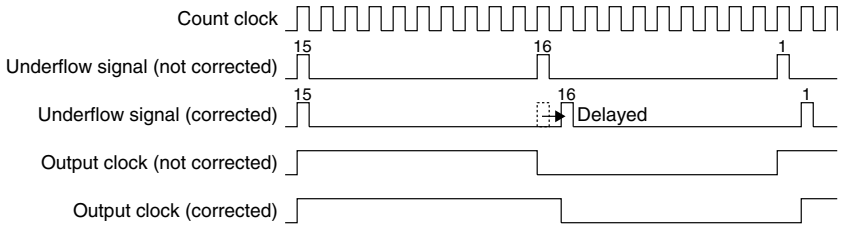


Figure 10.10.1 Delay Cycle Insertion in Fine Mode

D[7:5] Reserved

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 fine mode 16-bit timer stops automatically as soon as the counter underflows. This means only one interrupt can be generated after the timer starts. Note that the timer presets the counter to the reload data register value, then stops when an underflow occurs. Set the 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.

T16F Ch.*x* Interrupt Control Registers (T16F_INT*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16F Ch.*x* Interrupt Control Register (T16F_INT*x*)	0x4208 0x4288 (16 bits)	D15–9	–	reserved	–	–	–	0 when being read.
		D8	T16FIE	T16F interrupt enable	1 Enable 0 Disable	0	R/W	
		D7–1	–	reserved	–	–	–	0 when being read.
		D0	T16FIF	T16F 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 T16FIE: T16F Interrupt Enable Bit

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

Setting T16FIE to 1 enables T16F interrupt requests to the ITC; setting to 0 disables interrupts.

D[7:1] Reserved

D0 T16FIF: T16F 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

T16FIF is the T16F module interrupt flag that is set to 1 when the counter underflows.
T16FIF is reset by writing 1.

11 16-bit PWM Timers (T16A)

11.1 T16A Module Overview

The S1C17554/564 includes a 16-bit PWM timer (T16A) module that consists of four channels 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 T16A are listed below.

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

Figure 11.1.1 shows the T16A configuration.

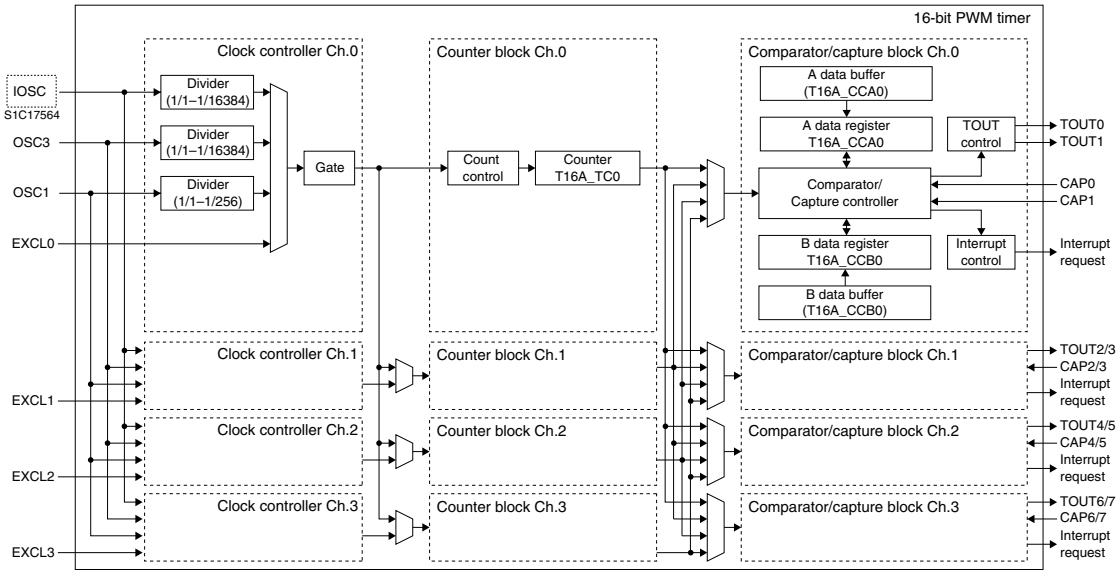


Figure 11.1.1 T16A Configuration

Clock controller

T16A includes four channels of clock controllers that generate the count clock for the counters. The clock source and division ratio can be selected with software.

Counter block

The counter block includes a 16-bit up-counter that operates with an IOSC (S1C17564 only), OSC3, or OSC1 division clock, or the external count clock input from outside the IC. The T16A module allows software to run and stop the counter of each channel, and to reset the counter value (cleared to 0) as well as selection of the count clock. The counter can also be reset by the compare B signal output from the comparator/capture block.

Comparator/capture block

The comparator/capture block includes two systems (units A and B) of comparators that compare between the counter value and the specified comparison value and capture circuits that capture the counter value by an external trigger signal. Note, however, that the comparator and capture functions cannot be used at the same time in each system. One of the two functions must be selected by the software switch.

When using the comparator function, set the value(s) to be compared with the counter value to the compare A and/or compare B registers. When the counter reaches the value set in the compare A or compare B register, the comparator asserts the compare A or compare B signal. These signals can generate interrupts. Also the signals control the cycle time and duty ratio of the timer output signal allowing the timer to output a PWM or other waveform. In addition to these functions, the compare B signal is used to reset the counter.
Comparison data can be read or written directly from/to the compare A and compare B registers. The compare buffers are separately provided to load data to the compare A and compare B registers automatically by the compare B signal. Software can select which of the compare register and buffer the comparison values are written to.

When the capture function is enabled, the compare A and compare B registers are used as the capture A and capture B registers, respectively. The capture A and capture B circuits can input a trigger signal individually, and the counter value is loaded to the respective capture register at the selected edge of the trigger signal.
The capturing operation can generate an interrupt, this make it possible to read the captured data in the interrupt handler routine. Also an overwrite interrupt can be generated for the error handling when the counter value is captured before reading the previous captured data.

Combination of counter block channel and comparator/capture block channel

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

Note: Each channel of the T16A module has the same functions except for the control register addresses. The description in this section applies to all channels of the T16A module otherwise a channel number is specified. The '*x*' in the register name refers to the channel number (0 to 3).
Example: T16A_CTL*x* register
Ch.0: T16A_CTL0 register
Ch.1: T16A_CTL1 register
Ch.2: T16A_CTL2 register
Ch.3: T16A_CTL3 register

11.2 T16A Input/Output Pins

Table 11.2.1 lists the input/output pins for the T16A module.

Table 11.2.1 List of T16A Pins

Pin name		I/O	Qty	Function
EXCL0 EXCL1 EXCL2 EXCL3	(for Ch.0) (for Ch.1) (for Ch.2) (for Ch.3)	I	4	External clock input pins Inputs an external clock for the event counter function.
CAP0, CAP1 CAP2, CAP3 CAP4, CAP5 CAP6, CAP7	(for Ch.0) (for Ch.1) (for Ch.2) (for Ch.3)	I	8	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 CAP0/2/4/6 pin captures the counter data into the capture A register. The CAP1/3/5/7 pin input signal captures the counter data into the capture B register.
TOUT0, TOUT1 TOUT2, TOUT3 TOUT4, TOUT5 TOUT6, TOUT7	(for Ch.0) (for Ch.1) (for Ch.2) (for Ch.3)	O	8	Timer generating signal output pins (effective in comparator mode) Each channel has two output pins and the signals generated in different conditions can be output.

The T16A input/output pins (EXCL*x*, CAP*x*, TOUT*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 T16A input/output pins.
For detailed information on pin function switching, see the "I/O Ports (P)" chapter.

11.3 Count Clock

The clock controller includes a clock source selector, dividers, and a gate circuit for controlling the count clock. The count clock can be controlled in each channel individually.

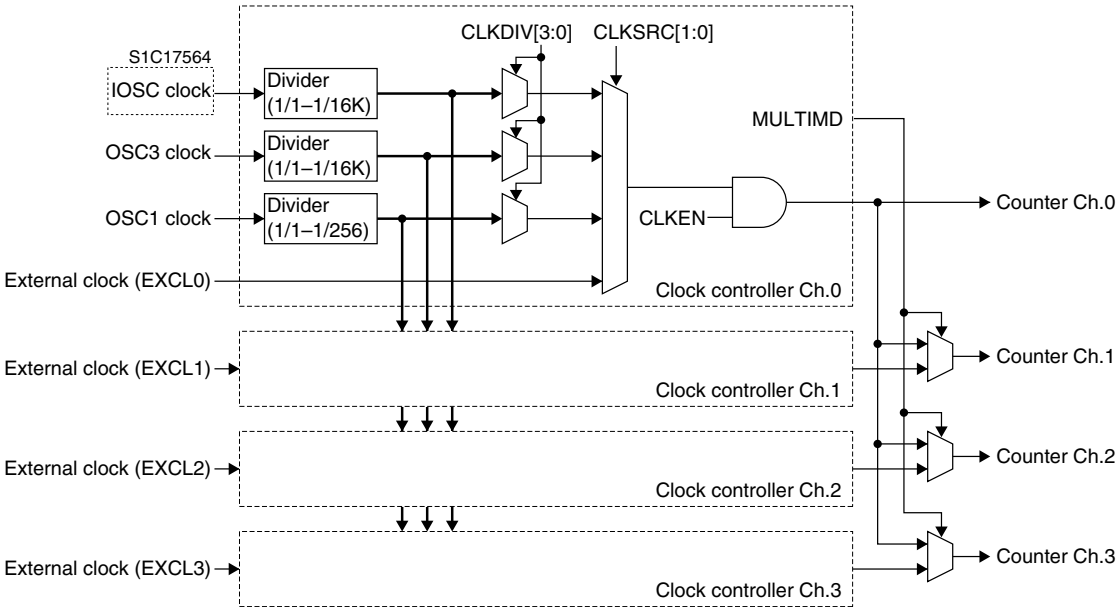


Figure 11.3.1 Clock Controller

Clock source selection

The clock source can be selected from IOSC (S1C17564), OSC3, OSC1, or external clock using CLKSRC[1:0]/ T16A_CLK*x* register.

Table 11.3.1 Clock Source Selection

CLKSRC[1:0]	Clock source
0x3	External clock (EXCL*x*)
0x2	OSC3
0x1	OSC1
0x0	IOSC (S1C17564)

(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 11.3.2 External Clock Input Pins

Channel	External clock input pin
T16A Ch.0	EXCL0
T16A Ch.1	EXCL1
T16A Ch.2	EXCL2
T16A Ch.3	EXCL3

Internal clock division ratio selection

When an internal clock (IOSC, OSC3, or OSC1) is selected, use CLKDIV[3:0]/T16A_CLK*x* register to select the division ratio.

Table 11.3.3 Internal Clock Division Ratio Selection

CLKDIV[3:0]	Division ratio	
	Clock source = IOSC or OSC3	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	Reserved
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 CLKEN/T16A_CLK*x* register. The CLKEN default setting is 0, which disables the clock supply. Setting CLKEN to 1 sends the clock generated as above to the counter. If T16A is not required, disable the clock supply to reduce current consumption.

Multi-comparator/capture mode

The T16A module allows a counter channel to be connected to multiple comparator/capture channels (multi-comparator/capture mode). In this case, all channels must be clocked with the Ch.0 clock. Use MULTIMD/T16A_CLK0 register to supply the Ch.0 clock to all channels. When using T16A in multi-comparator/capture mode, set MULTIMD to 1. When connecting the counter and comparator/capture block in the same channel (normal channel mode), set MULTIMD to 0 (default).

Note: Make sure the T16A count is stopped before setting the count clock.

11.4 T16A Operating Modes

The T16A module provides some operating modes to support various usages. This section describes the functions of each operating mode and how to enter the mode.

11.4.1 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 11.6, "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

Comparison data can be read or written directly from/to the compare registers. Comparison data for system A or B can also be written to the compare buffer so that it will be loaded to the compare A or compare B register when the compare B signal is generated. The CBUFEN/T16A_CTL*x* register is used to select whether comparison data is written to the compare register or buffer.
Setting CBUFEN to 0 (default) selects the compare registers. Setting it to 1 selects the compare buffers.
Although the T16A_CCA*x* and T16A_CCB*x* registers are used to read compare data regardless of the CBUFEN setting, compare registers will be accessed.

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 11.4.1.1 List of Counter Capture Trigger Signal Input Pins

Channel	Trigger input pins	
	Capture A	Capture B
T16A Ch.0	CAP0	CAP1
T16A Ch.1	CAP2	CAP3
T16A Ch.2	CAP4	CAP5
T16A Ch.3	CAP6	CAP7

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 11.4.1.2 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.

11.4.2 Repeat Mode and One-Shot Mode

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

Repeat mode (TRMD = 0, default)

Setting TRMD to 0 sets the corresponding counter to repeat mode.

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

One-shot mode (TRMD = 1)

Setting TRMD to 1 sets the corresponding counter to one-shot mode.

In this mode, the counter stops automatically as soon as the compare B signal is generated. The counter should be set to this mode to set a specific wait time or for pulse width measurement.

11.4.3 Normal Channel Mode and Multi-Comparator/Capture Mode

One channel of the T16A module basically consists of a counter block and a comparator/capture block. The T16A module also allows the application to use expanded comparator/capture function by connecting two or more comparator/capture blocks to one counter block. To support this expansion, two operating modes are provided: normal channel mode and multi-comparator/capture mode. This operating mode can be selected using MULTIMD/T16_CLK0 register.

Normal channel mode (MULTIMD = 0, default)

Set the T16A module to this mode when using it as four channels of different timers by connecting a counter block with the comparator/capture block of the same channel. In this mode, the counters can use different count clocks.

Each timer channel provides CCABCNT[1:0]/T16A_CTL*x* register to select a counter channel to be connected to the comparator/capture block.

Table 11.4.3.1 Counter Selection

CCABCNT[1:0]	Counter channel
0x3	Ch.3 (Counter 3)
0x2	Ch.2 (Counter 2)
0x1	Ch.1 (Counter 1)
0x0	Ch.0 (Counter 0)

(Default: 0x0)

When using the T16A module in normal channel mode, be sure to connect the counter block to the comparator/capture block in the same channel.

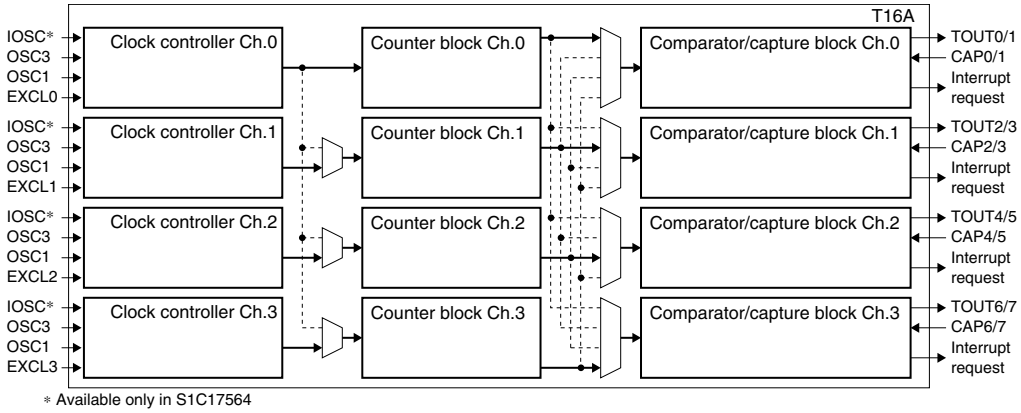
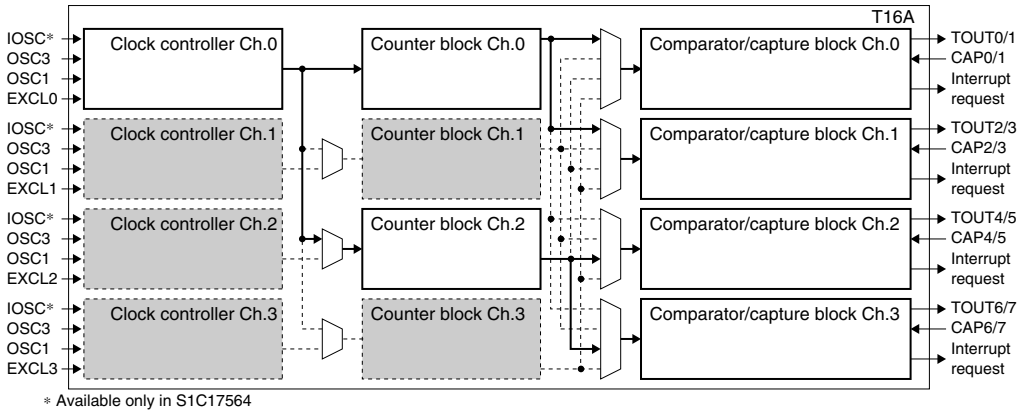


Figure 11.4.3.1 Timer Configuration in Normal Channel Mode (two comparator/capture blocks × four channels)

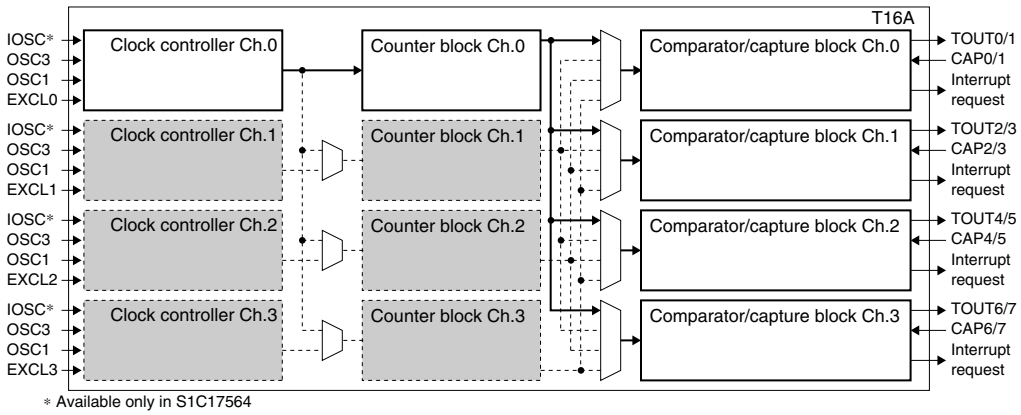
Note: Do not connect a counter block to a comparator/capture block in a different channel in normal channel mode (MULTIMD = 0), as normal operation cannot be guaranteed.

Multi-comparator/capture mode (MULTIMD = 1)

In order to set three or more comparison values for one counter or to capture the contents of one counter using three or more trigger signals, two or more comparator/capture blocks can be connected to one counter. Multi-comparator/capture mode is provided for this purpose. In this mode, any counter block can be combined with the comparator/capture blocks using CCABCNT[1:0] described above. Note, however, that the count clock is fixed at one type for counter Ch.0, regardless of the counter to be used. The clock settings for Ch.1 to Ch.3 are ineffective.



(1) Configuration Example 1 (four comparator/capture blocks × two channels)



(2) Configuration Example 2 (eight comparator/capture blocks × one channel)

Figure 11.4.3.2 Timer Configuration Example in Multi-Comparator/Capture Mode

11.5 Counter Control

11.5.1 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.

11.5.2 Counter RUN/STOP Control

Make the following settings before starting the count operation.
(1) Switch the input/output pin functions to be used for T16A. Refer to the "I/O Port (P)" chapter.
(2) Select operating modes. See Section 11.4.
(3) Select the clock source. See Section 11.3.
(4) Configure the timer outputs (TOUT). See Section 11.6.
(5) If using interrupts, set the interrupt level and enable the T16A interrupts. See Section 11.7.
(6) Reset the counter to 0. See Section 11.5.1.
(7) Set comparison data (in comparator mode). See Section 11.4.1.

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

11.5.3 Reading Counter Values

The counter value can be read from T16ATC[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.

11.5.4 Timing Charts

Comparator mode

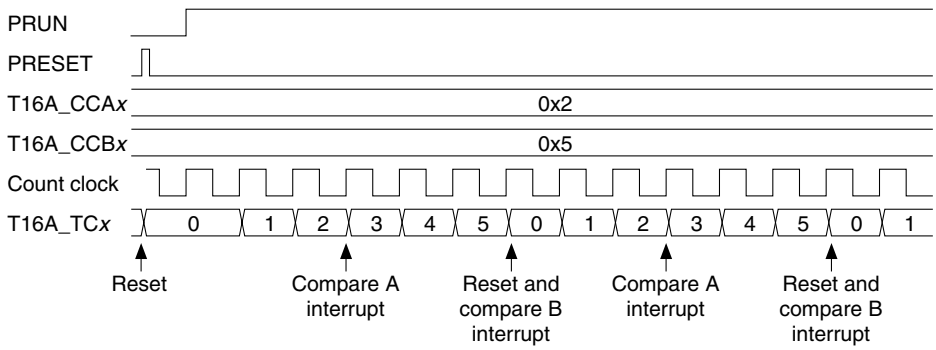


Figure 11.5.4.1 Operation Timing in Comparator Mode

Capture mode

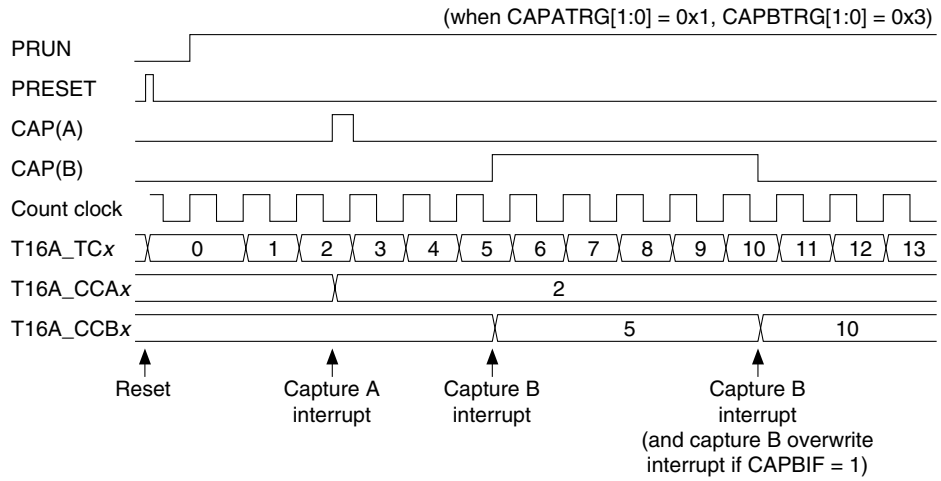


Figure 11.5.4.2 Operation Timing in Capture Mode

11.6 Timer Output Control

The timer that has been set in comparator mode can generate TOUT signals using the compare A and compare B signals and can output it to external devices. Each timer channel provides two TOUT outputs, thus the T16A module can output up to eight TOUT signals.
Figure 11.6.1 shows the TOUT output circuit (one timer channel).

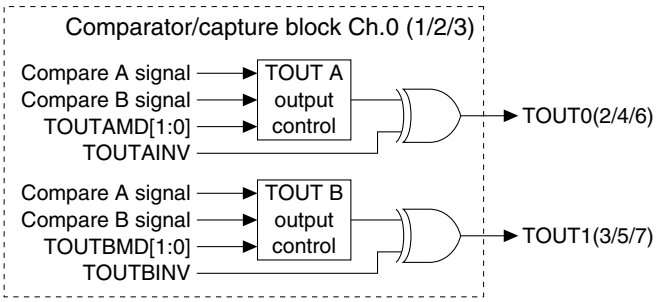


Figure 11.6.1 TOUT Output Circuit

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

TOUT output pins

Table 11.6.1 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 11.6.1 List of TOUT Output Pins

Channel	TOUT output pin	
	System A	System B
T16A Ch.0	TOUT0	TOUT1
T16A Ch.1	TOUT2	TOUT3
T16A Ch.2	TOUT4	TOUT5
T16A Ch.3	TOUT6	TOUT7

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 11.6.2 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 11.6.2 illustrates the TOUT output waveform.

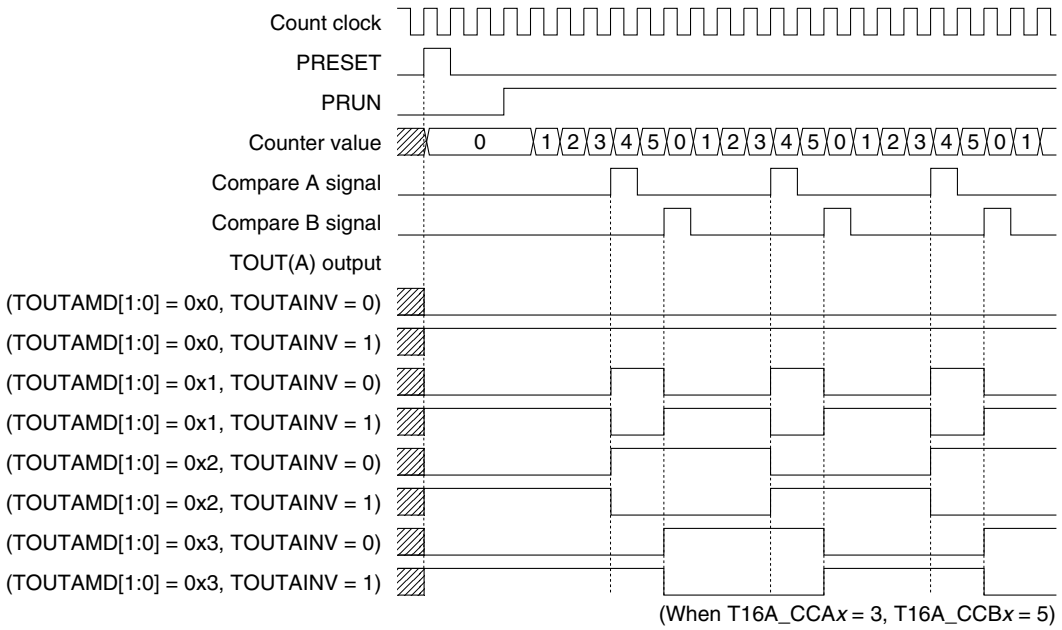


Figure 11.6.2 TOUT Output Waveform

11.7 T16A Interrupts

The T16A module can generate the following six kinds of interrupts:

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

Each timer channel outputs a single interrupt signal shared by the above interrupt causes to the interrupt controller (ITC). Read the interrupt flags in the T16A 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 T16A 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 T16A 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 T16A 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 T16A 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 T16A 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 T16A 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 T16A 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 T16A module must be reset in the interrupt handler routine.

11.8 Control Register Details

Table 11.8.1 List of T16A Registers

Address	Register name		Function
0x5068	T16A_CLK0	T16A Clock Control Register Ch.0	Controls the T16A Ch.0 clock.
0x5069	T16A_CLK1	T16A Clock Control Register Ch.1	Controls the T16A Ch.1 clock.
0x506a	T16A_CLK2	T16A Clock Control Register Ch.2	Controls the T16A Ch.2 clock.
0x506b	T16A_CLK3	T16A Clock Control Register Ch.3	Controls the T16A Ch.3 clock.
0x5400	T16A_CTL0	T16A Counter Ch.0 Control Register	Controls the counter.
0x5402	T16A_TC0	T16A Counter Ch.0 Data Register	Counter data
0x5404	T16A_CCCTL0	T16A Comparator/Capture Ch.0 Control Register	Controls the comparator/capture block and TOUT.
0x5406	T16A_CCA0	T16A Compare/Capture Ch.0 A Data Register	Compare A/capture A data
0x5408	T16A_CCB0	T16A Compare/Capture Ch.0 B Data Register	Compare B/capture B data
0x540a	T16A_IEN0	T16A Compare/Capture Ch.0 Interrupt Enable Register	Enables/disables interrupts.
0x540c	T16A_IFLG0	T16A Compare/Capture Ch.0 Interrupt Flag Register	Displays/sets interrupt occurrence status.
0x5420	T16A_CTL1	T16A Counter Ch.1 Control Register	Controls the counter.
0x5422	T16A_TC1	T16A Counter Ch.1 Data Register	Counter data
0x5424	T16A_CCCTL1	T16A Comparator/Capture Ch.1 Control Register	Controls the comparator/capture block and TOUT.
0x5426	T16A_CCA1	T16A Compare/Capture Ch.1 A Data Register	Compare A/capture A data
0x5428	T16A_CCB1	T16A Compare/Capture Ch.1 B Data Register	Compare B/capture B data
0x542a	T16A_IEN1	T16A Compare/Capture Ch.1 Interrupt Enable Register	Enables/disables interrupts.
0x542c	T16A_IFLG1	T16A Compare/Capture Ch.1 Interrupt Flag Register	Displays/sets interrupt occurrence status.
0x5440	T16A_CTL2	T16A Counter Ch.2 Control Register	Controls the counter.
0x5442	T16A_TC2	T16A Counter Ch.2 Data Register	Counter data
0x5444	T16A_CCCTL2	T16A Comparator/Capture Ch.2 Control Register	Controls the comparator/capture block and TOUT.
0x5446	T16A_CCA2	T16A Compare/Capture Ch.2 A Data Register	Compare A/capture A data
0x5448	T16A_CCB2	T16A Compare/Capture Ch.2 B Data Register	Compare B/capture B data
0x544a	T16A_IEN2	T16A Compare/Capture Ch.2 Interrupt Enable Register	Enables/disables interrupts.
0x544c	T16A_IFLG2	T16A Compare/Capture Ch.2 Interrupt Flag Register	Displays/sets interrupt occurrence status.
0x5460	T16A_CTL3	T16A Counter Ch.3 Control Register	Controls the counter.
0x5462	T16A_TC3	T16A Counter Ch.3 Data Register	Counter data
0x5464	T16A_CCCTL3	T16A Comparator/Capture Ch.3 Control Register	Controls the comparator/capture block and TOUT.
0x5466	T16A_CCA3	T16A Compare/Capture Ch.3 A Data Register	Compare A/capture A data
0x5468	T16A_CCB3	T16A Compare/Capture Ch.3 B Data Register	Compare B/capture B data
0x546a	T16A_IEN3	T16A Compare/Capture Ch.3 Interrupt Enable Register	Enables/disables interrupts.
0x546c	T16A_IFLG3	T16A Compare/Capture Ch.3 Interrupt Flag Register	Displays/sets interrupt occurrence status.

The T16A 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 0x5069 0x506a 0x506b** (8 bits)	D7–4	**CLKDIV [3:0]**	Clock division ratio select	CLKDIV[3:0] / Division ratio: OSC3 or IOSC / OSC1 0xf / – / – 0xe / 1/16384 / – 0xd / 1/8192 / – 0xc / 1/4096 / – 0xb / 1/2048 / – 0xa / 1/1024 / – 0x9 / 1/512 / – 0x8 / 1/256 / 1/256 0x7 / 1/128 / 1/128 0x6 / 1/64 / 1/64 0x5 / 1/32 / 1/32 0x4 / 1/16 / 1/16 0x3 / 1/8 / 1/8 0x2 / 1/4 / 1/4 0x1 / 1/2 / 1/2 0x0 / 1/1 / 1/1	0x0	R/W	
		D3–2	**CLKSRC [1:0]**	Clock source select	CLKSRC[1:0] / Clock source 0x3 / External clock 0x2 / OSC3 0x1 / OSC1 0x0 / IOSC*	0x0	R/W	* S1C17564 only
		D1	**MULTIMD**	Multi-comparator/capture mode select	1 Multi / 0 Normal	0	R/W	T16A_CLK0
			–	reserved	–	–	–	T16A_CLK1–3 0 when being read.
		D0	**CLKEN**	Count clock enable	1 Enable / 0 Disable	0	R/W	

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

Selects the division ratio for generating the count clock when an internal clock (IOSC, OSC3, or OSC1) is used.

Table 11.8.2 Internal Clock Division Ratio Selection

CLKDIV[3:0]	Division ratio	
	Clock source = IOSC or OSC3	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	Reserved
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] CLKSRC[1:0]: Clock Source Select Bits

Selects the count clock source.

Table 11.8.3 Clock Source Selection

CLKSRC[1:0]	Clock source
0x3	External clock (EXCL*x*)
0x2	OSC3
0x1	OSC1
0x0	IOSC (S1C17564)

(Default: 0x0)

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

D1 MULTIMD: Multi-Comparator/Capture Mode Select Bit (T16A_CLK0 register)

Sets the T16A module to multi-comparator/capture mode.
1 (R/W): Multi-comparator/capture mode
0 (R/W): Normal channel mode (default)

In multi-comparator/capture mode, the clock for Ch.0 configured in the T16A_CLK0 register is supplied to all timer channels. In normal channel mode, different clock configured for each channel individually is supplied to the respective counter.

D1 Reserved (T16A_CLK1–3 registers)

D0 CLKEN: 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 CLKEN default setting is 0, which disables the clock supply. Setting CLKEN to 1 sends the clock selected as above to the counter. If timer operation is not required, disable the clock supply to reduce current consumption.

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

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
T16A Counter Ch.*x* Control Register (T16A_CTL*x*)	0x5400 0x5420 0x5440 0x5460 (16 bits)	D15–6	–	reserved	–				–	–	0 when being read.
		D5–4	CCABCNT [1:0]	Counter select	CCABCNT[1:0] 0x3 0x2 0x1 0x0	Counter Ch. Ch.3 Ch.2 Ch.1 Ch.0			0x0	R/W	
		D3	CBUFEN	Compare buffer enable	1	Enable	0	Disable	0	R/W	
		D2	TRMD	Count mode select	1	One-shot	0	Repeat	0	R/W	
		D1	PRESET	Counter reset	1	Reset	0	Ignored	0	W	0 when being read.
		D0	PRUN	Counter run/stop control	1	Run	0	Stop	0	R/W	

D[15:6] Reserved

D[5:4] CCABCNT[1:0]: Counter Select Bits

Selects a counter to be connected to the comparator/capture block of each channel in multi-comparator/capture mode (MULTIMD/T16A_CLK0 register = 1).

Table 11.8.4 Counter Selection

CCABCNT[1:0]	Counter channel
0x3	Ch.3 (Counter 3)
0x2	Ch.2 (Counter 2)
0x1	Ch.1 (Counter 1)
0x0	Ch.0 (Counter 0)

(Default: 0x0)

When using the T16A module in normal channel mode (T16AMULTIMD = 0), be sure to connect the counter of the same channel to each comparator/capture block.

D3 CBUFEN: Compare Buffer Enable Bit

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

When CBUFEN is set to 1, compare data is written via the compare data buffer. The buffer contents are loaded into the compare A and compare B registers when the compare B signal is generated.
When CBUFEN is set to 0, compare data is written directly to the compare A and compare B registers.

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

D2 TRMD: Count Mode Select Bit

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

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

D1 PRESET: Counter Reset Bit

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

Writing 1 to this bit resets the counter to 0.

D0 PRUN: Counter Run/Stop Control Bit

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

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

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

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16A Counter Ch.*x* Data Register (T16A_TC*x*)	**0x5402** **0x5422** **0x5442** **0x5462** (16 bits)	D15–0	**T16ATC [15:0]**	Counter data T16ATC15 = MSB T16ATC0 = LSB	0x0 to 0xffff	0x0	R	

D[15:0] T16ATC[15:0]: Counter Data Bits

Counter data can be read out. (Default: 0x0)
The counter value can be read out even if the counter is running. However, the counter value should be read at once using a 16-bit transfer instruction. If data is read twice using an 8-bit transfer instruction, the correct value may not be obtained due to occurrence of count up between readings.

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

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16A Comparator/ Capture Ch.*x* Control Register (T16A_CCCTL*x*)	**0x5404 0x5424 0x5444 0x5464** (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 (CAP1/3/5/7) at which the counter value is captured in the capture B register.

Table 11.8.5 Capture B Trigger Edge Selection

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

(Default: 0x0)

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

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

Configures how the TOUT B signal waveform (TOUT1/3/5/7 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 11.8.6 TOUT B Generation Mode

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

(Default: 0x0)

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

D[11:10] Reserved

D9 TOUTBINV: TOUT B Invert Bit

Selects the TOUT B signal (TOUT1/3/5/7 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 (CAP0/2/4/6) at which the counter value is captured in the capture A register.

Table 11.8.7 Capture A Trigger Edge Selection

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

(Default: 0x0)

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

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

Configures how the TOUT A signal waveform (TOUT0/2/4/6 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 11.8.8 TOUT A Generation Mode

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

(Default: 0x0)

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

D[3:2] Reserved

D1 TOUTAINV: TOUT A Invert Bit

Selects the TOUT A signal (TOUT0/2/4/6 output) polarity.
1 (R/W): Inverted (active Low)
0 (R/W): Normal (active High) (default)

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

D0 CCAMD: T16A_CCA Register Mode Select Bit

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

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

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

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16A Comparator/ Capture Ch.*x* A Data Register (T16A_CCA*x*)	0x5406 0x5426 0x5446 0x5466 (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 T16A 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 (CAP0/2/4/6) edge selected using CAPATRG[1:0]/T16A_CCCTL*x* register, the captured value is loaded to this register. At the same time a capture A interrupt can be generated, thus the captured counter value can be read out in the interrupt handler.

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

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16A Comparator/ Capture Ch.*x* B Data Register (T16A_CCB*x*)	0x5408 0x5428 0x5448 0x5468 (16 bits)	D15–0	CCB[15:0]	Compare/capture B data CCB15 = MSB CCB0 = LSB	0x0 to 0xffff	0x0	R/W	

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

In comparator mode (CCBMD/ T16A_CCCTL*x* register = 0)

Sets a compare B data, which will be compared with the counter value, through this register.

When CBUFEN/T16A_CTL*x* register is set to 0, compare B data will be set to the compare B register after a lapse of two T16A 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 (CAP1/3/5/7) 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.

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 T16A count clock cycles.

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

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
T16A	0x540a	D15–6	–	reserved	–				–	–	0 when being read.
Comparator/	0x542a	D5	**CAPBOWIE**	Capture B overwrite interrupt enable	1	Enable	0	Disable	0	R/W	
Capture Ch.*x*	0x544a	D4	**CAPAOWIE**	Capture A overwrite interrupt enable	1	Enable	0	Disable	0	R/W	
Interrupt Enable	0x546a	D3	**CAPBIE**	Capture B interrupt enable	1	Enable	0	Disable	0	R/W	
Register	(16 bits)	D2	**CAPAIE**	Capture A interrupt enable	1	Enable	0	Disable	0	R/W	
(T16A_IEN*x*)		D1	**CBIE**	Compare B interrupt enable	1	Enable	0	Disable	0	R/W	
		D0	**CAIE**	Compare A interrupt enable	1	Enable	0	Disable	0	R/W	

D[15:6] Reserved

D5 CAPBOWIE: Capture B Overwrite Interrupt Enable Bit

Enables or disables capture B overwrite interrupts.
1 (R/W): Interrupt enabled
0 (R/W): Interrupt disabled (default)

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

D4 CAPAOWIE: Capture A Overwrite Interrupt Enable Bit

Enables or disables capture A overwrite interrupts.
1 (R/W): Interrupt enabled
0 (R/W): Interrupt disabled (default)

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

D3 CAPBIE: Capture B Interrupt Enable Bit

Enables or disables capture B interrupts.
1 (R/W): Interrupt enabled
0 (R/W): Interrupt disabled (default)

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

D2 CAPAIE: Capture A Interrupt Enable Bit

Enables or disables capture A interrupts.
1 (R/W): Interrupt enabled
0 (R/W): Interrupt disabled (default)

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

D1 CBIE: Compare B Interrupt Enable Bit

Enables or disables compare B interrupts.
1 (R/W): Interrupt enabled
0 (R/W): Interrupt disabled (default)

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

D0 CAIE: Compare A Interrupt Enable Bit

Enables or disables compare A interrupts.
1 (R/W): Interrupt enabled
0 (R/W): Interrupt disabled (default)

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

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

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
T16A	0x540c	D15–6	–	reserved	–				–	–	0 when being read.
Comparator/	0x542c	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.
Capture Ch.*x*	0x544c	D4	CAPAOWIF	Capture A overwrite interrupt flag					0	R/W	
Interrupt Flag	0x546c	D3	CAPBIF	Capture B interrupt flag					0	R/W	
Register	(16 bits)	D2	CAPAIF	Capture A interrupt flag					0	R/W	
(T16A_IFLG*x*)		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 T16A 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 T16A 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 T16A 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 T16A 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 T16A 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 T16A 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.

12 Clock Timer (CT)

12.1 CT Module Overview

The S1C17554/564 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 12.1.1 shows the CT configuration.

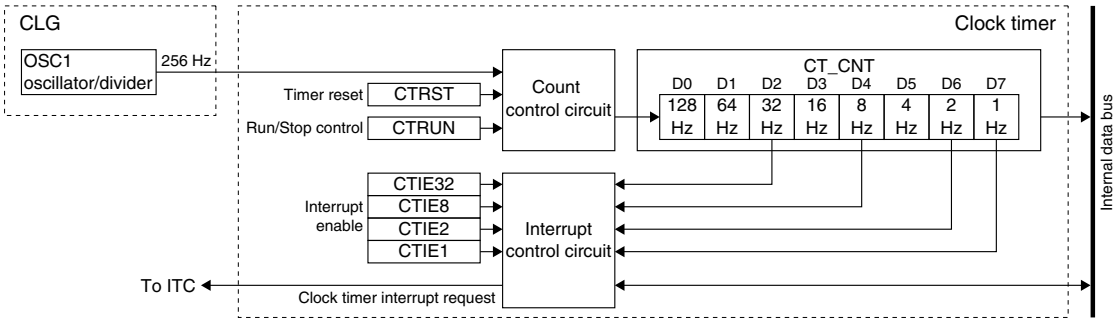


Figure 12.1.1 CT Configuration

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.

12.2 Operation Clock

The CT module uses the 256 Hz clock output by the CLG module as the operation clock. The CLG module generates this operation clock by dividing the OSC1 clock into 1/128, resulting in a frequency of 256 Hz when the OSC1 clock frequency is 32.768 kHz. The frequency described in this chapter will vary accordingly for other OSC1 clock frequencies.
The CLG module does not include a 256 Hz clock output control bit. The 256 Hz clock is normally supplied to the clock timer when the OSC1 oscillation is on.
For detailed information on OSC1 oscillator control, see the "Clock Generator (CLG)" chapter.

Note: The OSC1 oscillator must be turned on before the CT module can operate.

12.3 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.

12.4 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 12.5.
(2) Reset the timer. See Section 12.3.

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).

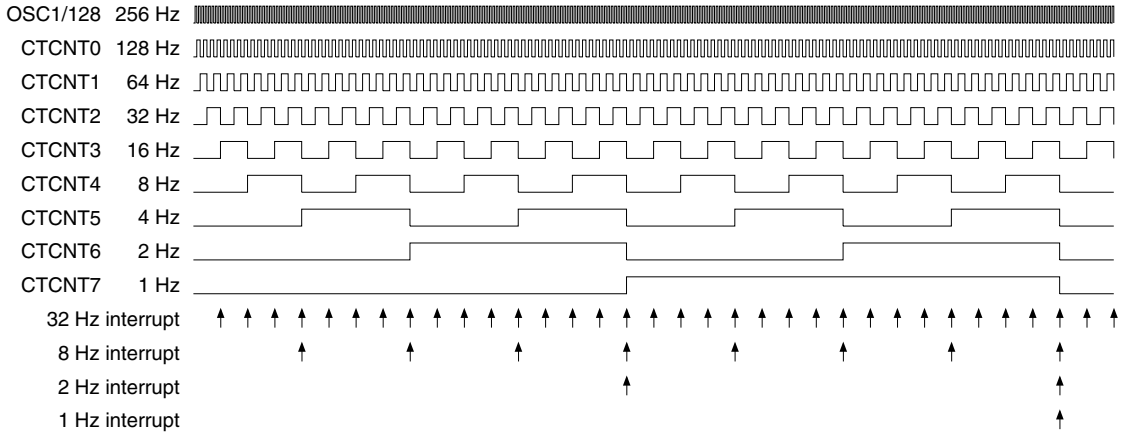


Figure 12.4.1 Clock Timer Timing Chart

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 12.4.2 shows the Run/Stop control timing chart.

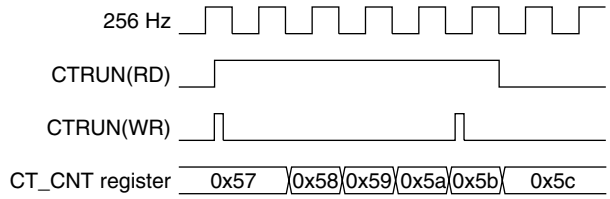


Figure 12.4.2 Run/Stop Control Timing Chart

• 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.

12.5 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 12.5.1 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.

12.6 Control Register Details

Table 12.6.1 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. These are 8-bit registers.

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

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

13 Stopwatch Timer (SWT)

13.1 SWT Module Overview

The S1C17554/564 includes a 1/100-second stopwatch timer module (SWT) that uses the OSC1 oscillator as its clock source. This timer can be used to implement a software stopwatch function.
The features of the SWT module are listed below.

- Two 4-bit BCD counters (approximately 1/100 and 1/10-second counters)
- Approximately 100 Hz, approximately 10 Hz, and 1 Hz interrupts can be generated.

Figure 13.1.1 shows the SWT configuration.

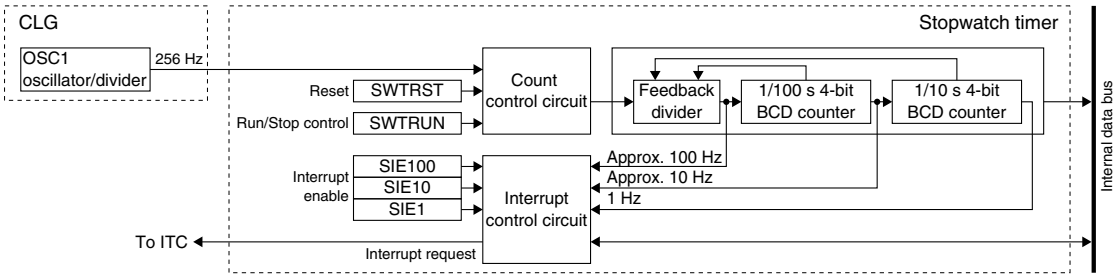


Figure 13.1.1 SWT Configuration

The SWT module consists of two 4-bit BCD counters (1/100 and 1/10 second) that use the 256 Hz signal divided from the OSC1 clock as the input clock and allows count data to be read out by software.
The SWT module can also generate interrupts using the 100 Hz (approximately 100 Hz), 10 Hz (approximately 10 Hz), and 1 Hz signals.

13.2 Operation Clock

The SWT module uses the 256 Hz clock output by the CLG module as the operation clock. The CLG module generates this operation clock by dividing the OSC1 clock into 1/128, resulting in a frequency of 256 Hz when the OSC1 clock frequency is 32.768 kHz. The frequency described in this chapter will vary accordingly for other OSC1 clock frequencies. The CLG module does not include a 256 Hz clock output control bit. The 256 Hz clock is normally supplied to the SWT module when the OSC1 oscillation is on.
For detailed information on OSC1 oscillator control, see the “Clock Generator (CLG)” chapter.

Note: The OSC1 oscillator must be turned on before the SWT module can operate.

13.3 BCD Counters

The SWT module consists of 1/100-second and 1/10-second 4-bit BCD counters.
The 1/100-second and 1/10-second counter values can be read from BCD100[3:0]/SWT_BCNT register and BCD10[3:0]/SWT_BCNT register, respectively.

Count-up Pattern

A feedback divider is used to generate 100 Hz, 10 Hz, and 1 Hz signals from the 256 Hz clock. The counter count-up pattern varies as shown in Figure 13.3.1.

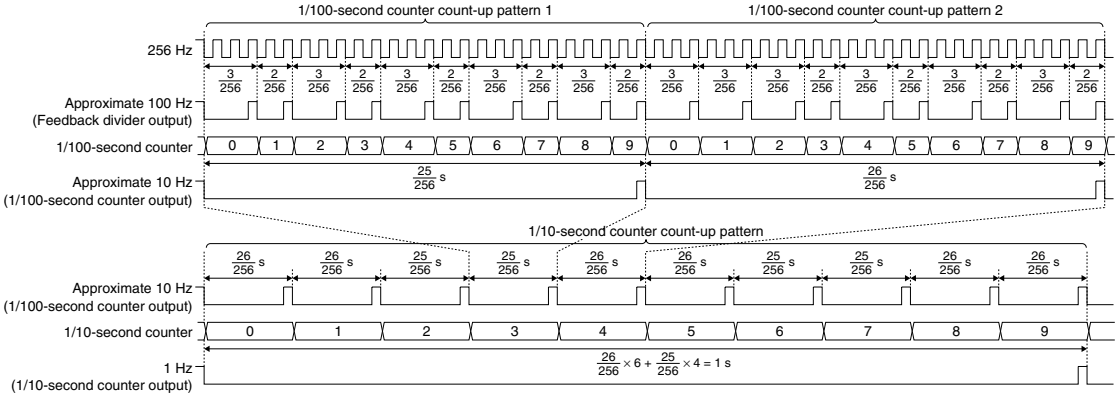


Figure 13.3.1 SWT Count-Up Patterns

The feedback divider generates an approximate 100 Hz signal at 2/256-second and 3/256-second intervals from the 256 Hz signal supplied from the CLG module.

The 1/100-second counter counts the approximate 100 Hz signal output by the feedback divider and generates an approximate 10 Hz signal at 25/256-second and 26/256-second intervals. Count-up will be pseudo 1/100-second counting at 2/256-second and 3/256-second intervals.

The 1/10-second counter counts the approximate 10 Hz signal generated by the 1/100-second counter at a ratio of 4:6, and generates a 1 Hz signal. Count-up will be pseudo 1/10-second counting at 25/256-second and 26/256-second intervals.

13.4 Timer Reset

Reset the SWT module by writing 1 to SWTRST/SWT_CTL register. This clears the counter to 0.
Apart from this operation, the counter is also cleared by initial reset.

13.5 Timer RUN/STOP Control

Make the following settings before starting SWT.

(1) If using interrupts, set the interrupt level and enable interrupts for the SWT module. See Section 13.6.
(2) Reset the timer. See Section 13.4.

The SWT module includes SWTRUN/SWT_CTL register for Run/Stop control.
The timer starts operating when 1 is written to SWTRUN. Writing 0 to SWTRUN disables clock input and stops the operation. This control does not affect the counter (SWT_BCNT register) data. The counter data is retained even when the count is halted, allowing resumption of the count from that data. If 1 is written to both SWTRUN and SWTRST simultaneously, the timer starts counting after resetting.
A cause of interrupt occurs during counting at the 100 Hz (approximate 100 Hz), 10 Hz (approximate 10 Hz), and 1 Hz signal falling edges. If interrupts are enabled, an interrupt request is sent to the interrupt controller (ITC).

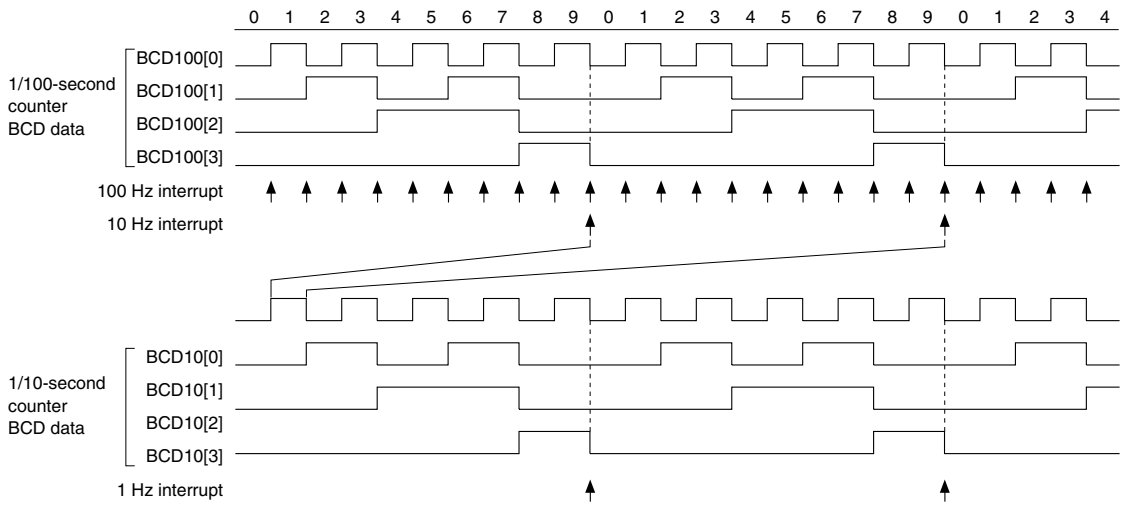


Figure 13.5.1 SWT Timing Chart

Notes: • The timer switches to Run/Stop status synchronized with the 256 Hz signal falling edge after data is written to SWTRUN. When 0 is written to SWTRUN, the timer stops after counting an additional "+1." 1 is retained for SWTRUN reading until the timer actually stops.
Figure 13.5.2 shows the Run/Stop control timing chart.

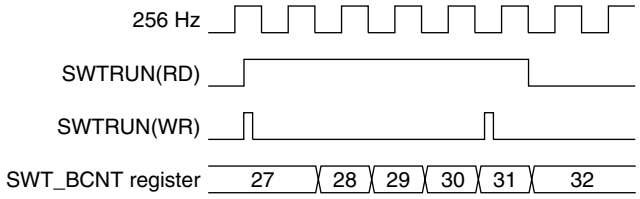


Figure 13.5.2 Run/Stop Control Timing Chart

• Executing the `slp` instruction while the timer is running (SWTRUN = 1) will destabilize the timer operation during restarting from SLEEP status. When switching to SLEEP status, stop the timer (SWTRUN = 0) before executing the `slp` instruction.

13.6 SWT Interrupts

The SWT module includes functions for generating the following three kinds of interrupts:
100 Hz, 10 Hz, and 1 Hz interrupts

The SWT module outputs a single interrupt signal shared by the above three interrupt causes to the interrupt controller (ITC). The interrupt flag in the SWT module should be read to identify the cause of interrupt that occurred.

100 Hz, 10 Hz, 1 Hz interrupts

The 100 Hz (approximate 100 Hz), 10 Hz (approximate 10 Hz), and 1 Hz signal falling edges set the corresponding interrupt flag in the SWT module to 1. At the same time, an interrupt request is sent to the ITC if the corresponding interrupt enable bit has been set to 1 (interrupt enabled). An interrupt is generated if the ITC and S1C17 Core interrupt conditions are satisfied.
If the interrupt enable bit is set to 0 (interrupt disabled, default), no interrupt request will be sent to the ITC.

Table 13.6.1 SWT Interrupt Flags and Interrupt Enable Bits

Cause of interrupt	Interrupt flag	Interrupt enable bit
100 Hz Interrupt	SIF100/SWT_IFLG register	SIE100/SWT_IMSK register
10 Hz Interrupt	SIF10/SWT_IFLG register	SIE10/SWT_IMSK register
1 Hz Interrupt	SIF1/SWT_IFLG register	SIE1/SWT_IMSK register

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

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

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

13.7 Control Register Details

Table 13.7.1 List of SWT Registers

Address	Register name		Function
0x5020	SWT_CTL	Stopwatch Timer Control Register	Resets and starts/stops the timer.
0x5021	SWT_BCNT	Stopwatch Timer BCD Counter Register	BCD counter data
0x5022	SWT_IMSK	Stopwatch Timer Interrupt Mask Register	Enables/disables interrupt.
0x5023	SWT_IFLG	Stopwatch Timer Interrupt Flag Register	Indicates/resets interrupt occurrence status.

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

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

Stopwatch Timer Control Register (SWT_CTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Stopwatch Timer Control Register (SWT_CTL)	0x5020 (8 bits)	D7–5	–	reserved	–	–	–	0 when being read.
		D4	SWTRST	Stopwatch timer reset	1 Reset 0 Ignored	0	W	
		D3–1	–	reserved	–	–	–	
		D0	SWTRUN	Stopwatch timer run/stop control	1 Run 0 Stop	0	R/W	

D[7:5] Reserved

D4 SWTRST: Stopwatch Timer Reset Bit

Resets the SWT module.

1 (W): Reset

0 (W): Ignored

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

Writing 1 to this bit resets the counter to 0x0. When reset in Run state, the timer restarts immediately after resetting. The reset data 0x0 is retained when in Stop state.

D[3:1] Reserved

D0 SWTRUN: Stopwatch Timer Run/Stop Control Bit

Controls the timer Run/Stop.

1 (R/W): Run

0 (R/W): Stop (default)

The timer starts counting when SWTRUN is written as 1 and stops when written as 0. The counter data is retained at Stop state until a reset or the next Run state.

Stopwatch Timer BCD Counter Register (SWT_BCNT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Stopwatch Timer BCD Counter Register (SWT_BCNT)	0x5021 (8 bits)	D7–4	BCD10[3:0]	1/10 sec. BCD counter value	0 to 9	0	R	
		D3–0	BCD100[3:0]	1/100 sec. BCD counter value	0 to 9	0	R	

D[7:4] BCD10[3:0]: 1/10 Sec. BCD Counter Value Bits

The 1/10-second counter BCD data can be read out. (Default: 0)

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

D[3:0] BCD100[3:0]: 1/100 Sec. BCD Counter Value Bits

The 1/100-second counter BCD data can be read out. (Default: 0)

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

Note: The correct counter value may not be read out (reading is unstable) if the register is read while counting is underway. Read the counter register twice in succession and treat the value as valid if the values read are identical.

Stopwatch Timer Interrupt Mask Register (SWT_IMSK)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Stopwatch Timer Interrupt Mask Register (SWT_IMSK)	0x5022 (8 bits)	D7–3	–	reserved	–				–	–	0 when being read.
		D2	SIE1	1 Hz interrupt enable	1	Enable	0	Disable	0	R/W	
		D1	SIE10	10 Hz interrupt enable	1	Enable	0	Disable	0	R/W	
		D0	SIE100	100 Hz interrupt enable	1	Enable	0	Disable	0	R/W	

This register enables or disables interrupt requests individually for the 100 Hz, 10 Hz, and 1 Hz signals. Setting SIE* to 1 enables SWT interrupts for the corresponding frequency signal falling edge, while setting to 0 disables interrupts.

D[7:3] Reserved

D2 SIE1: 1 Hz Interrupt Enable Bit

Enables or disables 1 Hz interrupts.
1 (R/W): Interrupt enabled
0 (R/W): Interrupt disabled (default)

D1 SIE10: 10 Hz Interrupt Enable Bit

Enables or disables 10 Hz interrupts.
1 (R/W): Interrupt enabled
0 (R/W): Interrupt disabled (default)

D0 SIE100: 100 Hz Interrupt Enable Bit

Enables or disables 100 Hz interrupts.
1 (R/W): Interrupt enabled
0 (R/W): Interrupt disabled (default)

Stopwatch Timer Interrupt Flag Register (SWT_IFLG)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Stopwatch Timer Interrupt Flag Register (SWT_IFLG)	0x5023 (8 bits)	D7–3	–	reserved	–				–	–	0 when being read.
		D2	SIF1	1 Hz interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D1	SIF10	10 Hz interrupt flag					0	R/W	
		D0	SIF100	100 Hz interrupt flag					0	R/W	

This register indicates the occurrence state of interrupt causes due to 100 Hz, 10 Hz, and 1 Hz signals. If an SWT interrupt occurs, identify the interrupt cause (frequency) by reading the interrupt flag in this register. SIF* is an SWT module interrupt flag that is set to 1 at the falling edge of the corresponding 100 Hz, 10 Hz, or 1 Hz interrupt. SIF* is reset by writing 1.

D[7:3] Reserved

D2 SIF1: 1 Hz Interrupt Flag Bit

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

D1 SIF10: 10 Hz Interrupt Flag Bit

Indicates whether the cause of 10 Hz interrupt has occurred or not.
1 (R): Cause of interrupt has occurred
0 (R): No cause of interrupt has occurred (default)
1 (W): Flag is reset
0 (W): Ignored

D0 SIF100: 100 Hz Interrupt Flag Bit

Indicates whether the cause of 100 Hz interrupt has occurred or not.

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

14 Watchdog Timer (WDT)

14.1 WDT Module Overview

The S1C17554/564 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.

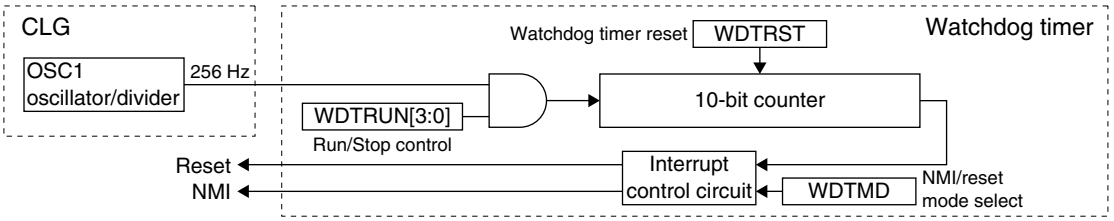


Figure 14.1.1 WDT Configuration

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. The CLG module generates this operation clock by dividing the OSC1 clock into 1/128, resulting in a frequency of 256 Hz when the OSC1 clock frequency is 32.768 kHz. The frequency described in this chapter will vary accordingly for other OSC1 clock frequencies. The CLG module does not include a 256 Hz clock output control bit. The 256 Hz clock is normally supplied to the WDT module when the OSC1 oscillation is on.
For detailed information on OSC1 oscillator control, see the "Clock Generator (CLG)" chapter.

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 WDT.
Since an NMI or reset may be generated immediately after running depending on the counter value, WDT should also be reset concurrently (before running WDT), 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. These are 8-bit registers.

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

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 fosc1 = 32.768 kHz). This resets the up-counter to 0 and starts counting with a new NMI/reset generation cycle.

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

Controls 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 S1C17554/564 includes a UART module with two asynchronous communication channels. 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:

- Transfer rate: 150 to 960 kbps (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.

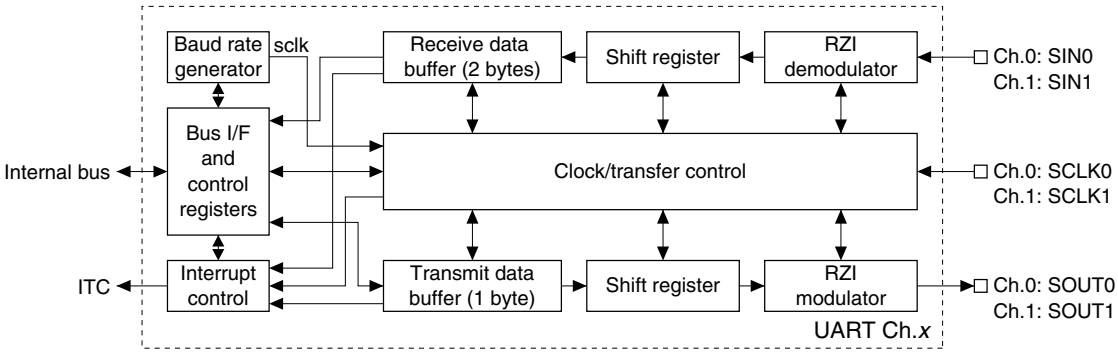


Figure 15.1.1 UART Configuration

Note: Two channels in the UART module have the same functions except for control register addresses. For this reason, the description in this chapter applies to both UART channels. The '*x*' in the register name indicates the channel number (0 or 1).
Example: UART_CTL*x* register
Ch.0: UART_CTL0 register
Ch.1: UART_CTL1 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) SIN1 (Ch.1)	I	2	UART Ch.*x* data input pin Inputs serial data sent from an external serial device.
SOUT0 (Ch.0) SOUT1 (Ch.1)	O	2	UART Ch.*x* data output pin Outputs serial data sent to an external serial device.
SCLK0 (Ch.0) SCLK1 (Ch.1)	I	2	UART Ch.*x* 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.

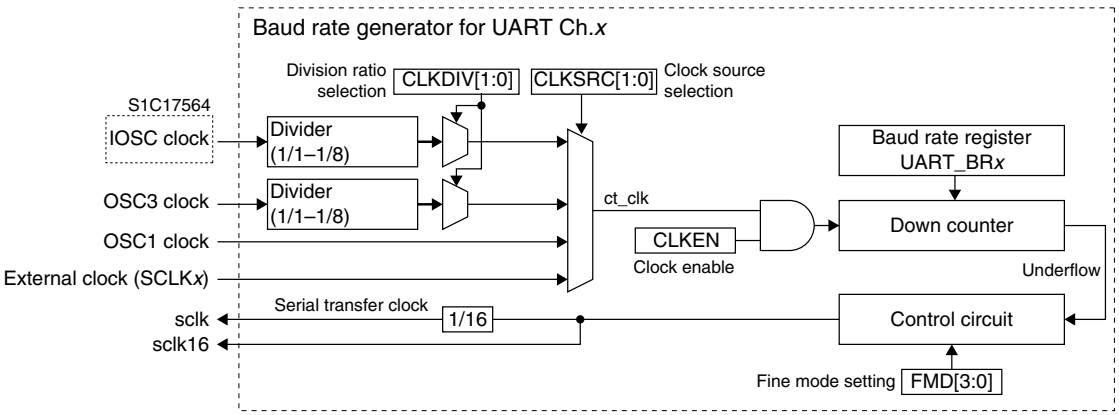


Figure 15.3.1 Baud Rate Generator

Clock source settings

The clock source can be selected from IOSC (S1C17564), OSC3, OSC1, or external clock using CLKSRC[1:0]/UART_CLK*x* register.

Table 15.3.1 Clock Source Selection

CLKSRC[1:0]	Clock source
0x3	External clock (SCLK*x*)
0x2	OSC3
0x1	OSC1
0x0	IOSC (S1C17564)

(Default: 0x0)

Note: When inputting the external clock via the SCLK*x* pin, the clock duty ratio must be 50%.

When IOSC or OSC3 is selected as the clock source, use CLKDIV[1:0]/UART_CLK*x* register to select the division ratio.

Table 15.3.2 IOSC/OSC3 Division Ratio Selection

CLKDIV[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 CLKEN/UART_CLK*x* register. The CLKEN default setting is 0, which disables the clock supply. Setting CLKEN 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.

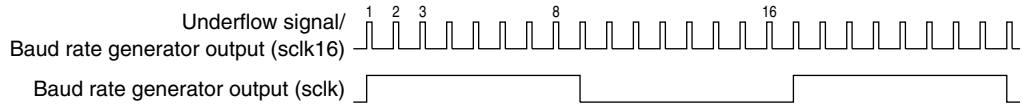


Figure 15.3.2 Counter Underflow and Clock Generated

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 960 kbps (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.

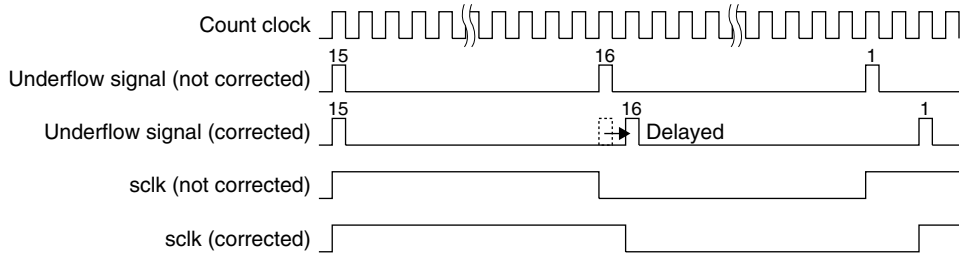


Figure 15.3.3 Delay Cycle Insertion in Fine Mode

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.

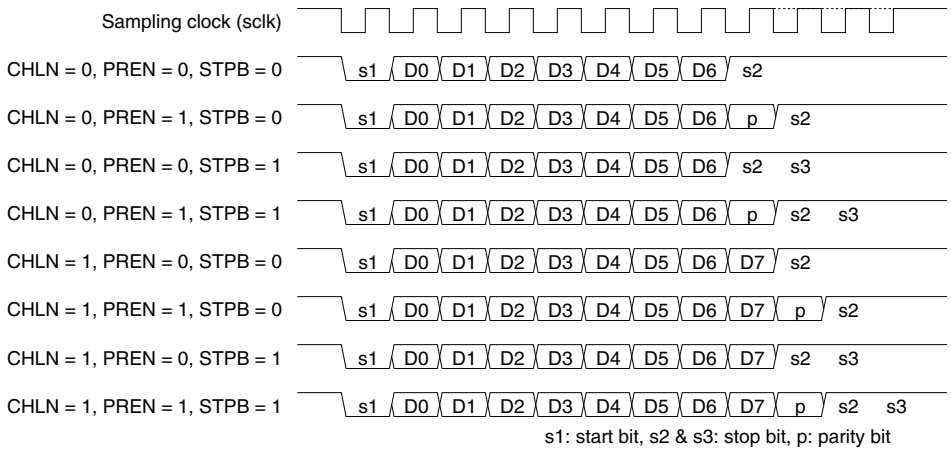


Figure 15.4.1 Transfer Data Format

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.

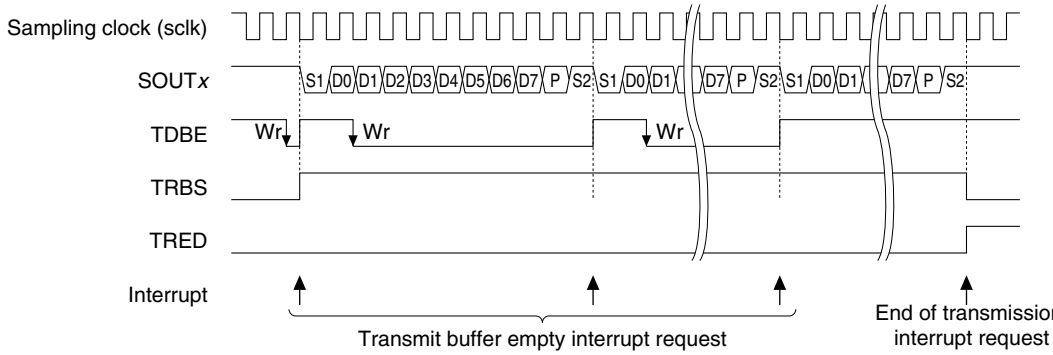
Note: Do not set RXEN to 0 while the UART is sending or receiving data.

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.

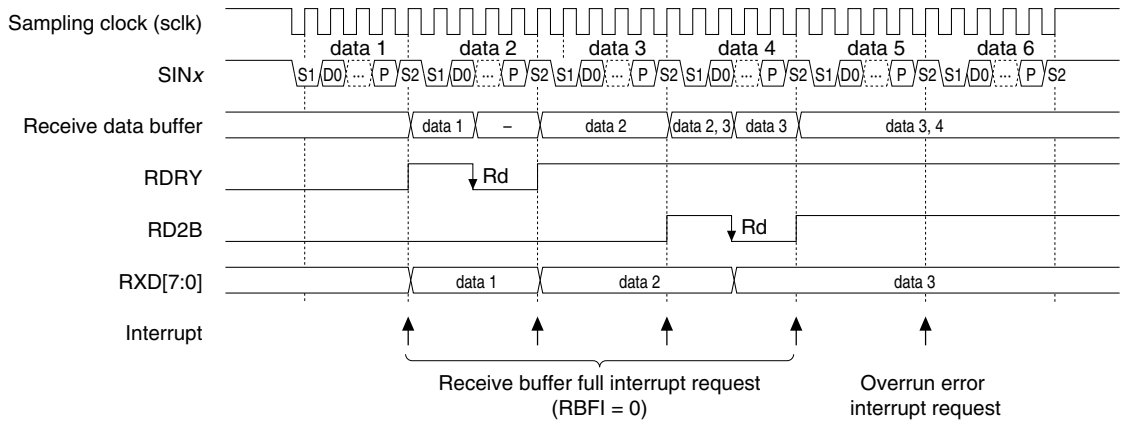
(1) RDRY = 0, RD2B = 0
The receive data buffer contents need not be read, since no data has been received.

(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

After a data transfer is completed (both transmission and reception), write 0 to RXEN to disable data transfers.

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 an overrun error will occur. 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

Each UART channel outputs one interrupt signal shared by the four above interrupt causes to the interrupt controller (ITC). Inspect the status flag and error flag to determine the interrupt cause occurred.

Transmit buffer empty interrupt

To use this interrupt, set TIEN/UART_CTL*x* register to 1. If TIEN is set to 1 while TDBE/UART_ST*x* register is 1 (transmit data buffer empty) or if TDBE is set to 1 (when the transmit data buffer becomes empty by loading the transmit data written to it to the shift register) while TIEN is 1, an interrupt request is sent to the ITC. An interrupt occurs if other interrupt conditions are met.

If TIEN is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

You can inspect the TDBE flag in the UART interrupt handler routine to determine whether the UART interrupt is attributable to a transmit buffer empty. If TDBE is 1, the next transmit data can be written to the transmit data buffer by the interrupt handler routine.

End of transmission interrupt

To use this interrupt, set TEIEN/UART_CTL*x* register to 1. If TEIEN is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

When the TRBS flag is reset to 0, the UART sets TRED/UART_ST*x* register to 1, indicating that the transmit operation has completed. If end of transmission interrupts are enabled (TEIEN = 1), an interrupt request is sent simultaneously to the ITC.

An interrupt occurs if other interrupt conditions are met. You can inspect the TRED flag in the UART interrupt handler routine to determine whether the UART interrupt is attributable to an end of transmission. If TRED is 1, the transmission processing can be terminated.

Receive buffer full interrupt

To use this interrupt, set RIEN/UART_CTL*x* register to 1. If RIEN is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

If the specified volume of received data is loaded into the receive data buffer when a receive buffer full interrupt is enabled (RIEN = 1), the UART outputs an interrupt request to the ITC. If RBFI/UART_CTL*x* register is 0, an interrupt request is output as soon as one received data is loaded into the receive data buffer (when RDRY/UART_ST*x* register is set to 1). If RBFI is 1, an interrupt request is output as soon as two received data are loaded into the receive data buffer (when RD2B/UART_ST*x* register is set to 1).

An interrupt occurs if other interrupt conditions are met. You can inspect the RDRY and RD2B flags in the UART interrupt handler routine to determine whether the UART interrupt is attributable to a receive buffer full. If RDRY or RD2B is 1, the received data can be read from the receive data buffer by the interrupt handler routine.

Receive error interrupt

To use this interrupt, set REIEN/UART_CTL*x* register to 1. If REIEN is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

The UART sets an error flag, PER, FER, or OER/UART_ST*x* register to 1 if a parity error, framing error, or overrun error is detected when receiving data. If receive error interrupts are enabled (REIEN = 1), an interrupt request is sent simultaneously to the ITC.

If other interrupt conditions are satisfied, an interrupt occurs. You can inspect the PER, FER, and OER flags in the UART interrupt handler routine to determine whether the UART interrupt was caused by a receive error. If any of the error flags has the value 1, the interrupt handler routine will proceed with error recovery.

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

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.

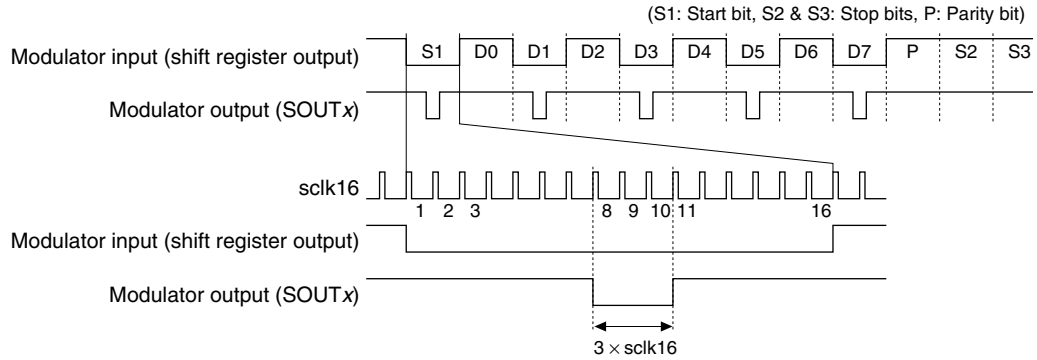


Figure 15.8.1 Transmission Signal Waveform

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).

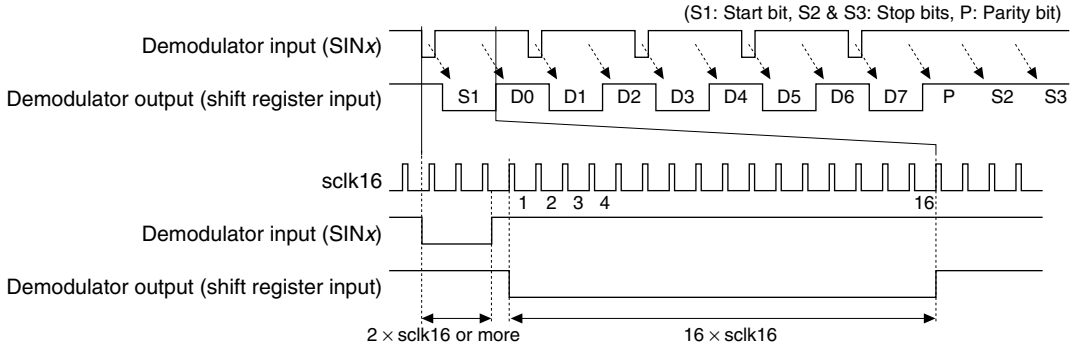


Figure 15.8.2 Receive Signal Waveform

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.

Address	Register name		Function
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.
0x4120	UART_ST1	UART Ch.1 Status Register	Indicates transfer, buffer and error statuses.
0x4121	UART_TXD1	UART Ch.1 Transmit Data Register	Transmit data
0x4122	UART_RXD1	UART Ch.1 Receive Data Register	Receive data
0x4123	UART_MOD1	UART Ch.1 Mode Register	Sets transfer data format.
0x4124	UART_CTL1	UART Ch.1 Control Register	Controls data transfer.
0x4125	UART_EXP1	UART Ch.1 Expansion Register	Sets IrDA mode.
0x4126	UART_BR1	UART Ch.1 Baud Rate Register	Sets baud rate.
0x4127	UART_FMD1	UART Ch.1 Fine Mode Register	Sets fine mode.
0x506c	UART_CLK0	UART Ch.0 Clock Control Register	Selects the baud rate generator clock.
0x506d	UART_CLK1	UART Ch.1 Clock Control Register	Selects the baud rate generator clock.

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

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 (CLKDIV[1:0], CLKSRC[1:0], CLKEN)

UART Ch.*x* Status Registers (UART_ST*x*)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
UART Ch.*x* Status Register (UART_ST*x*)	**0x4100** **0x4120** (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 when data is received in the shift register when the receive data buffer is already full and additional data is sent. The receive data buffer is not overwritten 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 Registers (UART_TXD*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
UART Ch.*x* Transmit Data Register (UART_TXD*x*)	**0x4101 0x4121** (8 bits)	D7–0	**TXD[7:0]**	Transmit data TXD7(6) = MSB TXD0 = LSB	0x0 to 0xff (0x7f)	0x0	R/W	

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 Registers (UART_RXD*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
UART Ch.*x* Receive Data Register (UART_RXD*x*)	**0x4102 0x4122** (8 bits)	D7–0	**RXD[7:0]**	Receive data in the receive data buffer RXD7(6) = MSB RXD0 = LSB	0x0 to 0xff (0x7f)	0x0	R	Older data in the buffer is read out first.

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 Registers (UART_MOD*x*)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
UART Ch.*x* Mode Register (UART_MOD*x*)	**0x4103 0x4123** (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 Registers (UART_CTL*x*)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
UART Ch.*x* Control Register (UART_CTL*x*)	0x4104 0x4124 (8 bits)	D7	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. 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.
The transfer conditions must be set while RXEN is 0.
Disabling transfers by writing 0 to RXEN also clears transmit data buffers.

UART Ch.*x* Expansion Registers (UART_EXP*x*)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
UART Ch.*x* Expansion Register (UART_EXP*x*)	0x4105	D7–1	–	reserved	–				–	–	0 when being read.
	0x4125 (8 bits)	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 Registers (UART_BR*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
UART Ch.*x* Baud Rate Register (UART_BR*x*)	0x4106 0x4126 (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 Registers (UART_FMD*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
UART Ch.*x*	**0x4107**	D7–4	–	reserved	–	–	–	0 when being read.
Fine Mode Register (UART_FMD*x*)	**0x4127** (8 bits)	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.

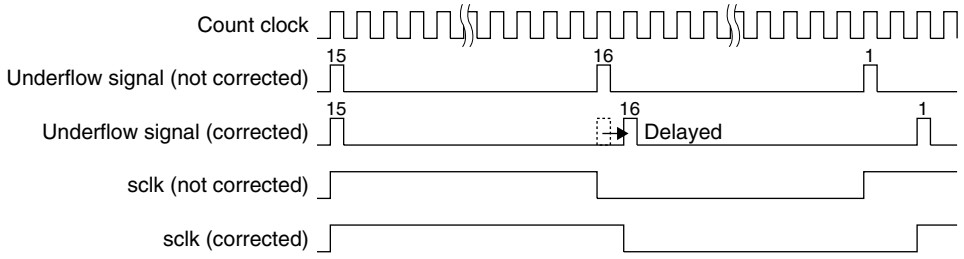


Figure 15.9.1 Delay Cycle Insertion in Fine Mode

UART Ch.*x* Clock Control Registers (UART_CLK*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
UART Ch.*x* Clock Control Register (UART_CLK*x*)	0x506c 0x506d (8 bits)	D7–6	–	reserved	–	–	–	0 when being read.
		D5–4	CLKDIV [1:0]	Clock division ratio select	CLKDIV[1:0] / Division ratio 0x3 / 1/8 0x2 / 1/4 0x1 / 1/2 0x0 / 1/1	0x0	R/W	When the clock source is IOSC or OSC3
		D3–2	CLKSRC [1:0]	Clock source select	CLKSRC[1:0] / Clock source 0x3 / External clock 0x2 / OSC3 0x1 / OSC1 0x0 / IOSC*	0x0	R/W	* S1C17564 only
		D1	–	reserved	–	–	–	0 when being read.
		D0	CLKEN	Count clock enable	1 Enable 0 Disable	0	R/W	

D[7:6] Reserved

D[5:4] CLKDIV[1:0]: Clock Division Ratio Select Bits

Selects the division ratio for generating the count clock of the baud rate generator when IOSC (S1C17564) or OSC3 is used as the clock source.

Table 15.9.3 IOSC/OSC3 Division Ratio Selection

CLKDIV[1:0]	Division ratio
0x3	1/8
0x2	1/4
0x1	1/2
0x0	1/1

(Default: 0x0)

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

Selects the count clock source for the baud rate generator.

Table 15.9.4 Clock Source Selection

CLKSRC[1:0]	Clock source
0x3	External clock (EXCL*x*)
0x2	OSC3
0x1	OSC1
0x0	IOSC (S1C17564)

(Default: 0x0)

D1 Reserved

D0 CLKEN: 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 CLKEN default setting is 0, which disables the clock supply. Setting CLKEN to 1 sends the clock selected to the counter.

16 SPI

16.1 SPI Module Overview

The S1C17554/564 includes a synchronized serial interface module (SPI) with three communication channels. The following shows the main features of the SPI:

- 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.

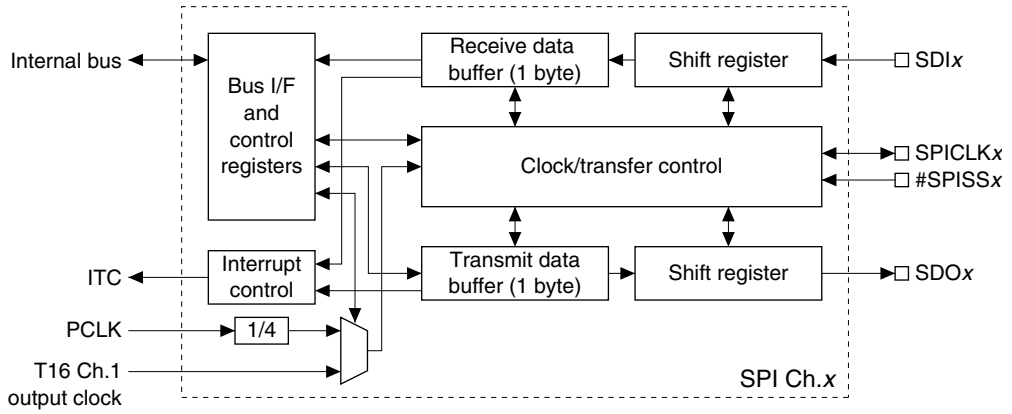


Figure 16.1.1 SPI Module Configuration (1 channel)

Note: Three channels in the SPI module have the same functions except for control register addresses. For this reason, the description in this chapter applies to all SPI channels. The '*x*' in the register name indicates the channel number (0 to 2).
Example: SPI_CTL*x* register
Ch.0: SPI_CTL0 register
Ch.1: SPI_CTL1 register
Ch.2: SPI_CTL2 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) SDI1 (Ch.1) SDI2 (Ch.2)	I	3	SPI Ch.*x* data input pin Inputs serial data from SPI bus.
SDO0 (Ch.0) SDO1 (Ch.1) SDO2 (Ch.2)	O	3	SPI Ch.*x* data output pin Outputs serial data to SPI bus.

Pin name	I/O	Qty	Function
SPICLK0 (Ch.0) SPICLK1 (Ch.1) SPICLK2 (Ch.2)	I/O	3	SPI Ch.*x* 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) #SPISS1 (Ch.1) #SPISS2 (Ch.2)	I	3	SPI Ch.*x* 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 16-bit timer (T16) Ch.1 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 T16 Ch.1 output clock or PCLK/4 clock is used.
Setting MCLK to 1 selects the T16 Ch.1 output clock; setting to 0 selects the PCLK/4 clock.
Using the T16 Ch.1 output clock enables programmable transfer rates. For more information on T16 control, see the "16-bit Timers (T16)" chapter.

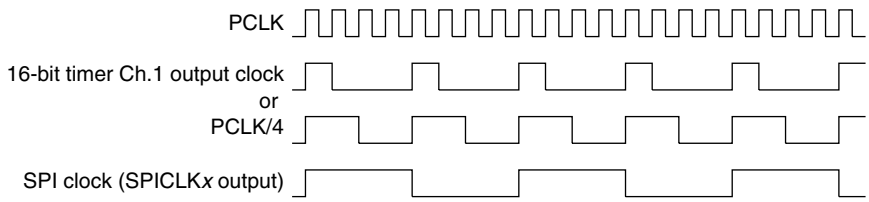


Figure 16.3.1 Master Mode SPI Clock

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.

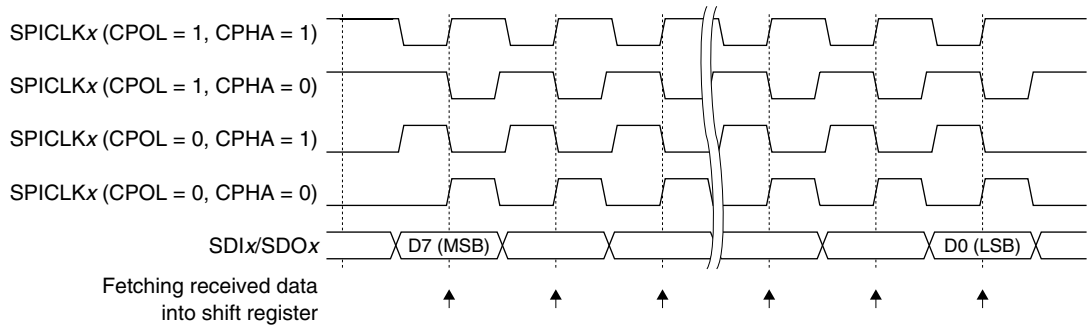


Figure 16.4.1 Clock and Data Transfer Timing

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.

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.

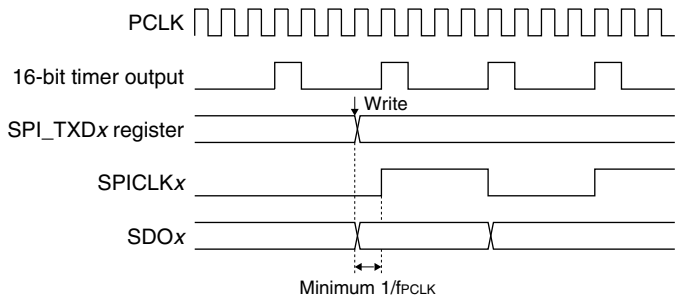


Figure 16.5.1 SDO*x* and SPICLK*x* Change Timings when CPHA = 0

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.

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.

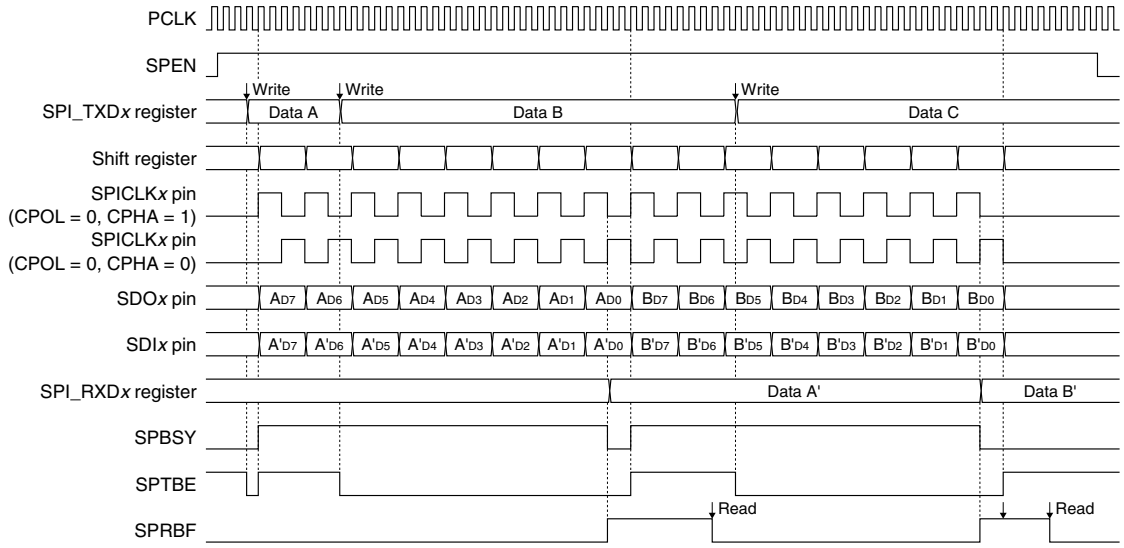


Figure 16.5.2 Data Transmission/Receiving Timing Chart (MSB first)

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.
0x4380	SPI_ST1	SPI Ch.1 Status Register	Indicates transfer and buffer statuses.
0x4382	SPI_TXD1	SPI Ch.1 Transmit Data Register	Transmit data
0x4384	SPI_RXD1	SPI Ch.1 Receive Data Register	Receive data
0x4386	SPI_CTL1	SPI Ch.1 Control Register	Sets the SPI mode and enables data transfer.
0x43a0	SPI_ST2	SPI Ch.2 Status Register	Indicates transfer and buffer statuses.
0x43a2	SPI_TXD2	SPI Ch.2 Transmit Data Register	Transmit data
0x43a4	SPI_RXD2	SPI Ch.2 Receive Data Register	Receive data
0x43a6	SPI_CTL2	SPI Ch.2 Control Register	Sets the SPI mode and enables data transfer.

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

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

SPI Ch.*x* Status Registers (SPI_ST*x*)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
SPI Ch.*x* Status Register (SPI_ST*x*)	0x4320 0x4380 0x43a0 (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 Registers (SPI_TXD*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SPI Ch.*x*	**0x4322**	D15–8	**–**	reserved	–	–	–	0 when being read.
Transmit Data	**0x4382**	D7–0	**SPTDB[7:0]**	SPI transmit data buffer	0x0 to 0xff	0x0	R/W	
Register	**0x43a2**			SPTDB7 = MSB				
(SPI_TXD*x*)	(16 bits)			SPTDB0 = LSB				

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 Registers (SPI_RXD*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SPI Ch.*x*	**0x4324**	D15–8	**–**	reserved	–	–	–	0 when being read.
Receive Data	**0x4384**	D7–0	**SPRDB[7:0]**	SPI receive data buffer	0x0 to 0xff	0x0	R	
Register	**0x43a4**			SPRDB7 = MSB				
(SPI_RXD*x*)	(16 bits)			SPRDB0 = LSB				

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 Registers (SPI_CTL*x*)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
SPI Ch.*x* Control Register (SPI_CTL*x*)	**0x4326** **0x4386** **0x43a6** (16 bits)	D15–10	**–**	reserved	–				–	–	0 when being read.
		D9	**MCLK**	SPI clock source select	1	T16 Ch.1	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): 16-bit timer Ch.1
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.)

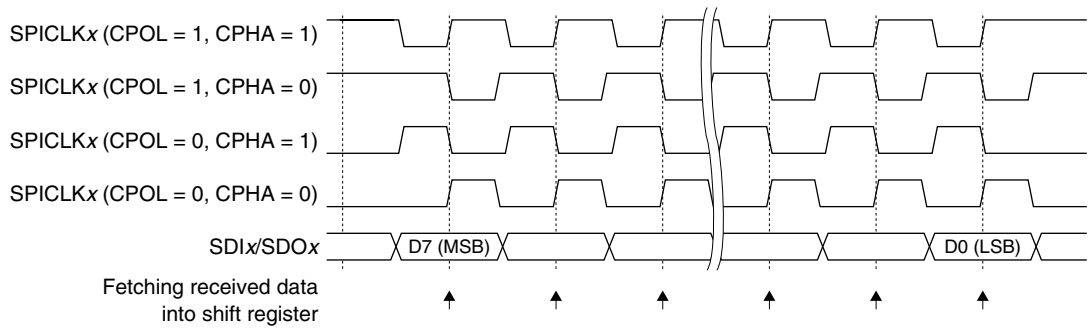


Figure 16.7.1 Clock and Data Transfer Timing

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 I²C Master (I2CM)

17.1 I2CM Module Overview

The S1C17554/564 includes an I²C master (I2CM) module that supports two-wire communications. The I2CM module operates as an I²C bus master device and can communicate with I²C-compliant slave devices.
The following shows the main features of I2CM:

- Operates as an I²C bus master device (as single master only).
- Supports standard (100 kbps) and fast (400 kbps) modes.
- Supports 8-bit data length only (MSB first).
- 7-bit addressing mode (10-bit addressing is possible by software control.)
- Includes one-byte receive data buffer and one-byte transmit data buffer.
- Can generate start, repeated start, and stop conditions.
- Supports half-duplex communications.
- Supports clock stretch function.
- Includes a noise filter function to help improve the reliability of data transfers.
- Can generate transmit buffer empty and receive buffer full interrupts.

Figure 17.1.1 shows the I2CM configuration.

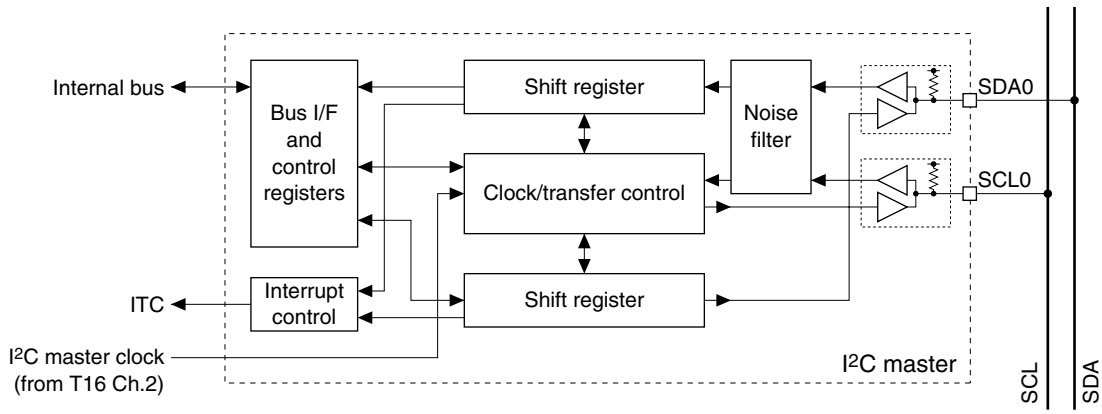


Figure 17.1.1 I2CM Module Configuration

17.2 I2CM Input/Output Pins

Table 17.2.1 lists the I2CM pins.

Table 17.2.1 List of I2CM Pins

Pin name	I/O	Qty	Function
SDA0	I/O	1	I2CM data input/output pin (see Note below) Inputs serial data from the I²C bus. Also outputs serial data to the I²C bus.
SCL0	I/O	1	I2CM clock input/output pin (see Note below) Inputs SCL line status. Also outputs a serial clock.

The I2CM input/output pins (SDA0, SCL0) are shared with I/O ports and are initially set as general purpose I/O port pins. The pin functions must be switched using the port function select bits to use the general purpose I/O port pins as I2CM input/output pins.
For detailed information on pin function switching, see the “I/O Ports (P)” chapter.

Note: The pins go to high impedance status when the port function is switched.
The SCL0 and SDA0 pins do not output a high level, so these lines should be pulled up to HV_{DD} with an external pull-up resistor. Be sure to avoid pulling these pins up to a voltage that exceeds the HV_{DD} level.

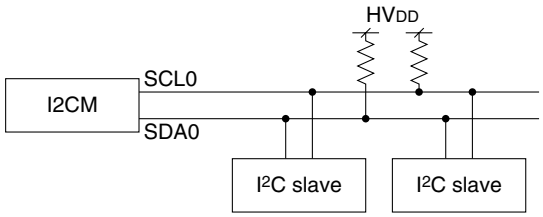


Figure 17.2.1 I²C Connection Example

17.3 Synchronization Clock

The I2CM module uses the internal clock (I2CM clock) output by the 16-bit timer (T16) Ch.2 as the synchronization clock. This clock is output from the SCL0 pin to the slave device while also driving the shift register. The clock should be programmed to output a signal matching the transfer rate from T16 Ch.2. For more information on T16 control, see the "16-bit Timers (T16)" chapter.

When the I2CM module is used to communicate with a slave device that performs clock stretching, the maximum transfer rate is limited to 50 kbps in standard mode or 200 kbps in fast mode.

The I2CM module does not function as a slave device. The SCL0 input pin is used to check the I²C bus SCL signal status. It is not used for synchronization clock input.

17.4 Settings Before Data Transfer

The I2CM module includes an optional noise filter function that can be selected via the application program.

Noise filter function

The I2CM module includes a function for filtering noise from the SDA0 and SCL0 pin input signals. This function is enabled by setting NSERM/I2CM_CTL register to 1. Note that using this function requires setting the I2CM clock (T16 Ch.2 output clock) frequency to 1/6 or less of PCLK.

17.5 Data Transfer Control

Make the following settings before starting data transfers.
(1) Configure T16 Ch.2 to output the I2CM clock. (See the T16 module chapter.)
(2) Select the option function. (See Section 17.4.)
(3) Set the interrupt conditions to use I2CM interrupts. (See Section 17.6.)

Note: Make sure the I2CM module is halted (I2CMEN/I2CM_EN register = 0) before changing the above settings.

Enabling data transfers

Set I2CMEN/I2CM_EN register to 1 to enable I2CM operations. This enables I2CM transfers and clock input/output.

Note: Do not set I2CMEN to 0 when the I2CM module is transferring data.

Starting Data transfer

To start data transfers, the I²C master (this module) must generate a start condition. The slave address is then sent to establish communications.

(1) Generating start condition

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

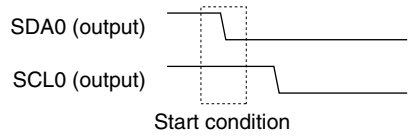


Figure 17.5.1 Start Condition

The start condition is generated by setting STRT/I2CM_CTL register to 1.
STRT is automatically reset to 0 once the start condition is generated. The I²C bus is busy from this point on.

(2) Slave address transmission

Once the start condition has been generated, the I²C master (this module) sends a bit indicating the slave address and transfer direction for communications. I²C slave addresses are either 7-bit or 10-bit. This module uses an 8-bit transfer data register to send the slave address and transfer direction bit, enabling single transfers in 7-bit address mode. In 10-bit mode, data is sent twice or three times under software control. Figure 17.5.2 shows the configuration of the address data.

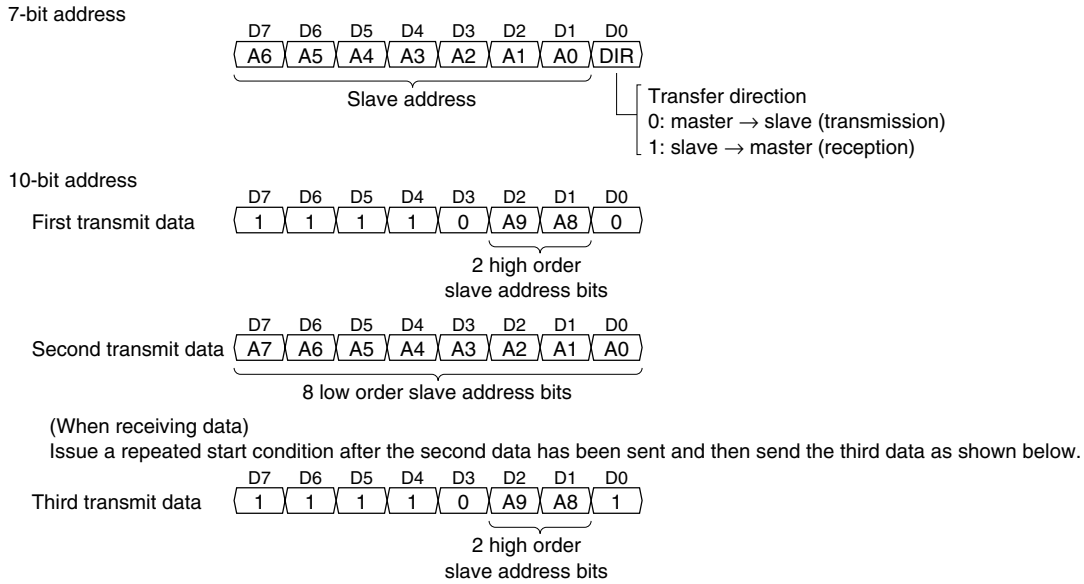


Figure 17.5.2 Transmit Data Specifying Slave Address and Transfer Direction

The transfer direction bit indicates the data transfer direction after the slave address has been sent. This is set to 0 when sending data from the master to the slave and to 1 when receiving data from the slave. To send a slave address, set the address with the transfer direction bit to RTDT[7:0]/I2CM_DAT register. At the same time, set TXE/I2CM_DAT register transmitting the address to 1.
After the slave address has been output, data can be sent and received as many times as required. Data must be sent or received according to the transfer direction set together with the slave address.

Data transmission control

The procedure for transmitting data is described below. Data transmission is performed by the same procedure as for slave address transmission.
To send byte data, set the transmit data to RTDT[7:0] and set TXE to 1 to transmit 1 byte.
When TXE is set to 1, the I2CM module begins data transmission in sync with the clock. If the previous data is currently being transmitted, data transmission starts after this has been completed. The I2CM module first transfers the data written to the shift register, then starts outputting the clock from the SCL0 pin. TXE is reset to 0 at this point and a cause of interrupt occurs, enabling the subsequent transmission data and TXE to be set.

The data bits in the shift register are shifted in sequence at the clock falling edge and output via the SDA0 pin with the MSB leading. The I2CM module outputs 9 clocks with each data transmission. In the 9th clock cycle, the I2CM module sets the SDA line into high impedance to receive an ACK or NAK sent from the slave device. The slave device returns ACK (0) to the master if the data is received. If the data is not received, the SDA line is not pulled down, which the I2CM module interprets to mean a NAK (1) (transmission failed).

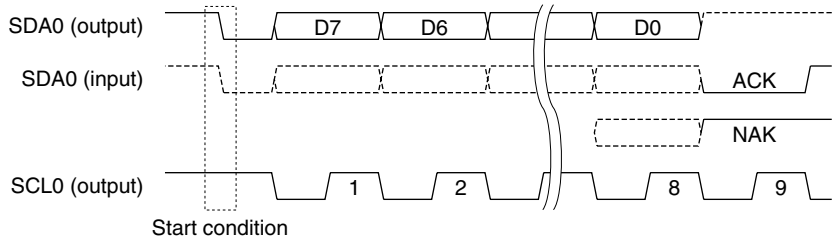


Figure 17.5.3 ACK and NAK

The I2CM module includes two status bits for transmission control: TBUSY/I2CM_CTL register and RTACK/I2CM_DAT register.
The TBUSY flag indicates the data transmission status. This flag becomes 1 when transmission starts (including slave address transmission) and reverts to 0 once data transmission ends. Inspect the flag to check whether the I2CM module is currently transmitting or at standby.
The RTACK bit indicates whether or not the slave device returned an ACK for the previous transmission. RTACK is 0 if an ACK was returned and 1 if ACK was not returned.

Data reception control

The procedure for receiving data is described below. When receiving data, the slave address must be sent with the transfer direction bit set to 1.
To receive data, set RXE/I2CM_DAT register to 1 for receiving 1 byte. When TXE/I2CM_DAT register is set to 1 for sending the slave address, RXE can also be set to 1 at the same time. If both TXE and RXE are set to 1, TXE takes priority.
When RXE is set to 1, allowing receiving to start, the I2CM module starts outputting the clock from the SCL0 pin with the SDA line at high impedance. The data is shifted into the shift register from the MSB first in sync with the clock.
RXE is reset to 0 when D7 is loaded.
The received data is loaded to RTDT[7:0] once the 8-bit data has been received in the shift register.

The I2CM module includes two status bits for receive control: RBRDY/I2CM_DAT register and RBUSY/I2CM_CTL register.
The RBRDY flag indicates the received data status. This flag becomes 1 when the data received in the shift register is loaded to RTDT[7:0] and reverts to 0 when the received data is read out from RTDT[7:0]. Interrupts can also be generated once the flag value becomes 1.
The RBUSY flag indicates the receiving operation status. This flag is 1 when receiving starts and reverts to 0 when the data is received. Inspect the flag to determine whether the I2CM module is currently receiving or in standby.

The I2CM module outputs 9 clocks with each data reception. In the 9th clock cycle, an ACK or NAK is sent to the slave via the SDA0 pin. The bit state sent can be set in RTACK/I2CM_DAT register. To send ACK, set RTACK to 0. To send NAK, set RTACK to 1.

End of data transfers (Generating stop condition)

To end data transfers after all data has been transferred, the I²C master (this module) must generate a stop condition. The stop condition applies when the SCL line is maintained at High and the SDA line is pulled up from Low to High.

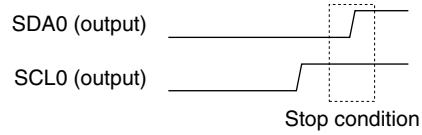


Figure 17.5.4 Stop Condition

The stop condition is generated by setting STP/I2CM_CTL register to 1.
When STP is set to 1, the I2CM module pulls up the I²C bus SDA line from Low to High with the SCL line maintained at High to generates a stop condition. The I²C bus subsequently switches to free state.
Before STP can be set to 1, confirm that TBUSY or RBUSY is reset to 0 from 1 (this indicates that the I2CM module has finished data transmit/receive operation) and then make the wait time longer than 1/4 of the I²C clock cycle set. When generating a stop condition to the slave device with a clock stretch function, STP must be set to 1 after data transfer (including ACK/NAK transfer) has finished and the time for the slave device to finish clock stretching has elapsed. STP is reset to 0 when the stop condition is generated.

Continuing data transfer (Generating repeated start condition)

To make it possible to continue with a different data transfer after data transfer completion, the I²C master (this module) can generate a repeated start condition.

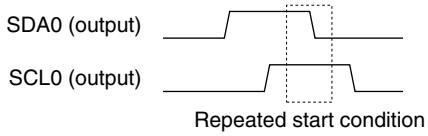


Figure 17.5.5 Repeated Start Condition

The repeated start condition is generated by setting STRT/I2CM_CTL register to 1 when the I²C bus is busy.
STRT is automatically reset to 0 once the repeated start condition is generated. Slave address transmission is subsequently possible with the I²C bus remaining in the busy state.

Disabling data transfer

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

Timing chart

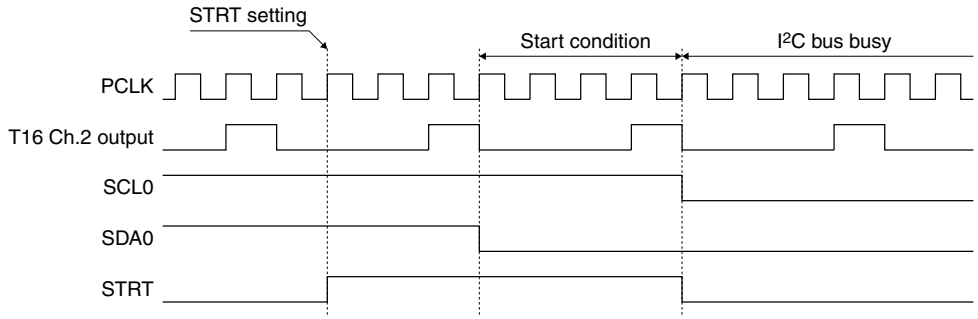


Figure 17.5.6 Start Condition Generation

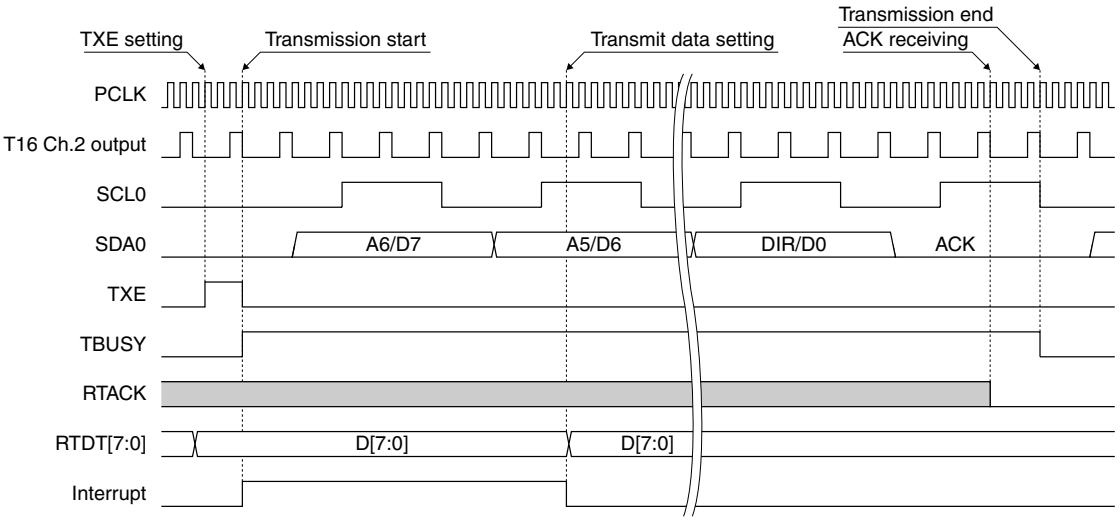


Figure 17.5.7 Slave Address Transmission/Data Transmission

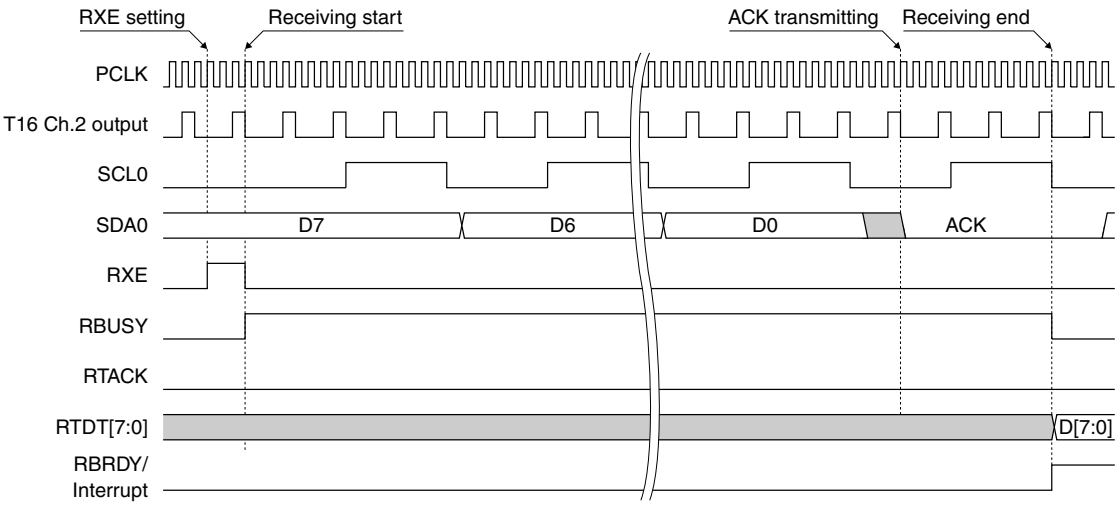


Figure 17.5.8 Data Receiving

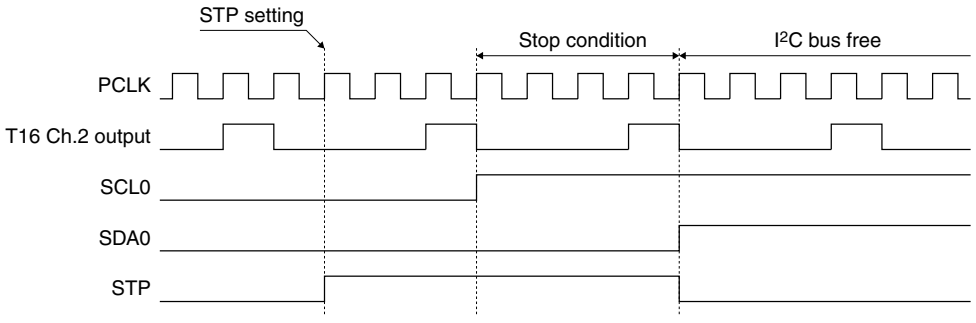


Figure 17.5.9 Stop Condition Generation

17.6 I2CM Interrupts

The I2CM module includes a function for generating the following two different types of interrupts.

- Transmit buffer empty interrupt
- Receive buffer full interrupt

The I2CM module outputs one interrupt signal shared by the two above interrupt causes to the interrupt controller (ITC).

Transmit buffer empty interrupt

To use this interrupt, set TINTE/I2CM_ICTL register to 1. If TINTE is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.
If transmit buffer empty interrupts are enabled (TINTE = 1), an interrupt request is output to the ITC as soon as the transmit data set in RTDT[7:0]/I2CM_DAT register is transferred to the shift register.
An interrupt occurs if other interrupt conditions are satisfied.

Checking whether a transmit buffer empty interrupt has occurred or not

A transmit buffer empty interrupt has occurred if TXE/I2CM_DAT register is read as 0 in the procedure shown below.

(1) Set TINTE/I2CM_ICTL register to 1.
(2) Write data to RTDT[7:0]/I2CM_DAT register.
(3) Set TXE/I2CM_DAT register to 1. (This can be performed simultaneously with Step 2 above.)
(4) An I2CM interrupt occurs.
(5) Read TXE/I2CM_DAT register.

Clearing the cause of transmit buffer empty interrupt

Write data to RTDT[7:0]/I2CM_DAT register.

Notes: • Data will not be sent if TXE/I2CM_DAT register is set to 0.

• If RTDT[7:0] contains data received from the I²C bus, it will be overwritten.

Receive buffer full interrupt

To use this interrupt, set RINTE/I2CM_ICTL register to 1. If RINTE is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.
If receive buffer full interrupts are enabled (RINTE = 1), an interrupt request is output to the ITC as soon as the data received in the shift register is loaded to RTDT[7:0].
An interrupt occurs if other interrupt conditions are met.

Checking whether a receive buffer full interrupt has occurred or not

A receive buffer full interrupt has occurred if RBRDY/I2CM_DAT register is read as 1 in the procedure shown below.

(1) Set RINTE/I2CM_ICTL register to 1.
(2) An I2CM interrupt occurs.
(3) Read RBRDY/I2CM_DAT register.

Clearing the cause of receive buffer full interrupt

Read data from RTDT[7:0]/I2CM_DAT register.

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

17.7 Control Register Details

Table 17.7.1 List of I2CM Registers

Address	Register name		Function
0x4340	I2CM_EN	I²C Master Enable Register	Enables the I²C master module.
0x4342	I2CM_CTL	I²C Master Control Register	Controls the I²C master operation and indicates transfer status.
0x4344	I2CM_DAT	I²C Master Data Register	Transmit/receive data
0x4346	I2CM_ICTL	I²C Master Interrupt Control Register	Controls the I²C master interrupt.

The I2CM module registers are described in detail below. These are 16-bit registers.

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

I²C Master Enable Register (I2CM_EN)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
I²C Master Enable Register (I2CM_EN)	0x4340 (16 bits)	D15–1	–	reserved	–				–	–	0 when being read.
		D0	I2CMEN	I²C master enable	1	Enable	0	Disable	0	R/W	

D[15:1] Reserved

D0 I2CMEN: I²C Master Enable Bit

Enables or disables I2CM module operation.

1 (R/W): Enabled

0 (R/W): Disabled (default)

Setting I2CMEN to 1 starts the I2CM module operation, enabling data transfer. Setting I2CMEN to 0 stops the I2CM module operation.

I²C Master Control Register (I2CM_CTL)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
I²C Master Control Register (I2CM_CTL)	0x4342 (16 bits)	D15–10	–	reserved	–				–	–	0 when being read.
		D9	RBUSY	Receive busy flag	1	Busy	0	Idle	0	R	
		D8	TBUSY	Transmit busy flag	1	Busy	0	Idle	0	R	
		D7–5	–	reserved	–				–	–	0 when being read.
		D4	NSERM	Noise remove on/off	1	On	0	Off	0	R/W	
		D3–2	–	reserved	–				–	–	0 when being read.
		D1	STP	Stop control	1	Stop	0	Ignored	0	R/W	
		D0	STRT	Start control	1	Start	0	Ignored	0	R/W	

D[15:10] Reserved

D9 RBUSY: Receive Busy Flag Bit

Indicates the I2CM receiving status.

1 (R): Operating

0 (R): Standby (default)

RBUSY is set to 1 when the I2CM starts data receiving and is maintained at 1 while receiving is underway. It is cleared to 0 once reception is completed.

D8 TBUSY: Transmit Busy Flag Bit

Indicates the I2CM transmission status.

1 (R): Operating

0 (R): Standby (default)

TBUSY is set to 1 when the I2CM starts data transmission and is maintained at 1 while transmission is underway. It is cleared to 0 once transmission is completed.

D[7:5] Reserved

D4 NSERM: Noise Remove On/Off Bit

Turns the noise filter function on or off.

1 (R/W): On

0 (R/W): Off (default)

The I2CM module includes a function for filtering noise from the SDA0 and SCL0 pin input signals. This function is enabled by setting NSERM to 1. Note that using this function requires setting the I2CM clock (T16 Ch.2 output clock) frequency to 1/6 or less of PCLK.

D[3:2] Reserved

D1 STP: Stop Control Bit

Generates the stop condition.

1 (R/W): Stop condition generated

0 (R/W): Ineffective (default)

By setting STP to 1, the I2CM module generates the stop condition by pulling up the I²C bus SDA line from Low to High with the SCL line maintaining at High. The I²C bus subsequently becomes free. Note that the stop condition will be generated only if STP is 1 and TXE/I2CM_DAT register, RXE/I2CM_DAT register, and STRT are set to 0 when data transfer is completed (including ACK transfer). STP is automatically reset to 0 if the stop condition is generated.

D0 STRT: Start Control Bit

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

By setting STRT to 1, the I2CM module generates the start condition by pulling down the I²C bus SDA line to Low with SCL line maintaining at High.
The repeated start condition can be generated by setting STRT to 1 when the I²C bus is busy.
STRT is automatically reset to 0 once the start condition or repeated start condition is generated. The I²C bus subsequently becomes busy.

I²C Master Data Register (I2CM_DAT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
I²C Master Data Register (I2CM_DAT)	0x4344 (16 bits)	D15–12	–	reserved	–	–	–	0 when being read.
		D11	**RBRDY**	Receive buffer ready flag	1 Ready 0 Empty	0	R	
		D10	**RXE**	Receive execution	1 Receive 0 Ignored	0	R/W	
		D9	**TXE**	Transmit execution	1 Transmit 0 Ignored	0	R/W	
		D8	**RTACK**	Receive/transmit ACK	1 Error 0 ACK	0	R/W	
		D7–0	**RTDT[7:0]**	Receive/transmit data RTDT7 = MSB RTDT0 = LSB	0x0 to 0xff	0x0	R/W	

D[15:12] Reserved

D11 RBRDY: Receive Buffer Ready Flag Bit

Indicates the receive buffer status.
1 (R): Receive data exists
0 (R): No receive data (default)

The RBRDY flag becomes 1 when the data received in the shift register is loaded to RTDT[7:0] and reverts to 0 when the receive data is read out from RTDT[7:0]. Interrupts can also be generated once the flag value becomes 1.

D10 RXE: Receive Execution Bit

Receives 1 byte of data.
1 (R/W): Data reception start
0 (R/W): Ineffective (default)

Setting RXE to 1 and TXE to 0 starts receiving for 1 byte of data. RXE can be set to 1 for subsequent reception, even if the slave address is being sent or data is being received. RXE is reset to 0 as soon as D7 is loaded to the shift register.

D9 TXE: Transmit Execution Bit

Transmits 1 byte of data.
1 (R/W): Data transmission start
0 (R/W): Ineffective (default)

Transmission is started by setting the transmit data to RTDT[7:0] and writing 1 to TXE. TXE can be set to 1 for subsequent transmission, even if the slave address or data is being sent. TXE is reset to 0 as soon as the data set in RTDT[7:0] is transferred to the shift register.

D8 RTACK: Receive/Transmit ACK Bit

When transmitting data

Indicates the response bit status.
1 (R/W): Error (NAK)
0 (R/W): ACK (default)

RTACK becomes 0 when ACK is returned from the slave after 1 byte of data is sent, indicating that the slave has received the data correctly. If RTACK is 1, the slave device is not operating or the data was not received correctly.

When receiving data

Sets the response bit sent to the slave.
1 (R/W): Error (NAK)
0 (R/W): ACK (default)

To return an ACK after data has been received, RTACK should be set to 0 before the I2CM module sends the response bit. To return a NAK, set RTACK to 1.

D[7:0] RTDT[7:0]: Receive/Transmit Data Bits

When transmitting data

Sets the transmit data. (Default: 0x0)
Data transmission is started by setting TXE to 1. If a slave address or data is currently being transmitted, transmission begins once the previous transmission is completed. Serial converted data is output from the SDA0 pin with MSB leading and bits set to 0 as Low level. A cause of transmit buffer empty interrupt is generated as soon as the data written to this register is transferred to the shift register, after which the subsequent transmission data can be written.

When receiving data

The received data can be read out. (Default: 0x0)
Data reception is started by setting RXE to 1. If a slave address is currently being transmitted or data is currently being received, the new reception starts once the previous data has been transferred. The RBRDY flag is set and a cause of receive buffer full interrupt generated as soon as reception is completed and the shift register data is transferred to this register. Data can then be read until the subsequent data has been received. If the subsequent data is received before this register is read out, the contents are overwritten by the most recent received data. Serial data input from the SDA0 pin with MSB leading is converted to parallel, with the High level bit set to 1 and the Low level bit set to 0, then loaded to this register.

I²C Master Interrupt Control Register (I2CM_ICTL)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
I²C Master Interrupt Control Register (I2CM_ICTL)	0x4346 (16 bits)	D15–2	–	reserved	–				–	–	0 when being read.
		D1	RINTE	Receive interrupt enable	1	Enable	0	Disable	0	R/W	
		D0	TINTE	Transmit interrupt enable	1	Enable	0	Disable	0	R/W	

D[15:2] Reserved

D1 RINTE: Receive Interrupt Enable Bit

Enables or disables I2CM receive buffer full interrupts.
1 (R/W): Enabled
0 (R/W): Disabled (default)

Setting RINTE to 1 enables the output of I2CM interrupt requests to the ITC due to a receive data buffer full. These interrupt requests are generated when the data received in the shift register is transferred to RTDT[7:0]/I2CM_DAT register (when reception is completed).
I2CM interrupts are not generated by receive data buffer full if RINTE is set to 0.

D0 TINTE: Transmit Interrupt Enable Bit

Enables or disables I2CM transmit buffer empty interrupts.
1 (R/W): Enabled
0 (R/W): Disabled (default)

Setting TINTE to 1 enables the output of I2CM interrupt requests to the ITC due to a transmit buffer empty. These interrupt requests are generated when the data written to RTDT[7:0] is transferred to the shift register.
I2CM interrupts are not generated by transmit buffer empty if TINTE is set to 0.

18 I2C Slave (I2CS)

18.1 I2CS Module Overview

The S1C17554/564 includes an I2C slave (I2CS) module that supports two-wire communications. The I2CS module operates as an I2C bus slave device and can communicate with an I2C-compliant master device.
The following shows the main features of I2CS:

- Operates as an I2C bus slave device.
- Supports standard (100 kbps) and fast (400 kbps) modes.
- Supports 8-bit data length only (MSB first).
- Supports 7-bit addressing mode.
- Includes one-byte receive data buffer and one-byte transmit data buffer.
- Can detect start and stop conditions.
- Supports half-duplex communications.
- Supports clock stretch function.
- Supports forced bus release function.
- Includes a noise filter function to help improve the reliability of data transfers.
- Can generate transmit buffer empty, receive buffer full, and bus status interrupts.

Figure 18.1.1 shows the I2CS configuration.

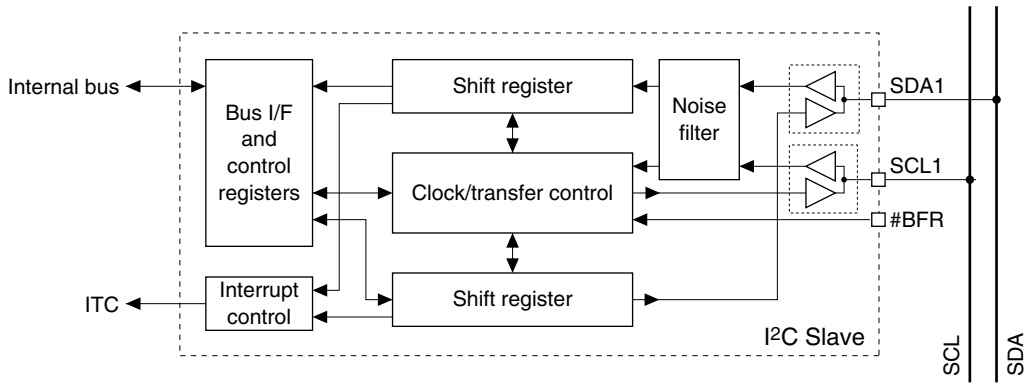


Figure 18.1.1 I2CS Module Configuration

Note: The I2CS module does not support general call address and 10-bit address mode.

18.2 I2CS Input/Output Pins

Table 18.2.1 lists the I2CS pins.

Table 18.2.1 List of I2CS Pins

Pin name	I/O	Qty	Function
SDA1	I/O	1	I2CS data input/output pin (see Note below) Inputs serial data from the I2C bus. Also outputs serial data to the I2C bus.
SCL1	I/O	1	I2CS clock input/output pin (see Note below) Inputs SCL line status from the I2C bus. Also outputs a low level to put the I2C bus into clock stretch status.
#BFR	I	1	I2C bus free request input pin A Low pulse input to this pin requests the I2CS to release the I2C bus. When the bus free request input has been enabled with software, a Low pulse initializes the communication process of the I2CS module and sets the SDA1 and SCL1 pins into high impedance.

The I2CS input/output pins (SDA1, SCL1, #BFR) are shared with I/O ports and are initially set as general purpose I/O port pins. The pin functions must be switched using the port function select bits to use the general purpose I/O port pins as I2CS input/output pins. For detailed information on pin function switching, see the "I/O Ports (P)" chapter.

Note: The pins go to high impedance status when the port function is switched.
The SCL1 and SDA1 pins do not output a high level, so these lines should be pulled up to HVDD with an external pull-up resistor. Be sure to avoid pulling these pins up to a voltage that exceeds the HVDD level.

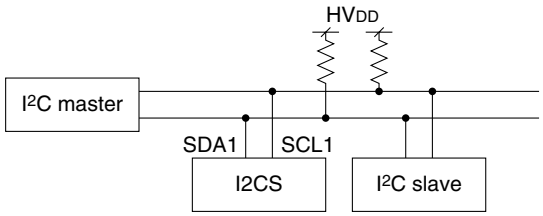


Figure 18.2.1 I2C Connection Example

18.3 Operation Clock

The I2CS module operates with the clock output from the external I2C master device by inputting it from the SCL1 pin.
The I2CS module also uses the peripheral module clock (PCLK) for its operations. The PCLK frequency must be set eight-times or higher than the SCL1 input clock frequency during data transfer. In standby status, use of the asynchronous address detection function allows the application to lower the PCLK clock frequency to reduce current consumption. For more information, see "Asynchronous address detection function" in Section 18.4.3.

18.4 Initializing I2CS

18.4.1 Reset

The I2CS module must be reset to initialize the communication process and to set the I2C bus into free status (high impedance). The following shows two methods for resetting the module:

(1) Software reset

The I2CS module can be reset using SOFTRESET/I2CS_CTL register.
To reset the I2CS module, write 1 to SOFTRESET to place the I2CS module into reset status, then write 0 to SOFTRESET to release it from reset status. It is not necessary to insert a waiting time between writing 1 and 0.
The I2CS module initializes the I2C communication process and put the SDA1 and SCL1 pins into high-impedance to be ready to detect a start condition. Furthermore, the I2CS control bits except for SOFTRESET are initialized. Perform the software reset in the initial setting process before staring communication.

(2) Bus free request with an input from the #BFR pin

The I2CS module can accept bus free requests via the #BFR pin. The bus free request support is disabled by default. To enable this function, set BFREQ_EN/I2CS_CTL register to 1.
When this function is enabled, a low pulse (One peripheral module clock (PCLK) cycle or more pulse width is required. Two PCLK clock cycles or more pulse width is recommended.) input to the #BFR pin sets BFREQ/I2CS_STAT register to 1. This initializes the I2C communication process and puts the SDA1 and SCL1 pins into high-impedance. The control registers will not be initialized as distinct from the software reset described above.

Note: When BFREQ is set to 1 (an interrupt can be used for checking this status), perform a software reset and set the registers again.

18.4.2 Setting Slave Address

I2C devices have a unique slave address to identify each device.
The I2CS module supports 7-bit address (does not support 10-bit address), and the address of this module must be set to SADRS[6:0]/I2CS_SADRS register.

18.4.3 Optional Functions

The I2CS module has a clock stretch, asynchronous address detection, and noise filter optional functions selectable in the application program.

Clock stretch function

After data and ACK are transmitted or received, the slave device may issue a wait request to the master device until it is ready to transmit/receive by pulling the I2C bus SCL line down to low. The I2CS module supports this clock stretch function.
The master device enters a standby state until the wait request is canceled (the SCL line goes high). The clock stretch function in this module is disabled by default. When using the clock stretch function, set CLKSTR_EN/I2CS_CTL register to 1 before starting data communication. Note that the data setup time (after the SDA1 pin outputs the MSB of SDATA[7:0]/I2CS_TRNS register until I2CS turns the SCL1 pin pull-down resistor off) while the I2CS module is operating with the clock stretch function enabled varies depending on the I2CS module operating clock (PCLK) frequency.

Asynchronous address detection function

The I2CS module operation clock (PCLK) frequency must be set eight-times or higher than the transfer rate during data transfer. However, the PCLK frequency can be lowered to reduce current consumption if no other processing is required during standby for data transfer. The asynchronous address detection function is provided to detect the I2C slave address sent from the master in this status.
The asynchronous address detection function in this module is disabled by default. When using the asynchronous address detection function, set ASDET_EN/I2CS_CTL register to 1.
If the slave address sent from the master has matched with one that has been set in this I2CS module when the asynchronous address detection function has been enabled, the I2CS module generates a bus status interrupt and returns NAK to the I2C master to request for resending the slave address.
Set the PCLK frequency to eight-times or higher than the transfer rate and reset ASDET_EN to 0 in the interrupt handler routine. Data transfer will be able to resume normally after the master retries transmission. After the master generates a stop condition to put the I2C bus into free status, the asynchronous address detection function can be enabled again to lower the operating speed.

Notes: • When the asynchronous address detection function is enabled, the I2C bus signals are input without passing through the noise filter. Therefore, the slave address may not be detected in a high-noise environment.

• When the asynchronous address detection function is enabled, data transfer cannot be performed even if the PCLK frequency is eight-times or higher than the transfer rate. Be sure to disable the asynchronous address detection function during normal operation.

Noise filter

The I2CS module includes a function to remove noise from the SDA1 and SCL1 input signals. This function is enabled by setting NF_EN/I2CS_CTL register to 1.

18.5 Data Transfer Control

Make the following settings before starting data transfers.
(1) Initialize the I2CS module. See Section 18.4.
(2) Set the interrupt conditions to use I2CS interrupt. See Section 18.6.

Note: Make sure that the I2CS module is disabled (I2CSEN/I2CS_CTL register = 0) before setting the conditions above.

Enabling data transfers

First, set I2CSEN/I2CS_CTL register to 1 to enable I2CS operation. This makes the I2CS in ready-to-transmit/receive status in which a start condition can be detected.

Note: Do not set the I2CSEN bit to 0 while the I2CS module is transmitting/receiving data.

Starting data transfer

To start data transmission/reception, set COM_MODE/I2CS_CTL register to 1 to enable data communications. When the slave address for this module that has been sent from the master is received after a start condition is detected, the I2CS module returns an ACK (SDA1 = low) and starts operating for data reception or data transmission according to the transfer direction bit that has been received with the slave address.

When COM_MODE is 0 (default), the I2CS module does not send back a response if the master has sent the slave address of this module (it is regarded as that the I2CS module has returned a NAK to the master).

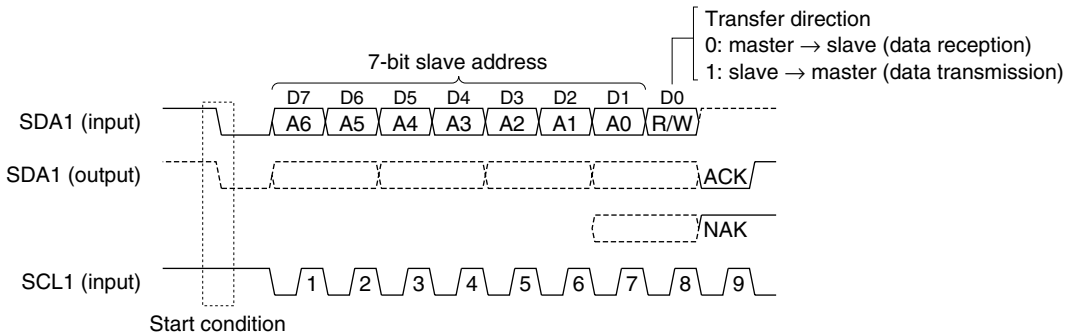


Figure 18.5.1 Receiving Slave Address and Data Direction Bit

When a start condition is detected, BUSY/I2CS_ASTAT register is set to 1 to indicate that the I²C bus is put into busy status. When the slave address of this module is received, SELECTED/I2CS_ASTAT register is set to 1 to indicate that this module has been selected as the I²C slave device. BUSY is maintained at 1 until a stop condition is detected. SELECTED is maintained at 1 until a stop condition or repeated start condition is detected.
The value of the transfer direction bit is set to R/W/I2CS_ASTAT register, so use R/W to select the transmit- or receive-handling.

If the slave address of this module is detected when the asynchronous address detection function has been enabled, ASDET/I2CS_STAT register is set to 1. The I2CS module generates a bus status interrupt and returns NAK to the I²C master to request for resending the slave address. Set the PCLK frequency to eight-times or higher than the transfer rate and disable the asynchronous address detection function in the interrupt handler routine. Data transfer will be able to resume normally after the master retries transmission. ASDET can be cleared by writing 1.

Data transmission

The following describes a data transmission procedure.
The I2CS module starts data transmission process when both SELECTED and R/W are set to 1. It sets TXEMP/I2CS_ASTAT register to 1 to issue a request to the application program to write transmit data. Write transmit data to SDATA[7:0]/I2CS_TRNS register.

When setting the first transmit data after this module has been selected as the slave device, follow the precautions described below.

When the clock stretch function is disabled (default)

Transmit data must be written to SDATA[7:0] within 1 cycle of the I²C clock (SCL1 input clock) after TXEMP has been set to 1. This time is not enough for data preparation, so write transmit data before TXEMP has been set to 1. If the previous transmit data is still stored in SDATA[7:0], it is overwritten with the new data to be transferred. Therefore, the clear operation (see below) using TBUF_CLR is unnecessary.

When the asynchronous address detection function is used, the data written before ASDET_EN is reset to 0 becomes invalid. Therefore, transmission data must be written after TXEMP has been set to 1.

When the clock stretch function is enabled

The master device is placed into wait status by the clock stretch function, so transmit data can be written after TXEMP is set. However, if the previous transmit data is still stored in SDATA[7:0], it will be sent immediately after TXEMP has been set. In order to avoid this problem, clear the I2CS_TRNS register using TBUF_CLR/I2CS_CTL register before this module is selected as the slave device. The I2CS_TRNS register is cleared by writing 1 to TBUF_CLR then writing 0 to it.

It is not necessary to clear the I2CS_TRNS register if the first transmit data is written before TXEMP has been set.

When the asynchronous address detection function is used, the data written before ASDET_EN is reset to 0 becomes invalid. Therefore, transmission data must be written after TXEMP has been set to 1.

For writing transmit data other than the first time, use an interrupt that can be generated when TXEMP is set to 1. TXEMP is also set to 1 when the transmit data written to SDATA[7:0] is loaded to the sift register during transmission. TXEMP is cleared by writing transmit data to SDATA[7:0].

When the clock stretch function is disabled (default)

When the clock stretch function has been disabled, data must be written to the I2CS_TRNS register within 7 cycles of the I²C clock (SCL1 input clock) from TXEMP being set to 1.

If data has not been written in this period, the current register value (previous transmit data) will be sent. In this case, TXUDF/I2CS_STAT register is set to 1 to indicate that invalid data has been sent. An interrupt can be generated when TXUDF is set to 1, so an error handling should be performed in the interrupt handler routine. TXUDF is cleared by writing 1.

When the clock stretch function is enabled

When the clock stretch function has been enabled, the I2CS module pulls down the SCL1 pin to low to generate a clock stretch (wait) status until transmit data is written to the I2CS_TRNS register.

Transmit data bits are output from the SDA1 pin in sync with the SCL1 input clock sent from the master. The MSB is output first. After the eight bits has been output, the master sends back an ACK or NAK in the ninth clock cycle.

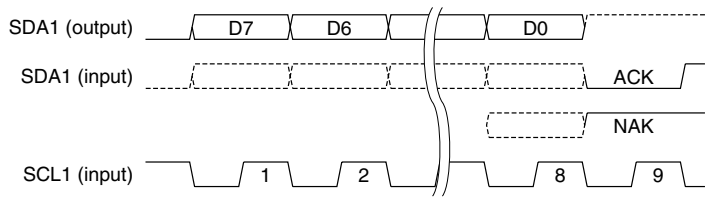


Figure 18.5.2 ACK and NAK

The ACK bit indicates that the master could receive data. It is also a transmit request bit, therefore, the next transmit data must be written in advance. Receiving an ACK generates a clock stretch status when the clock stretch function has been enabled, so data can be written after an ACK is received.

A NAK will be returned from the master if the master could not receive data or when the master terminates data reception. In this case a clock stretch status is not generated even if the clock stretch function has been enabled.

Read DA_NAK/I2CS_STAT register to check if an ACK is returned or if a NAK is returned. DA_NAK is set to 0 when an ACK is returned or set to 1 when a NAK is returned. An interrupt can be generated when DA_NAK is set to 1, so an error or termination handling can be performed in the interrupt handler routine. DA_NAK is cleared by writing 1.

The SDA line status during data transmission is input in the module and is compare with the output data. The comparison results are set to DMS/I2CS_STAT register. DMS is set to 0 when data is output correctly. If the SDA line status is different from the output data, DMS is set to 1. This may be caused by a low pull-up resistor value or another device that is controlling the SDA line. An interrupt can be generated when DMS is set to 1, so an error handling can be performed in the interrupt handler routine. DMS is cleared by writing 1.

Note: If the I2CS module has sent back a NAK as the response to the address sent by the master when the conditions shown below are all met, the master must wait for 33 µs or more before it can send another slave address (except when the master sends the I2CS slave address again).
1. The transfer rate is set to 320 kbps or higher.
2. The asynchronous address detection function is enabled.
3. The I2CS module is placed into transfer standby state and OSC1 is used as the operating clock (PCLK).

Data reception

The following describes a data receive procedure.
The I2CS module starts data receiving process when SELECTED is set to 1 and R/W is set to 0. The received data bits are input from the SDA1 pin in sync with the SCL1 input clock sent from the master. When the eight-bit data (MSB first) is received in the shift register, the received data is loaded to RDATA[7:0]/I2CS_RECV register.
When the received data is loaded to RDATA[7:0], RXRDY/I2CS_ASTAT register is set to 1 to issue a request to the application program to read RDATA[7:0]. An interrupt can be generated when RXRDY is set to 1, so the received data should be read in the interrupt handler routine. RXRDY is cleared by reading the received data.

When the clock stretch function is disabled (default)

When the clock stretch function has been disabled, data must be read from the I2CS_RECV register within 7 cycles of the I²C clock (SCL1 input clock) from RXRDY being set to 1.

When the clock stretch function is enabled

When the clock stretch function has been enabled, the I2CS module pulls down the SCL1 pin to low to generate a clock stretch (wait) status until the received data is read from the I2CS_RECV register.

If the next data has been received without reading the received data, RDATA[7:0] will be overwritten. In this case, RXOVF/I2CS_STAT register is set to 1 to indicate that the received data has been overwritten. An interrupt can be generated when RXOVF is set to 1, so an error handling should be performed in the interrupt handler routine. RXOVF is cleared by writing 1.

To return NAK during data reception

During data reception (master transmission), the I2CS module sends back an ACK (SDA1 = low) every time an 8-bit data has been received (by default setting). The response code can be changed to NAK (SDA1 = Hi-Z) by setting NAK_ANS/I2CS_CTL register. An ACK will be sent when NAK_ANS is 0 or a NAK will be sent when NAK_ANS is set to 1.
NAK_ANS should be set within 7 cycles of the I²C clock (SCL1 input clock) after RXRDY has been set to 1 by receiving data just prior to one required for returning NAK.

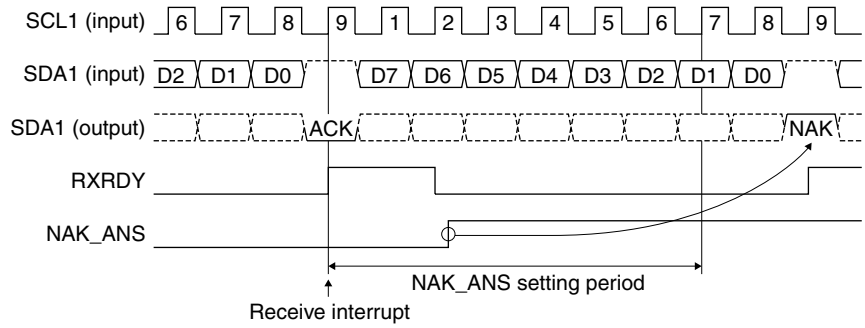


Figure 18.5.3 NAK_ANS Setting and NAK Response Timing

End of data transfer (detecting stop condition)

Data transfers will be terminated when the master generates a stop condition. The stop condition is a state in which the SDA line is pulled up from Low to High with the SCL line maintained at High.

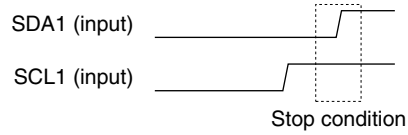


Figure 18.5.4 Stop Condition

If a stop condition is detected while the I2CS module is selected as the slave device (SELECTED = 1), the I2CS module sets DA_STOP/I2CS_STAT register to 1. At the same time, it sets the SDA1 and SCL1 pins into high-impedance and initializes the I2C communication process to enter standby state that is ready to detect the next start condition. Also SELECTED and BUSY are reset to 0.

An interrupt can be generated when DA_STOP is set to 1, so a communication terminating process should be performed in the interrupt handler routine. DA_STOP is cleared by writing 1.

Disabling data transfer

After data transfer has finished, write 0 to the COM_MODE/I2CS_CTL register to disable data transfer.

Always make sure that BUSY and SELECTED are 0 before disabling data transfer.

To deactivate the I2CS module, set I2CSEN/I2CS_CTL register to 0.

Timing charts

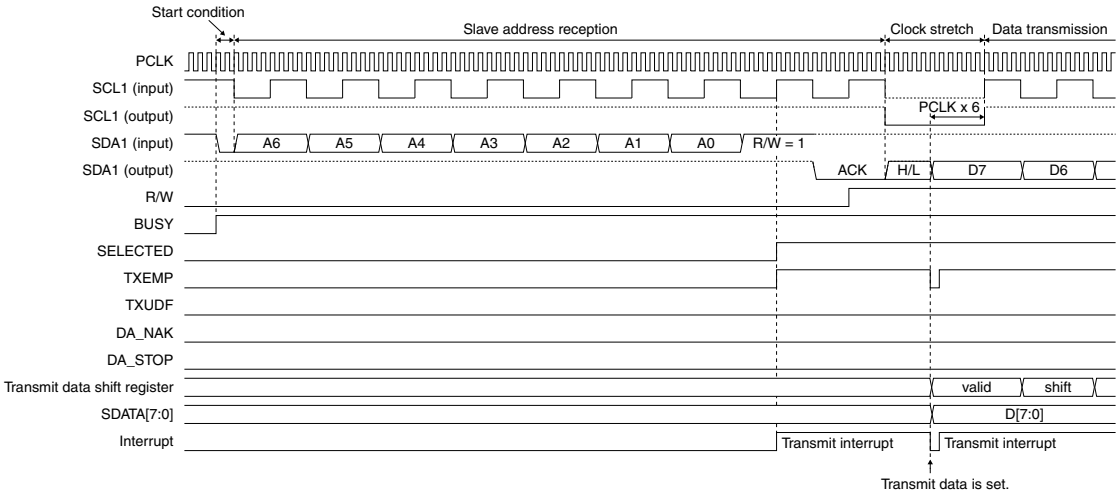


Figure 18.5.5 I2CS Timing Chart 1 (start condition → data transmission)

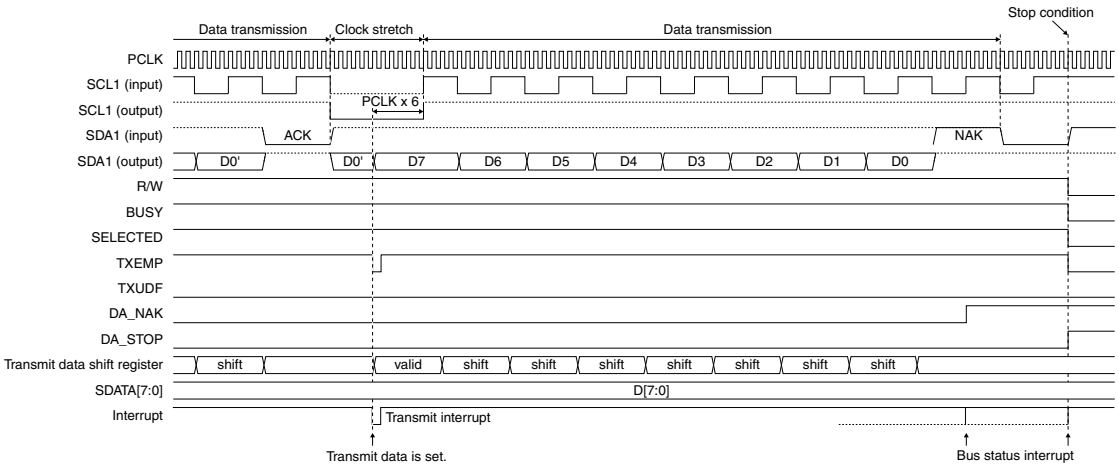


Figure 18.5.6 I2CS Timing Chart 2 (data transmission → stop condition)

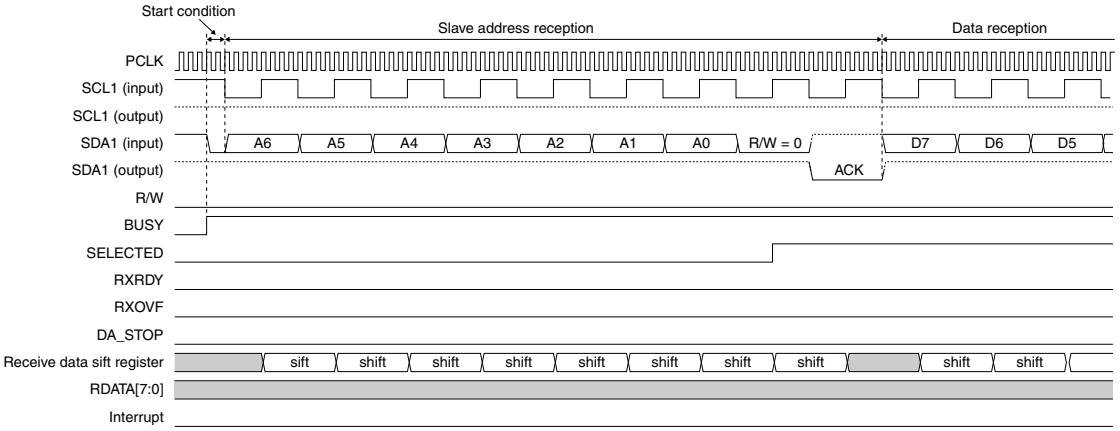


Figure 18.5.7 I2CS Timing Chart 3 (start condition → data reception)

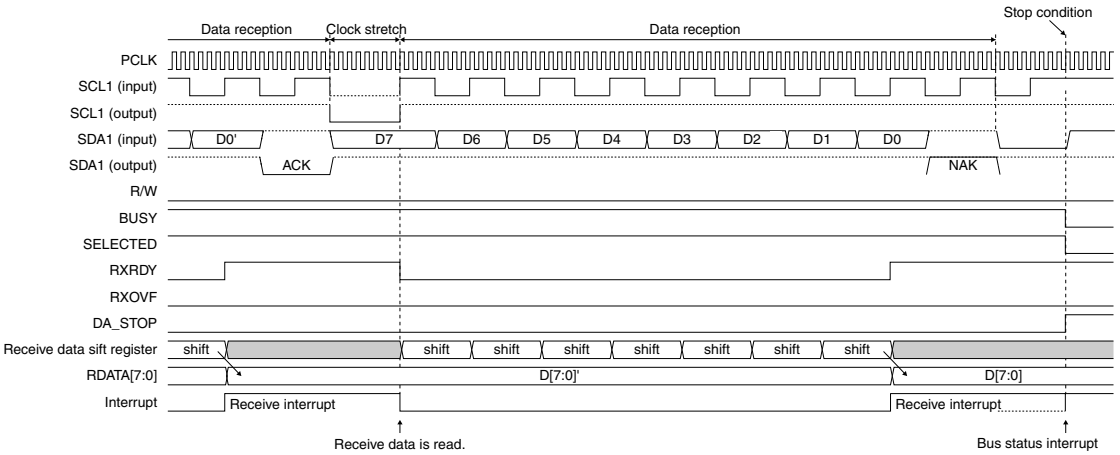


Figure 18.5.8 I2CS Timing Chart 4 (data reception → stop condition)

18.6 I2CS Interrupts

The I2CS module includes a function for generating the following three different types of interrupts.

- Transmit interrupt
- Receive interrupt
- Bus status interrupt

The I2CS module outputs one interrupt signal shared by the three above interrupt causes to the interrupt controller (ITC).

Transmit interrupt

When the transmit data written to SDATA[7:0]/I2CS_TRNS register is sent to the shift register, TXEMP/I2CS_ASTAT register is set to 1 and an interrupt signal is output to the ITC. An interrupt occurs if other interrupt conditions are satisfied. This interrupt can be used to write the next transmit data to SDATA[7:0].

Set TXEMP_IEN/I2CS_ICTL register to 1 when using this interrupt. If TXEMP_IEN is set to 0 (default), interrupt requests by this cause will not be sent to the ITC.

Receive interrupt

When the received data is loaded to RDATA[7:0]/I2CS_RECV register, RXRDY/I2CS_ASTAT register is set to 1 and an interrupt signal is output to the ITC. An interrupt occurs if other interrupt conditions are satisfied. This interrupt can be used to read the received data from RDATA[7:0].

Set RXRDY_IEN/I2CS_ICTL register to 1 when using this interrupt. If RXRDY_IEN is set to 0 (default), interrupt requests by this cause will not be sent to the ITC.

Bus status interrupt

The I2CS module provides the status bits listed below to represent the transmit/receive and I²C bus statuses (see Section 18.5 for details of each function).

1. ASDET/I2CS_STAT register: This bit is set to 1 when the slave address is detected by the asynchronous address detection function.
2. TXUDF/I2CS_STAT register: This bit is set to 1 when a transmit operation has started before transmit data is written. (When the clock stretch function is disabled)
3. DA_NAK/I2CS_STAT register: This bit is set to 1 when a NAK is returned from the master during transmission.
4. DMS/I2CS_STAT register: This bit is set to 1 when the SDA line status is different from transfer data.
   DMS will also be set to 1 when another slave device issues ACK to this I²C slave address (when ASDET_EN/I2CS_CTL register = 0).

   Note: When the master device of the I²C bus, which has multiple slave devices connected including this IC, starts communication with another slave device, the I2CS module issues NAK in response to the sent slave address. On the other hand, the selected slave device issues ACK. Therefore, DMS may be set due to a difference between the output value of this IC and the SDA line status. When SELECTED/I2CS_ASTAT register is set to 0, you can ignore DMS without a problem even if it is set to 1 as there is a difference in the response code (ACK/NAK) from the selected slave device.
   When the I2CS module is placed into asynchronous address detection mode (ASDET_EN = 1), a DMS does not occur as in the condition above.
5. RXOVF/I2CS_STAT register: This bit is set to 1 when the next data has been received before the received data is read (the received data is overwritten). (When the clock stretch function is disabled)
6. BFREQ/I2CS_STAT register: This bit is set to 1 when a bus free request is accepted.
7. DA_STOP/I2CS_STAT register: This bit is set to 1 if a stop condition or a repeated start condition is detected while this module is selected as the slave device.

When one of the bits listed above is set to 1, BSTAT/I2CS_STAT register is set to 1 and an interrupt signal is output to the ITC. An interrupt occurs if other interrupt conditions are satisfied. This interrupt can be used to perform an error or terminate handling.
Set BSTAT_IEN/I2CS_ICTL register to 1 when using this interrupt. If BSTAT_IEN is set to 0 (default), interrupt requests by this cause will not be sent to the ITC.

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

18.7 Control Register Details

Table 18.7.1 List of I2CS Registers

Address	Register name		Function
0x4360	I2CS_TRNS	I²C Slave Transmit Data Register	I²C slave transmit data
0x4362	I2CS_RECV	I²C Slave Receive Data Register	I²C slave receive data
0x4364	I2CS_SADRS	I²C Slave Address Setup Register	Sets the I²C slave address.
0x4366	I2CS_CTL	I²C Slave Control Register	Controls the I²C slave module.
0x4368	I2CS_STAT	I²C Slave Status Register	Indicates the I²C bus status.
0x436a	I2CS_ASTAT	I²C Slave Access Status Register	Indicates the I²C slave access status.
0x436c	I2CS_ICTL	I²C Slave Interrupt Control Register	Controls the I²C slave interrupt.

The I2CS module registers are described in detail below. These are 16-bit registers.

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

I²C Slave Transmit Data Register (I2CS_TRNS)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
I²C Slave Transmit Data Register (I2CS_TRNS)	0x4360 (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	SDATA[7:0]	I²C slave transmit data	0–0xff	0x0	R/W	

D[15:8] Reserved

D[7:0] SDATA[7:0]: I²C Slave Transmit Data Bits

Sets a transmit data in this register. (Default: 0x0)

The serial-converted data is output from the SDA1 pin beginning with the MSB, in which the bits set to 0 are output as Low-level signals. When the data set in this register is sent to the shift register, a transmit interrupt occurs. The next transmit data can be written to the register after that.

If the clock stretch function has been disabled, data must be written to this register within 7 cycles of the I²C clock (SCL1 input clock) after a transmit interrupt has been occurred.

However, when setting the first transmit data after this module has been selected as the slave device, follow the precautions described below.

When the clock stretch function is disabled (default)

Transmit data must be written to SDATA[7:0] within 1 cycle of the I²C clock (SCL1 input clock) after TXEMP has been set to 1. This time is not enough for data preparation, so write transmit data before TXEMP has been set to 1. If the previous transmit data is still stored in SDATA[7:0], it is overwritten with the new data to be transferred. Therefore, the clear operation (see below) using TBUF_CLR is unnecessary.

When the asynchronous address detection function is used, the data written before ASDET_EN is reset to 0 becomes invalid. Therefore, transmission data must be written after TXEMP has been set to 1.

When the clock stretch function is enabled

The master device is placed into wait status by the clock stretch function, so transmit data can be written after TXEMP is set. However, if the previous transmit data is still stored in SDATA[7:0], it will be sent immediately after TXEMP has been set. In order to avoid this problem, clear the I2CS_TRNS register using TBUF_CLR/I2CS_CTL register before this module is selected as the slave device. The I2CS_TRNS register is cleared by writing 1 to TBUF_CLR then writing 0 to it.

It is not necessary to clear the I2CS_TRNS register if the first transmit data is written before TXEMP has been set.

When the asynchronous address detection function is used, the data written before ASDET_EN is reset to 0 becomes invalid. Therefore, transmission data must be written after TXEMP has been set to 1.

I²C Slave Receive Data Register (I2CS_RECV)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
I²C Slave Receive Data Register (I2CS_RECV)	0x4362 (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	RDATA[7:0]	I²C slave receive data	0–0xff	0x0	R	

D[15:8] Reserved

D[7:0] RDATA[7:0]: I²C Slave Receive Data Bits

The received data can be read from this register. (Default: 0x0)

The serial data input from the SDA1 pin beginning with the MSB is converted into parallel data, with the high-level signals changed to 1 and the low-level signals changed to 0. The resulting data is stored in this register.

When a receive operation is completed and the data received in the shift register is loaded to this register, RXRDY/I2CS_ASTAT register is set and a receive interrupt occurs. Thereafter, the data can be read out.

When the clock stretch function has been disabled, data must be read from this register within 7 cycles of the I²C clock (SCL1 input clock) after RXRDY is set to 1. If the next data has been received without reading the received data, this register will be overwritten with the newly received data.

I2C Slave Address Setup Register (I2CS_SADRS)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
I2C Slave Address Setup Register (I2CS_SADRS)	0x4364 (16 bits)	D15–7	–	reserved	–	–	–	0 when being read.
		D6–0	SADRS[6:0]	I2C slave address	0–0x7f	0x0	R/W	

D[15:7] Reserved

D[6:0] SADRS[6:0]: I2CS Address Bits

Sets the slave address of the I2CS module to this register. (Default: 0x0)

I2C Slave Control Register (I2CS_CTL)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
I2C Slave Control Register (I2CS_CTL)	0x4366 (16 bits)	D15–9	–	reserved	–				–	–	0 when being read.
		D8	TBUF_CLR	I2CS_TRNS register clear	1	Clear state	0	Normal	0	R/W	
		D7	I2CSEN	I2C slave enable	1	Enable	0	Disable	0	R/W	
		D6	SOFTRESET	Software reset	1	Reset	0	Cancel	0	R/W	
		D5	NAK_ANS	NAK answer	1	NAK	0	ACK	0	R/W	
		D4	BFREQ_EN	Bus free request enable	1	Enable	0	Disable	0	R/W	
		D3	CLKSTR_EN	Clock stretch On/Off	1	On	0	Off	0	R/W	
		D2	NF_EN	Noise filter On/Off	1	On	0	Off	0	R/W	
		D1	ASDET_EN	Async.address detection On/Off	1	On	0	Off	0	R/W	
		D0	COM_MODE	I2C slave communication mode	1	Active	0	Standby	0	R/W	

D[15:9] Reserved

D8 TBUF_CLR: I2CS_TRNS Register Clear Bit

Clears the I2CS_TRNS register.

1 (R/W): Clear state

0 (R/W): Normal state (clear state cancellation) (default)

When TBUF_CLR is set to 1, the I2CS_TRNS register enters clear state. After that writing 0 to TBUF_CLR returns the I2CS_TRNS register to normal state. It is not necessary to insert a waiting time between writing 1 and 0.

If a new transmission is started when the I2CS_TRNS register still stores data for the previous transmission that has already finished, the data will be sent when TXEMP/I2CS_ASTAT register is set. In order to avoid this problem, clear the I2CS_TRNS register using TBUF_CLR before starting transmission (before slave selection). The clear operation is not required if transmit data is written to the I2CS_TRNS register before TXEMP is set to 1.

Data can be written to the I2CS_TRNS register even if it is placed into clear state (TBUF_CLR = 1). However, this writing does not reset TXEMP to 0. Note that TXEMP is not reset to 0 when TBUF_CLR is set back to 0. Therefore, data must be written to the I2CS_TRNS register when TBUF_CLR = 0.

D7 I2CSEN: I2C Slave Enable Bit

Enables or disables operations of the I2CS module.

1 (R/W): Enabled

0 (R/W): Disabled (default)

When I2CSEN is set to 1, the I2CS module is activated and data transfer is enabled.

When I2CSEN is set to 0, the I2CS module goes off.

D6 SOFTRESET: Software Reset Bit

Resets the I2CS module.

1 (R/W): Reset

0 (R/W): Cancel reset state (default)

To reset the I2CS module, write 1 to SOFTRESET to place the I2CS module into reset status, then write 0 to SOFTRESET to release it from reset status. It is not necessary to insert a waiting time between writing 1 and 0. The I2CS module initializes the I2C communication process and put the SDA1 and SCL1 pins into high-impedance to be ready to detect a start condition. Furthermore, the I2CS control bits except for SOFTRESET are initialized. Perform the software reset in the initial setting process before staring communication.

D5 NAK_ANS: NAK Answer Bit

Specifies the acknowledge bit to be sent after data reception.
1 (R/W): NAK
0 (R/W): ACK (default)

When an eight-bit data is received, the I2CS module sends back an ACK (SDA1 = low) or a NAK (SDA1 = Hi-Z). Either ACK or NAK should be specified using NAK_ANS within 7 cycles of the I2C clock (SCL1 input clock) after RXRDY has been set to 1 by receiving the previous data.

D4 BFREQ_EN: Bus Free Request Enable Bit

Enables or disables I2C bus free requests by inputting a low pulse to the #BFR pin.
1 (R/W): Enabled
0 (R/W): Disabled (default)

To accept I2C bus free requests, set BFREQ_EN to 1. When a bus free request is accepted, BFREQ/I2CS_STAT register is set to 1. This initializes the I2C communication process and puts the SDA1 and SCL1 pins into high-impedance. The control registers will not be initialized in this process.
When BFREQ_EN is set to 0, low pulse inputs to the #BFR pin are ignored and BFREQ is not set to 1.

D3 CLKSTR_EN: Clock Stretch On/Off Bit

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

After data and ACK are transmitted or received, the slave device may issue a wait request to the master device until it is ready to transmit/receive by pulling the I2C bus SCL line down to Low. The I2CS module supports this clock stretch function. The master device enters a standby state until the wait request is canceled (the SCL line goes high). When using the clock stretch function, set CLKSTR_EN to 1 before starting data communication.

D2 NF_EN: Noise Filter On/Off Bit

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

The I2CS module contains a function to remove noise from the SDA1 and SCL1 input signals. This function is enabled by setting NF_EN to 1.

D1 ASDET_EN: Async. Address Detection On/Off Bit

Turns the asynchronous address detection function on or off.
1 (R/W): On
0 (R/W): Off (default)

The I2CS module operation clock (PCLK) frequency must be set eight-times or higher than the transfer rate during data transfer. However, the PCLK frequency can be lowered to reduce current consumption if no other processing is required during standby for data transfer.
The asynchronous address detection function is provided to detect the I2C slave address sent from the master in this status. This function is enabled by setting ASDET_EN to 1. If the slave address sent from the master has matched with one that has been set in this I2CS module when the asynchronous address detection function has been enabled, the I2CS module generates a bus status interrupt and returns NAK to the I2C master to request for resending the slave address. Set the PCLK frequency to eight-times or higher than the transfer rate and reset ASDET_EN to 0 in the interrupt handler routine. Data transfer will be able to resume normally after the master retries transmission. After the master generates a stop condition to put the I2C bus into free status, the asynchronous address detection function can be enabled again to lower the operating speed.

Notes: • When the asynchronous address detection function is enabled, the I2C bus signals are input without passing through the noise filter. Therefore, the slave address may not be detected in a high-noise environment.

- When the asynchronous address detection function is enabled, data transfer cannot be performed even if the PCLK frequency is eight-times or higher than the transfer rate. Be sure to disable the asynchronous address detection function during normal operation.

D0 COM_MODE: I²C Slave Communication Mode Bit

Enables or disables data communication.
1 (R/W): Enabled
0 (R/W): Disabled (default)

Set COM_MODE to 1 to enable data communication after setting I2CSEN to 1 to enable I2CS operation. When COM_MODE is 0 (default), the I2CS module does not send back a response if the master has sent the slave address of this module (it is regarded as that the I2CS module has returned a NAK to the master).

I²C Slave Status Register (I2CS_STAT)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
I²C Slave Status Register (I2CS_STAT)	0x4368 (16 bits)	D15–8	–	reserved	–				–	–	0 when being read.
		D7	BSTAT	Bus status transition	1	Changed	0	Unchanged	0	R	
		D6	–	reserved	–				–	–	0 when being read.
		D5	TXUDF	Transmit data underflow	1	Occurred	0	Not occurred	0	R/W	Reset by writing 1.
			RXOVF	Receive data overflow							
		D4	BFREQ	Bus free request	1	Occurred	0	Not occurred	0	R/W	
		D3	DMS	Output data mismatch	1	Error	0	Normal	0	R/W	
		D2	ASDET	Async. address detection status	1	Detected	0	Not detected	0	R/W	
		D1	DA_NAK	NAK receive status	1	NAK	0	ACK	0	R/W	
		D0	DA_STOP	STOP condition detect	1	Detected	0	Not detected	0	R/W	

D[15:8] Reserved

D7 BSTAT: Bus Status Transition Bit

Indicates transition of the bus status.
1 (R): Changed
0 (R): Unchanged (default)

When one of the TXUDF/RXOVF, BFREQ, DMS, ASDET, DA_NAK, and DA_STOP bits is set to 1, BSTAT is also set to 1 and an interrupt signal is output to the ITC if the interrupt is enabled with BSTAT_IEN/I2CS_ICTL register. This interrupt can be used to perform an error or terminate handling. BSTAT will be reset to 0 when the TXUDF/RXOVF, BFREQ, DMS, ASDET, DA_NAK, and DA_STOP bits are all reset to 0.

D6 Reserved

D5 TXUDF: Transmit Data Underflow Bit (for transmission)
RXOVF: Receive Data Overflow Bit (for reception)

Indicates the transmit/receive data register status.
1 (R/W): Data underflow/overflow has been occurred
0 (R/W): Data underflow/overflow has not been occurred (default)

This bit is effective during transmission/reception when the clock stretch function is disabled. If a data transmission begins before transmit data is written to the I2CS_TRNS register, it is regarded as a transmit data underflow and TXUDF is set to 1. If the next data reception has completed before the received data is read from the I2CS_RECV register and the I2CS_RECV register value is overwritten with the newly received data, it is regarded as a data overflow and RXOVF is set to 1.
At the same time, an interrupt signal is output to the ITC if the interrupt is enabled with BSTAT_IEN/I2CS_ICTL register. This interrupt can be used to perform an error handling.
After TXUDF/RXOVF is set to 1, it is reset to 0 by writing 1.

D4 BFREQ: Bus Free Request Bit

Indicates the I²C bus free request input status.
1 (R/W): Request has been issued
0 (R/W): Request has not been issued (default)

If BFREQ_EN/I2CS_CTL register has been set to 1 (bus free request enabled), a low pulse longer than five peripheral module clock (PCLK) cycles input to the #BFR pin sets BFREQ to 1 and the bus free request is accepted. When a bus free request is accepted, the I2CS module initializes the I2C communication process and puts the SDA1 and SCL1 pins into high-impedance. The control registers will not be initialized in this process.
At the same time, an interrupt signal is output to the ITC if the interrupt is enabled with BSTAT_IEN/I2CS_ICTL register. This interrupt can be used to perform an error handling.
After BFREQ is set to 1, it is reset to 0 by writing 1.
If BFREQ_EN is set to 0, low pulse inputs to the #BFR pin are ignored and BFREQ is not set to 1.

D3 DMS: Output Data Mismatch Bit

Represents the results of comparison between output data and SDA line status.
1 (R/W): Error has been occurred
0 (R/W): Error has not been occurred (default)

The I2C bus SDA line status during data transmission is input in the module and is compared with the output data. The comparison results are set to DMS. DMS is set to 0 when data is output correctly. If the SDA line status is different from the output data, DMS is set to 1. This may be caused by a low pull-up resistor value or another device that is controlling the SDA line. At the same time, an interrupt signal is output to the ITC if the interrupt is enabled with BSTAT_IEN/I2CS_ICTL register. This interrupt can be used to perform an error handling. After DMS is set to 1, it is reset to 0 by writing 1.

Note: When the master device of the I2C bus, which has multiple slave devices connected including this IC, starts communication with another slave device, the I2CS module of this IC issues NAK in response to the sent slave address. On the other hand, the selected slave device issues ACK. Therefore, DMS may be set due to a difference between the output value of this IC and the SDA line status. When SELECTED/I2CS_ASTAT register is set to 0, you can ignore DMS without a problem even if it is set to 1 as there is a difference in the response code (ACK/NAK) from the selected slave device.
When the I2CS module is placed into asynchronous address detection mode (ASDET_EN = 1), a DMS does not occur as in the condition above.

D2 ASDET: Async. Address Detection Status Bit

Indicates the asynchronous address detection status.
1 (R/W): Detected
0 (R/W): Not detected (default)

The I2CS module operation clock (PCLK) frequency must be set eight-times or higher than the transfer rate during data transfer. However, the PCLK frequency can be lowered to reduce current consumption if no other processing is required during standby for data transfer. The asynchronous address detection function is provided to detect the I2C slave address sent from the master in this status. ASDET is set to 1 if the slave address of the I2CS module is detected when the asynchronous address detection function has been enabled by setting ASDET_EN/I2CS_CTL register.
The I2CS module returns a NAK to the I2C master to request for resending the slave address. At the same time, an interrupt signal is output to the ITC if the interrupt is enabled with BSTAT_IEN/I2CS_ICTL register. Set the PCLK frequency to eight-times or higher than the transfer rate and reset ASDET_EN to 0 in the interrupt handler routine. Data transfer will be able to resume normally after the master retries transmission. After ASDET is set to 1, it is reset to 0 by writing 1.

D1 DA_NAK: NAK Receive Status Bit

Indicates the acknowledge bit returned from the master.
1 (R/W): NAK
0 (R/W): ACK (default)

DA_NAK is set to 0 when an ACK is returned from the master after an eight-bit data has been sent. This indicates that the master could receive data. If DA_NAK is 1, it indicates that the master could not receive data or the master terminates data reception. At the same time DA_NAK is set to 1, an interrupt signal is output to the ITC if the interrupt is enabled with BSTAT_IEN/I2CS_ICTL register. This interrupt can be used to perform an error handling. After DA_NAK is set to 1, it is reset to 0 by writing 1.

D0 DA_STOP: Stop Condition Detect Bit

Indicates that a stop condition or a repeated start condition is detected.
1 (R/W): Detected
0 (R/W): Not detected (default)

If a stop condition or a repeated start condition is detected while the I2CS module is selected as the slave device (SELECTED/I2CS_ASTAT register = 1), the I2CS module sets DA_STOP to 1. At the same time, it initializes the I²C communication process.
When DA_STOP is set to 1, an interrupt signal is output to the ITC if the interrupt is enabled with BSTAT_IEN/I2CS_ICTL register. This interrupt can be used to perform a terminate handling. After DA_STOP is set to 1, it is reset to 0 by writing 1.

I²C Slave Access Status Register (I2CS_ASTAT)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
I²C Slave Access Status Register (I2CS_ASTAT)	0x436a (16 bits)	D15–5	–	reserved	–				–	–	0 when being read.
		D4	RXRDY	Receive data ready	1	Ready	0	Not ready	0	R	
		D3	TXEMP	Transmit data empty	1	Empty	0	Not empty	0	R	
		D2	BUSY	I²C bus status	1	Busy	0	Free	0	R	
		D1	SELECTED	I²C slave select status	1	Selected	0	Not selected	0	R	
		D0	R/W	Read/write direction	1	Output	0	Input	0	R	

D[15:5] Reserved

D4 RXRDY: Receive Data Ready Bit

Indicates that the received data is ready to read.
1 (R): Received data ready
0 (R): No received data (default)

When the received data is loaded to the I2CS_RECV register, RXRDY is set to 1. At the same time, an interrupt signal is output to the ITC if the interrupt is enabled with RXRDY_IEN/I2CS_ICTL register. This interrupt can be used to read the received data from the I2CS_RECV register.
After RXRDY is set to 1, it is reset to 0 when the I2CS_RECV register is read.

D3 TXEMP: Transmit Data Empty Bit

Indicates that transmit data can be written.
1 (R): Transmit data empty (data can be written)
0 (R): Transmit data still stored (data cannot be written) (default)

When the transmit data written to the I2CS_TRNS register is sent, TXEMP is set to 1. At the same time, an interrupt signal is output to the ITC if the interrupt is enabled with TXEMP_IEN/I2CS_ICTL register. This interrupt can be used to write the next transmit data to the I2CS_TRNS register.
After TXEMP is set to 1, it is reset to 0 when data is written to the I2CS_TRNS register.

D2 BUSY: I²C Bus Status Bit

Indicates the I²C bus status.
1 (R): Bus busy status
0 (R): Bus free status (default)

When the I2CS module detects a start condition or detects that the SCL1 or SDA1 signal goes low, BUSY is set to 1 to indicate that the I²C bus enters busy status. The slave select status whether this module is selected as the slave device or not does not affect the BUSY status. After BUSY is set to 1, it is reset to 0 when a STOP condition is detected.

D1 SELECTED: I²C Slave Select Status Bit

Indicates that this module is selected as the I²C slave device.
1 (R): Selected
0 (R): Not selected (default)

When the slave address that is set in this module is received, SELECTED is set to 1 to indicate that this module is selected as the I²C slave device. After SELECTED is set to 1, it is reset to 0 when a stop condition or a repeated start condition is detected.

D0 R/W: Read/Write Direction Bit

Represents the transfer direction bit value.

1 (R): Output (master read operation)

0 (R): Input (master write operation) (default)

The transfer direction bit value that has been received with the slave address is set to R/W. Use R/W to select the transmit- or receive-handling.

I2C Slave Interrupt Control Register (I2CS_ICTL)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
I2C Slave Interrupt Control Register (I2CS_ICTL)	0x436c (16 bits)	D15–3	–	reserved	–				–	–	0 when being read.
		D2	BSTAT_IEN	Bus status interrupt enable	1	Enable	0	Disable	0	R/W	
		D1	RXRDY_IEN	Receive interrupt enable	1	Enable	0	Disable	0	R/W	
		D0	TXEMP_IEN	Transmit interrupt enable	1	Enable	0	Disable	0	R/W	

D[15:3] Reserved

D2 BSTAT_IEN: Bus Status Interrupt Enable Bit

Enables or disables the bus status interrupt.

1 (R/W): Enabled

0 (R/W): Disabled (default)

When BSTAT_IEN is set to 1, I2C bus status interrupt requests to the ITC are enabled. A bus status interrupt request occurs when BSTAT/I2CS_STAT register is set to 1. (See description of BSTAT.)
When BSTAT_IEN is set to 0, a bus status interrupt will not be generated.

D1 RXRDY_IEN: Receive Interrupt Enable Bit

Enables or disables the I2CS receive interrupt.

1 (R/W): Enabled

0 (R/W): Disabled (default)

When RXRDY_IEN is set to 1, I2CS receive interrupt requests to the ITC are enabled. A receive interrupt request occurs when the data received in the shift register is loaded to the I2CS_RECV register (receive operation completed). When RXRDY_IEN is set to 0, a receive interrupt will not be generated.

D0 TXEMP_IEN: Transmit Interrupt Enable Bit

Enables or disables the I2CS transmit interrupt.

1 (R/W): Enabled

0 (R/W): Disabled (default)

When TXEMP_IEN is set to 1, I2CS transmit interrupt requests to the ITC are enabled. A transmit interrupt request occurs when the data written to the I2CS_TRNS register is transferred to the shift register. When TXEMP_IEN is set to 0, a transmit interrupt will not be generated.

19 Universal Serial Interface (USI) [S1C17564]

Note: The universal serial interface (USI) is unavailable in the S1C17554.

19.1 USI Module Overview

The S1C17564 incorporates a two-channel universal serial interface (USI) module in which each channel can be configured as a UART, SPI, or I²C interface unit by the software switch.
The following shows the main features of USI:

- Supports four interface modes: UART, SPI master, I²C master, and I²C slave modes.
- Two channels can be configured to different interface modes.
- Contains one-byte receive data buffer and one-byte transmit data buffer.
- Supports both MSB first and LSB first modes.
- UART mode
  - Character length: 7 or 8 bits
  - Parity mode: even, odd, or no parity
  - Stop bit: 1 or 2 bits
  - Start bit: 1 bit fixed
  - Parity error, framing error, and overrun error detectable
  - Can generate receive buffer full, transmit buffer empty, and receive error interrupts.
- SPI master mode
  - Data length: 8 or 9 bits
  - Supports both fast and normal modes.
  - Data transfer timing (clock phase and polarity variations) is selectable from among 4 types.
  - Can generate receive buffer full, transmit buffer empty, and overrun error interrupts.
- I²C master/slave mode
  - 7-bit addressing mode (10-bit addressing is possible by software control.)
  - Supports single master configuration only (master mode).
  - Supports clock stretch/wait functions.
  - Can generate start/stop, data transfer, ACK/NAK transfer, and overrun error interrupts.

Figure 19.1.1 shows the USI configuration.

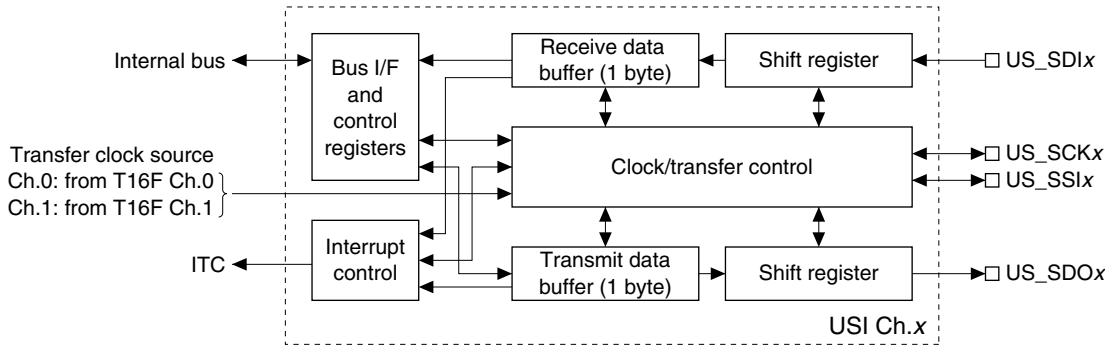


Figure 19.1.1 USI Configuration (one channel)

Note: Two channels in the USI module have the same functions except for control register addresses. For this reason, the description in this chapter applies to all USI channels. The '*x*' in the register name indicates the channel number (0 or 1).
Example: USI_GCFG*x* register
Ch.0: USI_GCFG0 register
Ch.1: USI_GCFG1 register

19.2 USI Pins

Table 19.2.1 lists the USI input/output pins.

Table 19.2.1 List of USI Pins

Pin name	USI mode	Signal name	I/O	Function
US_SDI0	UART	uart_rx	I	Data input pins
US_SDI1	SPI master	spi_sdi	I	Inputs serial data sent from an external serial device.
	I²C master	i2c_sda	I/O	Data input/output pins
	I²C slave	i2c_sda	I/O	Inputs/outputs serial data from/to the I²C bus. (∗1)
US_SDO0	UART	uart_tx	O	Data output pins
US_SDO1	SPI master	spi_sdo	O	Outputs serial data sent to an external serial device.
	I²C master	–	–	Not used
	I²C slave	–	–	
US_SCK0	UART	–	–	Not used
US_SCK1	SPI master	spi_sck	O	Clock output pins Outputs the SPI clock.
	I²C master	i2c_scl	I/O	SCL input/output pins Inputs SCL line status from the I²C bus. Also outputs the I²C clock.
	I²C slave	i2c_scl	I/O	SCL input/output pins Inputs SCL line status from the I²C bus. Also outputs a clock stretch condition.
US_SSI0	UART	–	–	Not used
US_SSI1	SPI master	–	–	
	I²C master	i2c_sda	I/O	Data input/output pins
	I²C slave	i2c_sda	I/O	Inputs/outputs serial data from/to the I²C bus. (∗1)

∗1: When USI Ch.*x* is configured to I²C master or slave mode, either the US_SDI*x* pin or the US_SSI*x* pin can be used as the data input/output pin. Note, however, that both the US_SDI*x* and US_SSI*x* pins cannot be used as the data input/output pin simultaneously.

Note: Use an I/O (P) port to output the slave select signal when USI Ch.*x* is configured to SPI master mode.

The USI input/output pins (US_SDI*x*, US_SDO*x*, US_SCK*x*, US_SSI*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 USI input/output pins.
For detailed information on pin function switching, see the "I/O Ports (P)" chapter.

19.3 USI Clock Sources

Operating clock

The USI module uses PCLK as the operating clock. Therefore, PCLK must be supplied from the CLG before starting the USI including setting the control registers. For more information on the PCLK supply, refer to the "Clock Generator (CLG)."

Transfer clock

When the USI is configured to a UART, SPI master (normal mode), or I²C master device, the source clock for transfer is supplied by the fine mode 16-bit timer (T16F). Program the T16F according to the transfer rate and enable supplying the source clock to the USI module. The USI module divides the source clock to generate the transfer clock (or sampling clock). Be aware that the division ratio in the USI depends on the interface mode.
When the USI is configured to an SPI master (fast mode) device, PCLK is used as the source clock.
When the USI is configured to an I²C slave device, the transfer clock is supplied from the external master device. However, I²C slave mode uses the T16F output clock to generate the sampling signal.

Table 19.3.1 USI Clocks

Clock	Interface mode	USI Ch.0	USI Ch.1
Operating clock	UART	PCLK	PCLK
	SPI master	PCLK	PCLK
	I²C master	PCLK	PCLK
	I²C slave	PCLK	PCLK
Transfer/sampling clock source (division ratio in USI)	UART	T16F Ch.0 (fSOURCE/8)	T16F Ch.1 (fSOURCE/8)
	SPI master	Normal mode: T16F Ch.0 (fSOURCE/2) Fast mode: PCLK (fPCLK)	Normal mode: T16F Ch.1 (fSOURCE/2) Fast mode: PCLK (fPCLK)
	I²C master	T16F Ch.0 (fSOURCE/8)	T16F Ch.1 (fSOURCE/8)
	I²C slave	T16F Ch.0 (fSOURCE) for sampling	T16F Ch.1 (fSOURCE) for sampling

For controlling the T16F module and setting the output clock, refer to the "Fine Mode 16-bit Timers (T16F)" chapter.

Note: When the USI module is set to I²C slave mode, i2c_scl (I²C clock) is supplied from the external I²C master. The T16F output clock frequency (fSOURCE) must be set to eight times the i2c_scl frequency.
In I²C slave mode, the USI module sets i2c_scl to low to put the external I²C master into waiting status until USI becomes ready to operate.
The i2c_scl goes low when an I²C slave operation is completed with the interrupt flag (ISIF/USI_ISIF*x* register) set to 1. (However, if the status bits (ISSTA[2:0]/USI_ISIF*x* register) are 0x1 (a stop condition is detected) when ISIF is set to 1, i2c_scl is maintained at high-impedance status.)
The USI module sets i2c_scl to high-impedance status after two T16F output clock cycles from the point an I²C slave operation specified with ISTGMOD[2:0]/USI_ISTG*x* register is started by writing 1 to ISTG/USI_ISTG*x* register. If data transmission or ACK/NAK transmission is specified as an I²C slave operation, the USI module output data after one clock cycle from writing 1 to ISTG.

19.4 USI Module Settings

Make the following settings before starting data transfers using the USI module.

SPI master and I²C master/slave modes

(1) Configure the pins to be used for USI according to the interface mode. (See Section 19.2.)
(2) Program the clock source module to supply the clock required to the USI module. (See Section 19.3.)
(3) Reset the USI module.
(4) Set the USI interface mode and a general condition (MSB first/LSB first) to be applied to all interface modes.
(5) Set the data format and operating conditions for the interface mode selected.
(6) Set interrupt conditions if necessary. (See Section 19.7.)

UART mode

(1) Program the clock source module to supply the clock required to the USI module. (See Section 19.3.)
(2) Reset the USI module.
(3) Set the USI interface mode and a general condition (MSB first/LSB first) to be applied to all interface modes.
(4) Configure the pins to be used for USI according to the interface mode. (See Section 19.2.)
(5) Set the data format and operating conditions for the interface mode selected.
(6) Set interrupt conditions if necessary. (See Section 19.7.)

The USI pins pulled down to low in the initial status.
When using USI in UART mode, configure USI before switching the pin functions to prevent undesired start bit to be generated.

19.4.1 USI Module Software Reset

Writing 0x0 to USIMOD[2:0]/USI_GCFG*x* register resets the USI module circuits. Be sure to perform software reset before setting the interface mode.

19.4.2 Interface Mode

The USI module provides five serial interface functions shown in Section 19.1. Each channel can be configured to one of them using the USIMOD[2:0]/USI_GCFG*x* register.

Table 19.4.2.1 Interface Mode Selection

USIMOD[2:0]	Interface mode
0x7–0x6	Reserved
0x5	I²C slave
0x4	I²C master
0x3	Reserved
0x2	SPI master
0x1	UART
0x0	Software reset

(Default: 0x0)

Note: Be sure to perform software reset and set the interface mode before changing other USI configurations.

19.4.3 General Settings for All Interface Modes

MSB first/LSB first selection

Use LSBFST/USI_GCFG*x* register to select whether the data MSB or LSB is input/output first.
LSB first is selected when LSBFST is set to 0 (default). MSB first is selected when LSBFST is set to 1.

19.4.4 Settings for UART Mode

When the USI is used in UART mode, configure the data length, stop bit, and parity bit. The start bit length is fixed at 1 bit.

Data length

Use UCHLN/USI_UCFG*x* register to select the data length. Setting UCHLN to 0 (default) configures the data length to 7 bits. Setting UCHLN to 1 configures it to 8 bits.

Stop bit

Use USTPB/USI_UCFG*x* register to select the stop bit length. Setting USTPB to 0 (default) configures the stop bit length to 1 bit. Setting USTPB to 1 configures it to 2 bits.

Parity bit

Use UPREN/USI_UCFG*x* register to select whether the parity function is enabled or not. Setting UPREN to 0 (default) disables the parity function. In this case, no parity bit will be added to transfer data and receive data will not be checked for parity. Setting UPREN to 1 enables the parity function. In this case, a parity bit will be added to transfer data and receive data will be checked for parity.
When the parity function is enabled, the parity mode should be selected using UPMD/USI_UCFG*x* register. Setting UPMD to 0 (default) adds a parity bit and checks for odd parity. Setting UPMD to 1 adds a parity bit and checks for even parity.

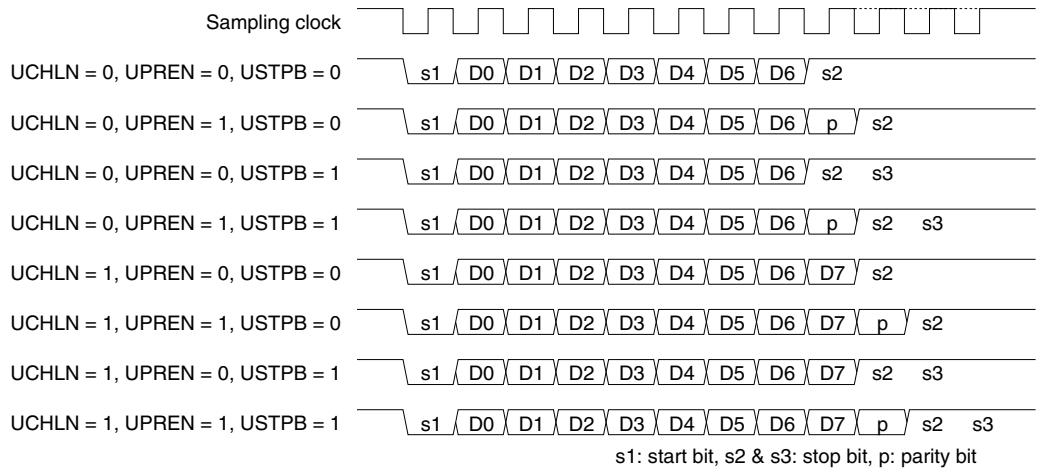


Figure 19.4.4.1 Transfer Data Format in UART Mode (LSB first)

19.4.5 Settings for SPI Master Mode

When the USI is used in SPI master mode, configure the SPI clock polarity/phase, clock mode, and data length.

SPI clock polarity and phase settings

Use SCPOL/USI_SCFG*x* register to select the SPI clock polarity. Setting SCPOL to 1 treats the SPI clock as active low. Setting it to 0 (default) treats it as active high.
The SPI clock phase can be selected using SCPHA/USI_SCFG*x* register.
These control bits set transfer timing as shown in Figure 19.4.5.1.

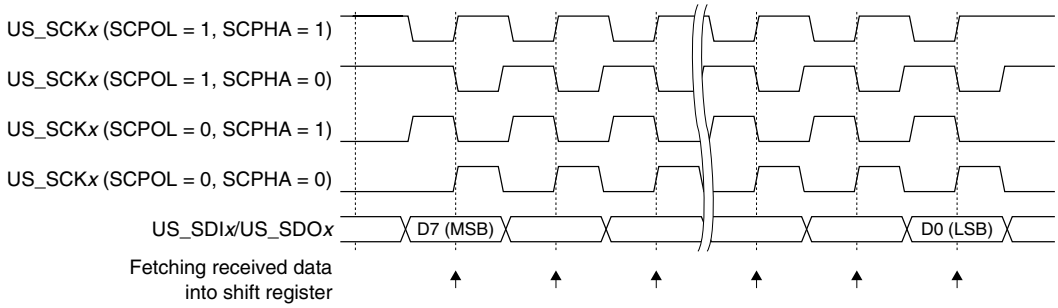


Figure 19.4.5.1 Clock and Data Transfer Timing (MSB first)

Clock mode

Either normal or fast clock mode can be selected using SFSTMOD/USI_SCFG*x* register. Setting SFSTMOD to 0 (default) places the USI into normal mode and the USI generates the transfer clock by dividing the T16F output by 2. Setting SFSTMOD to 1 places the USI into fast mode and the USI uses PCLK supplied from the CLG directly as the transfer clock. The fast mode does not use the T16F.

Data length

The data length can be selected using SCHLN/USI_SCFG*x* register. Setting SCHLN to 0 (default) configures the data length to 8 bits.
Setting SCHLN to 1 configures the data length to 9 bits. In 9-bit mode, 8-bit data is prefixed with a command bit (1 bit). The command bit is used for controlling the SPI LCD controller connected to the USI. The command bit value to be transmitted can be specified using SCMD/USI_SCFG*x* register. Setting SCMD to 1 configures the command bit to high. Setting SCMD to 0 configures the command bit to low.

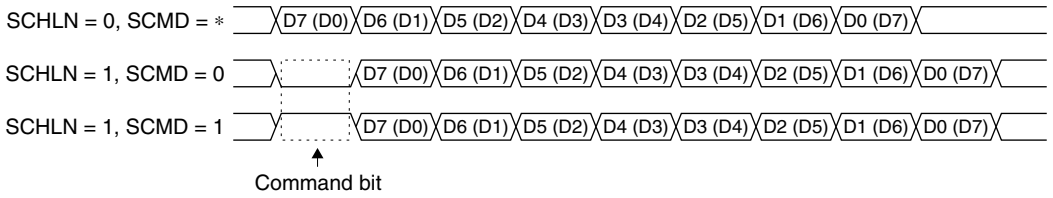


Figure 19.4.5.2 9-bit Transfer Data Format in SPI Master Mode

19.4.6 Settings for I²C Mode

The I²C mode does not need to set data format and other conditions. The data length in I²C mode is fixed at 8 bits.

19.5 Data Transfer Control

This section describes how to control data transfers. The following explanations assume that the configurations described above and interrupt settings have already been finished.

19.5.1 Data Transfer in UART Mode

Data transmission

To start data transmission in UART mode, write the transmit data to the transmit data buffer (TD[7:0]/USI_TD*x* register).

The buffer data is sent to the transmit shift register, and the start bit is output from the US_SDO*x* pin. The data in the shift register is then output in sequence. Following output of the eighth data bit, the parity bit (if parity is enabled) and the stop bit are output.

The transmitter circuit includes two status flags: UTDIF/USI_UIF*x* register and UTBSY/USI_UIF*x* register.

The UTDIF flag indicates the transmit data buffer status. This flag is set to 1 indicating that the transmit data buffer becomes empty when data written to the transmit data buffer is sent to the transmit shift register. UTDIF is an interrupt flag. An interrupt request can be generated when this flag is set to 1 (see Section 19.7). Write subsequent data to the transmit data buffer to start the following transmission using this interrupt. 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. If an interrupt is not used for transmission, be sure to confirm that the transmit data buffer is empty before writing transmit data. Writing data before UTDIF has been set will overwrite earlier transmit data inside the transmit data buffer. After UTDIF is set to 1, it can be reset to 0 by writing 1.

The UTBSY 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.

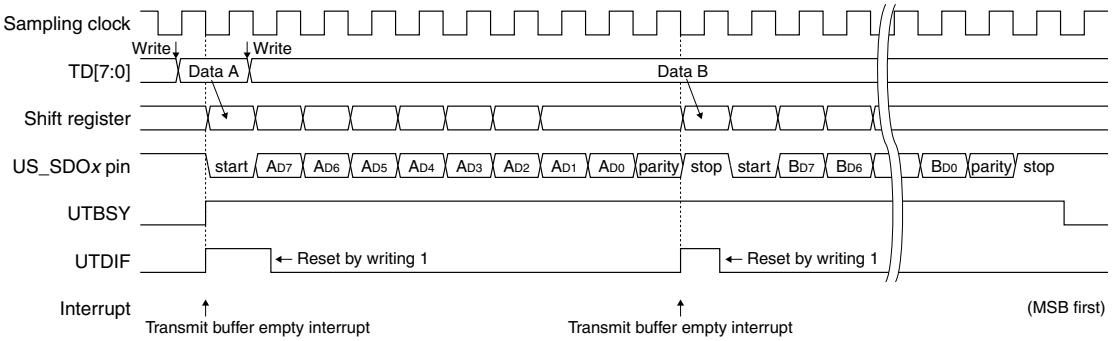


Figure 19.5.1.1 Data Transmission Timing Chart (UART mode)

Data reception

When the external serial device sends a start bit, the receiver circuit detects its low level and starts sampling the following data bits. Once the 8-bit data has been received into the shift register, the received data is loaded into the receive data buffer (RD[7:0]/USI_RD*x* register). 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 eighth data bit.

The receiver circuit includes two status flags: URDIF/USI_UIF*x* register and URBSY/USI_UIF*x* register.

The URDIF flag indicates the receive data buffer status. This flag is set to 1 indicating that the received data can be read out when data received in the shift register is loaded to the receive data buffer. URDIF is an interrupt flag. An interrupt request can be generated when this flag is set to 1 (see Section 19.7). Read the received data from the receive data buffer using this interrupt. The receive data buffer size is 1 byte, therefore the received data must be read before the subsequent data reception has completed. Furthermore, URDIF must be reset by writing 1. If the subsequent receive data is written to the receive data buffer when URDIF is 1, an overrun error occurs.

The URBSY flag indicates the shift register status. This flag is set to 1 while data is being received in the shift register and reverts to 0 once the received data is loaded to the receive data buffer. Read this flag to check whether the receiver circuit is operating or at standby.

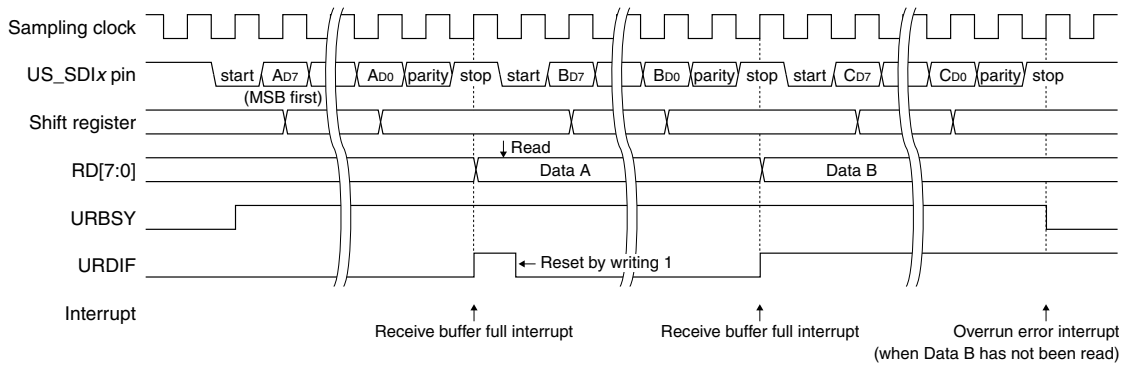


Figure 19.5.1.2 Data Receiving Timing Chart (UART mode)

19.5.2 Data Transfer in SPI Master Mode

Data transmission

To start data transmission in SPI master mode, write the transmit data to the transmit data buffer (TD[7:0]/USI_TD*x* register).

The buffer data is sent to the transmit shift register, and the module starts clock output from the US_SCK*x* pin. The data in the shift register is shifted in sequence at the clock rising or falling edge (see Figure 19.4.5.1) and sent from the US_SDO*x* pin.

The SPI controller includes STDIF/USI_SIF*x* register for transfer control.

The STDIF flag indicates the transmit data buffer status. STDIF is set to 1 indicating that the transmit data buffer becomes empty when data written to the transmit data buffer is sent to the transmit shift register. STDIF is an interrupt flag. An interrupt request can be generated when this flag is set to 1 (see Section 19.7). Write subsequent data to the transmit data buffer to start the following transmission using this interrupt. 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. If an interrupt is not used for transmission, be sure to confirm that the transmit data buffer is empty before writing transmit data. Writing data before STDIF has been set will overwrite earlier transmit data inside the transmit data buffer.

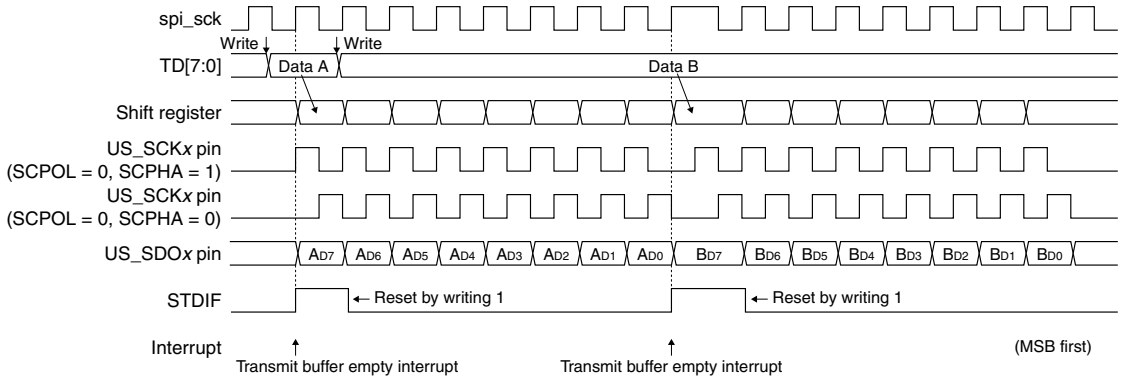


Figure 19.5.2.1 Data Transmission Timing Chart (SPI master mode)

Data reception

Write dummy data to the transmit data buffer. Writing to the transmit data buffer 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 US_SCK*x* pin.

The data is received in sequence in the shift register at the SPI clock edge (see Figure 19.4.5.1). 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 RD[7:0]/USI_RD*x* register.

The SPI controller includes SRDIF/USI_SIF*x* register for transfer control.

The SRDIF 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. SRDIF is an interrupt flag. An interrupt request can be generated when this flag is set to 1 (see Section 19.7). Read the received data from the receive data buffer using this interrupt. The receive data buffer size is 1 byte, therefore the received data must be read before the subsequent data reception has completed. Furthermore, SRDIF must be reset by writing 1. If the subsequent receive data is written to the receive data buffer when SRDIF is 1, an overrun error occurs.

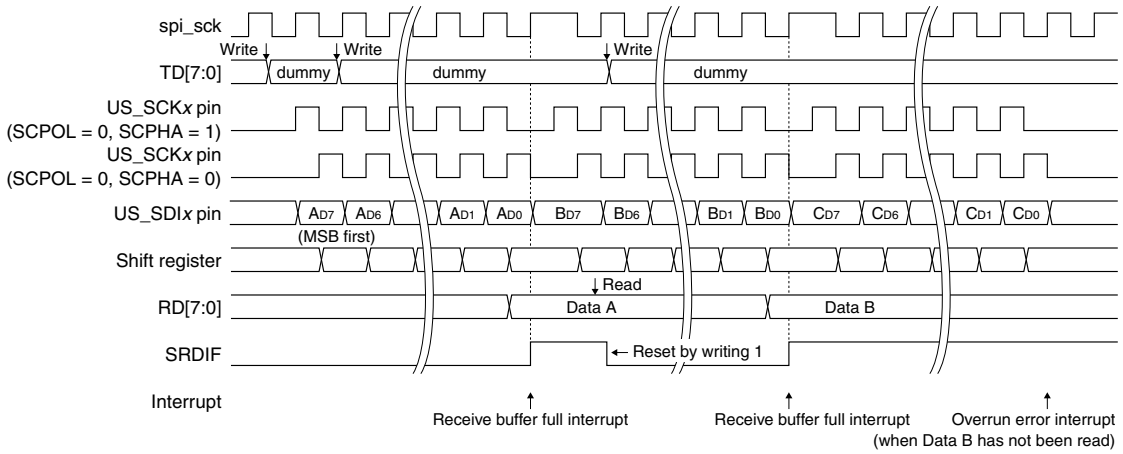
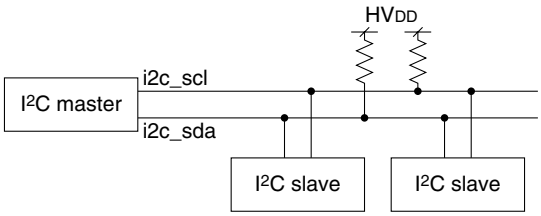


Figure 19.5.2.2 Data Receiving Timing Chart (SPI master mode)

Slave select signal

If a slave select output is required in SPI master mode, use a general-purpose I/O port and control its output with software.

19.5.3 Data Transfer in I²C Mode



The i2c_scl and i2c_sda pins go to a low level or high impedance status when the USI module is set to I²C mode. These pins do not output a high level, so the i2c_scl and i2c_sda lines should be pulled up to HVDD with an external pull-up resistor.

Note: Be sure to avoid pulling these pins up to a voltage that exceeds the HVDD level.

Figure 19.5.3.1 I²C Mode Connection Example

Control method in I²C master mode

Data transfer in I²C master mode is controlled using IMTGMOD[2:0]/USI_IMTG*x* register and IMTG/USI_IMTG*x* register. Select an I²C master operation using IMTGMOD[2:0] and write 1 to IMTG as the trigger. The I²C controller controls the I²C bus to generate the specified operating status.

Table 19.5.3.1 Trigger List in I²C Master Mode

IMTGMOD[2:0]	Trigger
0x7	Reserved
0x6	ACK/NAK reception
0x5	NAK transmission
0x4	ACK transmission
0x3	Data reception
0x2	Data transmission
0x1	Stop condition
0x0	Start condition

(Default: 0x0)

Writing 1 to IMTG sets IMBSY/USI_IMIF*x* register to 1 indicating that the I²C controller is busy (operating). When the specified operation has finished, IMBSY is reset to 0. At the same time, the interrupt flag (IMIF/USI_IMIF*x* register) is also set to 1. After an interrupt occurs, read the status bits (IMSTA[2:0]/USI_IMIF*x* register) to check the operation finished. Then clear IMIF by writing 1. IMSTA[2:0] will be automatically cleared to 0x0.

Table 19.5.3.2 I²C Master Status Bits

IMSTA[2:0]	Status
0x7	Reserved
0x6	NAK has been received.
0x5	ACK has been received.
0x4	ACK or NAK has been transmitted.
0x3	Receive data buffer is full.
0x2	Transmit data buffer is empty.
0x1	Stop condition has been generated.
0x0	Start condition has been generated.

(Default: 0x0)

Data transmission in I²C master mode

The following describes the data transmission procedure in I²C master mode.

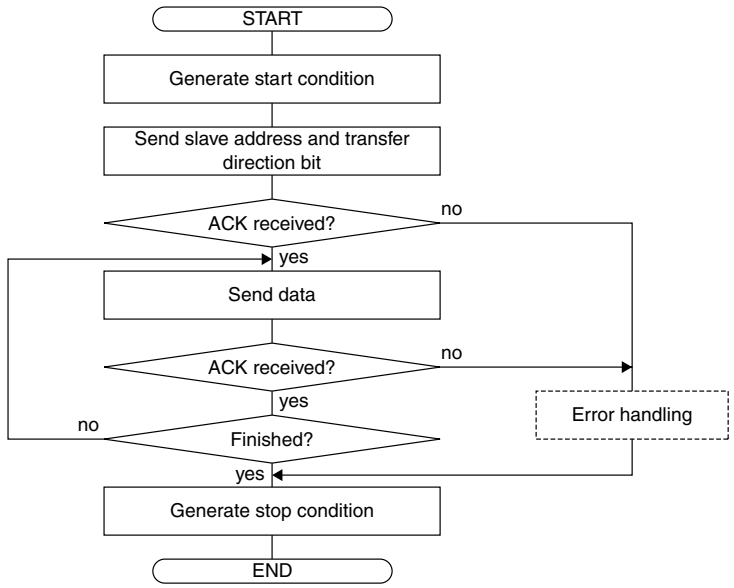
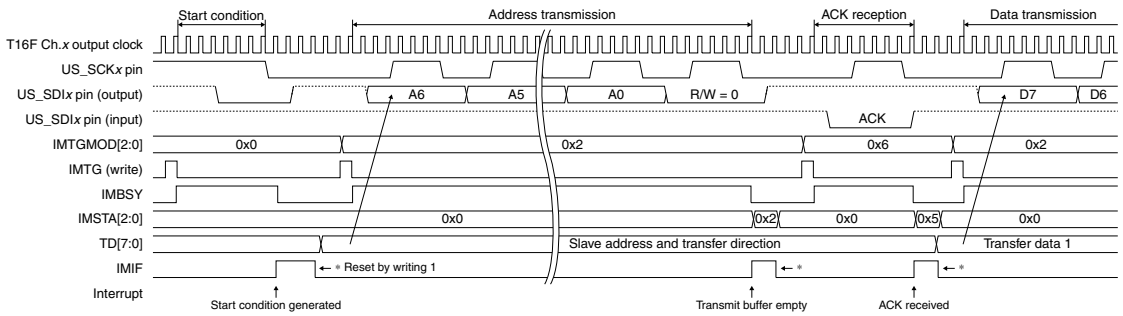
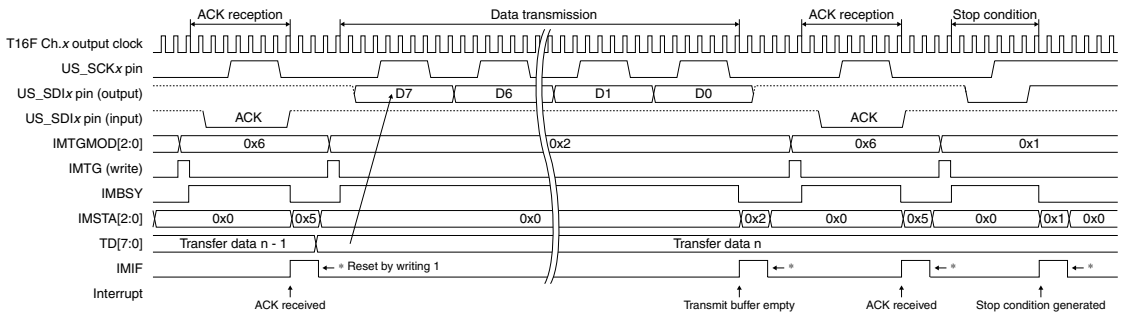


Figure 19.5.3.2 I²C Master Data Transmission Flow Chart



(1) Start condition → Data transmission



(2) Data transmission → Stop condition

Figure 19.5.3.3 I²C Master Data Transmission Timing Chart

(1) Generating start condition

I²C data transfer starts when the I²C master device generates a start condition. The start condition applies when the SCL line is maintained at high and the SDA line is pulled down to low.

To generate a start condition in this I²C master, set IMTGMOD[2:0] to 0x0 (default) and write 1 to IMTG.

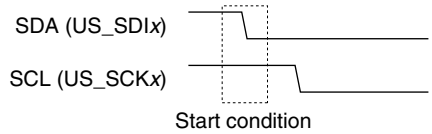


Figure 19.5.3.4 Start Condition

IMBSY is set to 1 while a start condition is being generated. When the start condition is generated, IMBSY is reset to 0 and IMSTA[2:0] is set to 0x0. The I²C bus is busy from this point on.

Note: Other operations cannot be started before a start condition is generated.

(2) Sending slave address and transfer direction bit

After a start condition has been generated, send the address of the slave device to be communicated and a transfer direction bit. I²C slave addresses are either 7-bit or 10-bit. This module uses an 8-bit transfer data buffer to send the slave address and transfer direction bit, enabling single transfers in 7-bit address mode. In 10-bit mode, data is sent twice or three times under software control. Figure 19.5.3.5 shows the configuration of the address data.

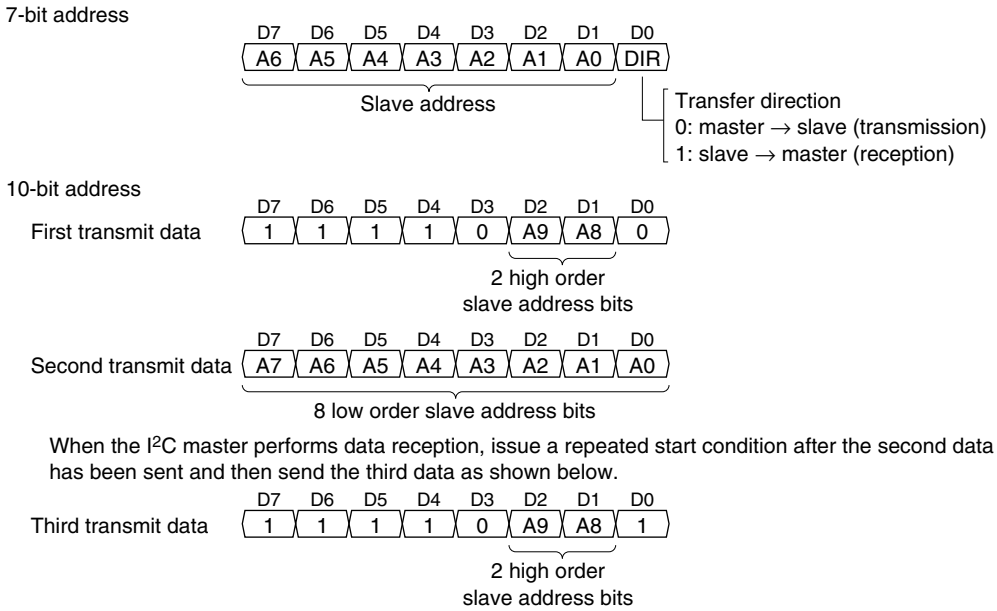


Figure 19.5.3.5 Transmit Data Specifying Slave Address and Transfer Direction

The transfer direction bit indicates the data transfer direction after the slave address has been sent. Set this bit to 0 when sending data from the master to the slave.

To send a slave address, set the address with the transfer direction bit to the transmit data buffer (TD[7:0]/USI_TD*x* register). Then set IMTGMOD[2:0] to 0x2 and write 1 to IMTG.

To send a 10-bit address, execute this procedure twice or three times as shown in Figure 19.5.3.5.

Writing 1 to IMTG sets IMBSY to 1. When data in the transmit data buffer is sent to the transmit shift register, IMBSY reverts to 0 and IMSTA[2:0] is set to 0x2. Confirm that the slave address (each byte) has been sent by reading IMBSY or using an interrupt.

After a slave address has been sent, the selected slave device sends back an ACK by pulling down the SCL line to low. If the SCL line maintains high, it is regarded as a NAK. In this case, the I²C controller cannot communicate with the slave device specified.

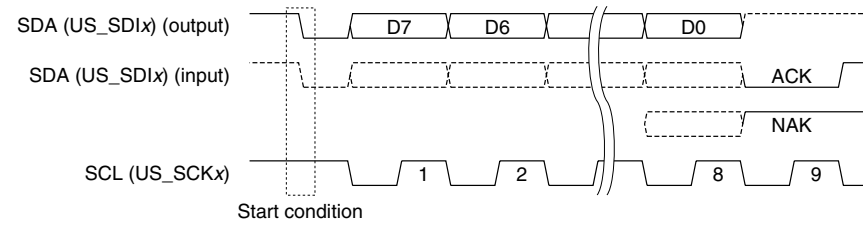


Figure 19.5.3.6 ACK and NAK

It is necessary to check that an ACK has been received before sending data. To do this, set IMTGMOD[2:0] to 0x6 and write 1 to IMTG after the slave address has been sent.
IMBSY is set to 1 while an ACK/NAK is being detected and it reverts to 0 when the detection has completed. Receiving an ACK sets IMSTA[2:0] to 0x5; receiving a NAK sets it to 0x6. Check IMSTA[2:0] after confirming IMBSY or using an interrupt. When an ACK has been received, perform data transmission. When a NAK has been received, perform an error handling.

(3) Data transmission

The data transmission procedure is the same as that of the slave address transmission.
1. Write an 8-bit transmit data to the transmit data buffer (TD[7:0]).
2. Set IMTGMOD[2:0] to 0x2 and IMTG to 1.

This trigger transfers the buffer data to the transmit shift register to start transmission. The module starts clock output from the US_SCK*x* pin. The data in the shift register is shifted in sequence with the clock and sent from the US_SDI*x* pin.
Writing 1 to IMTG sets IMBSY to 1. When data in the transmit data buffer is sent to the transmit shift register, IMBSY reverts to 0 and IMSTA[2:0] is set to 0x2 (transmit data buffer empty). An interrupt request can be generated at this point. Write subsequent data to the transmit data buffer to start the following transmission using this interrupt.
However, as in the case of the slave address transmission, check that the slave device has sent back an ACK (by setting IMTGMOD[2:0] to 0x6 and IMTG to 1) before starting the following 8-bit data transmission.
Repeat an 8-bit data transmission and ACK receiving check for the required number of times.

(4) Generating stop condition

To end I²C communication after all data has been sent, the I²C master must generate a stop condition. The stop condition applies when the SCL line is maintained at high and the SDA line is pulled up from low to high. To generate a stop condition in this I²C master, set IMTGMOD[2:0] to 0x1 and write 1 to IMTG.

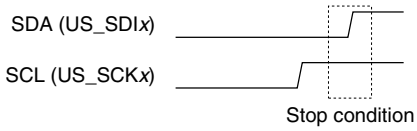


Figure 19.5.3.7 Stop Condition

IMBSY is set to 1 while a stop condition is being generated. When the stop condition is generated, IMBSY is reset to 0 and IMSTA[2:0] is set to 0x1. Read IMBSY or use an interrupt to check that a stop condition has been generated. The I²C bus subsequently switches to free state.

(5) Generating repeated start condition

To make it possible to continue with a different data transfer after a data transmission has completed, the I²C master can omit stop condition generation and generate a repeated start condition. To generate a repeated start condition, perform a start condition generation procedure described in Step (1). Slave address transmission is subsequently possible with the I²C bus remaining in the busy state.

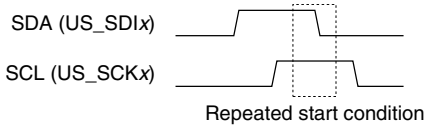


Figure 19.5.3.8 Repeated Start Condition

Data reception in I²C master mode

The following describes the data receiving procedure in I²C master mode.

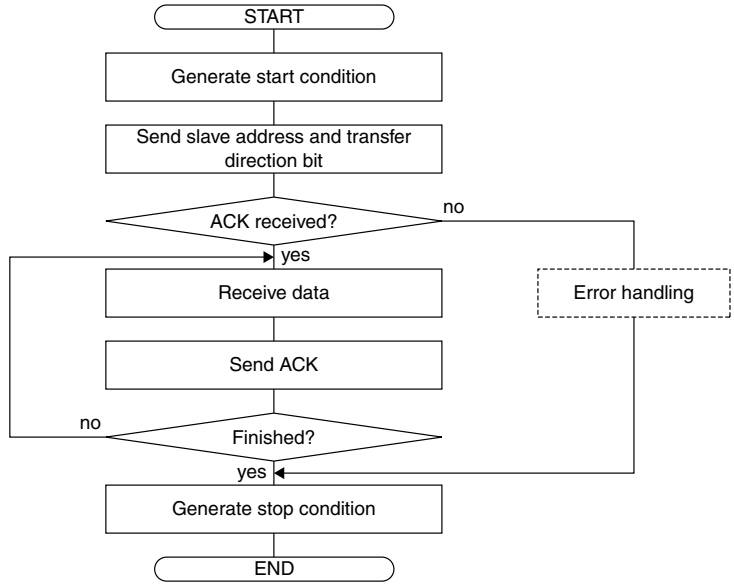
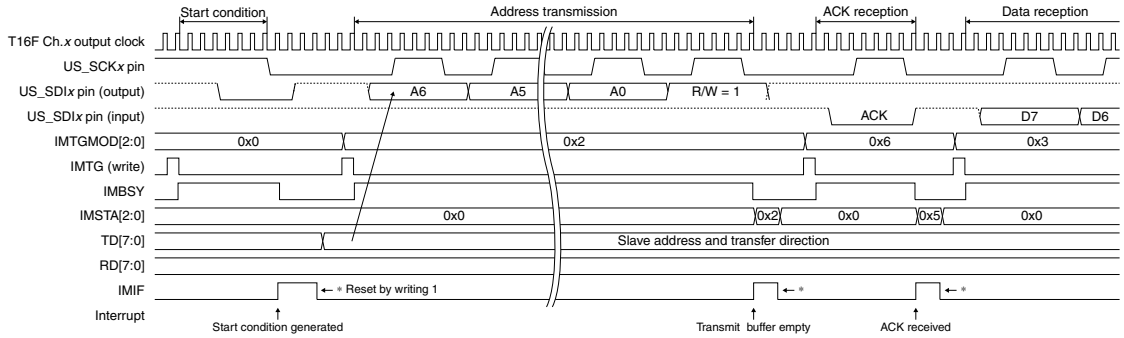
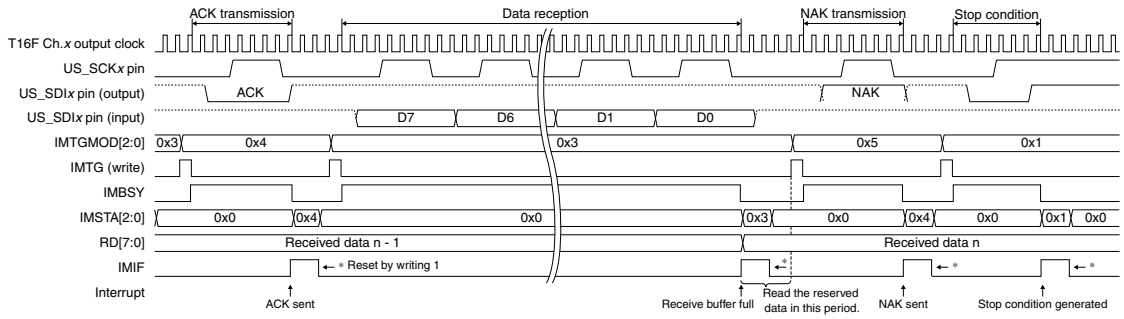


Figure 19.5.3.9 I²C Master Data Receiving Flow Chart]



(1) Start condition → Data reception



(2) Data reception → Stop condition

Figure 19.5.3.10 I²C Master Data Receiving Timing Chart

Note: The timing chart above shows a basic transfer operation that does not include an actual I²C transfer procedure. See “Receiving control byte in I²C slave mode” in “19.9 Precautions.”

(1) Generating start condition
The procedure is the same as that of data transmission in I²C master mode.

(2) Sending slave address and transfer direction bit
The procedure is the same as that of data transmission in I²C master mode. However, send the slave address with the transfer direction bit set to 1. Then check that the slave device sends back an ACK.

(3) Data reception
To start data reception, set IMTGMOD[2:0] to 0x3 and write 1 to IMTG.
This trigger starts outputting 8 clocks from the US_SCK*x* pin. The US_SDI*x* pin status is sampled in sync with the clock and loaded to the shift register. The received data is loaded to the receive data buffer (RD[7:0]/USI_RD*x* register) once the 8-bit data has been received in the shift register.
Writing 1 to IMTG sets IMBSY to 1. When the received data is loaded to the receive data buffer, IMBSY reverts to 0 and IMSTA[2:0] is set to 0x3 (receive data buffer full). An interrupt request can be generated at this point. Read the received data from the receive data buffer using this interrupt.

It is necessary to send back an ACK or NAK to the slave device after an 8-bit data has been received (perform this operation after the received data is read). To send back an ACK, set IMTGMOD[2:0] to 0x4 and write 1 to IMTG. To send back a NAK, set IMTGMOD[2:0] to 0x5 and write 1 to IMTG.
IMBSY is set to 1 while an ACK/NAK is being sent and it reverts to 0 when the transmission has completed. An interrupt request can be generated at this point. When an ACK or NAK has been sent, IMSTA[2:0] is set to 0x4.
Repeat an 8-bit data reception and ACK (NAK) transmission for the required number of times.

(4) Generating stop condition
The procedure is the same as that of data transmission in I²C master mode.

(5) Generating repeated start condition
The procedure is the same as that of data transmission in I²C master mode.

Clock stretch function

During transmitting/receiving data, the slave device may issue a wait request to the master device by pulling down the SCL line to low until the slave device becomes ready to transmit/receive the subsequent data. The master device enters a standby state until the wait request is canceled (the SCL line goes high).
This I²C controller supports this clock stretch function. When a clock stretch condition is detected after a slave address or data has been sent/received, this module enters a waiting status and it does not start operating even if it accepts a trigger for data transfer until the clock stretch status is canceled. IMBSY is maintained at 1 until the triggered operation has completed including a waiting status.

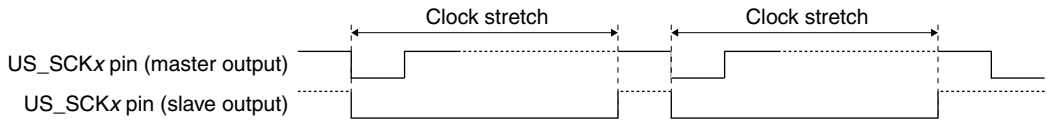


Figure 19.5.3.11 Clock Stretch

Control method in I²C slave mode

Data transfer in I²C slave mode is controlled using ISTGMOD[2:0]/USI_ISTG*x* register and ISTG/USI_ISTG*x* register. Select an I²C slave operation using ISTGMOD[2:0] and write 1 to ISTG as the trigger. The I²C controller controls the I²C bus to generate the specified operating status.

Table 19.5.3.3 Trigger List in I²C Slave Mode

ISTGMOD[2:0]	Trigger
0x7	Reserved
0x6	ACK/NAK reception
0x5	NAK transmission
0x4	ACK transmission
0x3	Data reception
0x2	Data transmission
0x1	Reserved
0x0	Wait for start condition

(Default: 0x0)

Writing 1 to ISTG sets ISBSY/USI_ISIF*x* register to 1 indicating that the I²C controller is busy (operating). When the specified operation has finished, ISBSY is reset to 0. At the same time, the interrupt flag (ISIF/USI_ISIF*x* register) is also set to 1. After an interrupt occurs, read the status bits (ISSTA[2:0]/USI_ISIF*x* register) to check the operation finished. Then, clear ISIF by writing 1. This also automatically reset ISSTA[2:0] to 0x0.

Table 19.5.3.4 I²C Slave Status Bits

ISSTA[2:0]	Status
0x7	Reserved
0x6	NAK has been received.
0x5	ACK has been received.
0x4	ACK or NAK has been transmitted.
0x3	Receive data buffer is full.
0x2	Transmit data buffer is empty.
0x1	Stop condition has been detected.
0x0	Start condition has been detected.

(Default: 0x0)

Data transmission in I²C slave mode

The following describes the data transmission procedure in I²C slave mode.

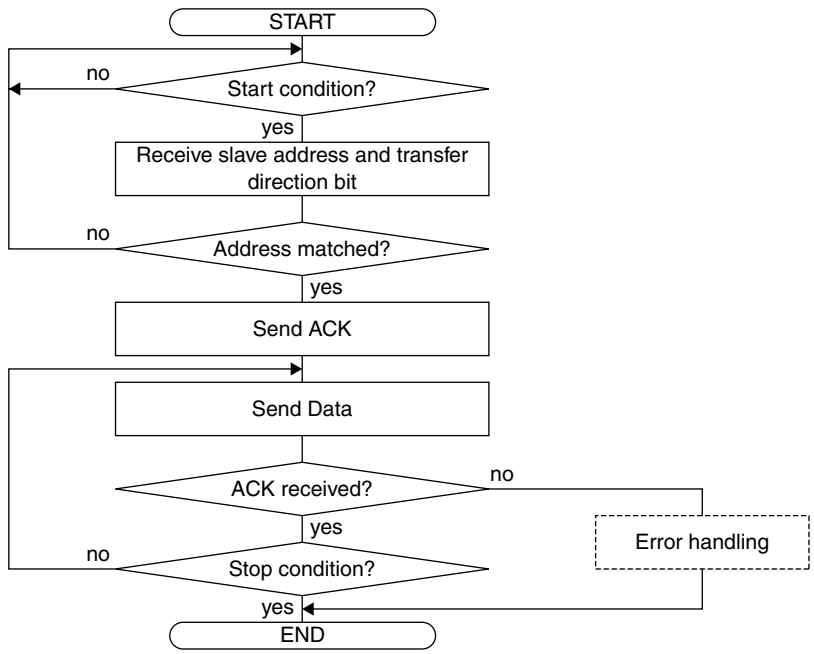


Figure 19.5.3.12 I²C Slave Data Transmission Flow Chart

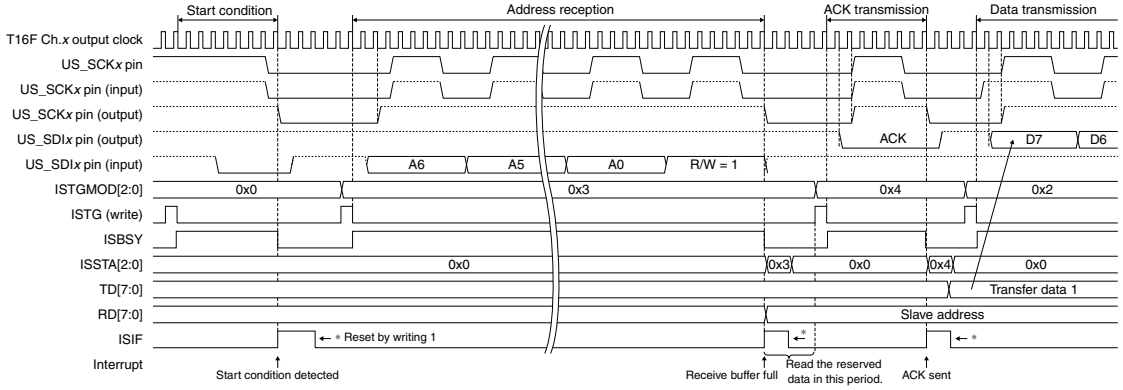


Figure 19.5.3.13 I²C Slave Data Transmission Timing Chart

Note: The timing chart above shows a basic transfer operation that does not include an actual I²C transfer procedure. See "Receiving control byte in I²C slave mode" in "19.9 Precautions."

(1) Waiting for start condition

I²C data transfer starts when the I²C master device generates a start condition (see Figure 19.5.3.4).

First enable this I²C slave to detect a start condition by setting ISTGMOD[2:0] to 0x0 (default) and writing 1 to ISTG. The I²C controller starts detecting a start condition and sets ISBSY to 1. ISBSY is set to 1 while a start condition is being detected. ISBSY reverts to 0 and ISSTA[2:0] is set to 0x0 when the detection has completed. Check if a start condition is generated by reading ISBSY or using an interrupt.

Note: Other operations cannot be started before a start condition is detected.

(2) Receiving slave address and transfer direction data bit

The I²C master sends the address of the slave device to be communicated and a transfer direction bit (see Figure 19.5.3.5) after it has generated a start condition. Set this I²C slave into receiving status to receive the slave address. To start reception, set ISTGMOD[2:0] to 0x3 and write 1 to ISTG.

This trigger starts sampling clocks input from the US_SCK*x* pin. When clocks are input, the I²C controller loads the US_SDI*x* pin status to the shift register in sync with each clock. The received data is loaded to the receive data buffer (RD[7:0]/USI_RD*x* register) once the 8-bit data has been received in the shift register.

Writing 1 to ISTG sets ISBSY to 1. When the received data is loaded to the receive data buffer, ISBSY reverts to 0 and ISSTA[2:0] is set to 0x3 (receive data buffer full). An interrupt request can be generated at this point. Read the received data from the receive data buffer using this interrupt.

When a 7-bit address is used, the slave address and transfer direction bit can be obtained in one operation. When a 10-bit address is used, save the first data received in the receive data buffer into the memory and perform data reception again to obtain the remaining address bits.

Check whether the received address is matched to this I²C slave address or not. When they are matched, send back an ACK to the I²C master by setting ISTGMOD[2:0] to 0x4 and write 1 to ISTG. ISBSY is set to 1 while an ACK is being sent and it reverts to 0 when the transmission has completed. An interrupt request can be generated at this point. When an ACK has been sent, ISSTA[2:0] is set to 0x4.
If the received address is not for this I²C slave, abort data reception and return to Step (1) to wait the subsequent start condition.

(3) Data transmission
When the transfer direction bit received with the slave address in Step (2) is 1, start data transmission by the following procedure:
1. Write an 8-bit transmit data to the transmit data buffer (TD[7:0]).
2. Set ISTGMOD[2:0] to 0x2 and ISTG to 1.

This trigger transfers the buffer data to the transmit shift register to start transmission. When clocks are input from the US_SCK*x* pin, the data in the shift register is shifted in sequence with the clock and sent from the US_SDI*x* pin.
Writing 1 to ISTG sets ISBSY to 1. When data in the transmit data buffer is sent to the transmit shift register, ISBSY reverts to 0 and ISSTA[2:0] is set to 0x2 (transmit data buffer empty). An interrupt request can be generated at this point. Write subsequent data to the transmit data buffer to start the following transmission using this interrupt.
However, check that the master device has sent back an ACK or NAK (by setting ISTGMOD[2:0] to 0x6 and ISTG to 1) before starting the following 8-bit data transmission.
ISBSY is set to 1 while an ACK/NAK is being detected and it reverts to 0 when the detection has completed. Receiving an ACK sets ISSTA[2:0] to 0x5; receiving a NAK sets it to 0x6. Check ISSTA[2:0] after confirming ISBSY or using an interrupt. When an ACK has been received, perform data transmission. When a NAK has been received, perform the appropriate handling.

(4) When a stop condition is received
If the ISSTA[2:0] value read during data transmission is 0x1, the I²C master device has generated a stop condition (see Figure 19.5.3.7). In this case, abort data transmission.
The stop condition can be received only when ISTGMOD[2:0] is set to an operation mode shown below. When the SDA and SCL lines are placed into a stop condition status as shown in Figure 19.5.3.7 after 1 is written to ISTG with one of the operation modes shown below specified, this I²C slave detects it as a stop condition.

ISTGMOD[2:0] = 0x2 (Data transmission)
0x3 (Data reception)
0x5 (NAK transmission)
0x6 (ACK/NAK reception)

Data reception in I²C slave mode

The following describes the data receiving procedure in I²C slave mode.

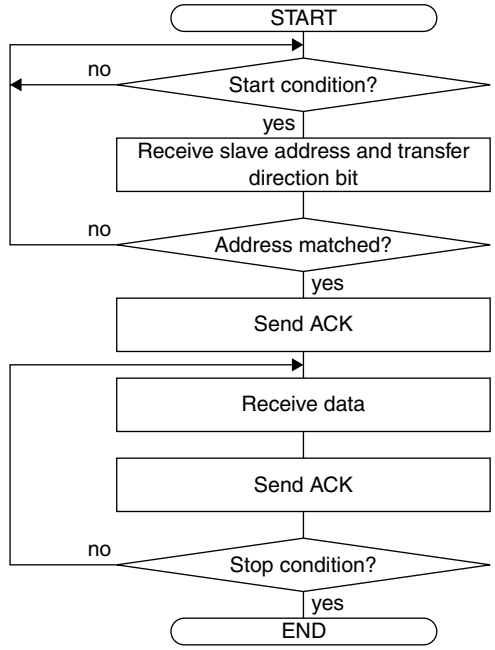
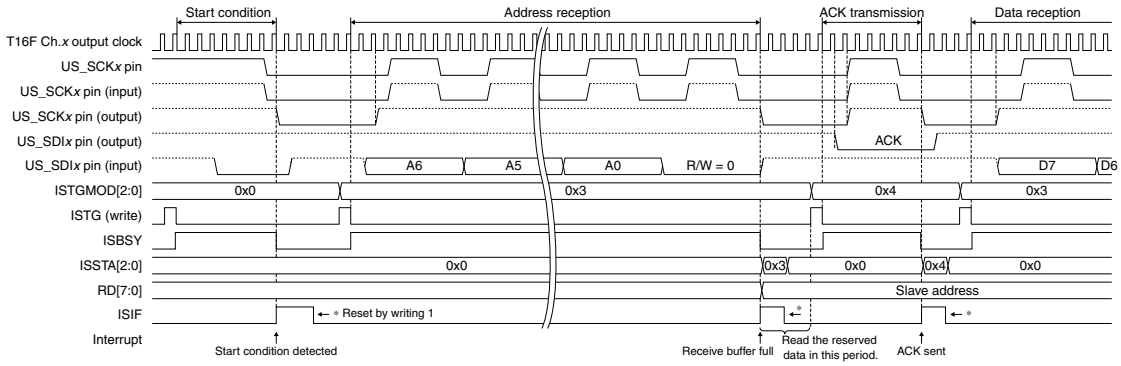
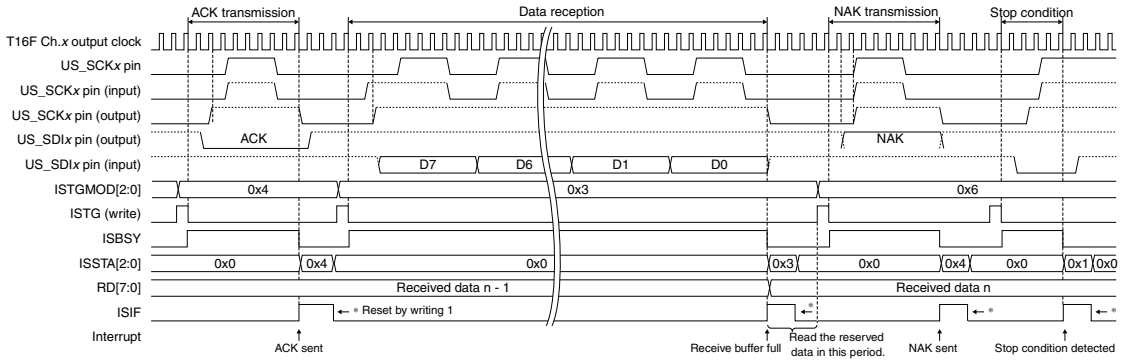


Figure 19.5.3.14 I²C Slave Data Receiving Flow Chart



(1) Start condition → Data reception



(2) Data reception → Stop condition

Figure 19.5.3.15 I²C Slave Data Receiving Timing Chart

Note: The timing chart above shows a basic transfer operation that does not include an actual I²C transfer procedure. See "Receiving control byte in I²C slave mode" in "19.9 Precautions."

(1) Waiting for start condition

The procedure is the same as that of data transmission in I²C slave mode.

(2) Receiving slave address and transfer direction data bit
The procedure is the same as that of data transmission in I²C slave mode.

(3) Data reception
When the transfer direction bit received with the slave address in Step (2) is 0, start data reception by setting ISTGMOD[2:0] to 0x3 and writing 1 to ISTG.
When clocks are input, the I²C controller loads the US_SDI*x* pin status to the shift register in sync with each clock. The received data is loaded to the receive data buffer (RD[7:0]/USI_RD*x* register) once the 8-bit data has been received in the shift register.
Writing 1 to ISTG sets ISBSY to 1. When the received data is loaded to the receive data buffer, ISBSY reverts to 0 and ISSTA[2:0] is set to 0x3 (receive data buffer full). An interrupt request can be generated at this point. Read the received data from the receive data buffer using this interrupt.

It is necessary to send back an ACK or NAK to the master device after an 8-bit data has been received (perform this operation after the received data is read). To send back an ACK, set ISTGMOD[2:0] to 0x4 and write 1 to ISTG. To send back a NAK, set ISTGMOD[2:0] to 0x5 and write 1 to ISTG.
ISBSY is set to 1 while an ACK/NAK is being sent and it reverts to 0 when the transmission has completed. An interrupt request can be generated at this point. When an ACK or NAK has been sent, ISSTA[2:0] is set to 0x4.
Repeat an 8-bit data reception and ACK (NAK) transmission for the required number of times.

(4) When a stop condition is received
If the ISSTA[2:0] value read during data reception is 0x1, the I²C master device has generated a stop condition (see Figure 19.5.3.7). In this case, abort data reception.
The stop condition can be received only when ISTGMOD[2:0] is set to an operation mode shown below. When the SDA and SCL lines are placed into a stop condition status as shown in Figure 19.5.3.7 after 1 is written to ISTG with one of the operation modes shown below specified, this I²C slave detects it as a stop condition.

ISTGMOD[2:0] = 0x2 (Data transmission)
0x3 (Data reception)
0x5 (NAK transmission)
0x6 (ACK/NAK reception)

Clock stretch function

While data is being sent/received, this I²C slave generates a clock stretch status by pulling down the SCL line to low to make a wait request to the master device after an ACK is sent/received until the following data transfer is started.

19.6 Receive Errors

In UART mode, three different receive errors (overrun error, framing error, and parity error) may be detected while receiving data. In SPI master and I²C modes, overrun errors may be detected while receiving data.
Since receive errors are interrupt causes, they can be processed by generating interrupts. For more information on interrupt control, see Section 19.7.

Overrun error (all interface modes)

If data is received before the previously received data in the receive data buffer has not been read, the receive data buffer is overwritten and an overrun error occurs. When an overrun error occurs, the overrun error flag for the current interface mode is set to 1.

Overrun error flags: UOEIF/USI_UIF*x* register (UART mode)
SEIF/USI_SIF*x* register (SPI master mode)
IMEIF/USI_IMIF*x* register (I²C master mode)
ISEIF/USI_ISIF*x* register (I²C slave mode)

The receiving operation continues even if this error occurs. The overrun error flag is reset to 0 by writing 1.

Framing error (UART mode only)

If the stop bit is received as 0 in UART mode, the UART controller determines loss of sync and a framing error occurs. If the stop bit is configured to two bits, only the first bit is checked.
The framing error flag (USEIF/USI_UIF*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 framing error flag is reset to 0 by writing 1.

Parity error (UART mode only)

If UPREN/USI_UCFG*x* register has been set to 1 (parity enabled), data received is checked for parity in UART mode. Data received in the shift register is checked for parity when sent to the receive data buffer. The matching is checked against the UPMD/USI_UCFG*x* register setting (odd or even parity). If the result is a non-match, a parity error is issued, and the parity error flag (UPEIF/USI_UIF*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 UPEIF flag is reset to 0 by writing 1.

19.7 USI Interrupts

This section describes the USI interrupts generated in each interface mode.
For more information on interrupt processing, see the “Interrupt Controller (ITC)” chapter.

Each USI channel outputs one interrupt signal (two signals for two channels) shared by the all interrupt causes to the interrupt controller (ITC). Inspect the interrupt flags available in each mode to determine the interrupt cause occurred.

19.7.1 Interrupts in UART Mode

The UART mode includes a function for generating the following three different types of interrupts.
- Transmit buffer empty interrupt
- Receive buffer full interrupt
- Receive error interrupt

Transmit buffer empty interrupt

To use this interrupt, set UTDIE/USI_UIE*x* register to 1. If UTDIE 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 USI module sets UTDIF/USI_UIF*x* register to 1, indicating that the transmit data buffer is empty. If transmit buffer empty interrupts are enabled (UTDIE = 1), an interrupt request is sent simultaneously to the ITC. An interrupt occurs if other interrupt conditions are met. You can inspect the UTDIF flag in the interrupt handler routine to determine whether the USI (UART mode) interrupt is attributable to a transmit buffer empty. If UTDIF 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 URDIE/USI_UIE*x* register to 1. If URDIE is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.
If a received data is loaded into the receive data buffer, the USI module sets URDIF/USI_UIF*x* register to 1. If receive buffer full interrupts are enabled (URDIE = 1), an interrupt request is sent simultaneously to the ITC. An interrupt occurs if other interrupt conditions are met. You can inspect the URDIF flag in the interrupt handler routine to determine whether the USI (UART mode) interrupt is attributable to a receive buffer full. If URDIF is 1, the received data can be read from the receive data buffer by the interrupt handler routine. However, be sure to check whether a receive error has occurred or not.

Receive error interrupt

To use this interrupt, set UEIE/USI_UIE*x* register to 1. If UEIE is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

The USI module sets an error flag (UPEIF/USI_UIF*x* register, USEIF/USI_UIF*x* register, or UOEIF/USI_UIF*x* register) to 1 if a parity error, framing error, or overrun error is detected when receiving data. If receive error interrupts are enabled (UEIE = 1), an interrupt request is sent simultaneously to the ITC. If other interrupt conditions are satisfied, an interrupt occurs. You can inspect the UPEIF, USEIF, and UOEIF flags in the interrupt handler routine to determine whether the USI (UART mode) 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.

To reset an overrun error, perform USI software reset (write 0x0 to USIMOD[2:0]/USI_GCFG*x* register) to initialize USI.

19.7.2 Interrupts in SPI Master Mode

The SPI master mode include a function for generating the following three different types of interrupts.

- Transmit buffer empty interrupt
- Receive buffer full interrupt
- Receive error interrupt

Transmit buffer empty interrupt

To use this interrupt, set STDIE/USI_SIE*x* register to 1. If STDIE 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 USI module sets STDIF/USI_SIF*x* register to 1, indicating that the transmit data buffer is empty. If transmit buffer empty interrupts are enabled (STDIE = 1), an interrupt request is sent simultaneously to the ITC. An interrupt occurs if other interrupt conditions are met. You can inspect the STDIF flag in the interrupt handler routine to determine whether the USI (SPI master mode) interrupt is attributable to a transmit buffer empty. If STDIF 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 SRDIE/USI_SIE*x* register to 1. If SRDIE is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

If a received data is loaded into the receive data buffer, the USI module sets SRDIF/USI_SIF*x* register to 1. If receive buffer full interrupts are enabled (SRDIE = 1), an interrupt request is sent simultaneously to the ITC. An interrupt occurs if other interrupt conditions are met. You can inspect the SRDIF flag in the interrupt handler routine to determine whether the USI (SPI master mode) interrupt is attributable to a receive buffer full. If SRDIF is 1, the received data can be read from the receive data buffer by the interrupt handler routine. However, be sure to check whether a receive error has occurred or not.

Receive error interrupt

To use this interrupt, set SEIE/USI_SIE*x* register to 1. If SEIE is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

The USI module sets SEIF/USI_SIF*x* register to 1 if an overrun error is detected when receiving data. If receive error interrupts are enabled (SEIE = 1), an interrupt request is sent simultaneously to the ITC. An interrupt occurs if other interrupt conditions are met. You can inspect the SEIF flags in the interrupt handler routine to determine whether the USI (SPI master mode) interrupt was caused by a receive error. If SEIF is 1, the interrupt handler routine will proceed with error recovery.

To reset an overrun error, clear SEIF by writing 1 and then read the receive data buffer (USI_RD*x* register) twice.

19.7.3 Interrupts in I²C Master Mode

The I²C master mode includes a function for generating the following two different types of interrupts.
- Operation completion interrupt
- Receive error interrupt

Operation completion interrupt

To use this interrupt, set IMIE/USI_IMIE*x* register to 1. If IMIE is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

When the operation that initiated by a software trigger has completed, the USI module sets IMIF/USI_IMIF*x* register to 1. If operation completion interrupts are enabled (IMIE = 1), an interrupt request is sent simultaneously to the ITC. An interrupt occurs if other interrupt conditions are met. You can inspect the IMSTA[2:0]/USI_IMIF*x* register in the interrupt handler routine to determine the I²C operation/status that causes the interrupt.

Table 19.7.3.1 I²C Master Status Bits

IMSTA[2:0]	Status
0x7	Reserved
0x6	NAK has been received.
0x5	ACK has been received.
0x4	ACK or NAK has been transmitted.
0x3	Receive data buffer is full.
0x2	Transmit data buffer is empty.
0x1	Stop condition has been generated.
0x0	Start condition has been generated.

(Default: 0x0)

Receive error interrupt

To use this interrupt, set IMEIE/USI_IMIE*x* register to 1. If IMEIE is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

An overrun error occurs at the time a transmit or receive trigger is issued after two-byte data has been received without reading the receive data buffer.

The USI module sets IMEIF/USI_IMIF*x* register to 1 if an overrun error is detected when receiving data. If receive error interrupts are enabled (IMEIE = 1), an interrupt request is sent simultaneously to the ITC. An interrupt occurs if other interrupt conditions are met. You can inspect the IMEIF flags in the interrupt handler routine to determine whether the USI (I²C master mode) interrupt was caused by a receive error. If IMEIF is 1, the interrupt handler routine will proceed with error recovery.

To reset an overrun error, clear IMEIF by writing 1, and then read the receive data buffer (USI_RD*x* register) twice.

19.7.4 Interrupts in I²C Slave Mode

The I²C slave mode includes a function for generating the following two different types of interrupts.
- Operation completion interrupt
- Receive error interrupt

Operation completion interrupt

To use this interrupt, set ISIE/USI_ISIE*x* register to 1. If ISIE is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

When the operation that initiated by a software trigger has completed, the USI module sets ISIF/USI_ISIF*x* register to 1. If operation completion interrupts are enabled (ISIE = 1), an interrupt request is sent simultaneously to the ITC. An interrupt occurs if other interrupt conditions are met. You can inspect the ISSTA[2:0]/USI_ISIF*x* register in the interrupt handler routine to determine the I²C operation/status that causes the interrupt.

Table 19.7.4.1 I²C Slave Status Bits

ISSTA[2:0]	Status
0x7	Reserved
0x6	NAK has been received.
0x5	ACK has been received.
0x4	ACK or NAK has been transmitted.
0x3	Receive data buffer is full.
0x2	Transmit data buffer is empty.
0x1	Stop condition has been detected.
0x0	Start condition has been detected.

(Default: 0x0)

Receive error interrupt

To use this interrupt, set ISEIE/USI_ISIE*x* register to 1. If ISEIE is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

An overrun error occurs at the time a transmit or receive trigger is issued after two-byte data has been received without reading the receive data buffer.

The USI module sets ISEIF/USI_ISIF*x* register to 1 if an overrun error is detected when receiving data. If receive error interrupts are enabled (ISEIE = 1), an interrupt request is sent simultaneously to the ITC. An interrupt occurs if other interrupt conditions are met. You can inspect the ISEIF flags in the interrupt handler routine to determine whether the USI (I²C slave mode) interrupt was caused by a receive error. If ISEIF is 1, the interrupt handler routine will proceed with error recovery.

To reset an overrun error, clear ISEIF by writing 1, and then read the receive data buffer (USI_RD*x* register) twice.

19.8 Control Register Details

Table 19.8.1 List of USI Registers

Address	Register name		Function
0x50c0	USI_GCFG0	USI Ch.0 Global Configuration Register	Sets interface and MSB/LSB mode.
0x50c1	USI_TD0	USI Ch.0 Transmit Data Buffer Register	Transmit data buffer
0x50c2	USI_RD0	USI Ch.0 Receive Data Buffer Register	Receive data buffer
0x50c3	USI_UCFG0	USI Ch.0 UART Mode Configuration Register	Sets UART transfer conditions.
0x50c4	USI_UIE0	USI Ch.0 UART Mode Interrupt Enable Register	Enables interrupts.
0x50c5	USI_UIF0	USI Ch.0 UART Mode Interrupt Flag Register	Indicates interrupt occurrence status.
0x50c6	USI_SCFG0	USI Ch.0 SPI Master Mode Configuration Register	Sets SPI transfer conditions.
0x50c7	USI_SIE0	USI Ch.0 SPI Master Mode Interrupt Enable Register	Enables interrupts.
0x50c8	USI_SIF0	USI Ch.0 SPI Master Mode Interrupt Flag Register	Indicates interrupt occurrence status.
0x50ca	USI_IMTG0	USI Ch.0 I²C Master Mode Trigger Register	Starts I²C master operations.
0x50cb	USI_IMIE0	USI Ch.0 I²C Master Mode Interrupt Enable Register	Enables interrupts.
0x50cc	USI_IMIF0	USI Ch.0 I²C Master Mode Interrupt Flag Register	Indicates interrupt occurrence status.
0x50cd	USI_ISTG0	USI Ch.0 I²C Slave Mode Trigger Register	Starts I²C slave operations.
0x50ce	USI_ISIE0	USI Ch.0 I²C Slave Mode Interrupt Enable Register	Enables interrupts.
0x50cf	USI_ISIF0	USI Ch.0 I²C Slave Mode Interrupt Flag Register	Indicates interrupt occurrence status.
0x50e0	USI_GCFG1	USI Ch.1 Global Configuration Register	Sets interface and MSB/LSB mode.
0x50e1	USI_TD1	USI Ch.1 Transmit Data Buffer Register	Transmit data buffer
0x50e2	USI_RD1	USI Ch.1 Receive Data Buffer Register	Receive data buffer
0x50e3	USI_UCFG1	USI Ch.1 UART Mode Configuration Register	Sets UART transfer conditions.
0x50e4	USI_UIE1	USI Ch.1 UART Mode Interrupt Enable Register	Enables interrupts.
0x50e5	USI_UIF1	USI Ch.1 UART Mode Interrupt Flag Register	Indicates interrupt occurrence status.
0x50e6	USI_SCFG1	USI Ch.1 SPI Master Mode Configuration Register	Sets SPI transfer conditions.
0x50e7	USI_SIE1	USI Ch.1 SPI Master Mode Interrupt Enable Register	Enables interrupts.
0x50e8	USI_SIF1	USI Ch.1 SPI Master Mode Interrupt Flag Register	Indicates interrupt occurrence status.
0x50ea	USI_IMTG1	USI Ch.1 I²C Master Mode Trigger Register	Starts I²C master operations.
0x50eb	USI_IMIE1	USI Ch.1 I²C Master Mode Interrupt Enable Register	Enables interrupts.
0x50ec	USI_IMIF1	USI Ch.1 I²C Master Mode Interrupt Flag Register	Indicates interrupt occurrence status.
0x50ed	USI_ISTG1	USI Ch.1 I²C Slave Mode Trigger Register	Starts I²C slave operations.
0x50ee	USI_ISIE1	USI Ch.1 I²C Slave Mode Interrupt Enable Register	Enables interrupts.
0x50ef	USI_ISIF1	USI Ch.1 I²C Slave Mode Interrupt Flag Register	Indicates interrupt occurrence status.

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

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

USI Ch.*x* Global Configuration Registers (USI_GCFG*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
USI Ch.*x* Global Configuration Register (USI_GCFG*x*)	**0x50c0** **0x50e0** (8 bits)	D7–4	**–**	reserved	–	–	–	0 when being read.
		D3	**LSBFST**	MSB/LSB first mode select	1 MSB first 0 LSB first	0	R/W	
		D2–0	**USIMOD [2:0]**	Interface mode configuration	USIMOD[2:0] / I/F mode: 0x7–0x6 reserved 0x5 I²C slave 0x4 I²C master 0x3 reserved 0x2 SPI master 0x1 UART 0x0 Software reset	0x0	R/W	

Note: This register must be configured before setting other USI registers.

D[7:4] Reserved

D3 LSBFST: MSB/LSB First Mode Select Bit

Selects whether serial data will be transferred from the MSB or LSB.
1 (R/W): MSB first
0 (R/W): LSB first (default)

This setting affects all interface modes.

D[2:0] USIMOD[2:0]: Interface Mode Configuration Bits

Selects an interface mode.

Table 19.8.2 Interface Mode Selection

USIMOD[2:0]	Interface mode
0x7–0x6	Reserved
0x5	I²C slave
0x4	I²C master
0x3	Reserved
0x2	SPI master
0x1	UART
0x0	Software reset

(Default: 0x0)

Perform software reset (set USIMOD[2:0] to 0x0) and then set the interface mode before changing other USI configurations.

USI Ch.*x* Transmit Data Buffer Registers (USI_TD*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
USI Ch.*x* Transmit Data Buffer Register (USI_TD*x*)	**0x50c1** **0x50e1** (8 bits)	D7–0	**TD[7:0]**	USI transmit data buffer TD7 = MSB TD0 = LSB	0x0 to 0xff	0x0	R/W	

D[7:0] TD[7:0]: USI Transmit Data Buffer Bits

Sets transmit data to be written to the transmit data buffer. (Default: 0x0)
In UART and SPI master mode, transmission begins immediately after writing data to this register.
In I²C master/slave mode, transmission begins by the software trigger for data transmission.
The data written to this register is converted into serial data through the shift register and is output from the US_SDO*x*/US_SDI*x* pin with the bit set to 1 as high level and the bit set to 0 as low level.
A transmit buffer empty interrupt can be generated when data written to this register has been transferred to the shift register. The subsequent transmit data can then be written, even while data is being sent.

USI Ch.*x* Receive Data Buffer Registers (USI_RD*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
USI Ch.*x* Receive Data Buffer Register (USI_RD*x*)	**0x50c2** **0x50e2** (8 bits)	D7–0	**RD[7:0]**	USI receive data buffer RD7 = MSB RD0 = LSB	0x0 to 0xff	0x0	R	

D[7:0] RD[7:0]: USI Receive Data Buffer Bits

Contains the received data. (Default: 0x0)

Serial data input from the US_SDI*x* pin is converted to parallel, with the high level bit set to 1 and the low level bit set to 0, and then it is loaded to this register.

A receive buffer full interrupt can be generated when the data received in the shift register has been loaded to this register. 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.

This register is read-only.

USI Ch.*x* UART Mode Configuration Registers (USI_UCFG*x*)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
USI Ch.*x* UART Mode Configuration Register (USI_UCFG*x*)	**0x50c3** **0x50e3** (8 bits)	D7–4	**–**	reserved	–				–	–	0 when being read.
		D3	**UCHLN**	Character length select	1	8 bits	0	7 bits	0	R/W	
		D2	**USTPB**	Stop bit select	1	2 bits	0	1 bit	0	R/W	
		D1	**UPMD**	Parity mode select	1	Even	0	Odd	0	R/W	
		D0	**UPREN**	Parity enable	1	With parity	0	No parity	0	R/W	

Note: This register is effective only in UART mode. Configure the USI channel to UART mode before setting this register.

D[7:4] Reserved

D3 UCHLN: Character Length Select Bit

Selects the serial transfer data length.

1 (R/W): 8 bits

0 (R/W): 7 bits (default)

When 7-bit data length is selected, D7 in the transmit data buffer is ignored and D7 in the receive data buffer is always set to 0.

D2 USTPB: Stop Bit Select Bit

Selects the stop bit length.

1 (R/W): 2 bits

0 (R/W): 1 bit (default)

Writing 1 to USTPB selects 2 stop bits; writing 0 to it selects 1 bit. The start bit is fixed at 1 bit.

D1 UPMD: Parity Mode Select Bit

Selects the parity mode.

1 (R/W): Even parity

0 (R/W): Odd parity (default)

Parity checking and parity bit addition are enabled only when UPREN is set to 1. The UPMD setting is disabled if UPREN is 0.

D0 UPREN: Parity Enable Bit

Enables the parity function.

1 (R/W): With parity

0 (R/W): No parity (default)

UPREN is used to select whether received data parity checking is performed and whether a parity bit is added to transmit data. Setting UPREN to 1 parity-checks the received data. A parity bit is automatically added to the transmit data. If UPREN is set to 0, no parity bit is checked or added.

USI Ch.*x* UART Mode Interrupt Enable Registers (USI_UIE*x*)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
USI Ch.*x* UART Mode Interrupt Enable Register (USI_UIE*x*)	0x50c4 0x50e4 (8 bits)	D7–3	–	reserved	–				–	–	0 when being read.
		D2	UEIE	Receive error interrupt enable	1	Enable	0	Disable	0	R/W	
		D1	URDIE	Receive buffer full interrupt enable	1	Enable	0	Disable	0	R/W	
		D0	UTDIE	Transmit buffer empty int. enable	1	Enable	0	Disable	0	R/W	

Note: This register is effective only in UART mode. Configure the USI channel to UART mode before this register can be used.

D[7:3] Reserved

D2 UEIE: 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.

D1 URDIE: Receive Buffer Full Interrupt Enable Bit

Enables interrupt requests to the ITC when received data is loaded to the receive data buffer.
1 (R/W): Enabled
0 (R/W): Disabled (default)

Set this bit to 1 to read received data using interrupts.

D0 UTDIE: Transmit Buffer Empty Interrupt Enable Bit

Enables interrupt requests to the ITC when data written to 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.

USI Ch.*x* UART Mode Interrupt Flag Registers (USI_UIF*x*)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
USI Ch.*x* UART Mode Interrupt Flag Register (USI_UIF*x*)	0x50c5 0x50e5 (8 bits)	D7	–	reserved	–				–	–	0 when being read.
		D6	URBSY	Receive busy flag	1	Busy	0	Idle	0	R	
		D5	UTBSY	Transmit busy flag	1	Busy	0	Idle	0	R	
		D4	UPEIF	Parity error flag	1	Error	0	Normal	0	R/W	Reset by writing 1.
		D3	USEIF	Framing error flag	1	Error	0	Normal	0	R/W	
		D2	UOEIF	Overrun error flag	1	Error	0	Normal	0	R/W	
		D1	URDIF	Receive buffer full flag	1	Full	0	Not full	0	R/W	
		D0	UTDIF	Transmit buffer empty flag	1	Empty	0	Not empty	0	R/W	

Note: This register is effective only in UART mode. Configure the USI channel to UART mode before this register can be used.

D7 Reserved

D6 URBSY: Receive Busy Flag Bit

Indicates the receive shift register status.
1 (R): Busy
0 (R): Idle (default)

URBSY is set to 1 when the first start bit is detected (when data reception begins) and is reset to 0 when the data received in the shift register is loaded into the receive data buffer. Inspect URBSY to determine whether the receiving circuit is operating or at standby.

D5 UTBSY: Transmit Busy Flag Bit

Indicates the transmit shift register status.
1 (R): Busy
0 (R): Idle (default)

UTBSY 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 UTBSY to determine whether the transmit circuit is operating or at standby.

D4 UPEIF: 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

UPEIF is set to 1 when a parity error occurs. At the same time a receive error interrupt request is sent to the ITC if UEIE/USI_UIE*x* register is 1. Parity checking is enabled only when UPREN/USI_UCFG*x* register is set to 1 and is performed when received data is transferred from the shift register to the receive data buffer. UPEIF is reset by writing 1.

D3 USEIF: 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

USEIF is set to 1 when a framing error occurs. At the same time a receive error interrupt request is sent to the ITC if UEIE/USI_UIE*x* register is 1. A framing error occurs when data is received with the stop bit set to 0. USEIF is reset by writing 1.

D2 UOEIF: 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

UOEIF is set to 1 when an overrun error occurs. At the same time a receive error interrupt request is sent to the ITC if UEIE/USI_UIE*x* register is 1. An overrun error occurs when the previous received data in the receive data buffer before reading is overwritten with a new received data. To reset UOEIF, perform USI software reset (write 0x0 to USIMOD[2:0]/USI_GCFG*x* register) to initialize USI.

D1 URDIF: Receive Buffer Full Flag Bit

Indicates the receive data buffer status.

1 (R): Data full
0 (R): No data (default)
1 (W): Reset to 0
0 (W): Ignored

URDIF 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. At the same time a receive buffer full interrupt request is sent to the ITC if URDIE/USI_UIE*x* register is 1. URDIF is reset by writing 1.

D0 UTDIF: Transmit Data Buffer Empty Flag Bit

Indicates the transmit data buffer status.

1 (R): Empty (default)
0 (R): Data exists
1 (W): Reset to 0
0 (W): Ignored

UTDIF is set to 1 when the transmit data written to the transmit data buffer is transferred to the shift register (when transmission starts), indicating that the next transmit data can be written to. At the same time a transmit buffer empty interrupt request is sent to the ITC if UTDIE/USI_UIE*x* register is 1. UTDIF is reset by writing 1.

USI Ch.*x* SPI Master Mode Configuration Registers (USI_SCFG*x*)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
USI Ch.*x* SPI Master Mode Configuration Register (USI_SCFG*x*)	0x50c6 0x50e6 (8 bits)	D7–6	–	reserved	–				–	–	0 when being read.
		D5	SCMD	Command bit (for 9-bit data)	1	High	0	Low	0	R/W	
		D4	SCHLN	Character length select	1	9 bits	0	8 bits	0	R/W	
		D3	SCPHA	Clock phase select	1	Phase 1	0	Phase 0	0	R/W	
		D2	SCPOL	Clock polarity select	1	Active L	0	Active H	0	R/W	
		D1	–	reserved	–				–	–	0 when being read.
		D0	SFSTMOD	Fast mode select	1	Fast	0	Normal	0	R/W	

Note: This register is effective only in SPI master mode. Configure the USI channel to SPI master mode before this register can be used.

D[7:6] Reserved

D5 SCMD: Command Bit (for 9-bit data)

Sets the command bit value for 9-bit data (see SCHLN below).
1 (R/W): High
0 (R/W): Low (default)

D4 SCHLN: Character Length Select Bit

Selects the serial transfer data length.
1 (R/W): 9 bits
0 (R/W): 8 bits (default)

In 9-bit mode, 8-bit data is prefixed with a command bit (1 bit). The command bit is used for controlling the SPI LCD controller connected to the USI. The command bit value to be transmitted can be specified using SCMD.

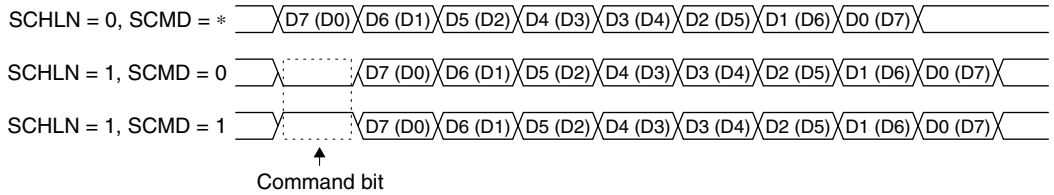


Figure 19.8.1 9-bit Transfer Data Format in SPI Master Mode

D3 SCPHA: Clock Phase Select Bit

Selects the SPI clock phase.
1 (R/W): Phase 1
0 (R/W): Phase 0 (default)

Set the data transfer timing together with SCPOL. (See Figure 19.8.2.)

D2 SCPOL: 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 SCPHA. (See Figure 19.8.2.)

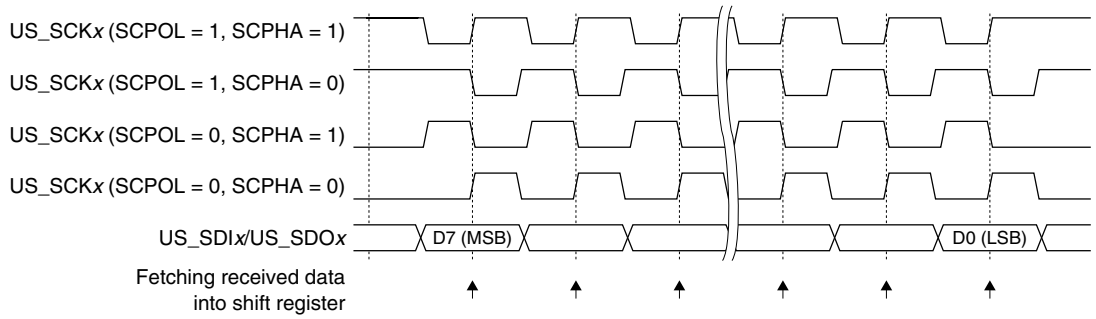


Figure 19.8.2 Clock and Data Transfer Timing

D1 Reserved

D0 SFSTMOD: Fast Mode Select Bit

Selects Fast mode.
1 (R/W): Fast mode
0 (R/W): Normal mode (default)

Either normal or fast clock mode can be selected using SFSTMOD. Setting SFSTMOD to 0 (default) places the USI into normal mode and the USI generates the transfer clock by dividing the T16F output by 2. Setting SFSTMOD to 1 places the USI into fast mode and the USI uses PCLK supplied from the CLG directly as the transfer clock. The fast mode does not use the T16F.

USI Ch.*x* SPI Master Mode Interrupt Enable Registers (USI_SIE*x*)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
USI Ch.*x* SPI Master Mode Interrupt Enable Register (USI_SIE*x*)	0x50c7 0x50e7 (8 bits)	D7–3	–	reserved	–				–	–	0 when being read.
		D2	SEIE	Receive error interrupt enable	1	Enable	0	Disable	0	R/W	
		D1	SRDIE	Receive buffer full interrupt enable	1	Enable	0	Disable	0	R/W	
		D0	STDIE	Transmit buffer empty int. enable	1	Enable	0	Disable	0	R/W	

Note: This register is effective only in SPI master mode. Configure the USI channel to SPI master mode before this register can be used.

D[7:3] Reserved

D2 SEIE: Receive Error Interrupt Enable Bit

Enables interrupt requests to the ITC when an overrun error occurs.
1 (R/W): Enabled
0 (R/W): Disabled (default)

Set this bit to 1 to process overrun errors using interrupts.

D1 SRDIE: Receive Buffer Full Interrupt Enable Bit

Enables interrupt requests to the ITC when received data is loaded to the receive data buffer.
1 (R/W): Enabled
0 (R/W): Disabled (default)

Set this bit to 1 to read received data using interrupts.

D0 STDIE: Transmit Buffer Empty Interrupt Enable Bit

Enables interrupt requests to the ITC when data written to 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.

USI Ch.*x* SPI Master Mode Interrupt Flag Registers (USI_SIF*x*)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
USI Ch.*x* SPI Master Mode Interrupt Flag Register (USI_SIF*x*)	0x50c8 0x50e8 (8 bits)	D7–3	–	reserved	–				–	–	0 when being read.
		D2	SEIF	Overrun error flag	1	Error	0	Normal	0	R/W	Reset by writing 1.
		D1	SRDIF	Receive buffer full flag	1	Full	0	Not full	0	R/W	
		D0	STDIF	Transmit buffer empty flag	1	Empty	0	Not empty	0	R/W	

Note: This register is effective only in SPI master mode. Configure the USI channel to SPI master mode before this register can be used.

D[7:3] Reserved

D2 SEIF: 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

SEIF is set to 1 when an overrun error occurs. At the same time a receive error interrupt request is sent to the ITC if SEIE/USI_SIE*x* register is 1. An overrun error occurs when the previous received data in the receive data buffer before reading is overwritten with a new received data. When data reception for one byte has completed, the received data is loaded into the receive data buffer (USI_RD*x* register). If the second byte of data is received before the buffer data is read out, the second byte data remains in the shift register. An overrun error occurs if the third byte data is received in this condition, as the second byte data in the shift register is corrupted (an overrun error occurs at the time the first bit of the third byte is fetched).
SEIF is reset by writing 1. To reset an overrun error, write 1 to SEIF and then read the receive data buffer (USI_RD*x* register) twice. The procedure that writes 1 to SEIF and reads USI_RD*x* register twice can be reversed.

D1 SRDIF: Receive Buffer Full Flag Bit

Indicates the receive data buffer status.

1 (R): Data full
0 (R): No data (default)
1 (W): Reset to 0
0 (W): Ignored

SRDIF 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. At the same time a receive buffer full interrupt request is sent to the ITC if SRDIE/USI_SIE*x* register is 1. SRDIF is reset by writing 1.

D0 STDIF: Transmit Data Buffer Empty Flag Bit

Indicates the transmit data buffer status.

1 (R): Empty (default)
0 (R): Data exists
1 (W): Reset to 0
0 (W): Ignored

STDIF is set to 1 when the transmit data written to the transmit data buffer is transferred to the shift register (when transmission starts), indicating that the next transmit data can be written to. At the same time a transmit buffer empty interrupt request is sent to the ITC if STDIE/USI_SIE*x* register is 1. STDIF is reset by writing 1.

USI Ch.*x* I²C Master Mode Trigger Registers (USI_IMTG*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
USI Ch.*x* I²C Master Mode Trigger Register (USI_IMTG*x*)	0x50ca 0x50ea (8 bits)	D7–5	–	reserved	–	–	–	0 when being read.
		D4	IMTG	I²C master operation trigger	1 Trigger 0 Ignored 1 Waiting 0 Finished	0	W R	
		D3	–	reserved	–	–	–	0 when being read.
		D2–0	IMTGMOD [2:0]	I²C master trigger mode select	IMTGMOD[2:0] Trigger mode 0x7 reserved 0x6 Receive ACK/NAK 0x5 Transmit NAK 0x4 Transmit ACK 0x3 Receive data 0x2 Transmit data 0x1 Stop condition 0x0 Start condition	0x0	R/W	

Note: This register is effective only in I²C master mode. Configure the USI channel to I²C master mode before this register can be used.

D[7:5] Reserved

D4 IMTG: I²C Master Operation Trigger Bit

Starts an I²C master operation.

1 (W): Trigger
0 (W): Ignored
1 (R): Waiting for starting operation
0 (R): Trigger has finished (default)

Select an I²C master operation using IMTGMOD[2:0] and write 1 to IMTG as the trigger. The I²C controller controls the I²C bus to generate the specified operating status.

D3 Reserved

D[2:0] IMTGMOD[2:0]: I²C Master Trigger Mode Select Bits

Selects an I²C master operation.

Table 19.8.3 Trigger List in I²C Master Mode

IMTGMOD[2:0]	Trigger
0x7	Reserved
0x6	ACK/NAK reception
0x5	NAK transmission
0x4	ACK transmission
0x3	Data reception
0x2	Data transmission
0x1	Stop condition
0x0	Start condition

(Default: 0x0)

USI Ch.*x* I²C Master Mode Interrupt Enable Registers (USI_IMIE*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
USI Ch.*x* I²C Master Mode Interrupt Enable Register (USI_IMIE*x*)	0x50cb 0x50eb (8 bits)	D7–2	–	reserved	–	–	–	0 when being read.
		D1	IMEIE	Receive error interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	IMIE	Operation completion int. enable	1 Enable 0 Disable	0	R/W	

Note: This register is effective only in I²C master mode. Configure the USI channel to I²C master mode before this register can be used.

D[7:2] Reserved

D1 IMEIE: Receive Error Interrupt Enable Bit

Enables interrupt requests to the ITC when an overrun error occurs.

1 (R/W): Enabled
0 (R/W): Disabled (default)

Set this bit to 1 to process overrun errors using interrupts.

D0 IMIE: Operation Completion Interrupt Enable Bit

Enables interrupt requests to the ITC when the triggered operation has completed.

1 (R/W): Enabled
0 (R/W): Disabled (default)

Set this bit to 1 to confirm whether the triggered operation has completed or not using interrupts.

USI Ch.*x* I²C Master Mode Interrupt Flag Registers (USI_IMIF*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
USI Ch.*x* I²C Master Mode Interrupt Flag Register (USI_IMIF*x*)	0x50cc 0x50ec (8 bits)	D7–6	–	reserved	–	–	–	0 when being read.
		D5	IMBSY	I²C master busy flag	1 Busy 0 Standby	0	R	
		D4–2	IMSTA[2:0]	I²C master status	IMSTA[2:0] Status 0x7 reserved 0x6 NAK received 0x5 ACK received 0x4 ACK/NAK sent 0x3 Rx buffer full 0x2 Tx buffer empty 0x1 Stop generated 0x0 Start generated	0x0	R	
		D1	IMEIF	Overrun error flag	1 Error 0 Normal	0	R/W	Reset by writing 1.
		D0	IMIF	Operation completion flag	1 Completed 0 Not completed	0	R/W	

Note: This register is effective only in I²C master mode. Configure the USI channel to I²C master mode before this register can be used.

D[7:6] Reserved

D5 IMBSY: I²C Master Busy Flag Bit

Indicates the I²C master operation status.

1 (R): Busy

0 (R): Standby (default)

Writing 1 to IMTG/USI_IMTG*x* register (starting an I²C master operation) sets IMBSY to 1 indicating that the I²C controller is busy (operating). When the specified operation has finished, IMBSY is reset to 0.

D[4:2] IMSTA[2:0]: I²C Master Status Bits

Indicates the I²C master status.

Table 19.8.4 I²C Master Status Bits

IMSTA[2:0]	Status
0x7	Reserved
0x6	NAK has been received.
0x5	ACK has been received.
0x4	ACK or NAK has been transmitted.
0x3	Receive data buffer is full.
0x2	Transmit data buffer is empty.
0x1	Stop condition has been generated.
0x0	Start condition has been generated.

(Default: 0x0)

When an operation completion interrupt occurs, read IMSTA[2:0] to check the operation that has been finished. IMSTA[2:0] is automatically reset to 0x0 by writing 1 to IMIF.

D1 IMEIF: 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

IMEIF is set to 1 when an overrun error occurs. At the same time a receive error interrupt request is sent to the ITC if IMEIE/USI_IMIE*x* register is 1.

An overrun error occurs when a transmit or receive trigger is issued after two-byte data has been received without the receive data buffer being read.

IMEIF is reset by writing 1.

To reset an overrun error, clear IMEIF by writing 1, and then read the receive data buffer (USI_RD*x* register) twice.

D0 IMIF: Operation Completion Flag Bit

Indicates whether the triggered operation has completed or not.

1 (R): Completed
0 (R): Not completed (default)
1 (W): Reset to 0
0 (W): Ignored

IMIF is set to 1 when the operation that is specified and triggered using the USI_IMTG*x* register has completed. At the same time an operation completion interrupt request is sent to the ITC if IMIE/USI_IMIE*x* register is 1. IMIF is reset by writing 1.

USI Ch.*x* I²C Slave Mode Trigger Registers (USI_ISTG*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
USI Ch.*x* I²C Slave Mode Trigger Register (USI_ISTG*x*)	0x50cd 0x50ed (8 bits)	D7–5	–	reserved	–	–	–	0 when being read.
		D4	ISTG	I²C slave operation trigger	1 Trigger / 0 Ignored 1 Waiting / 0 Finished	0	W R	
		D3	–	reserved	–	–	–	0 when being read.
		D2–0	ISTGMOD [2:0]	I²C slave trigger mode select	ISTGMOD[2:0] / Trigger mode 0x7 reserved 0x6 Receive ACK/NAK 0x5 Transmit NAK 0x4 Transmit ACK 0x3 Receive data 0x2 Transmit data 0x1 reserved 0x0 Wait for start	0x0	R/W	

Note: This register is effective only in I²C slave mode. Configure the USI channel to I²C slave mode before this register can be used.

D[7:5] Reserved

D4 ISTG: I²C Slave Operation Trigger Bit

Starts an I²C slave operation.

1 (W): Trigger
0 (W): Ignored
1 (R): Waiting for starting operation
0 (R): Trigger has finished (default)

Select an I²C slave operation using ISTGMOD[2:0] and write 1 to ISTG as the trigger. The I²C controller controls the I²C bus to generate the specified operating status.

D3 Reserved

D[2:0] ISTGMOD[2:0]: I²C Slave Trigger Mode Select Bits

Selects an I²C slave operation.

Table 19.8.5 Trigger List in I²C Slave Mode

ISTGMOD[2:0]	Trigger
0x7	Reserved
0x6	ACK/NAK reception
0x5	NAK transmission
0x4	ACK transmission
0x3	Data reception
0x2	Data transmission
0x1	Reserved
0x0	Wait for start condition

(Default: 0x0)

USI Ch.*x* I²C Slave Mode Interrupt Enable Registers (USI_ISIE*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
USI Ch.*x* I²C Slave Mode Interrupt Enable Register (USI_ISIE*x*)	0x50ce 0x50ee (8 bits)	D7–2	–	reserved	–	–	–	0 when being read.
		D1	ISEIE	Receive error interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	ISIE	Operation completion int. enable	1 Enable 0 Disable	0	R/W	

Note: This register is effective only in I²C slave mode. Configure the USI channel to I²C slave mode before this register can be used.

D[7:2] Reserved

D1 ISEIE: Receive Error Interrupt Enable Bit

Enables interrupt requests to the ITC when an overrun error occurs.

1 (R/W): Enabled

0 (R/W): Disabled (default)

Set this bit to 1 to process overrun errors using interrupts.

D0 ISIE: Operation Completion Interrupt Enable Bit

Enables interrupt requests to the ITC when the triggered operation has completed.

1 (R/W): Enabled

0 (R/W): Disabled (default)

Set this bit to 1 to confirm whether the triggered operation has completed or not using interrupts.

USI Ch.*x* I²C Slave Mode Interrupt Flag Registers (USI_ISIF*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
USI Ch.*x* I²C Slave Mode Interrupt Flag Register (USI_ISIF*x*)	0x50cf 0x50ef (8 bits)	D7–6	–	reserved	–	–	–	0 when being read.
		D5	ISBSY	I²C slave busy flag	1 Busy 0 Standby	0	R	
		D4–2	ISSTA[2:0]	I²C slave status	ISSTA[2:0] Status: 0x7 reserved; 0x6 NAK received; 0x5 ACK received; 0x4 ACK/NAK sent; 0x3 Rx buffer full; 0x2 Tx buffer empty; 0x1 Stop detected; 0x0 Start detected	0x0	R	
		D1	ISEIF	Overrun error flag	1 Error 0 Normal	0	R/W	Reset by writing 1.
		D0	ISIF	Operation completion flag	1 Completed 0 Not completed	0	R/W	

Note: This register is effective only in I²C slave mode. Configure the USI channel to I²C slave mode before this register can be used.

D[7:6] Reserved

D5 ISBSY: I²C Slave Busy Flag Bit

Indicates the I²C slave operation status.

1 (R): Busy

0 (R): Standby (default)

Writing 1 to ISTG/USI_ISTG*x* register (starting an I²C slave operation) sets ISBSY to 1 indicating that the I²C controller is busy (operating). When the specified operation has finished, ISBSY is reset to 0.

D[4:2] ISSTA[2:0]: I²C Slave Status Bits

Indicates the I²C slave status.

Table 19.8.6 I²C Slave Status Bits

ISSTA[2:0]	Status
0x7	Reserved
0x6	NAK has been received.
0x5	ACK has been received.
0x4	ACK or NAK has been transmitted.
0x3	Receive data buffer is full.
0x2	Transmit data buffer is empty.
0x1	Stop condition has been detected.
0x0	Start condition has been detected.

(Default: 0x0)

When an operation completion interrupt occurs, read ISSTA[2:0] to check the operation that has been finished. ISSTA[2:0] is automatically reset to 0x0 by writing 1 to ISIF.

D1 ISEIF: 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

ISEIF is set to 1 when an overrun error occurs. At the same time a receive error interrupt request is sent to the ITC if ISEIE/USI_ISIE*x* register is 1. An overrun error occurs at the time a transmit or receive trigger is issued after two-byte data has been received without reading the receive data buffer.
ISEIF is reset by writing 1.
To reset an overrun error, clear ISEIF by writing 1, and then read the receive data buffer (USI_RD*x* register) twice.

D0 ISIF: Operation Completion Flag Bit

Indicates whether the triggered operation has completed or not.

1 (R): Completed
0 (R): Not completed (default)
1 (W): Reset to 0
0 (W): Ignored

ISIF is set to 1 when the operation that is specified and triggered using the USI_ISTG*x* register has completed. At the same time an operation completion interrupt request is sent to the ITC if ISIE/USI_ISIE*x* register is 1. ISIF is reset by writing 1.

19.9 Precautions

Interface mode setting

Be sure to perform software reset (USIMOD[2:0]/USI_GCFG*x* register = 0x0) and set the interface mode (USIMOD[2:0]/USI_GCFG*x* register = 0x1, 0x2, 0x4, or 0x5) before changing other USI configurations.

Busy flags

The busy flags listed below may be set with delay. When checking the busy status after performing an operation that sets the busy flag, wait for at least one T16F output clock cycle before reading the flag. If the busy flag is read with no wait time inserted, the flag may not indicate the current status properly.

Table 19.9.1 Busy Flags and Delay Conditions

Interface mode	Busy flag	Timing with delay occurred
UART mode	UTBSY/USI_UIF*x* register	After transmit data is written to the transmit data buffer
I²C master mode	IMBSY/USI_IMIF*x* register	After the trigger bit is set
I²C slave mode	ISBSY/USI_ISIF*x* register	After the trigger bit is set

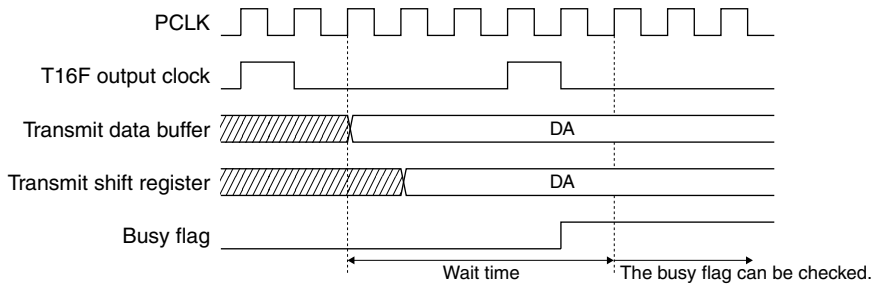


Figure 19.9.1 Waiting Before Reading Busy Flag

Receiving control byte in I²C slave mode

The external I²C master device sends a control byte to the I²C slave device when an ACK has been received after sending a slave address. The subsequent operations of the slave device are determined by the control byte.

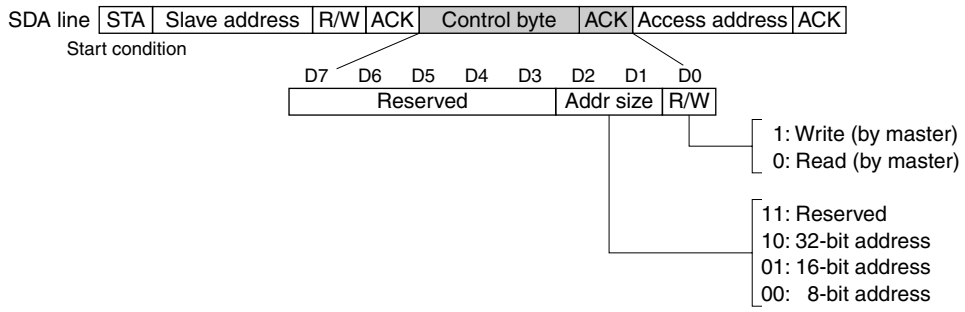


Figure 19.9.2 Control Byte Sent from I²C Master

I²C master write (data receiving from master)

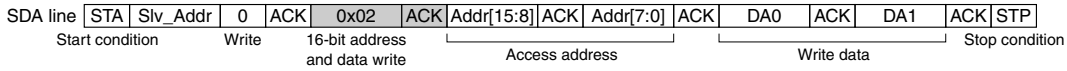


Figure 19.9.3 I²C Master Write (Data Receiving from Master)

The control byte specifies the access address size and writing operations. The received data that follow the control byte should be used as the address and the data to be written according to the access address size.

I²C master read (data transmission to master)

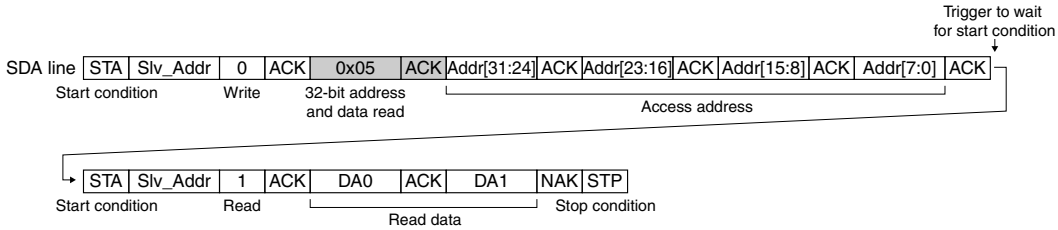


Figure 19.9.4 I²C Master Read (Data Transmission to Master)

The master sends the access address following the control byte. Perform data reception for the control byte and address data to determine the address from which transmit data is read. After sending an ACK for Addr 0, set ISTGMOD[2:0]/USI_ISTG*x* register to 0x0 and ISTG/USI_ISTG*x* register to 1 to wait for a start condition that will be sent from the master for reading data (for the slave to sent the read data).

20 IR Remote Controller (REMC)

20.1 REMC Module Overview

The S1C17554/564 includes an IR remote controller (REMC) module for transmitting/receiving infrared remote control communication signals.
The following shows the features of the REMC module:

- Supports input and output infrared remote control communication signals.
- Includes a carrier generator for generating a carrier signal.
- Includes an 8-bit down-counter for counting the transfer data length.
- Includes a modulator for generating transmission data of the specified carrier length.
- Includes an edge detector for detecting input signal rising and falling edges.
- Can generate counter underflow interrupts indicating that the specified data length has been sent and input rising/falling edge detection interrupts for data receive processing.

Figure 20.1.1 shows the configuration of the REMC module.

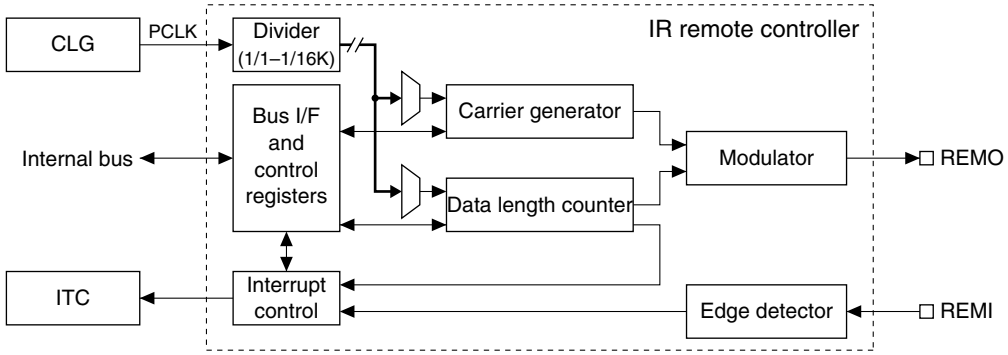


Figure 20.1.1 REMC Module Configuration

20.2 REMC Input/Output Pins

Table 20.2.1 lists the REMC input/output pins.

Table 20.2.1 List of REMC Pins

Pin name	I/O	Qty	Function
REMI	I	1	Remote control receive data input pin Inputs receive data.
REMO	O	1	Remote control transmit data output pin Outputs modulated remote control transmit data.

The REMC input/output pins (REMI, REMO) 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 REMC input/output pins.
For detailed information on pin function switching, see the "I/O Ports (P)" chapter.

20.3 Carrier Generation

The REMC module includes a carrier generator that generates a carrier signal for transmission in accordance with the clock set by software and carrier H and L section lengths.

The carrier generation clock is generated by dividing PCLK into 1/1 to 1/16K. The division ratio can be selected from the 15 types shown below using CGCLK[3:0]/REMC_CFG register.

Table 20.3.1 Carrier Generation Clock (PCLK Division Ratio) Selection

CGCLK[3:0]	Division ratio	CGCLK[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 REMC.

• Make sure the REMC is halted before setting the clock.

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

The carrier H and L section lengths are set by REMCH[5:0]/REMC_CAR register and REMCL[5:0]/REMC_CAR register, respectively. Set a value corresponding to the number of clock (selected as above) cycles + 1 to these registers.
The carrier H and L section lengths can be calculated as follows:

$$\text{Carrier H section length} = \frac{\text{REMCH} + 1}{\text{cg\_clk}} \text{ [s]}$$

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

REMCH: Carrier H section length data value
REMCL: Carrier L section length data value
cg_clk: Carrier generation clock frequency

The carrier signal is generated from these settings as shown in Figure 20.3.1.

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

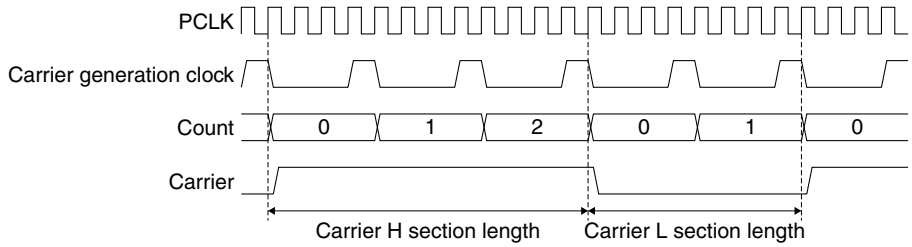


Figure 20.3.1 Carrier Signal Generation

20.4 Data Length Counter Clock Settings

The data length counter is an 8-bit counter for setting data lengths when transmitting data.
When a value corresponding to the data pulse width is written during data transmission, the data length counter begins counting down from that value and stops after generating an underflow interrupt cause when the counter reaches 0. The subsequent transmit data is set using this interrupt.

This counter is also used for data receiving, enabling measurement of the received data length. Interrupts can be generated at the input signal rising or falling edges when receiving data. The data pulse length can be obtained from the difference between data pulse edges by setting the data length counter to 0xff using the interrupt when the input changes and by reading out the count value when a subsequent interrupt occurs due to input changes.

This data length counter clock also uses a divided PCLK clock and can select one of 15 different types. The division ratio to generate the data length counter clock is selected by LCCLK[3:0]/REMC_CFG register provided separately to the carrier generation clock select bits.

Table 20.4.1 Data Length Counter Clock (PCLK Division Ratio) Selection

LCCLK[3:0]	Division ratio	LCCLK[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)

The data length counter can count up to 256. The count clock should be selected to ensure that the data length fits within this range.

20.5 Data Transfer Control

Make the following settings before starting data transfers.
(1) Configure the carrier signal. (See Section 20.3.)
(2) Select the data length counter clock. (See Section 20.4.)
(3) Set the interrupt conditions. (See Section 20.6.)

Note: Make sure the REMC module is halted (REMEN/REMC_CFG register = 0) before changing the above settings.

Data transmission control

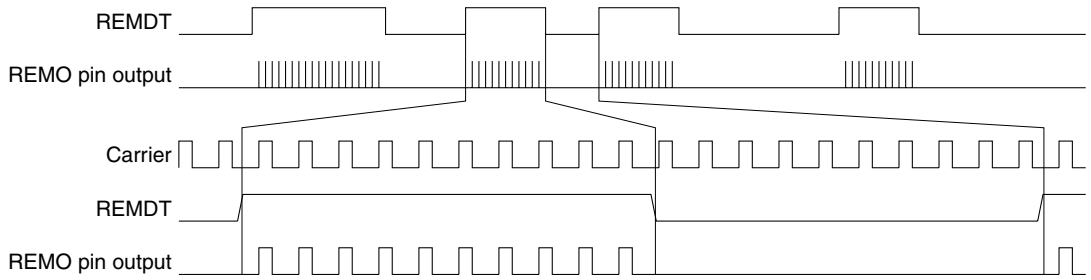


Figure 20.5.1 Data Transmission

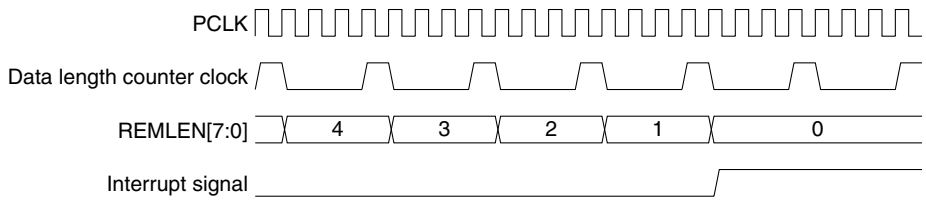


Figure 20.5.2 Underflow Interrupt Generation Timing

(1) Data transmit mode setting
Set REMC to transmit mode by writing 0 to REMMD/REMC_CFG register.

(2) Enabling data transmission
Enable REMC operation by setting REMEN/REMC_CFG register to 1. This initiates REMC transmission.

Set REMDT/REMC_LCNT register to 0 and REMLEN[7:0]/REMC_LCNT register to 0x0 before setting REMEN to 1 to prevent unnecessary data transmission.

(3) Transmission data setting

Set the data to be transmitted (High or Low) to REMDT/REMC_LCNT register.
Setting REMDT to 1 outputs High; setting it to 0 outputs Low from the REMO pin after being modulated by the carrier signal.

(4) Data pulse length setting

Set the value corresponding to the data pulse length (High or Low section) to REMLEN[7:0]/REMC_LCNT register to set to the data length counter.
Given below is the value to which the data length counter is set:

Setting value = Data pulse length (seconds) × Data length counter clock frequency (Hz)

The data length counter starts counting down from the value written using the data length counter clock selected. A cause of underflow interrupt occurs when the data length counter value reaches 0. If the interrupt is enabled, an REMC interrupt request is output to the interrupt controller (ITC). The data length counter stops counting at the same time with the counter value 0 maintained.

(5) Interrupt handling

To transmit the subsequent data, set the subsequent data (Step 3) and set the data pulse length (Step 4) in the interrupt handler routine executed by the data length counter underflow.

(6) Terminating data transmission

To terminate data transmission, set REMEN to 0 after the final data transmission has completed (after an underflow interrupt has occurred).

Data reception control

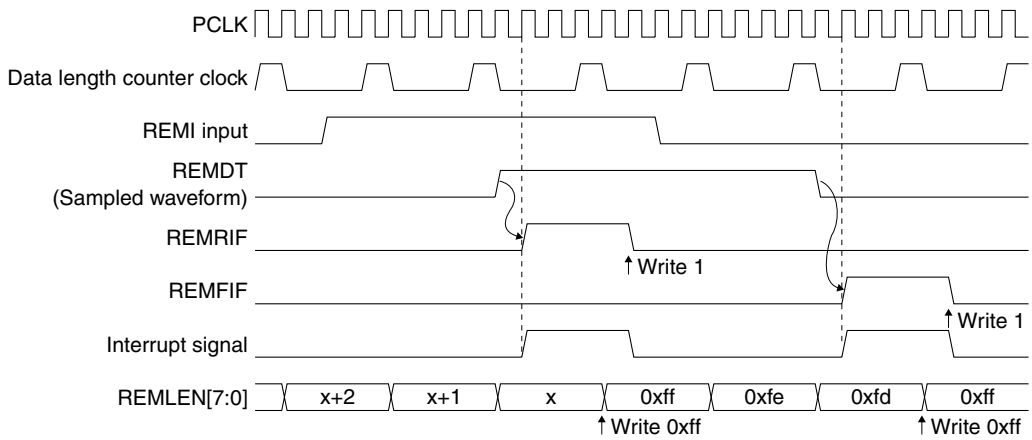


Figure 20.5.3 Data Reception

(1) Data receive mode setting

Set REMC to receive mode by writing 1 to REMMD/REMC_CFG register.

(2) Enabling data reception

Enable REMC operation by setting REMEN/REMC_CFG register to 1. This initiates REMC reception (input edge detection).
REMC detects an input transition (signal rising or falling edges) by sampling the input signal from the REMI pin using the carrier generation clock. If a signal edge is detected, a cause of rising or falling edge interrupt is generated. An REMC interrupt request is output to the ITC if the interrupt is enabled. Rising edge and falling edge interrupts can be individually enabled or disabled.

Note that if the signal level after the input has changed is not detected for at least two continuous sampling clock cycles, the input signal transition is interpreted as noise, and no rising or falling edge interrupt is generated.

(3) Interrupt handling

When a rising edge or falling edge interrupt occurs, write 0xff to REMLEN[7:0]/REMC_LCNT register in the interrupt handler routine to set the value to the data length counter.

The data length counter starts counting down using the selected data length counter clock from the value written.

The data received can be read out from REMDT/REMC_LCNT register.

The subsequent falling or rising edge interrupt is generated at the termination of the data pulse. Read the data length counter at that point. The data length can be calculated from the difference between 0xff and the value read. To receive the subsequent data, set the data length counter to 0xff once again, then wait for the subsequent interrupt.

If the data length counter becomes 0 after being set to 0xff without the occurrence of an edge interrupt, either no more data is left or a receive error has occurred. Data length counter underflow interrupts are generated even when receiving data and should be used for terminate/error handling.

(4) Terminating data reception

To terminate data reception, write 0 to REMEN after the final data has been received.

20.6 REMC Interrupts

The REMC module includes a function for generating the following three different types of interrupts.
- Underflow interrupt
- Rising edge interrupt
- Falling edge interrupt

The REMC module outputs one interrupt signal shared by the three interrupt causes above to the interrupt controller (ITC). To identify the cause of interrupt occurred, check the interrupt flag status in the REMC module.

Underflow interrupt

Generated when the data length counter has counted down to 0, this interrupt cause sets the interrupt flag REMUIF/REMC_INT register inside the REMC to 1.
When data is being transmitted, the underflow interrupt indicates that the specified data length has been transmitted. When receiving data, the underflow interrupt indicates that data has been received or a receive error has occurred.
To use this interrupt, set REMUIE/REMC_INT register to 1. If REMUIE is set to 0 (default), the interrupt request attributable to this cause will not be sent to the ITC.
When REMUIF is set to 1, REMC outputs an interrupt request to the ITC. An interrupt will be generated if the ITC and S1C17 Core interrupt conditions are met.
REMUIF should be inspected in the REMC interrupt handler routine to determine whether the REMC interrupt is attributable to data length counter underflow.
The interrupt cause should be cleared in the interrupt handler routine by resetting (writing 1 to) REMUIF.

Rising edge interrupt

Generated when the REMI pin input signal changes from Low to High, this interrupt cause sets the interrupt flag REMRIF/REMC_INT register to 1 within the REMC.
By running the data length counter between this interrupt and a falling edge interrupt when data is being received, the received data pulse width can be calculated from that count value.
To use this interrupt, set REMRIE/REMC_INT register to 1. If REMRIE is set to 0 (default), the interrupt request attributable to this cause will not be sent to the ITC.
When REMRIF is set to 1, REMC outputs an interrupt request to the ITC. An interrupt will be generated if the ITC and S1C17 Core interrupt conditions are met.
REMRIF should be inspected in the REMC interrupt handler routine to determine whether the REMC interrupt is attributable to input signal rising edge.
The interrupt cause should be cleared in the interrupt handler routine by resetting (writing 1 to) REMRIF.

Falling edge interrupt

Generated when the REMI pin input signal changes from High to Low, this interrupt cause sets the interrupt flag REMFIF/REMC_INT register to 1 within the REMC.

By running the data length counter between this interrupt and a rising edge interrupt when data is being received, the received data pulse width can be calculated from that count value.

To use this interrupt, set REMFIE/REMC_INT register to 1. If REMFIE is set to 0 (default), the interrupt request attributable to this cause will not be sent to the ITC.

When REMFIF is set to 1, REMC outputs an interrupt request to the ITC. An interrupt will be generated if the ITC and S1C17 Core interrupt conditions are met.

REMFIF should be inspected in the REMC interrupt handler routine to determine whether the REMC interrupt is attributable to input signal falling edge.

The interrupt cause should be cleared in the interrupt handler routine by resetting (writing 1 to) REMFIF.

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

20.7 Control Register Details

Table 20.7.1 List of REMC Registers

Address	Register name		Function
0x5340	REMC_CFG	REMC Configuration Register	Controls the clock and data transfer.
0x5342	REMC_CAR	REMC Carrier Length Setup Register	Sets the carrier H/L section lengths.
0x5344	REMC_LCNT	REMC Length Counter Register	Sets the transmit/receive data length.
0x5346	REMC_INT	REMC Interrupt Control Register	Controls interrupts.

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

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

REMC Configuration Register (REMC_CFG)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
REMC Configuration Register (REMC_CFG)	**0x5340** (16 bits)	D15–12	**CGCLK[3:0]**	Carrier generator clock division ratio select	CGCLK[3:0] / LCCLK[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	0x0	R/W	Source clock = PCLK
		D11–8	**LCCLK[3:0]**	Length counter clock division ratio select	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	
		D7–2	–	reserved	–	–	–	0 when being read.
		D1	**REMMD**	REMC mode select	1 Receive / 0 Transmit	0	R/W	
		D0	**REMEN**	REMC enable	1 Enable / 0 Disable	0	R/W	

D[15:12] CGCLK[3:0]: Carrier Generator Clock Division Ratio Select Bits

Selects a carrier generation clock (PCLK division ratio).

Table 20.7.2 Carrier Generation Clock (PCLK Division Ratio) Selection

CGCLK[3:0]	Division ratio	CGCLK[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)

D[11:8] LCCLK[3:0]: Length Counter Clock Division Ratio Select Bits

Selects a data length counter clock (PCLK division ratio).

Table 20.7.3 Data Length Counter Clock (PCLK Division Ratio) Selection

LCCLK[3:0]	Division ratio	LCCLK[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: The clock should be set only while the REMC module is stopped (REMEN = 0).

D[7:2] Reserved

D1 REMMD: REMC Mode Select Bit

Selects the transfer direction.
1 (R/W): Reception
0 (R/W): Transmission (default)

D0 REMEN: REMC Enable Bit

Enables or disables data transfer by the REMC module.
1 (R/W): Enabled
0 (R/W): Disabled (default)

Setting REMEN to 1 starts transmission or receiving in accordance with REMMD settings.
Setting REMEN to 0 disables REMC module operations.

REMC Carrier Length Setup Register (REMC_CAR)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
REMC Carrier Length Setup Register (REMC_CAR)	0x5342 (16 bits)	D15–14	–	reserved	–	–	–	0 when being read.
		D13–8	REMCL[5:0]	Carrier L length setup	0x0 to 0x3f	0x0	R/W	
		D7–6	–	reserved	–	–	–	0 when being read.
		D5–0	REMCH[5:0]	Carrier H length setup	0x0 to 0x3f	0x0	R/W	

D[15:14] Reserved

D[13:8] REMCL[5:0]: Carrier L Length Setup Bits

Sets the carrier signal L section length. (Default: 0x0)
Specify a value corresponding to the number of carrier generation clock cycles selected by CGCLK[3:0]/REMC_CFG register + 1. Calculate carrier L section length as follows:

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

REMCL: REMCL[5:0] setting
cg_clk: Carrier generation clock frequency

The H section length is specified by REMCH[5:0]. The carrier signal is generated from these settings as shown in Figure 20.7.1.

D[7:6] Reserved

D[5:0] REMCH[5:0]: Carrier H Length Setup Bits

Sets the carrier signal H section length. (Default: 0x0)
Specify a value corresponding to the number of carrier generation clock cycles selected by CGCLK[3:0]/REMC_CFG register + 1. Calculate carrier H section length as follows:

$$\text{Carrier H section length} = \frac{\text{REMCH} + 1}{\text{cg\_clk}} \text{ [s]}$$

REMCH: REMCH[5:0] setting
cg_clk: Carrier generation clock frequency

The L section length is specified by REMCL[5:0]. The carrier signal is generated from these settings as shown in Figure 20.7.1.

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

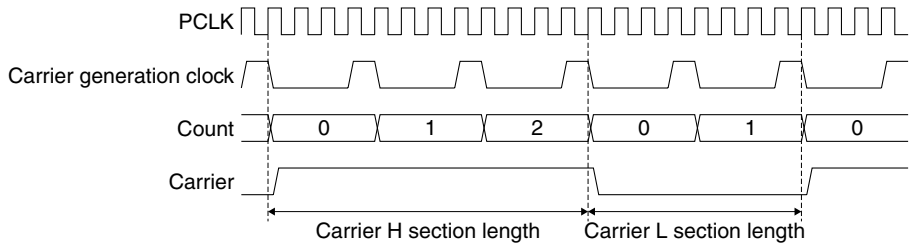


Figure 20.7.1 Carrier Signal Generation

REMC Length Counter Register (REMC_LCNT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
REMC Length Counter Register (REMC_LCNT)	0x5344 (16 bits)	D15–8	REMLEN[7:0]	Transmit/receive data length count (down counter)	0x0 to 0xff	0x0	R/W	
		D7–1	–	reserved	–	–	–	0 when being read.
		D0	REMDT	Transmit/receive data	1 1 (H) 0 0 (L)	0	R/W	

D[15:8] REMLEN[7:0]: Transmit/Receive Data Length Count Bits

Sets the data length counter value and starts counting. (Default: 0x0)
The counter stops when it reaches 0 and generates a cause of underflow interrupt.

For data transmission

Set the transmit data length for data transmission.
When a value corresponding to the data pulse width is written, the data length counter starts counting down from that value. The counter stops counting and generates a cause of underflow interrupt when it reaches 0. Set the subsequent transmit data using this interrupt.

For data receiving

Interrupts can be generated at the input signal rising or falling edges when receiving data. The data pulse length can be obtained from the difference between 0xff set to the data length counter using the interrupt when the input changes and the count value read out when the next interrupt occurs due to an input change.

D[7:1] Reserved

D0 REMDT: Transmit/Receive Data Bit

Sets the transmit data for data transmission. Receive data can be read when receiving data.
1 (R/W): 1 (H)
0 (R/W): 0 (L) (default)

If REMEN/REMC_CFG register is set to 1, the REMDT setting is modulated by the carrier signal for data transmission and output from the REMO pin. For data receiving, this bit is set to the value corresponding to the signal level of the data pulse input.

REMC Interrupt Control Register (REMC_INT)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
REMC Interrupt Control Register (REMC_INT)	0x5346 (16 bits)	D15–11	–	reserved	–				–	–	0 when being read.
		D10	REMFIF	Falling edge interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D9	REMRIF	Rising edge interrupt flag					0	R/W	
		D8	REMUIF	Underflow interrupt flag					0	R/W	
		D7–3	–	reserved	–				–	–	0 when being read.
		D2	REMFIE	Falling edge interrupt enable	1	Enable	0	Disable	0	R/W	
		D1	REMRIE	Rising edge interrupt enable	1	Enable	0	Disable	0	R/W	
		D0	REMUIE	Underflow interrupt enable	1	Enable	0	Disable	0	R/W	

This register controls the data length counter underflow, input signal rising edge, and input signal falling edge interrupts. The interrupt flag is set to 1 when the data length counter underflows, or when an input signal rising edge or falling edge is detected. If the corresponding interrupt enable bit has been set to 1, the REMC outputs an interrupt request signal to the ITC at the same time. An interrupt will be generated if the ITC and S1C17 Core interrupt conditions are met. When an REMC interrupt occurs, check the interrupt flag status in this register to identify the cause of interrupt occurred. If the interrupt enable bit is set to 0, the interrupt is disabled.

Notes: • To prevent interrupt recurrences, the REMC module interrupt flag must be reset in the interrupt handler routine after an REMC interrupt has occurred.

• To prevent generating unnecessary interrupts, reset the interrupt flag before enabling interrupts by the interrupt enable bit.

D[15:11] Reserved

D10 REMFIF: Falling Edge Interrupt Flag Bit

Indicates the falling edge interrupt cause occurrence status.

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

REMFIF is set to 1 at the input signal falling edge. REMFIF is reset to 0 by writing 1.

D9 REMRIF: Rising Edge Interrupt Flag Bit

Indicates the rising edge interrupt cause occurrence status.

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

REMRIF is set to 1 at the input signal rising edge. REMRIF is reset to 0 by writing 1.

D8 REMUIF: Underflow Interrupt Flag Bit

Indicates the underflow interrupt cause occurrence status.

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

REMUIF is set to 1 when a data length counter underflow occurs. REMUIF is reset to 0 by writing 1.

D[7:3] Reserved

D2 REMFIE: Falling Edge Interrupt Enable Bit

Enables or disables input signal falling edge interrupts.

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

D1 REMRIE: Rising Edge Interrupt Enable Bit

Enables or disables input signal rising edge interrupts.

1 (R/W): Interrupt enabled

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

D0 REMUIE: Underflow Interrupt Enable Bit

Enables or disables data length counter underflow interrupts.

1 (R/W): Interrupt enabled

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

21 A/D Converter (ADC10)

21.1 ADC10 Module Overview

The S1C17554/564 includes an A/D converter (ADC10) that converts analog input signals into 10-bit digital values. The following shows the features of the ADC10 module:

- Conversion method: Successive approximation type
- Resolution: 10 bits
- Input channels: Max. 4 channels
- A/D conversion clock: Max. 2 MHz
- Sampling rate: $f_{ADCLK}/13$ to $f_{ADCLK}/20$ [sps] (f_{ADCLK}: A/D conversion clock frequency)
- Analog input voltage range: V_{SS} to AV_{DD}
- Sampling & hold circuit included
- Supports two conversion modes: One-time conversion mode
  (for single channel or multi-channels)
  Continuous conversion mode
  (for single channel or multi-channels, terminated with software)
- Supports three conversion triggers: Software trigger
  External trigger (input from the #ADTRG pin)
  T16 Ch.0 underflow trigger
- The conversion results can be read as 16-bit data with the 10-bit converted data aligned to left or right.
- Two types of interrupts can be generated: Conversion completion interrupt
  Conversion data overwrite error interrupt

Figure 21.1.1 shows the ADC10 configuration.

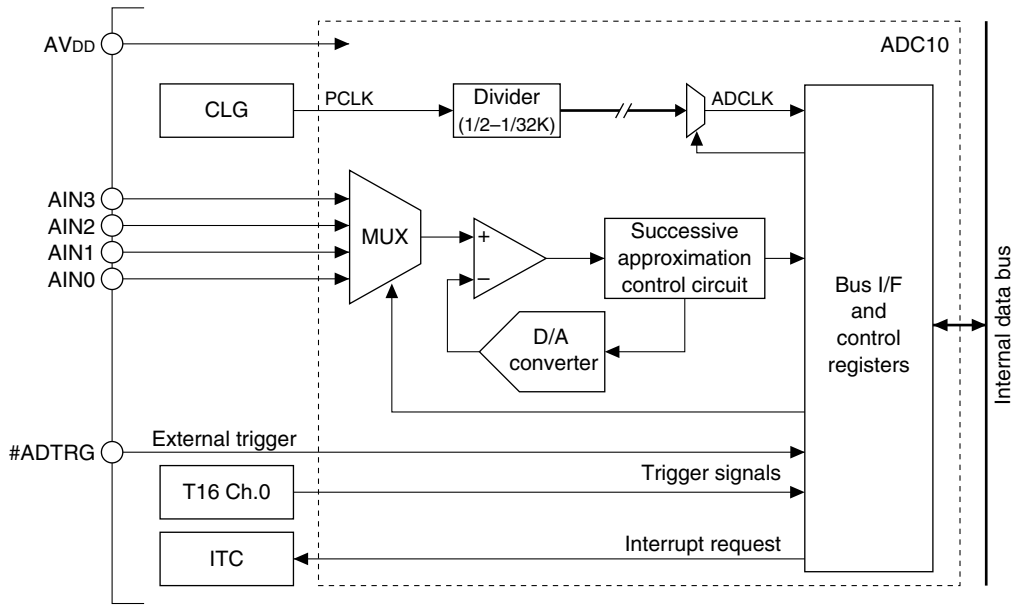


Figure 21.1.1 A/D Converter Configuration

21.2 ADC10 Input Pins

Table 21.2.1 lists the ADC10 input pins.

Table 21.2.1 List of ADC10 Input Pins

Pin name	I/O	Qty	Function
AIN[3:0]	I	4	Analog signal input pins AIN0 (Ch.0) to AIN3 (Ch.3) (see Note below) Input the analog signals to be A/D converted. The analog input voltage AV_{IN} must be within the range of $V_{SS} \leq AV_{IN} \leq AV_{DD}$.
#ADTRG	I	1	External trigger input pin Input a trigger signal to start A/D conversion from an external source.
AV_{DD}	–	1	Analog power-supply pin

Note: The pins go to high impedance status when the port function is switched.

The A/D converter input pins (AIN[3:0], #ADTRG) 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 A/D converter input pins.
For detailed information on pin function switching, see the "I/O Ports (P)" chapter.

21.3 A/D Converter Settings

Make the following settings before starting A/D conversion.
(1) Set the analog input pins. See Section 21.2.
(2) Adjust the comparator according to the A/D conversion voltage range used. See the ADC10_COM register descriptions in Section 21.6.
(3) Set the A/D conversion clock.
(4) Select the A/D conversion start and end channels.
(5) Select the A/D conversion mode.
(6) Select the A/D conversion trigger source.
(7) Set the sampling time.
(8) Select the conversion result storing mode.
(9) When using A/D converter interrupts, set interrupt conditions. See Section 21.5.

Note: Make sure the A/D converter is disabled (ADEN/ADC10_CTL register = 0) before changing the above settings. Changing the settings while the A/D converter is enabled may cause a malfunction.

21.3.1 A/D Conversion Clock Setting

To use the A/D converter, the clock used in the A/D converter must be supplied by turning on the peripheral module clock (PCLK) output from the clock generator (CLG). For more information on clock control, see the "Clock Generator (CLG)" chapters.
The A/D conversion is generated by dividing PCLK. The division ratio can be selected from the 15 types shown in Table 21.3.1.1 using ADDF[3:0]/ADC10_CLK register.

Note: For the A/D conversion clock frequency range that can be used for this A/D converter, see "A/D Converter Characteristics" in the "Electrical Characteristics" chapter.

Table 21.3.1.1 A/D Conversion Clock (PCLK Division Ratio) Selection

ADDF[3:0]	Division ratio
0xf	Reserved
0xe	1/32768
0xd	1/16384
0xc	1/8192
0xb	1/4096
0xa	1/2048
0x9	1/1024
0x8	1/512
0x7	1/256
0x6	1/128
0x5	1/64
0x4	1/32
0x3	1/16
0x2	1/8
0x1	1/4
0x0	1/2

(Default: 0x0)

21.3.2 Selecting A/D Conversion Start and End Channels

Select the channel in which the A/D conversion is to be performed from among the pins (channels) that have been set for analog input. To enable A/D conversions in multiple channels to be performed successively through one convert operation, specify the conversion start and conversion end channels using ADCS[2:0]/ADC10_TRG register and ADCE[2:0]/ADC10_TRG register, respectively.

Table 21.3.2.1 Relationship between ADCS/ADCE and Input Channels

ADCS[2:0]/ADCE[2:0]	Channel selected
0x7–0x4	Reserved
0x3	AIN3
0x2	AIN2
0x1	AIN1
0x0	AIN0

(Default: 0x0)

Example: Operation of one A/D conversion

ADCS[2:0] = 0, ADCE[2:0] = 0
Converted only in AIN0

ADCS[2:0] = 0, ADCE[2:0] = 3
Converted in the following order: AIN0→AIN1→AIN2→AIN3

ADCS[2:0] = 2, ADCE[2:0] = 1
Converted in the following order: AIN2→AIN3→(AIN4→AIN5→AIN6→AIN7)→AIN0→AIN1

Note: The control circuits in the A/D converter supports up to eight channels for expansion in the future, and it performs A/D conversion if a channel (AIN4–AIN7) without an analog input is specified. In this case, the results that will be stored to ADD[15:0]/ADC10_ADD register is 0x0. To avoid A/D conversion for the channels without an input, set the ADCS[2:0] to equal or smaller than ADCE[2:0] within the available analog inputs.

21.3.3 A/D Conversion Mode Setting

The A/D converter provides two conversion modes that can be selected using ADMS/ADC10_TRG register: one-time conversion mode and continuous conversion mode.

1. One-time conversion mode (ADMS = 0)

The A/D converter performs A/D conversion for all analog inputs within the range from the start channel specified by ADCS[2:0]/ADC10_TRG register to the end channel specified by the ADCE[2:0]/ADC10_TRG register once and then stops automatically.

2. Continuous conversion mode (ADMS = 1)

The A/D converter repeatedly performs A/D conversion for the channels in the range specified by ADCS[2:0] and ADCE[2:0] until stopped with software.

At initial reset, the A/D converter is set to one-time conversion mode.

21.3.4 Trigger Selection

Select a trigger source to start A/D conversion from among the three types listed in Table 21.3.4.1 using ADTS[1:0]/ADC10_TRG register.

Table 21.3.4.1 Trigger Selection

ADTS[1:0]	Trigger source
0x3	External trigger (#ADTRG)
0x2	Reserved
0x1	16-bit timer Ch.0
0x0	Software trigger

(Default: 0x0)

1. External trigger (#ADTRG)

The signal input to the #ADTRG pin is used as a trigger. To use this trigger source, the I/O port pin must be configured for the #ADTRG input using the port function select bit (see the "I/O Ports (P)" chapter). An A/D conversion starts when a falling edge of the #ADTRG signal is detected.

Note: When using an external trigger to start A/D conversion, ensure to maintain the Low period of the trigger signal input to the #ADTRG pin for two or more S1C17 Core operating clock cycles.

2. 16-bit timer (T16) Ch.0

The underflow signal of T16 Ch.0 is used as a trigger. Since the T16 underflow cycle can be programmed with flexibility, this trigger source is effective when periodic A/D conversions are required. For more information on timer settings, see the "16-bit Timers (T16)" chapter.

3. Software trigger

Writing 1 to ADCTL/ADC10_CTL register with software serves as a trigger to start A/D conversion.

21.3.5 Sampling Time Setting

The analog signal input sampling time in this A/D converter can be configured to eight steps (two to nine A/D conversion clock cycles) using ADST[2:0]/ADC10_TRG register.

Table 21.3.5.1 Sampling Time Settings

ADST[2:0]	Sampling time (in A/D conversion clock cycles)
0x7	9 cycles
0x6	8 cycles
0x5	7 cycles
0x4	6 cycles
0x3	5 cycles
0x2	4 cycles
0x1	3 cycles
0x0	2 cycles

(Default: 0x7)

The sampling time must satisfy the acquisition time condition (t_{ACQ}, time required for acquiring input voltage). Figure 21.3.5.1 shows an equivalent circuit of the analog input portion.

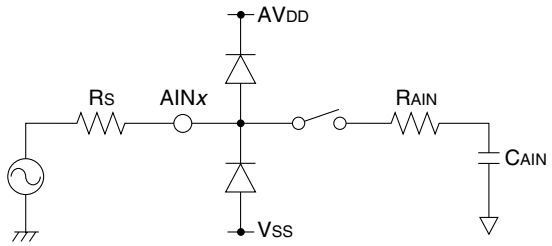


Figure 21.3.5.1 Equivalent Circuit of Analog Input Portion

Determine f_{ADCLK} and ADST[2:0] settings to satisfy the expression below.

$t_{ACQ} = 8 \times (R_S + R_{AIN}) \times C_{AIN}$ (See "Electrical Characteristics" for the R_{AIN} and C_{AIN} values.)

$$\frac{1}{f_{ADCLK}} \times (\text{Number of clock cycles set by ADST[2:0]}) > t_{ACQ}$$

f_{ADCLK}: A/D conversion clock frequency [Hz]

The following shows the relation between sampling time and sampling rate.

$$\text{Sampling rate [sps]} = \frac{f_{ADCLK}}{\text{Number of clock cycles set by ADST[2:0]} + 11}$$

21.3.6 Setting Conversion Result Storing Mode

The A/D converter loads the 10-bit conversion results into ADD[15:0]/ADC10_ADD register (16-bit register) after an A/D conversion has completed. At this time, the 10-bit conversion results are aligned in the 16-bit register according to the conversion result storing mode set with STMD/ADC10_TRG register either as the high-order 10 bits (left justify mode) or the low-order 10 bits (right justify mode). The remaining six bits are all set to 0.

ADD bit	15	...	10	9	...	6	5	...	0
Left justify mode (STMD = 1)	(MSB)	10-bit conversion results				(LSB)	0	...	0
Right justify mode (STMD = 0)	0	...	0	(MSB)	10-bit conversion results				(LSB)

Figure 21.3.6.1 Conversion Data Alignment

21.4 A/D Conversion Control and Operations

The A/D converter should be controlled in the sequence shown below.
1. Activate the A/D converter.
2. Start A/D conversion.
3. Read the A/D conversion results.
4. Terminate A/D conversion.

21.4.1 Activating A/D Converter

After the settings described in Section 21.3 have been completed, write 1 to ADEN/ADC10_CTL register to enable the A/D converter. The A/D converter is thereby ready to accept a trigger to start A/D conversion. To set up the A/D converter again, or when the A/D converter is not used, ADEN must be set to 0.

21.4.2 Starting A/D conversion

The A/D converter starts A/D conversion when a trigger is input while ADEN is 1. When software trigger is selected, an A/D conversion starts by writing 1 to ADCTL/ADC10_CTL register.
The A/D converter accepts triggers from only the trigger source selected by ADTS[1:0]/ADC10_TRG register.
Once a trigger is input, the A/D converter starts sampling of the analog input signal and A/D conversion beginning with the conversion start channel selected by ADCS[2:0]/ADC10_TRG register.
The software trigger bit ADCTL functions as an A/D conversion status bit that goes 1 while A/D conversion is underway even if it has started by another trigger source. The channel in which conversion is underway can be identified by reading ADICH[2:0]/ADC10_CTL register.

21.4.3 Reading A/D Conversion Results

Upon completion of the A/D conversion in the start channel, the A/D converter loads the conversion results into ADD[15:0]/ADC10_ADD register and sets the conversion completion flag ADCF/ADC10_CTL register. If multiple channels are specified using ADCS[2:0]/ADC10_TRG register and ADCE[2:0]/ADC10_TRG register, the A/D converter continues A/D conversions in the subsequent channels.
The results of A/D conversion are stored in ADD[15:0] each time conversion in one channel is completed. At the same time, a conversion completion interrupt can be generated, enabling to read out the converted data. If no conversion completion interrupt is used, read the conversion results from ADD[15:0] after confirming that ADCF is set to 1 indicating completion of conversion. ADCF is reset to 0 when ADD[15:0] is read.
When a single channel or multiple channels are being converted continuously, the conversion results must be read out from ADD[15:0] before the following conversion has completed. If the A/D conversion currently underway is completed while ADCF is set to 1 (before reading the previous conversion results), ADD[15:0] is overwritten and the overwrite error flag ADOWE/ADC10_CTL register is set to 1. At this time, a conversion data overwrite error interrupt can be generated. After the conversion results are read from ADD[15:0], ADOWE should be read to check whether the read data is valid or not. Or enable conversion data overwrite error interrupts and perform error handling using the interrupt. Once ADOWE is set, it will not be reset until software writes 1. Since ADCF is also set simultaneously with ADOWE, read out the converted data to reset ADCF.

Note: Occurrence of an overwrite error does not stop continuous conversion.

21.4.4 Terminating A/D Conversion

One-time conversion mode (ADMS = 0)

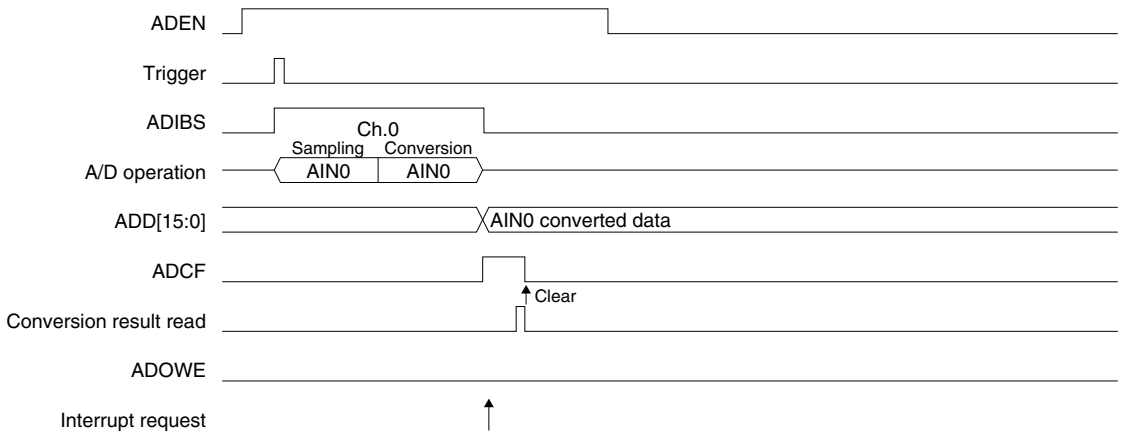
In one-time mode, the A/D converter performs A/D conversion within the channel range successively beginning with the conversion start channel specified by ADCS[2:0]/ADC10_TRG register and terminates once the conversion end channel specified by ADCE[2:0]/ADC10_TRG register has been completed. ADCTL/ADC10_CTL register is reset to 0 upon completion of the conversion sequence.

Continuous conversion mode (ADMS = 1)

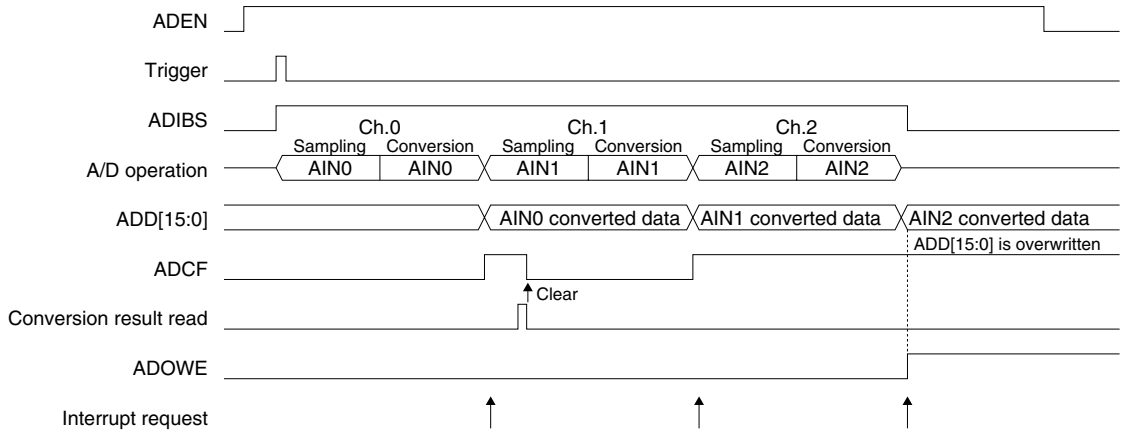
In continuous conversion mode, the A/D converter repeatedly performs A/D conversion from the conversion start channel to the conversion end channel. The hardware does not stop the conversion sequence. To stop A/D conversion, write 0 to ADCTL. Since the conversion sequence is forcibly terminated, the results of the conversion then underway cannot be obtained.

21.4.5 Timing Charts

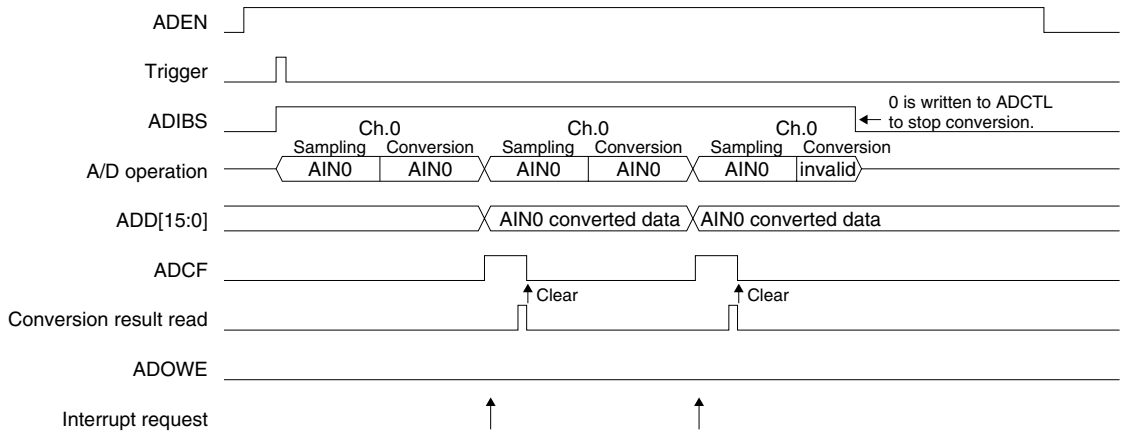
Figure 21.4.5.1 shows the operations of the A/D converter.



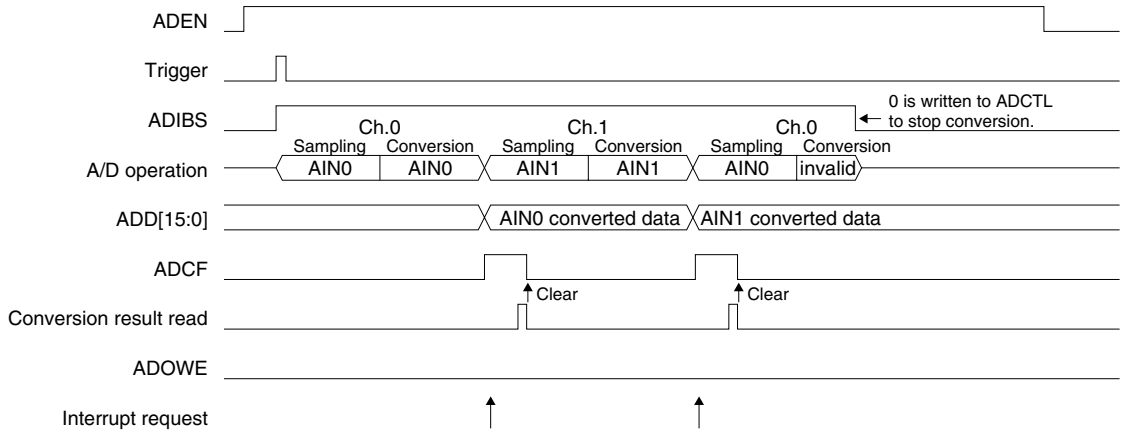
(1) Single channel (AIN0) one-time conversion mode (ADCS = 0, ADCE = 0, ADMS = 0)



(2) Multi-channel (AIN0–AIN2) one-time conversion mode (ADCS = 0, ADCE = 2, ADMS = 0)



(3) Single channel (AIN0) continuous conversion mode (ADCS = 0, ADCE = 0, ADMS = 1)



(4) Multi-channel (AIN0–AIN1) continuous conversion mode (ADCS = 0, ADCE = 1, ADMS = 1)

Figure 21.4.5.1 A/D Converter Operations

21.5 A/D Converter Interrupts

The A/D converter includes a function for generating the following two different types of interrupts.

- Conversion completion interrupt
- Conversion data overwrite error interrupt

The A/D converter 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.

Conversion completion interrupt

To use this interrupt, set ADCIE/ADC10_CTL register to 1. If ADCIE is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

When A/D conversion in a channel has completed, the A/D converter sets ADCF/ADC10_CTL register to 1, indicating that the converted data can be read out. If conversion completion interrupts are enabled (ADCIE = 1), an interrupt request is sent simultaneously to the ITC.

An interrupt occurs if other interrupt conditions are met.

You can inspect ADCF in the ADC10 interrupt handler routine to determine whether the ADC10 interrupt is attributable to a completion of conversion. If ADCF is 1, the converted data can be read out from ADD[15:0]/ADC10_ADD register by the interrupt handler routine. The interrupt cause ADCF is reset to 0 by reading ADD[15:0] and this interrupt will not be generated until the subsequent conversion has completed.

Conversion data overwrite error interrupt

To use this interrupt, set ADOIE/ADC10_CTL register to 1. If ADOIE is set to 0 (default), interrupt requests for this cause will not be sent to the ITC.

If the following A/D conversion has completed when ADD[15:0] has not been read (ADCF = 1), the A/D converter sets ADOWE/ADC10_CTL register to 1, indicating that ADD[15:0] is overwritten. If conversion data overwrite error interrupts are enabled (ADOIE = 1), an interrupt request is sent simultaneously to the ITC.

An interrupt occurs if other interrupt conditions are met.

You can inspect ADOWE in the ADC10 interrupt handler routine to determine whether the ADC10 interrupt is attributable to an overwrite error. If ADOWE is 1, perform error handling by the interrupt handler routine. The interrupt cause ADOWE is reset to 0 by writing 1.

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

Notes: • To prevent interrupt recurrences, the ADCF/ADC10_CTL register and ADOWE/ADC10_CTL register must be reset in the interrupt handler routine after an ADC10 interrupt has occurred.

• To prevent unwanted interrupts, reset ADCF and ADOWE before enabling interrupts with ADCIE/ADC10_CTL register and ADOIE/ADC10_CTL register.

21.6 Control Register Details

Table 21.6.1 List of ADC10 Registers

Address	Register name		Function
0x5380	ADC10_ADD	A/D Conversion Result Register	A/D converted data
0x5382	ADC10_TRG	A/D Trigger/Channel Select Register	Sets start/end channels and conversion mode.
0x5384	ADC10_CTL	A/D Control/Status Register	Controls A/D converter and indicates conversion status.
0x5386	ADC10_CLK	A/D Clock Control Register	Controls A/D converter clock.
0x5388	ADC10_COM	A/D Comparator Setting Register	Adjusts A/D conversion characteristics.

The A/D converter registers are described in detail below. These are 16-bit registers.

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

A/D Conversion Result Register (ADC10_ADD)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
A/D Conversion Result Register (ADC10_ADD)	**0x5380** (16 bits)	D15–0	**ADD[15:0]**	A/D converted data ADD[9:0] are effective when STMD = 0 (ADD[15:10] = 0) ADD[15:6] are effective when STMD = 1 (ADD[5:0] = 0)	0x0 to 0x3ff	0x0	R	

D[15:0] ADD[15:0]: A/D Converted Data Bits

The A/D conversion results are stored. (Default: 0x0)

The data alignment in this 16-bit register (conversion result storing mode) can be selected using the STMD/ADC10_TRG register.

ADD bit	15	...	10	9	...	6	5	...	0
Left justify mode (STMD = 1)	(MSB)	10-bit conversion results				(LSB)	0	...	0
Right justify mode (STMD = 0)	0	...	0	(MSB)	10-bit conversion results				(LSB)

Figure 21.6.1 Conversion Data Alignment

This register is a read-only, so writing to this register is ignored.

A/D Trigger/Channel Select Register (ADC10_TRG)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
A/D Trigger/ Channel Select Register (ADC10_TRG)	0x5382 (16 bits)	D15–14	–	reserved	–	–	–	0 when being read.
		D13–11	**ADCE[2:0]**	End channel select	0x0 to 0x3	0x0	R/W	
		D10–8	**ADCS[2:0]**	Start channel select	0x0 to 0x3	0x0	R/W	
		D7	**STMD**	Conversion result storing mode	1 ADD[15:6] 0 ADD[9:0]	0	R/W	
		D6	**ADMS**	Conversion mode select	1 Continuous 0 Single	0	R/W	
		D5–4	**ADTS[1:0]**	Conversion trigger select	ADTS[1:0] Trigger 0x3 #ADTRG pin 0x2 reserved 0x1 T16 Ch.0 0x0 Software	0x0	R/W	
		D3	–	reserved	–	–	–	0 when being read.
		D2–0	**ADST[2:0]**	Sampling time setting	ADST[2:0] Sampling time 0x7 9 cycles 0x6 8 cycles 0x5 7 cycles 0x4 6 cycles 0x3 5 cycles 0x2 4 cycles 0x1 3 cycles 0x0 2 cycles	0x7	R/W	

D[15:14] Reserved

D[13:11] ADCE[2:0]: End Channel Select Bits

Sets the conversion end channel with a channel number from 0 to 3. (Default: 0x0 = AIN0)
Analog inputs can be A/D-converted continuously from the channel set by ADCS[2:0] to the channel set by ADCE[2:0] in one A/D conversion. If only one channel is to be A/D converted, set the same channel number in both ADCS[2:0] and ADCE[2:0].

Table 21.6.2 Relationship between ADCS/ADCE and Input Channels

ADCS[2:0]/ADCE[2:0]	Channel selected
0x7–0x4	Reserved
0x3	AIN3
0x2	AIN2
0x1	AIN1
0x0	AIN0

(Default: 0x0)

D[10:8] ADCS[2:0]: Start Channel Select Bits

Sets the conversion start channel with a channel number from 0 to 3. (Default: 0x0 = AIN0)

D7 STMD: Conversion Result Storing Mode Bit

Selects the data alignment when the conversion results are loaded into ADD[15:0].
1 (R/W): Left justify mode (10-bit conversion results → ADD[15:6], ADD[5:0] = 0)
0 (R/W): Right justify mode (10-bit conversion results → ADD[9:0], ADD[15:10] = 0) (default)

D6 ADMS: Conversion Mode Select Bit

Selects an A/D conversion mode.
1 (R/W): Continuous conversion mode
0 (R/W): One-time conversion mode (default)

Writing 1 to ADMS sets the A/D converter to continuous conversion mode. In this mode, A/D conversions in the range of the channels selected by ADCS[2:0] and ADCE[2:0] are executed continuously until stopped with software.
When ADMS is 0, the A/D converter operates in one-time conversion mode. In this mode, A/D conversion is terminated after all inputs in the range of the channels selected by ADCS[2:0] and ADCE[2:0] have been converted once.

D[5:4] ADTS[1:0]: Conversion Trigger Select Bits

Selects a trigger source to start A/D conversion.

Table 21.6.3 Trigger Selection

ADTS[1:0]	Trigger source
0x3	External trigger (#ADTRG)
0x2	Reserved
0x1	16-bit timer Ch.0
0x0	Software trigger

(Default: 0x0)

When an external trigger is used, the #ADTRG pin must be configured in advance using the port function select bit (see the "I/O Ports (P)" chapter). A/D conversion is started when a falling edge of the #ADTRG signal is detected.
When 16-bit timer (T16) Ch.0 is used, since its underflow signal serves as a trigger, set the underflow cycle and other conditions for the timer.

D3 Reserved

D[2:0] ADST[2:0]: Sampling Time Setting Bits

Sets the analog input sampling time.

Table 21.6.4 Sampling Time Settings

ADST[2:0]	Sampling time (in A/D conversion clock cycles)
0x7	9 cycles
0x6	8 cycles
0x5	7 cycles
0x4	6 cycles
0x3	5 cycles
0x2	4 cycles
0x1	3 cycles
0x0	2 cycles

(Default: 0x7)

A/D Control/Status Register (ADC10_CTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
A/D Control/ Status Register (ADC10_CTL)	0x5384 (16 bits)	D15	–	reserved	–	–	–	0 when being read.
		D14–12	ADICH[2:0]	Conversion channel indicator	0x0 to 0x3	0x0	R	
		D11	–	reserved	–	–	–	0 when being read.
		D10	ADIBS	ADC10 status	1 Busy 0 Idle	0	R	
		D9	ADOWE	Overwrite error flag	1 Error 0 Normal	0	R/W	Reset by writing 1.
		D8	ADCF	Conversion completion flag	1 Completed 0 Run/Stand-by	0	R	Reset when ADC10_ADD is read.
		D7–6	–	reserved	–	–	–	0 when being read.
		D5	ADOIE	Overwrite error interrupt enable	1 Enable 0 Disable	0	R/W	
		D4	ADCIE	Conversion completion int. enable	1 Enable 0 Disable	0	R/W	
		D3–2	–	reserved	–	–	–	0 when being read.
		D1	ADCTL	A/D conversion control	1 Start 0 Stop	0	R/W	
		D0	ADEN	ADC10 enable	1 Enable 0 Disable	0	R/W	

D15 Reserved

D[14:12] ADICH[2:0]: Conversion Channel Indicator Bits

Indicates the channel number (0 to 3) currently being A/D-converted. (Default: 0x0 = AIN0)
When A/D conversion is performed in multiple channels, read this bit to identify the channel in which conversion is underway.

D11 Reserved

D10 ADIBS: ADC10 Status Bit

Indicates the A/D converter status.

1 (R): Being converted

0 (R): Conversion completed/standby (default)

ADIBS is set to 1 at the input trigger signal edge (at the beginning of sampling) and is reset to 0 upon completion of conversion (when ADCTL is set to 0).

D9 ADOWE: Overwrite Error Flag Bit

Indicates that the converted results in ADD[15:0]/ADC10_ADD register have been overwritten before reading.

1 (R): Overwrite error (cause of interrupt has occurred)

0 (R): Normal (cause of interrupt has not occurred) (default)

1 (W): Flag is reset

0 (W): Ignored

When a single channel or multiple channels are being converted continuously, ADD[15:0] is overwritten and ADOWE is set to 1 if the A/D conversion currently underway is completed while ADCF is set to 1 (before reading the previous conversion results). After the conversion results are read from ADD[15:0], ADOWE should be read to check whether the read data is valid or not.

ADOWE is a cause of ADC10 interrupt. When ADOWE is set to 1, a conversion data overwrite error interrupt request is output to the ITC if ADOIE has been set to 1 (interrupt enabled). An interrupt is generated if the ITC and S1C17 Core interrupt conditions are satisfied.

ADOWE is reset by writing 1.

D8 ADCF: Conversion Completion Flag Bit

Indicates that A/D conversion has been completed.

1 (R): Conversion completed (cause of interrupt has occurred)

0 (R): Being converted/standby (cause of interrupt has not occurred) (default)

ADCF is set to 1 when A/D conversion is completed, and the converted data is loaded into ADD[15:0]/ADC10_ADD register.

ADCF is a cause of ADC10 interrupt. When ADCF is set to 1, a conversion completion interrupt request is output to the ITC if ADCIE has been set to 1 (interrupt enabled). An interrupt is generated if the ITC and S1C17 Core interrupt conditions are satisfied. ADCF is reset to 0 by reading ADD[15:0]. An overwrite error occurs if the next A/D conversion is completed while ADCF is set (see ADOWE above), ADCF must be reset by reading ADD[15:0] before an overwrite occurs. When an overwrite error occurs, ADCF is also set due to completion of conversion.

D[7:6] Reserved

D5 ADOIE: Overwrite Error Interrupt Enable Bit

Enables or disables interrupts caused by occurrences of conversion data overwrite errors.

1 (R/W): Interrupt enabled

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

Setting ADOIE to 1 enables conversion data overwrite error interrupt requests to the ITC; setting to 0 disables interrupts.

D4 ADCIE: Conversion Completion Interrupt Enable Bit

Enables or disables interrupts caused by completion of conversion.

1 (R/W): Interrupt enabled

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

Setting ADCIE to 1 enables conversion completion interrupt requests to the ITC; setting to 0 disables interrupts.

D[3:2] Reserved

D1 ADCTL: A/D Conversion Control Bit

Controls A/D conversion.

1 (W): Software trigger
0 (W): Stop A/D conversion
1 (R): Being converted
0 (R): Conversion completed/standby (default)

Write 1 to ADCTL to start A/D conversion by a software trigger. If any other trigger is used, ADCTL is automatically set to 1 by the hardware.
ADCTL remains set while A/D conversion is underway. In one-time conversion mode, upon completion of A/D conversion in the specified channels, ADCTL is reset to 0 and the A/D conversion circuit stops operating. To stop A/D conversion during operation in continuous conversion mode, reset ADCTL by writing 0.
When ADEN is 0, no trigger will be accepted.

D0 ADEN: ADC10 Enable Bit

Enables or disables the A/D converter operations.

1 (R/W): Enabled
0 (R/W): Disabled (default)

Writing 1 to ADEN enables the A/D converter, meaning it is ready to start A/D conversion (i.e., ready to accept a trigger).
When ADEN is 0, the A/D converter is disabled, meaning it is unable to accept a trigger.
Before setting the modes, start/end channels, or other A/D converter conditions, be sure to reset ADEN to 0. This helps to prevent the A/D converter from operating erratically.

A/D Clock Control Register (ADC10_CLK)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
A/D Clock Control Register (ADC10_CLK)	**0x5386** (16 bits)	D15–4	–	reserved	–		–	–	0 when being read.
		D3–0	**ADDF[3:0]**	A/D converter clock division ratio select	ADDF[3:0]	Division ratio	0x0	R/W	Source clock = PCLK
					0xf	reserved			
					0xe	1/32768			
					0xd	1/16384			
					0xc	1/8192			
					0xb	1/4096			
					0xa	1/2048			
					0x9	1/1024			
					0x8	1/512			
					0x7	1/256			
					0x6	1/128			
					0x5	1/64			
					0x4	1/32			
					0x3	1/16			
					0x2	1/8			
					0x1	1/4			
					0x0	1/2			

D[15:4] Reserved

D[3:0] ADDF[3:0]: A/D Converter Clock Division Ratio Select Bits

Selects a PCLK division ratio to generate the A/D converter clock.

Table 21.6.5 A/D Conversion Clock (PCLK Division Ratio) Selection

ADDF[3:0]	Division ratio
0xf	Reserved
0xe	1/32768
0xd	1/16384
0xc	1/8192
0xb	1/4096
0xa	1/2048
0x9	1/1024
0x8	1/512
0x7	1/256
0x6	1/128
0x5	1/64
0x4	1/32
0x3	1/16
0x2	1/8
0x1	1/4
0x0	1/2

(Default: 0x0)

Note: To use the A/D converter, the clock used in the A/D converter must be supplied by turning on the peripheral module clock (PCLK) output from the clock generator (CLG).

A/D Comparator Setting Register (ADC10_COM)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
A/D Comparator Setting Register (ADC10_COM)	0x5388 (16 bits)	D15–6	–	reserved	–	–	–	0 when being read.
		D5–4	FSEL[1:0]	A/D comparator adjustment	0x0 to 0x3	0x0	R/W	
		D3–2	–	reserved	–	–	–	0 when being read.
		D1–0	XPD[1:0]	A/D comparator adjustment	0x0 to 0x3	0x3	R/W	

D[15:6], D[3:2]

Reserved

D[5:4], D[1:0]

FSEL[1:0], XPD[1:0]: A/D Comparator Adjustment Bits

Adjusts the comparator for optimal A/D conversion according to the A/D operating voltage used.

Table 21.6.6 A/D Comparator Adjustment

AV_{DD} range	FSEL[1:0]	XPD[1:0]
3.3 to 2.7 V	2	3
3.6 to 3.0 V	3	2
4.3 to 3.3 V	0	3
5.0 to 4.0 V	2	1
5.5 to 4.7 V	0	2

22 On-chip Debugger (DBG)

22.1 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 to the S1C17554/564 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
• S1C17 Family C compiler package (e.g., S5U1C17001C)

Debug pins

The following debug pins are used to connect ICDmini.

Table 22.1.1 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.
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.

22.2 Debug Break Operation Status

The S1C17 Core enters debug mode when the brk instruction is executed or a debug interrupt is generated by a break signal (Low) input to the DSIO pin. This state persists until the retd instruction is executed. During this time, hardware interrupts and NMIs are disabled.
The default setting halts peripheral circuit operations. This setting can be modified even when debugging is underway.
The peripheral circuits that operate with PCLK will continue running in debug mode by setting DBRUN1/MISC_DMODE1 register to 1. Setting DBRUN1 to 0 (default) will stop these peripheral circuits in debug mode.
The peripheral circuits that operate with a clock other than PCLK will continue running in debug mode by setting DBRUN2/MISC_DMODE2 register to 1. Setting DBRUN2 to 0 (default) will stop these peripheral circuits in debug mode.
Some peripheral circuits, such as SPI, I2CS, and T16A, that run with an external input clock will not stop operating even if the S1C17 Core enters debug mode.

22.3 Additional Debugging Function

The S1C17554/564 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 S1C17554/564 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 S1C17554/564 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.

22.4 Control Register Details

Table 22.4.1 List of Debug Registers

Address	Register name		Function
0x4020	MISC_DMODE1	Debug Mode Control Register 1	Enables peripheral operations in debug mode (PCLK).
0x5322	MISC_DMODE2	Debug Mode Control Register 2	Enables peripheral operations in debug mode (except PCLK).
0x5326	MISC_IRAMSZ	IRAM Size Select Register	Selects the IRAM size.
0xffff90	DBRAM	Debug RAM Base Register	Indicates the debug RAM base address.
0xffffa0	DCR	Debug Control Register	Controls debugging.
0xffffb8	IBAR2	Instruction Break Address Register 2	Sets Instruction break address #2.
0xffffbc	IBAR3	Instruction Break Address Register 3	Sets Instruction break address #3.
0xffffd0	IBAR4	Instruction Break Address Register 4	Sets Instruction break address #4.

The debug registers are described in detail below.

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

• For debug registers not described here, refer to the S1C17 Core Manual.

Debug Mode Control Register 1 (MISC_DMODE1)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Debug Mode Control Register 1 (MISC_DMODE1)	**0x4020** (8 bits)	D7–2	–	reserved	–	–	–	0 when being read.
		D1	**DBRUN1**	Run/stop select in debug mode	1 Run 0 Stop	0	R/W	
		D0	–	reserved	–	–	–	0 when being read.

D[7:2] Reserved

D1 DBRUN1: Run/Stop Select Bit in Debug Mode

Selects the operating status of the peripheral circuits that operate with PCLK in debug mode.

1 (R/W): Run

0 (R/W): Stop (default)

Setting DBRUN1 to 1 enables the peripheral circuits that operate with PCLK to run even in debug mode. Setting it to 0 will stop them when the S1C17 Core enters debug mode. Set DBRUN1 to 1 to maintain running status for these peripheral circuits in debug mode.

D0 Reserved

Debug Mode Control Register 2 (MISC_DMODE2)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Debug Mode Control Register 2 (MISC_DMODE2)	0x5322 (16 bits)	D15–1	–	reserved	–	–	–	0 when being read.
		D0	DBRUN2	Run/stop select in debug mode (except PCLK peripheral circuits)	1 Run 0 Stop	0	R/W	

D[15:1] Reserved

D0 DBRUN2: Run/Stop Select Bit in Debug Mode (except PCLK peripheral circuits)

Selects the operating status of the peripheral circuits that operate with a clock other than PCLK in debug mode.

1 (R/W): Run

0 (R/W): Stop (default)

Setting DBRUN2 to 1 enables the peripheral circuits that operate with a clock other than PCLK to run even in debug mode. Setting it to 0 will stop them when the S1C17 Core enters debug mode. Set DBRUN2 to 1 to maintain running status for these peripheral circuits in debug mode.

Some peripheral circuits, such as SPI, I2CS, and T16A, that run with an external input clock will not stop operating even if the S1C17 Core enters debug mode.

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	0x6 (= 16KB)	0x6	R	
		D3	–	reserved	–	–	–	0 when being read.
		D2–0	IRAMSZ[2:0]	IRAM size select	IRAMSZ[2:0] / Size 0x7 reserved 0x6 16KB 0x5 512B 0x4 1KB 0x3 2KB 0x2 4KB 0x1 8KB 0x0 12KB	0x6	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: 0x6)

D3 Reserved

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

Selects the size of the internal RAM to be used.

Table 22.4.2 Internal RAM Size Selection

IRAMSZ[2:0]	Internal RAM size
0x7	Reserved
0x6	16KB
0x5	512B
0x4	1KB
0x3	2KB
0x2	4KB
0x1	8KB
0x0	12KB

(Default: 0x6)

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	0x2fc0	0x2fc0	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)

23 Multiplier/Divider (COPRO)

23.1 Overview

The S1C17554/564 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.

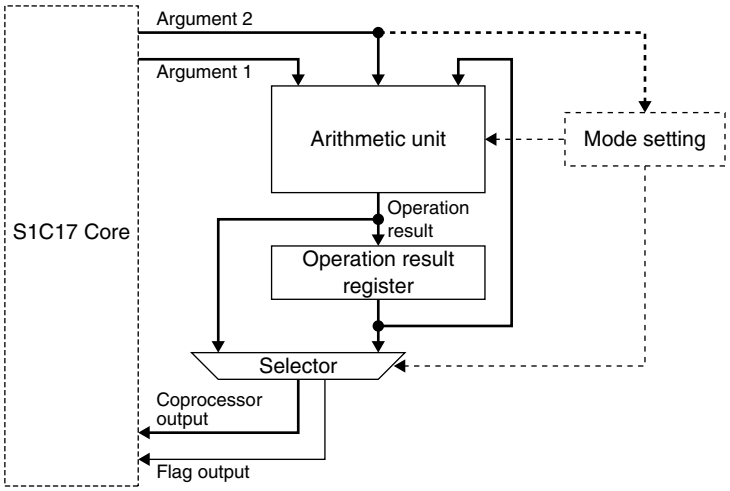


Figure 23.1.1 Multiplier/Divider Block Diagram

23.2 Operation Mode and Output Mode

The Multiplier/divider operates according to the operation mode specified by the application program. As listed in Table 23.2.1, 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)

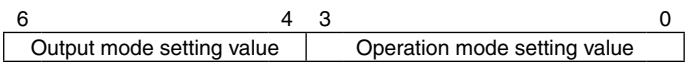


Figure 23.2.1 Mode Setting Register

Table 23.2.1 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

23.3 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.

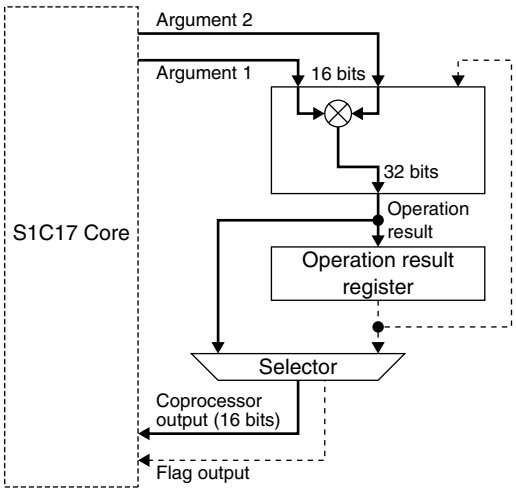


Figure 23.3.1 Data Path in Multiplication Mode

Table 23.3.1 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.
```

23.4 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.

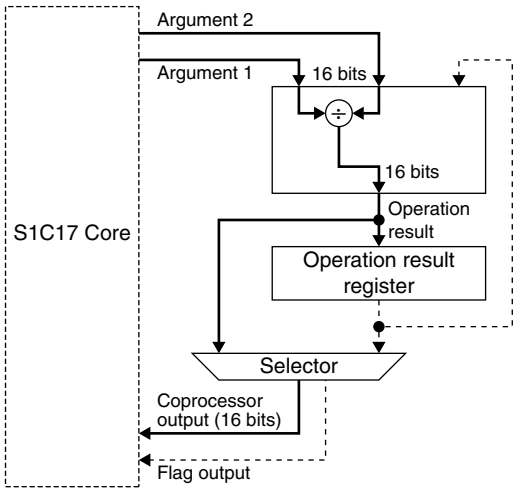


Figure 23.4.1 Data Path in Division Mode

Table 23.4.1 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.
```

23.5 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.

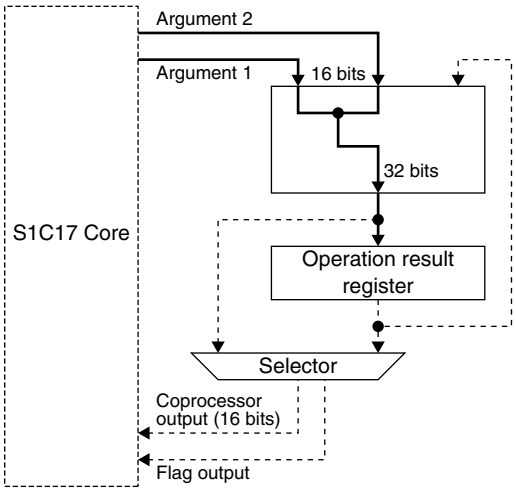


Figure 23.5.1 Data Path in Initialize Mode

Table 23.5.1 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.

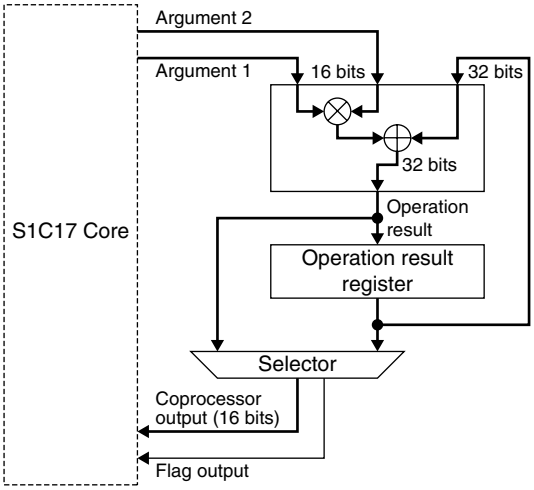


Figure 23.5.2 Data Path in MAC Mode

Table 23.5.2 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 23.5.3 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.

23.6 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.

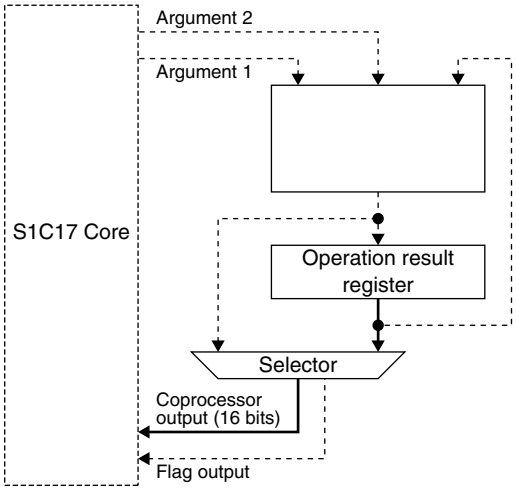


Figure 23.6.1 Data Path in Operation Result Read Mode

Table 23.6.1 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

24 Electrical Characteristics

24.1 Absolute Maximum Ratings

(V_{SS} = 0V)

Item	Symbol	Condition	Rated value	Unit
Core power supply voltage	LV_{DD}		-0.3 to 2.5 *2	V
I/O power supply voltage	HV_{DD}		-0.3 to 7.0 *2	V
Analog power supply voltage	AV_{DD}		-0.3 to 7.0	V
Flash programming voltage	V_{PP}		8	V
Input voltage	V_I		-0.3 to HV_{DD} + 0.5	V
Output voltage	V_O		-0.3 to HV_{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
Permissible loss *1	V_O		200	mW
Storage temperature	Tstg		-65 to 125	°C
Soldering temperature/time	Tsol		260°C, 10 seconds (lead section)	–

*1 In case of plastic package
*2 $HV_{DD} \geq LV_{DD}$

24.2 Recommended Operating Conditions

S1C17554

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Core power supply voltage	LV_{DD}		1.65	1.80	1.95	V
I/O power supply voltage	HV_{DD}		1.65	–	5.5	V
Analog power supply voltage	AV_{DD}	When ADC is used	2.7	–	5.5	V
		When ADC is not used	1.65	–	5.5	V
Flash programming voltage	$V_{PP}P$		6.8	7.0	7.2	V
Flash erasing voltage	$V_{PP}E$		7.3	7.5	7.7	V
Operating frequency	f_{OSC3}	Crystal/ceramic oscillation	0.2	–	24	MHz
	f_{OSC1}	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

S1C17564

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Core power supply voltage	LV_{DD}	When the regulator is not used	1.65	1.80	1.95	V
Regulator input voltage	V_{IN}	REGEN = V_{IN}	2.0	–	5.5	V
I/O power supply voltage	HV_{DD}	When the regulator is not used	1.65	–	5.5	V
		When the regulator is used	2.0	–	5.5	V
Analog power supply voltage	AV_{DD}	When ADC is used	2.7	–	5.5	V
		When ADC is not used	1.65	–	5.5	V
Flash programming voltage	$V_{PP}P$		6.8	7.0	7.2	V
Flash erasing voltage	$V_{PP}E$		7.3	7.5	7.7	V
Operating frequency	f_{OSC3}	Crystal/ceramic oscillation	0.2	–	24	MHz
	f_{OSC1}	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

24.3 Current Consumption

Unless otherwise specified: LV_{DD} = 1.8V, HV_{DD} = AV_{DD} = 5.5V, V_{SS} = 0V, Ta = 25°C, PCKEN[1:0] = 0x3 (ON), RDWAIT[1:0] = 0x1 (no wait), CCLKGR[1:0] = 0x0 (gear ratio 1/1)

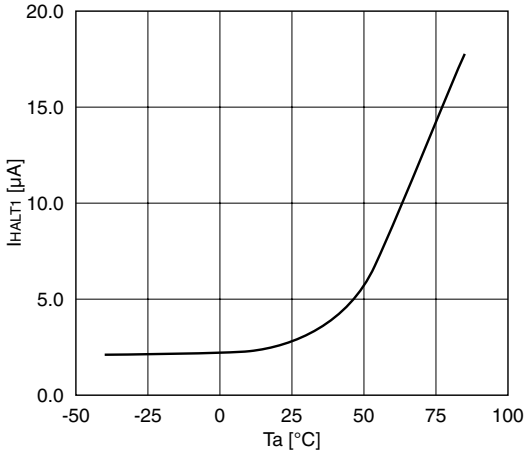
Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Current consumption in SLEEP mode	I_{SLP}	OSC1 = OFF, IOSC = OFF, OSC3 = OFF	–	0.8	4.5	µA
Current consumption in HALT mode	I_{HALT1}	OSC1 = 32kHz, IOSC = OFF, OSC3 = OFF, PCKEN[1:0] = 0x0 (OFF)	–	2.7	7	µA
		OSC1 = 32kHz, IOSC = OFF, OSC3 = OFF	–	4.7	9	µA
	I_{HALT2}	OSC1 = 32kHz, IOSC = OFF, OSC3 = 8MHz (ceramic)	–	650	750	µA
		OSC1 = 32kHz, IOSC = OFF, OSC3 = 24MHz (ceramic)		2000	2300	µA
	I_{HALT3} *2	OSC1 = 32kHz, IOSC = ON (12MHz), OSC3 = OFF	–	1400	1600	µA
Current consumption during execution *1	I_{EXE1}	OSC1 = 32kHz, IOSC = OFF, OSC3 = OFF, CPU = OSC1	–	16	20	µA
		OSC1 = 32kHz, IOSC = OFF, OSC3 = OFF, CCLKGR[1:0] = 0x2 (gear ratio 1/4), CPU = OSC1	–	9.5	14	µA
	I_{EXE2}	OSC1 = 32kHz, IOSC = OFF, OSC3 = 1MHz (ceramic), CPU = OSC3	–	450	550	µA
		OSC1 = 32kHz, IOSC = OFF, OSC3 = 8MHz (ceramic), CPU = OSC3	–	3000	3400	µA
		OSC1 = 32kHz, IOSC = OFF, OSC3 = 24MHz (ceramic), CPU = OSC3, RDWAIT[1:0] = 0x3 (2 wait)		6000	6900	µA
		OSC1 = 32kHz, IOSC = OFF, OSC3 = 8MHz (ceramic), CCLKGR[1:0] = 0x2 (gear ratio 1/4), CPU = OSC3	–	1600	2000	µA
	I_{EXE3} *2	OSC1 = 32kHz, IOSC = ON (2MHz), OSC3 = OFF, CPU = IOSC	–	1000	1300	µA
		OSC1 = 32kHz, IOSC = ON (12MHz), OSC3 = OFF, CPU = IOSC	–	4500	5300	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.

*2 S1C17564

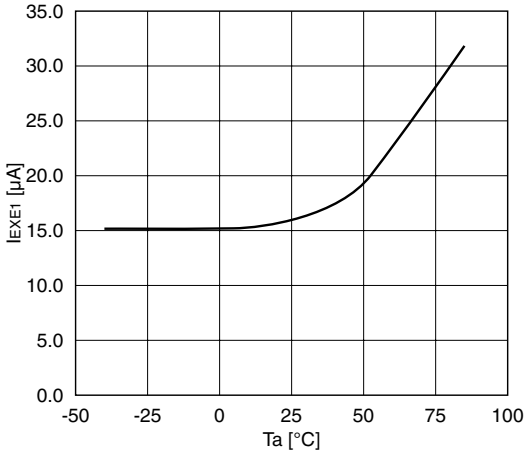
Current consumption-temperature characteristic in HALT mode (OSC1 operation)

OSC1 = 32.768kHz, IOSC = OFF, OSC3 = OFF,
PCKEN[1:0] = 0x0 (OFF), CCLKGR[1:0] = 0x0 (1/1), Typ. value



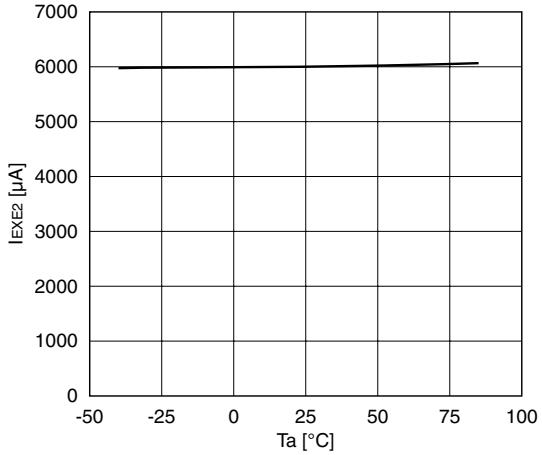
Current consumption-temperature characteristic during execution with OSC1

OSC1 = 32.768kHz, IOSC = OFF, OSC3 = OFF,
PCKEN[1:0] = 0x3 (ON), CCLKGR[1:0] = 0x0 (1/1), Typ. value



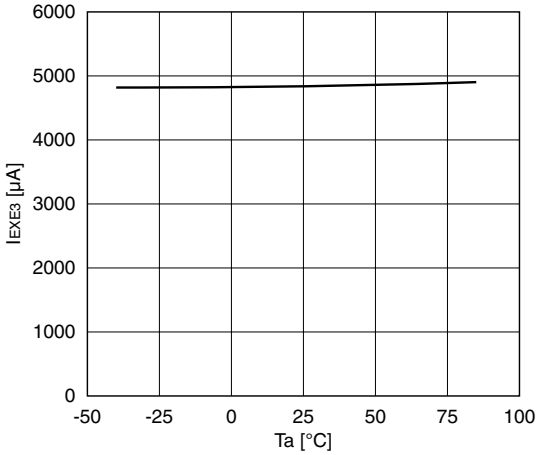
Current consumption-temperature characteristic during execution with OSC3

OSC1 = ON, IOSC = OFF, OSC3 = ON (24MHz, ceramic), PCKEN[1:0] = 0x3 (ON), Typ. value



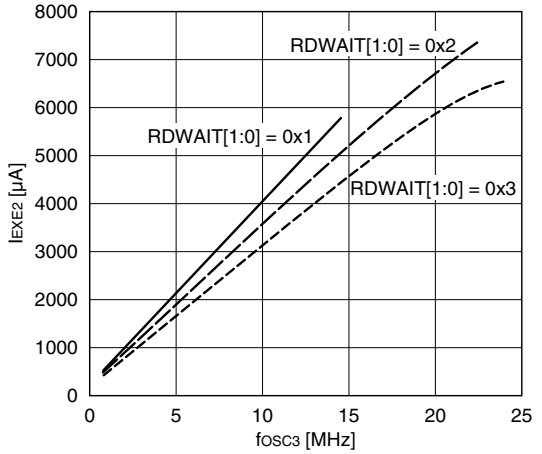
Current consumption-temperature characteristic during execution with IOSC

OSC1 = ON, IOSC = ON (12MHz), OSC3 = OFF, PCKEN[1:0] = 0x3 (ON), Typ. value



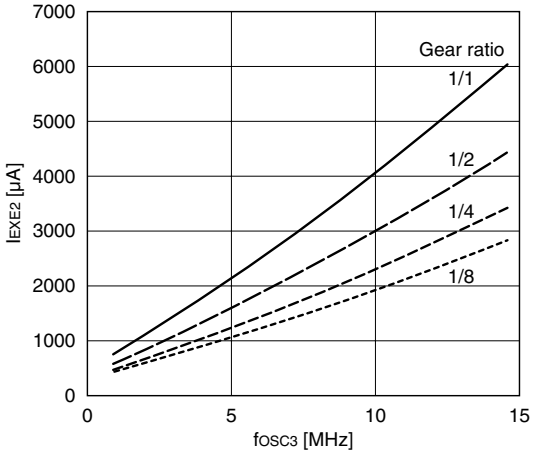
Flash read current consumption characteristic (OSC3 operation)

OSC1 = ON, IOSC = OFF, OSC3 = ON, PCKEN[1:0] = 0x3 (ON), Ta = 25°C, Typ. value



Clock gear characteristic (OSC3 operation)

OSC1 = ON, IOSC = OFF, OSC3 = ON, PCKEN[1:0] = 0x3 (ON), RDWAIT[1:0] = 0x1 (no wait), Ta = 25°C, Typ. value



24.4 DC Regulator Characteristics

DC regulator current consumption

Unless otherwise specified: Ta = 25°C, No load

Item	Symbol	Condition		Min.	Typ.	Max.	Unit
DC regulator current consumption *1	I_{REG1}	Economy mode	V_{IN} = 2.0 to 3.6V	–	0.4	0.7	µA
			V_{IN} = 3.6 to 5.5V	–	0.5	0.8	µA
	I_{REG2}	Normal mode	V_{IN} = 2.0 to 5.5V	–	22	33	µA

*1 The current value should be added to the current consumption during SLEEP/HALT/execution when the DC regulator is used.

24.5 Oscillation Characteristics

Oscillation characteristics change depending on conditions (board pattern, components used, etc.). Use the following characteristics as reference values.

OSC1 crystal oscillation

Unless otherwise specified: LV_{DD} = 1.65 to 1.95V, V_{SS} = 0V, Ta = 25°C, C_{G1} = C_{D1} = 10pF external, R_{f1} = 10MΩ external

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Oscillation start time *1 *2	t_{sta}		–	–	3	s

*1 Crystal resonator = MC-146: manufactured by EPSON TOYOCOM (R_1 = 65kΩ Max., C_L = 12.5pF)
*2 The oscillation start time varies according to the crystal resonator used and the C_{G1} and C_{D1} values.

OSC3 crystal oscillation

Unless otherwise specified: LV_{DD} = 1.65 to 1.95V, V_{SS} = 0V, Ta = 25°C, C_{G3} = C_{D3} = 8pF external, R_{f3} = 1MΩ external

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Oscillation start time *1 *2	t_{sta}		–	–	20	ms

*1 Crystal resonator = MA-406: manufactured by EPSON TOYOCOM (R_1 = 150Ω, C_L = 10pF)
*2 The oscillation start time varies according to the crystal resonator used and the C_{G3} and C_{D3} values.

OSC3 ceramic oscillation

Unless otherwise specified: LV_{DD} = 1.65 to 1.95V, V_{SS} = 0V, Ta = 25°C, R_{f3} = 1MΩ external

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Oscillation start time *1 *2	t_{sta}		–	–	1	ms

*1 Ceramic resonator = CSTCG24M0V51: manufactured by Murata Manufacturing Co., Ltd. (C_{G3} = C_{D3} = 5pF built-in)
*2 The oscillation start time varies according to the ceramic resonator used and the C_{G3} and C_{D3} values.

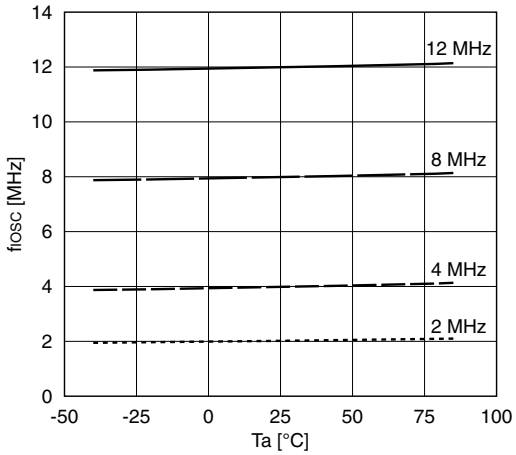
IOSC oscillation

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

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Oscillation start time	t_{sta}		–	–	5	µs
Oscillation frequency	f_{IOSC}	IOSCSEL[1:0] = 0x3	1.86	2	2.14	MHz
		IOSCSEL[1:0] = 0x2	3.76	4	4.24	MHz
		IOSCSEL[1:0] = 0x0	6.8	8	9.2	MHz
		IOSCSEL[1:0] = 0x1	11.4	12	12.6	MHz

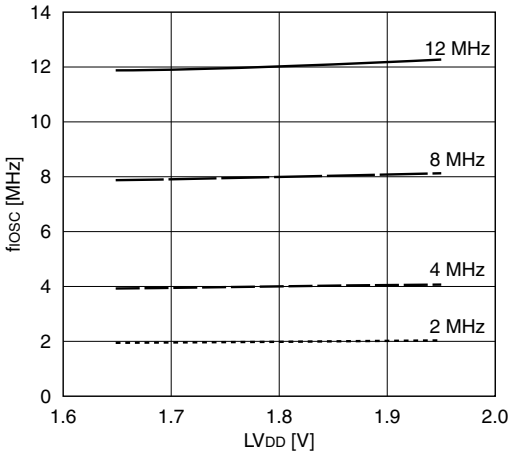
IOSC oscillation frequency-temperature characteristic

Typ. value



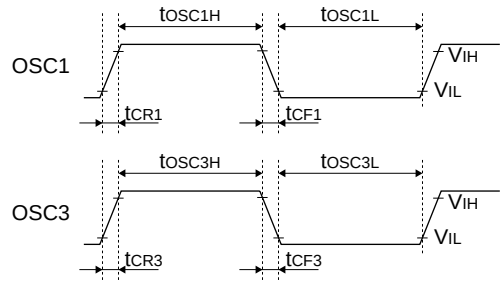
IOSC oscillation frequency-voltage characteristic

Typ. value



24.6 External Clock Input Characteristics

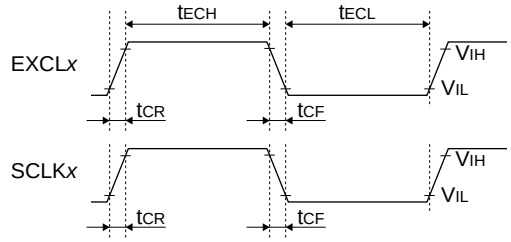
OSC1, OSC3



Unless otherwise specified: LV_{DD} = 1.65 to 1.95V, V_{SS} = 0V, V_{IH} = 0.8LV_{DD}, V_{IL} = 0.2LV_{DD}, Ta = -40 to 85°C

Item	Symbol	Min.	Typ.	Max.	Unit
OSC1 input High pulse width	t_{OSC1H}	9	–	–	µs
OSC1 input Low pulse width	t_{OSC1L}	9	–	–	µs
OSC1 input rise time	t_{CR1}	–	–	200	ns
OSC1 input fall time	t_{CF1}	–	–	200	ns
OSC3 input High pulse width	t_{OSC3H}	19	–	–	ns
OSC3 input Low pulse width	t_{OSC3L}	19	–	–	ns
OSC3 input rise time	t_{CR3}	–	–	10	ns
OSC3 input fall time	t_{CF3}	–	–	10	ns

EXCL, SCLK



Unless otherwise specified: HV_{DD} = 1.65 to 5.5V, V_{SS} = 0V, V_{IH} = 0.8HV_{DD}, V_{IL} = 0.2HV_{DD}, Ta = -40 to 85°C

Item	Symbol	Min.	Typ.	Max.	Unit
EXCL*x* input High pulse width	t_{ECH}	21	–	–	ns
EXCL*x* input Low pulse width	t_{ECL}	21	–	–	ns
UART transfer rate	R_U	–	–	960000	bps
UART transfer rate (IrDA mode)	R_{UIrDA}	–	–	115200	bps
Input rise time	t_{CR}	–	–	80	ns
Input fall time	t_{CF}	–	–	80	ns

24.7 System Clock Characteristics

Unless otherwise specified: LV_{DD} = 1.65 to 1.95V, V_{SS} = 0V

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
System clock frequency	f_{SYSCLK}	RDWAIT[1:0] = 0x1, Ta = 0 to 70°C	–	–	14.75	MHz
		RDWAIT[1:0] = 0x2, Ta = 0 to 70°C	–	–	22.5	MHz
		RDWAIT[1:0] = 0x3, Ta = 0 to 70°C	–	–	24	MHz
		RDWAIT[1:0] = 0x1, Ta = -40 to 85°C	–	–	14.6	MHz
		RDWAIT[1:0] = 0x2, Ta = -40 to 85°C	–	–	22.5	MHz
		RDWAIT[1:0] = 0x3, Ta = -40 to 85°C	–	–	24	MHz

24.8 Input/Output Pin Characteristics

Common characteristics

Unless otherwise specified: HV_{DD} = AV_{DD} = 1.65 to 5.5V, LV_{DD} = 1.65 to 1.95V, V_{SS} = 0V, Ta = -40 to 85°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Pin capacitance	C_{IN}	f = 1MHz, HV_{DD} = 0V	–	–	8	pF
Reset low pulse width	t_{SR}	V_{IH} = 0.8HV_{DD}, V_{IL} = 0.2HV_{DD}	100	–	–	µs
Input leakage current	I_{LI}	P*xx*, #RESET	-100	–	100	nA

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

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
High level output current	I_{OH}	V_{OH} = HV_{DD} - 0.4V, HV_{DD} = Min.	-0.4	–	–	mA
Low level output current	I_{OL}	V_{OL} = 0.4V, HV_{DD} = Min.	0.4	–	–	mA
Positive trigger voltage	V_{T+}	LVCMOS Schmitt	0.66	–	1.36	V
Negative trigger voltage	V_{T-}	LVCMOS Schmitt	0.42	–	1.07	V
Hysteresis voltage	ΔV	LVCMOS Schmitt	0.17	–	–	V
Pull-up resistance	R_{PU}	Type 1, V_I = 0V	120	300	1200	kΩ
Pull-down resistance	R_{PD}	Type 1, V_I = HV_{DD}	120	300	1200	kΩ

Unless otherwise specified: HV_{DD} = AV_{DD} = 2.2 to 2.6V, LV_{DD} = 1.65 to 1.95V, V_{SS} = 0V, Ta = -40 to 85°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
High level output current	I_{OH}	V_{OH} = HV_{DD} - 0.4V, HV_{DD} = Min.	-0.8	–	–	mA
Low level output current	I_{OL}	V_{OL} = 0.4V, HV_{DD} = Min.	0.8	–	–	mA
Positive trigger voltage	V_{T+}	LVCMOS Schmitt	0.88	–	1.82	V
Negative trigger voltage	V_{T-}	LVCMOS Schmitt	0.55	–	1.43	V
Hysteresis voltage	ΔV	LVCMOS Schmitt	0.22	–	–	V
Pull-up resistance	R_{PU}	Type 1, V_I = 0V	60	150	450	kΩ
Pull-down resistance	R_{PD}	Type 1, V_I = HV_{DD}	60	150	450	kΩ

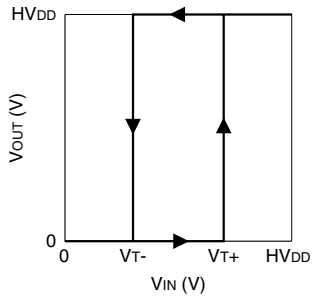
Unless otherwise specified: HV_{DD} = AV_{DD} = 3.0 to 3.6V, LV_{DD} = 1.65 to 1.95V, V_{SS} = 0V, Ta = -40 to 85°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
High level output current	I_{OH}	V_{OH} = HV_{DD} - 0.4V, HV_{DD} = Min.	-1.4	–	–	mA
Low level output current	I_{OL}	V_{OL} = 0.4V, HV_{DD} = Min.	1.4	–	–	mA
Positive trigger voltage	V_{T+}	LVCMOS Schmitt	1.2	–	2.52	V
Negative trigger voltage	V_{T-}	LVCMOS Schmitt	0.75	–	1.98	V
Hysteresis voltage	ΔV	LVCMOS Schmitt	0.3	–	–	V
Pull-up resistance	R_{PU}	Type 1, V_I = 0V	32	80	224	kΩ
Pull-down resistance	R_{PD}	Type 1, V_I = HV_{DD}	32	80	224	kΩ

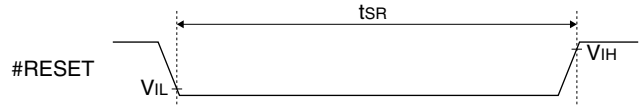
Unless otherwise specified: HV_{DD} = AV_{DD} = 4.5 to 5.5V, LV_{DD} = 1.65 to 1.95V, V_{SS} = 0V, Ta = -40 to 85°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
High level output current	I_{OH}	V_{OH} = HV_{DD} - 0.4V, HV_{DD} = Min.	-2	–	–	mA
Low level output current	I_{OL}	V_{OL} = 0.4V, HV_{DD} = Min.	2	–	–	mA
Positive trigger voltage	V_{T+}	LVCMOS Schmitt	2	–	4	V
Negative trigger voltage	V_{T-}	LVCMOS Schmitt	0.8	–	3.1	V
Hysteresis voltage	ΔV	LVCMOS Schmitt	0.3	–	–	V
Pull-up resistance	R_{PU}	Type 1, V_I = 0V	20	50	120	kΩ
Pull-down resistance	R_{PD}	Type 1, V_I = HV_{DD}	20	50	120	kΩ

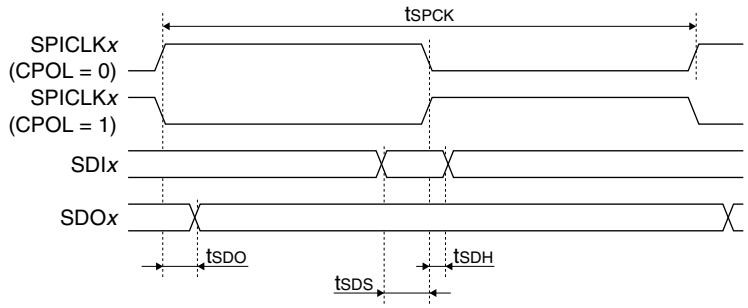
Schmitt input threshold voltage



Reset pulse



24.9 SPI Characteristics



Master mode

Unless otherwise specified: HV_{DD} = 1.65 to 5.5V, V_{SS} = 0V, Ta = -40 to 85°C

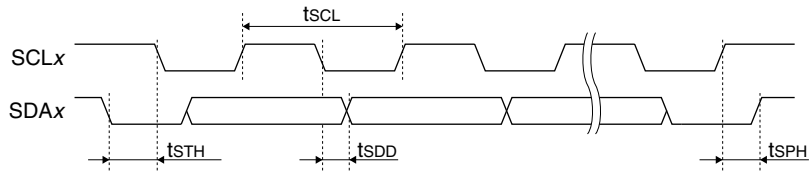
Item	Symbol	Min.	Typ.	Max.	Unit
SPICLK*x* cycle time	t_{SPCK}	250	–	–	ns
SDI*x* setup time	t_{SDS}	70	–	–	ns
SDI*x* hold time	t_{SDH}	10	–	–	ns
SDO*x* output delay time	t_{SDO}	–	–	20	ns

Slave mode

Unless otherwise specified: HV_{DD} = 1.65 to 5.5V, V_{SS} = 0V, Ta = -40 to 85°C

Item	Symbol	Min.	Typ.	Max.	Unit
SPICLK*x* cycle time	t_{SPCK}	250	–	–	ns
SDI*x* setup time	t_{SDS}	10	–	–	ns
SDI*x* hold time	t_{SDH}	10	–	–	ns
SDO*x* output delay time	t_{SDO}	–	–	80	ns

24.10 I^2C Characteristics



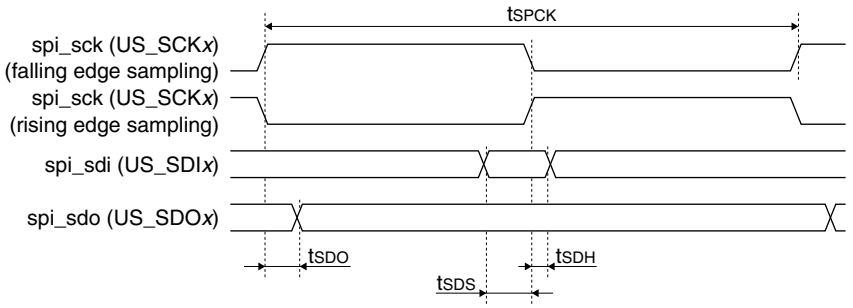
Unless otherwise specified: HV_{DD} = 1.65 to 5.5V, V_{SS} = 0V, Ta = -40 to 85°C

Item	Symbol	Min.	Typ.	Max.	Unit
SCL cycle time	t_{SCL}	2500	–	–	ns
Start condition hold time	t_{STH}	1/f_{SYS}	–	–	ns
Data output delay time	t_{SDD}	1/f_{SYS}	–	–	ns
Stop condition hold time	t_{SPH}	1/f_{SYS}	–	–	ns

∗ f_{SYS}: System operating clock frequency

24.11 USI Characteristics (S1C17564)

SPI master mode



SPI master mode (8 or 9 bits, normal mode)

Unless otherwise specified: HV_{DD} = 1.65 to 5.5V, V_{SS} = 0V, Ta = -40 to 85°C

Item	Symbol	Min.	Typ.	Max.	Unit
spi_sck cycle time	t_{SPCK}	85 + t_{PCLK}	–	–	ns
spi_sdi setup time	t_{SDS}	85 + t_{PCLK}	–	–	ns
spi_sdi hold time	t_{SDH}	0	–	–	ns
spi_sdo output delay time	t_{SDO}	–	–	10	ns

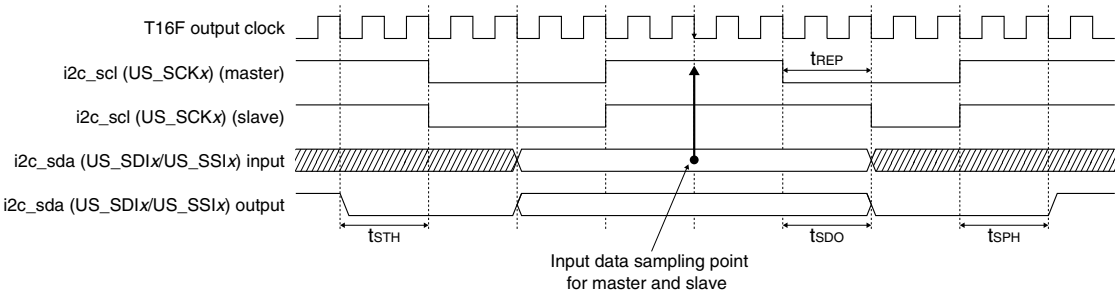
t_{PCLK}: PCLK (peripheral module clock supplied from the CLG) clock cycle time

SPI master mode (8 or 9 bits, fast mode)

Unless otherwise specified: HV_{DD} = 1.65 to 5.5V, V_{SS} = 0V, Ta = -40 to 85°C

Item	Symbol	Min.	Typ.	Max.	Unit
spi_sck cycle time	t_{SPCK}	85	–	–	ns
spi_sdi setup time	t_{SDS}	85	–	–	ns
spi_sdi hold time	t_{SDH}	0	–	–	ns
spi_sdo output delay time	t_{SDO}	–	–	10	ns

I²C master/slave mode



I²C master mode

Unless otherwise specified: HV_{DD} = 1.65 to 5.5V, V_{SS} = 0V, Ta = -40 to 85°C

Item	Symbol	Min.	Typ.	Max.	Unit
i2c_scl cycle time	t_{SCL}	2500	–	–	ns
i2c_sda output delay time	t_{SDO}	–	–	2∗t_{T16}	ns
Start condition hold time	t_{STH}	4∗t_{T16}	–	–	ns
Stop condition hold time	t_{SPH}	3∗t_{T16}	–	–	ns

I²C slave mode

Unless otherwise specified: HV_{DD} = 1.65 to 5.5V, V_{SS} = 0V, Ta = -40 to 85°C

Item	Symbol	Min.	Typ.	Max.	Unit
i2c_scl cycle time	t_{SCL}	2500	–	–	ns
i2c_scl input clock response delay time	t_{REP}	–	–	4∗t_{T16}	ns
i2c_sda output delay time	t_{SDO}	–	–	2∗t_{T16}	ns
Start condition hold time	t_{STH}	7∗t_{PCLK}	–	–	ns
Stop condition hold time	t_{SPH}	7∗t_{PCLK}	–	–	ns

t_{PCLK}: PCLK (peripheral module clock supplied from the CLG) clock cycle time
t_{T16} = T16F output clock cycle time

24.12 A/D Converter Characteristics

Analog characteristics

Unless otherwise specified: AV_{DD} = 2.7 to 5.5V, V_{SS} = 0V, Ta = -40 to 85°C, ADST[2:0] = 0x7 (9 cycles)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Resolution	–		–	10	–	bit
A/D conversion clock	f_{ADCLK}		10	–	2000	kHz
Sampling rate *1	f_{SMP}		0.5	–	100	ksps
Zero-scale error	E_{ZS}	AV_{DD} = 2.7 to 3.6V	–	–	±3	LSB
		AV_{DD} = 3.6 to 5.5V	–	–	±5	LSB
Full-scale error	E_{FS}	AV_{DD} = 2.7 to 3.6V	–	–	±3	LSB
		AV_{DD} = 3.6 to 5.5V	–	–	±5	LSB
Integral linearity error *2	E_{INL}		–	–	±1.5	LSB
Differential linearity error	E_{DNL}		–	–	±1	LSB
Analog input resistance	R_{AIN}		–	12	20	kΩ
Analog input capacitance	C_{AIN}		–	16	18	pF

*1 Condition for Max. value: A/D converter clock frequency f_{ADCLK} = 2MHz. Condition for Min. value: A/D converter clock frequency f_{ADCLK} = 10kHz.
*2 Integral linearity error is measured at the end point line.
*3 These characteristics assume that FSEL[1:0] and XPD[1:0] are set correctly according to the AV_{DD} voltage.

A/D converter current consumption

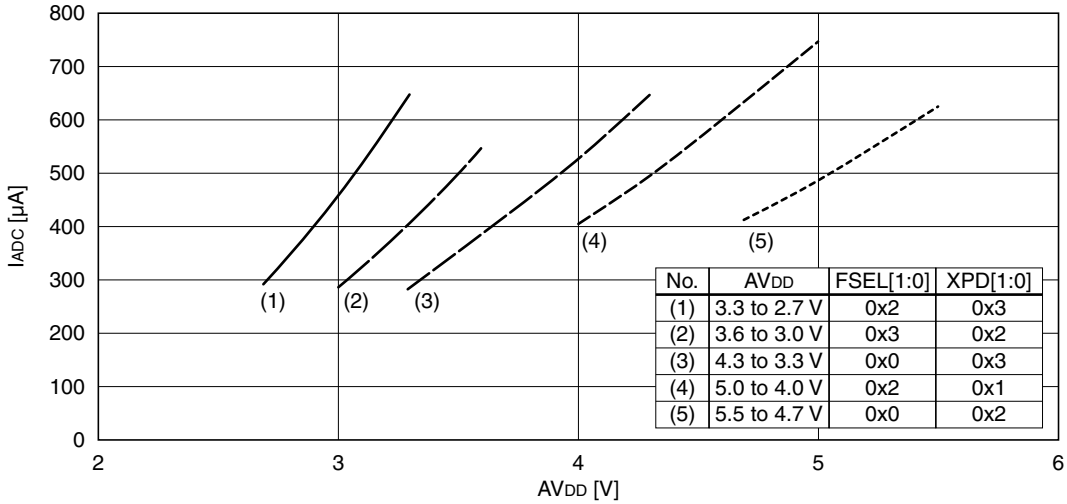
Unless otherwise specified: Ta = 25°C, ADST[2:0] = 0x7 (9 cycles), PCKEN[1:0] = 0x3 (ON), A_{IN} = AV_{DD}/2V, f_{SMP} = 100ksps

Item	Symbol	Condition			Min.	Typ.	Max.	Unit
		FSEL[1:0]	XPD[1:0]	AV_{DD}				
A/D converter operating current *1	I_{ADC}	0x2	0x3	2.7	–	290	–	µA
				3	–	450	–	µA
				3.3	–	640	–	µA
		0x3	0x2	3	–	280	–	µA
				3.3	–	400	–	µA
				3.6	–	540	–	µA
		0x0	0x3	3.3	–	280	–	µÅ
				3.6	–	380	–	µA
				4	–	520	–	µA
				4.3	–	640	–	µA
		0x2	0x1	4	–	400	–	µA
				4.3	–	490	–	µA
				4.7	–	630	–	µA
				5	–	740	–	µA
		0x0	0x2	4.7	–	410	–	µA
				5	–	480	–	µA
				5.5	–	620	–	µA

*1 This value is added to the current consumption in HALT mode (only when PCKEN[1:0] = 0x3 (ON)) or current consumption during execution when the A/D converter is active.

A/D converter current consumption-voltage characteristic

OSC3 = 4MHz, OSC1 = 32kHz, sampling = 9 clocks, AV_{DD} = HV_{DD}, LV_{DD} = 1.95V, PCKEN[1:0] = 0x3 (ON), Ta = 25°C, Typ. value



No.	AV_{DD}	FSEL[1:0]	XPD[1:0]
(1)	3.3 to 2.7 V	0x2	0x3
(2)	3.6 to 3.0 V	0x3	0x2
(3)	4.3 to 3.3 V	0x0	0x3
(4)	5.0 to 4.0 V	0x2	0x1
(5)	5.5 to 4.7 V	0x0	0x2

24.13 Flash Memory Characteristics

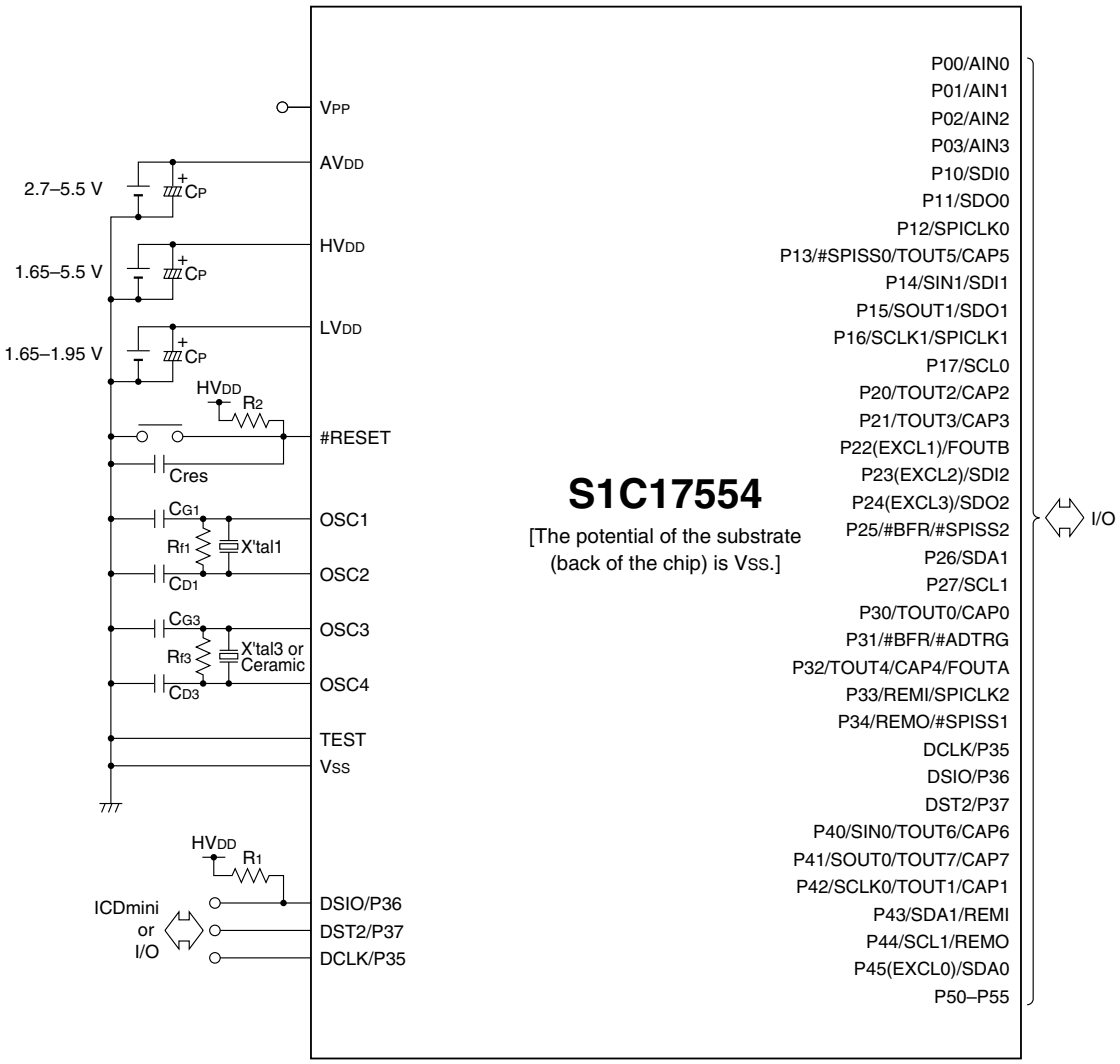
Unless otherwise specified: LV_{DD} = 1.65 to 1.95V, V_{PP} = 7.0V (for programming)/7.5V (for erasing), V_{SS} = 0V, Ta = 10 to 40°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Program count *1	C_{FEP}	Programmed data is guaranteed to be retained for 10 years.	10 *2	–	–	times

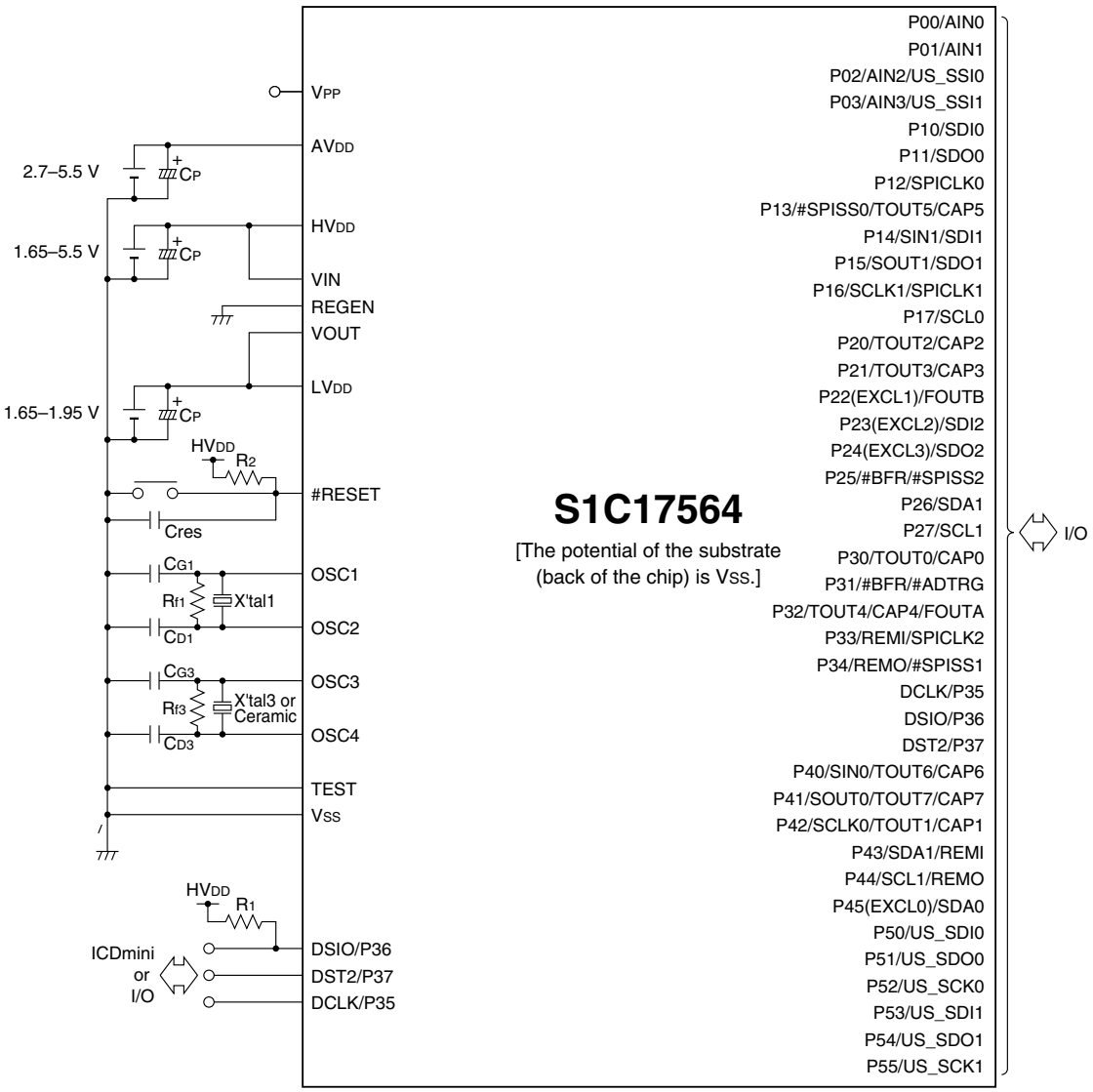
*1 Assumed that Erasing + Programming as count of 1. The count includes programming in the factory.
*2 Applied only when FLS V1.0 or later is used.

25 Basic External Connection Diagram

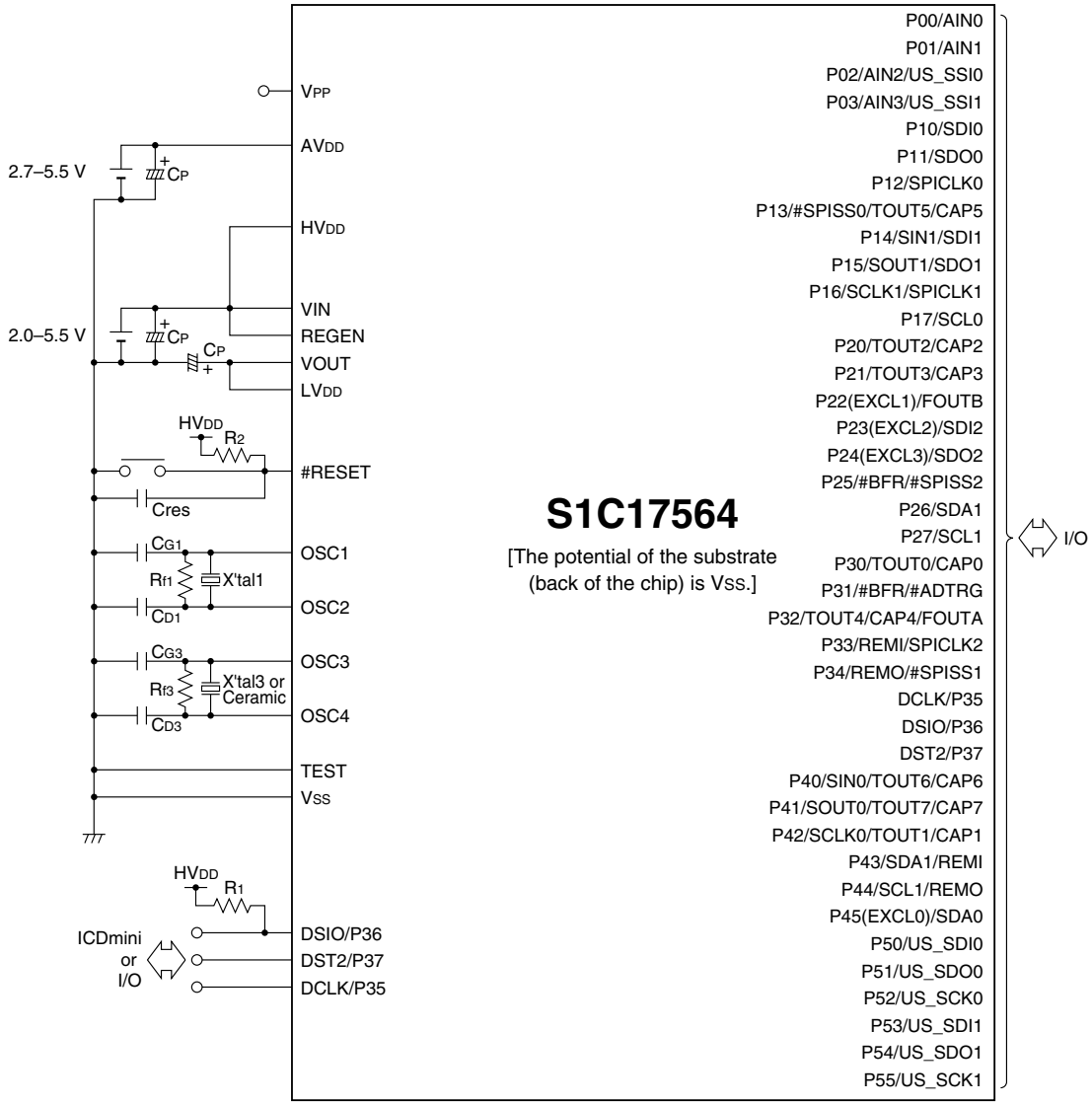
S1C17554



S1C17564 (when the regulator is not used)



S1C17564 (when the regulator is used)



Recommended values for external parts

Symbol	Name	Recommended value
X'tal1	Crystal resonator	32.768 kHz
C_{G1} *1	Gate capacitor	5 to 25 pF
C_{D1} *1	Drain capacitor	5 to 25 pF
R_{f1}	Feedback resistor	10 MΩ
X'tal3	Crystal resonator	0.2 to 24 MHz
Ceramic	Ceramic resonator	0.2 to 24 MHz
C_{G3} *1, 2	Gate capacitor	10 to 30 pF
C_{D3} *1, 2	Drain capacitor	10 to 30 pF
R_{f3}	Feedback resistor	1 MΩ
CP	Capacitor for power supply	1 µF
Cres	Capacitor for #RESET pin	0.47 µF
R_1	Pull-up resistor	10 kΩ
R_2	Pull-up resistor	10 kΩ

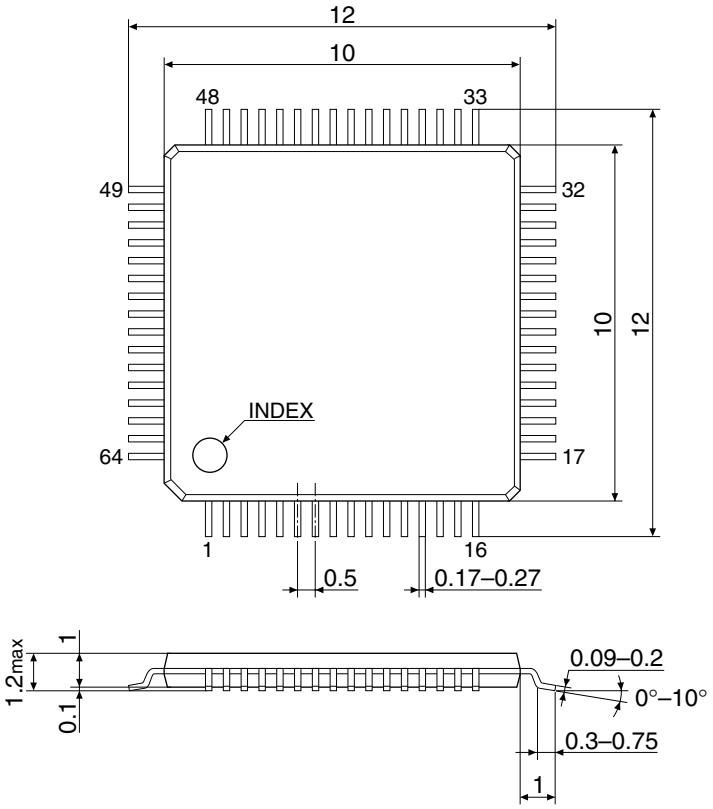
*1 The capacitances listed above include stray capacitance of the board.
Please contact the resonator manufacturer for an optimum capacitance.

*2 There are ceramic resonators with built-in capacitance available.

26 Package

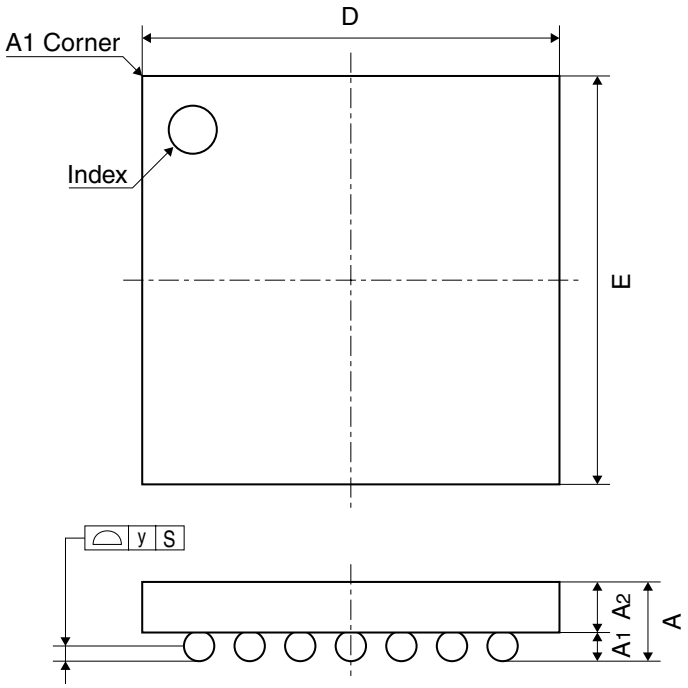
TQFP13-64pin package (S1C17554, S1C17564)

(Unit: mm)

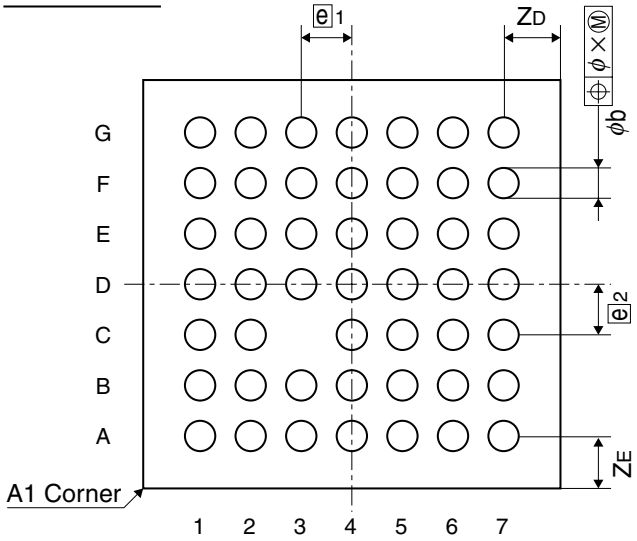


WCSP-48 package (S1C17554)

Top View



Bottom View



Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	3.037	3.137	3.197
E	3.037	3.137	3.197
A	0.630	0.690	0.745
A_1	0.170	0.200	0.225
A_2	0.460	0.490	0.520
e1	–	0.40	–
e2	–	0.40	–
b	0.200	0.230	0.260
X	–	–	0.08
y	–	–	0.05
Z_D	–	0.369	–
Z_E	–	0.369	–

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.
UART (with IrDA) Ch.1 (8-bit device)	0x4120	UART_ST1	UART Ch.1 Status Register	Indicates transfer, buffer and error statuses.
	0x4121	UART_TXD1	UART Ch.1 Transmit Data Register	Transmit data
	0x4122	UART_RXD1	UART Ch.1 Receive Data Register	Receive data
	0x4123	UART_MOD1	UART Ch.1 Mode Register	Sets transfer data format.
	0x4124	UART_CTL1	UART Ch.1 Control Register	Controls data transfer.
	0x4125	UART_EXP1	UART Ch.1 Expansion Register	Sets IrDA mode.
	0x4126	UART_BR1	UART Ch.1 Baud Rate Register	Sets baud rate.
	0x4127	UART_FMD1	UART Ch.1 Fine Mode Register	Sets fine mode.
Fine mode 16-bit timer Ch. 0 (16-bit device)	0x4200	T16F_CLK0	T16F Ch.0 Count Clock Select Register	Selects a count clock.
	0x4202	T16F_TR0	T16F Ch.0 Reload Data Register	Sets reload data.
	0x4204	T16F_TC0	T16F Ch.0 Counter Data Register	Counter data
	0x4206	T16F_CTL0	T16F Ch.0 Control Register	Sets the timer mode and starts/stops the timer.
	0x4208	T16F_INT0	T16F Ch.0 Interrupt Control Register	Controls the interrupt.
16-bit timer Ch. 0 (16-bit device)	0x4220	T16_CLK0	T16 Ch.0 Count Clock Select Register	Selects a count clock.
	0x4222	T16_TR0	T16 Ch.0 Reload Data Register	Sets reload data.
	0x4224	T16_TC0	T16 Ch.0 Counter Data Register	Counter data
	0x4226	T16_CTL0	T16 Ch.0 Control Register	Sets the timer mode and starts/stops the timer.
	0x4228	T16_INT0	T16 Ch.0 Interrupt Control Register	Controls the interrupt.
16-bit timer Ch. 1 (16-bit device)	0x4240	T16_CLK1	T16 Ch.1 Count Clock Select Register	Selects a count clock.
	0x4242	T16_TR1	T16 Ch.1 Reload Data Register	Sets reload data.
	0x4244	T16_TC1	T16 Ch.1 Counter Data Register	Counter data
	0x4246	T16_CTL1	T16 Ch.1 Control Register	Sets the timer mode and starts/stops the timer.
	0x4248	T16_INT1	T16 Ch.1 Interrupt Control Register	Controls the interrupt.
16-bit timer Ch. 2 (16-bit device)	0x4260	T16_CLK2	T16 Ch.2 Count Clock Select Register	Selects a count clock.
	0x4262	T16_TR2	T16 Ch.2 Reload Data Register	Sets reload data.
	0x4264	T16_TC2	T16 Ch.2 Counter Data Register	Counter data
	0x4266	T16_CTL2	T16 Ch.2 Control Register	Sets the timer mode and starts/stops the timer.
	0x4268	T16_INT2	T16 Ch.2 Interrupt Control Register	Controls the interrupt.
Fine mode 16-bit timer Ch. 1 (16-bit device)	0x4280	T16F_CLK1	T16F Ch.1 Count Clock Select Register	Selects a count clock.
	0x4282	T16F_TR1	T16F Ch.1 Reload Data Register	Sets reload data.
	0x4284	T16F_TC1	T16F Ch.1 Counter Data Register	Counter data
	0x4286	T16F_CTL1	T16F Ch.1 Control Register	Sets the timer mode and starts/stops the timer.
	0x4288	T16F_INT1	T16F Ch.1 Interrupt Control Register	Controls the interrupt.
Interrupt controller (16-bit device)	0x4306	ITC_LV0	Interrupt Level Setup Register 0	Sets the P0 and P1 interrupt levels.
	0x4308	ITC_LV1	Interrupt Level Setup Register 1	Sets the SWT and CT interrupt levels.
	0x430a	ITC_LV2	Interrupt Level Setup Register 2	Sets the T16A Ch.2 and P4 interrupt levels.
	0x430c	ITC_LV3	Interrupt Level Setup Register 3	Sets the SPI Ch.2 and T16A Ch.0 interrupt levels.
	0x430e	ITC_LV4	Interrupt Level Setup Register 4	Sets the T16F Ch.0 & Ch.1/USI Ch.0 & Ch.1 and T16 Ch.0 interrupt levels.
	0x4310	ITC_LV5	Interrupt Level Setup Register 5	Sets the T16 Ch.1 and T16 Ch.2/T16A Ch.3 interrupt levels.
	0x4312	ITC_LV6	Interrupt Level Setup Register 6	Sets the UART Ch.0 and Ch.1 interrupt levels.
	0x4314	ITC_LV7	Interrupt Level Setup Register 7	Sets the SPI Ch.0 and I2CM interrupt levels.
	0x4316	ITC_LV8	Interrupt Level Setup Register 8	Sets the REMC/SPI Ch.1 and T16A Ch.1 interrupt levels.
	0x4318	ITC_LV9	Interrupt Level Setup Register 9	Sets the ADC10 and P5 interrupt levels.
	0x431a	ITC_LV10	Interrupt Level Setup Register 10	Sets the P2 and P3 interrupt levels.
	0x431c	ITC_LV11	Interrupt Level Setup Register 11	Sets the I2CS interrupt level.

Peripheral	Address	Register name		Function
SPI Ch.0 (16-bit device)	0x4320	SPI_ST0	SPI Ch.0 Status Register	Indicates transfer and buffer statuses.
	0x4322	SPI_TXD0	SPI Ch.0 Transmit Data Register	Transmit data
	0x4324	SPI_RXD0	SPI Ch.0 Receive Data Register	Receive data
	0x4326	SPI_CTL0	SPI Ch.0 Control Register	Sets the SPI mode and enables data transfer.
I²C master (16-bit device)	0x4340	I2CM_EN	I²C Master Enable Register	Enables the I²C master module.
	0x4342	I2CM_CTL	I²C Master Control Register	Controls the I²C master operation and indicates transfer status.
	0x4344	I2CM_DAT	I²C Master Data Register	Transmit/receive data
	0x4346	I2CM_ICTL	I²C Master Interrupt Control Register	Controls the I²C master interrupt.
I²C slave (16-bit device)	0x4360	I2CS_TRNS	I²C Slave Transmit Data Register	I²C slave transmit data
	0x4362	I2CS_RECV	I²C Slave Receive Data Register	I²C slave receive data
	0x4364	I2CS_SADRS	I²C Slave Address Setup Register	Sets the I²C slave address.
	0x4366	I2CS_CTL	I²C Slave Control Register	Controls the I²C slave module.
	0x4368	I2CS_STAT	I²C Slave Status Register	Indicates the I²C bus status.
	0x436a	I2CS_ASTAT	I²C Slave Access Status Register	Indicates the I²C slave access status.
	0x436c	I2CS_ICTL	I²C Slave Interrupt Control Register	Controls the I²C slave interrupt.
SPI Ch.1 (16-bit device)	0x4380	SPI_ST1	SPI Ch.1 Status Register	Indicates transfer and buffer statuses.
	0x4382	SPI_TXD1	SPI Ch.1 Transmit Data Register	Transmit data
	0x4384	SPI_RXD1	SPI Ch.1 Receive Data Register	Receive data
	0x4386	SPI_CTL1	SPI Ch.1 Control Register	Sets the SPI mode and enables data transfer.
SPI Ch.2 (16-bit device)	0x43a0	SPI_ST2	SPI Ch.2 Status Register	Indicates transfer and buffer statuses.
	0x43a2	SPI_TXD2	SPI Ch.2 Transmit Data Register	Transmit data
	0x43a4	SPI_RXD2	SPI Ch.2 Receive Data Register	Receive data
	0x43a6	SPI_CTL2	SPI Ch.2 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.
Stopwatch timer (8-bit device)	0x5020	SWT_CTL	Stopwatch Timer Control Register	Resets and starts/stops the timer.
	0x5021	SWT_BCNT	Stopwatch Timer BCD Counter Register	BCD counter data
	0x5022	SWT_IMSK	Stopwatch Timer Interrupt Mask Register	Enables/disables interrupt.
	0x5023	SWT_IFLG	Stopwatch Timer Interrupt Flag Register	Indicates/resets interrupt occurrence status.
Watchdog timer (8-bit device)	0x5040	WDT_CTL	Watchdog Timer Control Register	Resets and starts/stops the timer.
	0x5041	WDT_ST	Watchdog Timer Status Register	Sets the timer mode and indicates NMI status.
Clock generator (8-bit device) (T16A, UART)	0x5060	CLG_SRC	Clock Source Select Register	Selects the clock source.
	0x5061	CLG_CTL	Oscillation Control Register	Controls oscillation.
	0x5062	CLG_NFEN	Noise Filter Enable Register	Turns oscillation stabilization wait circuit/ noise filter on/off.
	0x5064	CLG_FOUTA	FOUTA Control Register	Controls FOUTA clock output.
	0x5065	CLG_FOUTB	FOUTB Control Register	Controls FOUTB clock output.
	0x5068	T16A_CLK0	T16A Clock Control Register Ch.0	Controls the T16A Ch.0 clock.
	0x5069	T16A_CLK1	T16A Clock Control Register Ch.1	Controls the T16A Ch.1 clock.
	0x506a	T16A_CLK2	T16A Clock Control Register Ch.2	Controls the T16A Ch.2 clock.
	0x506b	T16A_CLK3	T16A Clock Control Register Ch.3	Controls the T16A Ch.3 clock.
	0x506c	UART_CLK0	UART Ch.0 Clock Control Register	Selects the baud rate generator clock.
	0x506d	UART_CLK1	UART Ch.1 Clock Control Register	Selects the baud rate generator clock.
	0x506e	CLG_IOSC	IOSC Control Register	Configures IOSC oscillation frequency.
	0x5080	CLG_PCLK	PCLK Control Register	Controls the PCLK supply.
	0x5081	CLG_CCLK	CCLK Control Register	Configures the CCLK division ratio.
USI Ch.0 (8-bit device)	0x50c0	USI_GCFG0	USI Ch.0 Global Configuration Register	Sets interface and MSB/LSB mode.
	0x50c1	USI_TD0	USI Ch.0 Transmit Data Buffer Register	Transmit data buffer
	0x50c2	USI_RD0	USI Ch.0 Receive Data Buffer Register	Receive data buffer
	0x50c3	USI_UCFG0	USI Ch.0 UART Mode Configuration Register	Sets UART transfer conditions.
	0x50c4	USI_UIE0	USI Ch.0 UART Mode Interrupt Enable Register	Enables interrupts.
	0x50c5	USI_UIF0	USI Ch.0 UART Mode Interrupt Flag Register	Indicates interrupt occurrence status.
	0x50c6	USI_SCFG0	USI Ch.0 SPI Master Mode Configuration Register	Sets SPI transfer conditions.
	0x50c7	USI_SIE0	USI Ch.0 SPI Master Mode Interrupt Enable Register	Enables interrupts.
	0x50c8	USI_SIF0	USI Ch.0 SPI Master Mode Interrupt Flag Register	Indicates interrupt occurrence status.
	0x50ca	USI_IMTG0	USI Ch.0 I²C Master Mode Trigger Register	Starts I²C master operations.
	0x50cb	USI_IMIE0	USI Ch.0 I²C Master Mode Interrupt Enable Register	Enables interrupts.

Peripheral	Address	Register name		Function
USI Ch.0 (8-bit device)	0x50cc	USI_IMIF0	USI Ch.0 I²C Master Mode Interrupt Flag Register	Indicates interrupt occurrence status.
	0x50cd	USI_ISTG0	USI Ch.0 I²C Slave Mode Trigger Register	Starts I²C slave operations.
	0x50ce	USI_ISIE0	USI Ch.0 I²C Slave Mode Interrupt Enable Register	Enables interrupts.
	0x50cf	USI_ISIF0	USI Ch.0 I²C Slave Mode Interrupt Flag Register	Indicates interrupt occurrence status.
USI Ch.1 (8-bit device)	0x50e0	USI_GCFG1	USI Ch.1 Global Configuration Register	Sets interface and MSB/LSB mode.
	0x50e1	USI_TD1	USI Ch.1 Transmit Data Buffer Register	Transmit data buffer
	0x50e2	USI_RD1	USI Ch.1 Receive Data Buffer Register	Receive data buffer
	0x50e3	USI_UCFG1	USI Ch.1 UART Mode Configuration Register	Sets UART transfer conditions.
	0x50e4	USI_UIE1	USI Ch.1 UART Mode Interrupt Enable Register	Enables interrupts.
	0x50e5	USI_UIF1	USI Ch.1 UART Mode Interrupt Flag Register	Indicates interrupt occurrence status.
	0x50e6	USI_SCFG1	USI Ch.1 SPI Master Mode Configuration Register	Sets SPI transfer conditions.
	0x50e7	USI_SIE1	USI Ch.1 SPI Master Mode Interrupt Enable Register	Enables interrupts.
	0x50e8	USI_SIF1	USI Ch.1 SPI Master Mode Interrupt Flag Register	Indicates interrupt occurrence status.
	0x50ea	USI_IMTG1	USI Ch.1 I²C Master Mode Trigger Register	Starts I²C master operations.
	0x50eb	USI_IMIE1	USI Ch.1 I²C Master Mode Interrupt Enable Register	Enables interrupts.
	0x50ec	USI_IMIF1	USI Ch.1 I²C Master Mode Interrupt Flag Register	Indicates interrupt occurrence status.
	0x50ed	USI_ISTG1	USI Ch.1 I²C Slave Mode Trigger Register	Starts I²C slave operations.
	0x50ee	USI_ISIE1	USI Ch.1 I²C Slave Mode Interrupt Enable Register	Enables interrupts.
	0x50ef	USI_ISIF1	USI Ch.1 I²C Slave Mode Interrupt Flag Register	Indicates interrupt occurrence status.
Power generator (8-bit device)	0x5121	VD1_CTL	V_{D1} Control Register	Controls the regulator operation mode.
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.
	0x5215	P1_IMSK	P1 Port Interrupt Mask Register	Enables P1 port interrupts.
	0x5216	P1_EDGE	P1 Port Interrupt Edge Select Register	Selects the signal edge for generating P1 port interrupts.
	0x5217	P1_IFLG	P1 Port Interrupt Flag Register	Indicates/resets the P1 port interrupt occurrence status.
	0x5218	P1_CHAT	P1 Port Chattering Filter Control Register	Controls the P1 port chattering filter.
	0x521a	P1_IEN	P1 Port Input Enable Register	Enables P1 port inputs.
	0x5220	P2_IN	P2 Port Input Data Register	P2 port input data
	0x5221	P2_OUT	P2 Port Output Data Register	P2 port output data
	0x5222	P2_OEN	P2 Output Enable Register	Enables P2 port outputs.
	0x5223	P2_PU	P2 Port Pull-up Control Register	Controls the P2 port pull-up resistor.
	0x5225	P2_IMSK	P2 Port Interrupt Mask Register	Enables P2 port interrupts.
	0x5226	P2_EDGE	P2 Port Interrupt Edge Select Register	Selects the signal edge for generating P2 port interrupts.
	0x5227	P2_IFLG	P2 Port Interrupt Flag Register	Indicates/resets the P2 port interrupt occurrence status.
	0x5228	P2_CHAT	P2 Port Chattering Filter Control Register	Controls the P2 port chattering filter.
	0x522a	P2_IEN	P2 Port Input Enable Register	Enables P2 port inputs.
	0x5230	P3_IN	P3 Port Input Data Register	P3 port input data
	0x5231	P3_OUT	P3 Port Output Data Register	P3 port output data
	0x5232	P3_OEN	P3 Port Output Enable Register	Enables P3 port outputs.
	0x5233	P3_PU	P3 Port Pull-up Control Register	Controls the P3 port pull-up resistor.
	0x5235	P3_IMSK	P3 Port Interrupt Mask Register	Enables P3 port interrupts.

Peripheral	Address	Register name		Function
P port & port MUX (8-bit device)	0x5236	P3_EDGE	P3 Port Interrupt Edge Select Register	Selects the signal edge for generating P3 port interrupts.
	0x5237	P3_IFLG	P3 Port Interrupt Flag Register	Indicates/resets the P3 port interrupt occurrence status.
	0x5238	P3_CHAT	P3 Port Chattering Filter Control Register	Controls the P3 port chattering filter.
	0x523a	P3_IEN	P3 Port Input Enable Register	Enables P3 port inputs.
	0x5240	P4_IN	P4 Port Input Data Register	P4 port input data
	0x5241	P4_OUT	P4 Port Output Data Register	P4 port output data
	0x5242	P4_OEN	P4 Port Output Enable Register	Enables P4 port outputs.
	0x5243	P4_PU	P4 Port Pull-up Control Register	Controls the P4 port pull-up resistor.
	0x5245	P4_IMSK	P4 Port Interrupt Mask Register	Enables P4 port interrupts.
	0x5246	P4_EDGE	P4 Port Interrupt Edge Select Register	Selects the signal edge for generating P4 port interrupts.
	0x5247	P4_IFLG	P4 Port Interrupt Flag Register	Indicates/resets the P4 port interrupt occurrence status.
	0x5248	P4_CHAT	P4 Port Chattering Filter Control Register	Controls the P4 port chattering filter.
	0x524a	P4_IEN	P4 Port Input Enable Register	Enables P4 port inputs.
	0x5250	P5_IN	P5 Port Input Data Register	P5 port input data
	0x5251	P5_OUT	P5 Port Output Data Register	P5 port output data
	0x5252	P5_OEN	P5 Port Output Enable Register	Enables P5 port outputs.
	0x5253	P5_PU	P5 Port Pull-up Control Register	Controls the P5 port pull-up resistor.
	0x5255	P5_IMSK	P5 Port Interrupt Mask Register	Enables P5 port interrupts.
	0x5256	P5_EDGE	P5 Port Interrupt Edge Select Register	Selects the signal edge for generating P5 port interrupts.
	0x5257	P5_IFLG	P5 Port Interrupt Flag Register	Indicates/resets the P5 port interrupt occurrence status.
	0x5258	P5_CHAT	P5 Port Chattering Filter Control Register	Controls the P5 port chattering filter.
	0x525a	P5_IEN	P5 Port Input Enable Register	Enables P5 port inputs.
	0x52a0	P00_03PMUX	P0[3:0] Port Function Select Register	Selects the P0[3:0] port functions.
	0x52a2	P10_13PMUX	P1[3:0] Port Function Select Register	Selects the P1[3:0] port functions.
	0x52a3	P14_17PMUX	P1[7:4] Port Function Select Register	Selects the P1[7:4] port functions.
	0x52a4	P20_23PMUX	P2[3:0] Port Function Select Register	Selects the P2[3:0] port functions.
	0x52a5	P24_27PMUX	P2[7:4] Port Function Select Register	Selects the P2[7:4] port functions.
	0x52a6	P30_33PMUX	P3[3:0] Port Function Select Register	Selects the P3[3:0] port functions.
	0x52a7	P34_37PMUX	P3[7:4] Port Function Select Register	Selects the P3[7:4] port functions.
	0x52a8	P40_43PMUX	P4[3:0] Port Function Select Register	Selects the P4[3:0] port functions.
	0x52a9	P44_45PMUX	P4[5:4] Port Function Select Register	Selects the P4[5:4] port functions.
	0x52aa	P50_53PMUX	P5[3:0] Port Function Select Register	Selects the P5[3:0] port functions.
	0x52ab	P54_55PMUX	P5[5:4] Port Function Select Register	Selects the P5[5:4] 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 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.
IR remote controller (16-bit device)	0x5340	REMC_CFG	REMC Configuration Register	Controls the clock and data transfer.
	0x5342	REMC_CAR	REMC Carrier Length Setup Register	Sets the carrier H/L section lengths.
	0x5344	REMC_LCNT	REMC Length Counter Register	Sets the transmit/receive data length.
	0x5346	REMC_INT	REMC Interrupt Control Register	Controls interrupts.
A/D converter (16-bit device)	0x5380	ADC10_ADD	A/D Conversion Result Register	A/D converted data
	0x5382	ADC10_TRG	A/D Trigger/Channel Select Register	Sets start/end channels and conversion mode.
	0x5384	ADC10_CTL	A/D Control/Status Register	Controls A/D converter and indicates conversion status.
	0x5386	ADC10_CLK	A/D Clock Control Register	Controls A/D converter clock.
	0x5388	ADC10_COM	A/D Comparator Setting Register	Adjusts A/D conversion characteristics.
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.

Peripheral	Address	Register name		Function
16-bit PWM timer Ch.1 (16-bit device)	0x5420	T16A_CTL1	T16A Counter Ch.1 Control Register	Controls the counter.
	0x5422	T16A_TC1	T16A Counter Ch.1 Data Register	Counter data
	0x5424	T16A_CCCTL1	T16A Comparator/Capture Ch.1 Control Register	Controls the comparator/capture block and TOUT.
	0x5426	T16A_CCA1	T16A Compare/Capture Ch.1 A Data Register	Compare A/capture A data
	0x5428	T16A_CCB1	T16A Compare/Capture Ch.1 B Data Register	Compare B/capture B data
	0x542a	T16A_IEN1	T16A Compare/Capture Ch.1 Interrupt Enable Register	Enables/disables interrupts.
	0x542c	T16A_IFLG1	T16A Compare/Capture Ch.1 Interrupt Flag Register	Displays/sets interrupt occurrence status.
16-bit PWM timer Ch.2 (16-bit device)	0x5440	T16A_CTL2	T16A Counter Ch.2 Control Register	Controls the counter.
	0x5442	T16A_TC2	T16A Counter Ch.2 Data Register	Counter data
	0x5444	T16A_CCCTL2	T16A Comparator/Capture Ch.2 Control Register	Controls the comparator/capture block and TOUT.
	0x5446	T16A_CCA2	T16A Compare/Capture Ch.2 A Data Register	Compare A/capture A data
	0x5448	T16A_CCB2	T16A Compare/Capture Ch.2 B Data Register	Compare B/capture B data
	0x544a	T16A_IEN2	T16A Compare/Capture Ch.2 Interrupt Enable Register	Enables/disables interrupts.
	0x544c	T16A_IFLG2	T16A Compare/Capture Ch.2 Interrupt Flag Register	Displays/sets interrupt occurrence status.
16-bit PWM timer Ch.3 (16-bit device)	0x5460	T16A_CTL3	T16A Counter Ch.3 Control Register	Controls the counter.
	0x5462	T16A_TC3	T16A Counter Ch.3 Data Register	Counter data
	0x5464	T16A_CCCTL3	T16A Comparator/Capture Ch.3 Control Register	Controls the comparator/capture block and TOUT.
	0x5466	T16A_CCA3	T16A Compare/Capture Ch.3 A Data Register	Compare A/capture A data
	0x5468	T16A_CCB3	T16A Compare/Capture Ch.3 B Data Register	Compare B/capture B data
	0x546a	T16A_IEN3	T16A Compare/Capture Ch.3 Interrupt Enable Register	Enables/disables interrupts.
	0x546c	T16A_IFLG3	T16A Compare/Capture Ch.3 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.

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.

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
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	**CLKDIV [1:0]**	Clock division ratio select	CLKDIV[1:0] / Division ratio: 0x3 / 1/8; 0x2 / 1/4; 0x1 / 1/2; 0x0 / 1/1	0x0	R/W	When the clock source is IOSC or OSC3
		D3–2	**CLKSRC [1:0]**	Clock source select	CLKSRC[1:0] / Clock source: 0x3 / External clock; 0x2 / OSC3; 0x1 / OSC1; 0x0 / IOSC*	0x0	R/W	* S1C17564 only
		D1	**–**	reserved	–	–	–	0 when being read.
		D0	**CLKEN**	Count clock enable	1 Enable 0 Disable	0	R/W	

0x4120–0x4127, 0x506d UART (with IrDA) Ch.1

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
UART Ch.1 Status Register (UART_ST1)	**0x4120** (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.1 Transmit Data Register (UART_TXD1)	**0x4121** (8 bits)	D7–0	**TXD[7:0]**	Transmit data TXD7(6) = MSB TXD0 = LSB	0x0 to 0xff (0x7f)	0x0	R/W	
UART Ch.1 Receive Data Register (UART_RXD1)	**0x4122** (8 bits)	D7–0	**RXD[7:0]**	Receive data in the receive data buffer RXD7(6) = MSB RXD0 = LSB	0x0 to 0xff (0x7f)	0x0	R	Older data in the buffer is read out first.
UART Ch.1 Mode Register (UART_MOD1)	**0x4123** (8 bits)	D7–5	**–**	reserved	–	–	–	0 when being read.
		D4	**CHLN**	Character length 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.1 Control Register (UART_CTL1)	**0x4124** (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	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
UART Ch.1 Expansion Register (UART_EXP1)	**0x4125** (8 bits)	D7–1	**–**	reserved	–	–	–	0 when being read.
		D0	**IRMD**	IrDA mode select	1 On / 0 Off	0	R/W	
UART Ch.1 Baud Rate Register (UART_BR1)	**0x4126** (8 bits)	D7–0	**BR[7:0]**	Baud rate setting	0x0 to 0xff	0x0	R/W	
UART Ch.1 Fine Mode Register (UART_FMD1)	**0x4127** (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.1 Clock Control Register (UART_CLK1)	**0x506d** (8 bits)	D7–6	**–**	reserved	–	–	–	0 when being read.
		D5–4	**CLKDIV [1:0]**	Clock division ratio select	CLKDIV[1:0] / Division ratio: 0x3 1/8; 0x2 1/4; 0x1 1/2; 0x0 1/1	0x0	R/W	When the clock source is IOSC or OSC3
		D3–2	**CLKSRC [1:0]**	Clock source select	CLKSRC[1:0] / Clock source: 0x3 External clock; 0x2 OSC3; 0x1 OSC1; 0x0 IOSC*	0x0	R/W	* S1C17564 only
		D1	**–**	reserved	–	–	–	0 when being read.
		D0	**CLKEN**	Count clock enable	1 Enable / 0 Disable	0	R/W	

0x4200–0x4208 Fine Mode 16-bit Timer Ch.0

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16F Ch.0 Count Clock Select Register (T16F_CLK0)	**0x4200** (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
T16F Ch.0 Reload Data Register (T16F_TR0)	**0x4202** (16 bits)	D15–0	**TR[15:0]**	Reload data TR15 = MSB TR0 = LSB	0x0 to 0xffff	0x0	R/W	
T16F Ch.0 Counter Data Register (T16F_TC0)	**0x4204** (16 bits)	D15–0	**TC[15:0]**	Counter data TC15 = MSB TC0 = LSB	0x0 to 0xffff	0xffff	R	
T16F Ch.0 Control Register (T16F_CTL0)	**0x4206** (16 bits)	D15–12	**–**	reserved	–	–	–	0 when being read.
		D11–8	**TFMD[3:0]**	Fine mode setup	0x0 to 0xf	0x0	R/W	Set a number of times to insert delay into a 16-underflow period.
		D7–5	**–**	reserved	–	–	–	0 when being read.
		D4	**TRMD**	Count mode select	1 One shot / 0 Repeat	0	R/W	
		D3–2	**–**	reserved	–	–	–	0 when being read.
		D1	**PRESER**	Timer reset	1 Reset / 0 Ignored	0	W	
		D0	**PRUN**	Timer run/stop control	1 Run / 0 Stop	0	R/W	
T16F Ch.0 Interrupt Control Register (T16F_INT0)	**0x4208** (16 bits)	D15–9	**–**	reserved	–	–	–	0 when being read.
		D8	**T16FIE**	T16F interrupt enable	1 Enable / 0 Disable	0	R/W	
		D7–1	**–**	reserved	–	–	–	0 when being read.
		D0	**T16FIF**	T16F interrupt flag	1 Cause of interrupt occurred / 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.

0x4220–0x4228 16-bit Timer Ch.0

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16 Ch.0 Count Clock Select Register (T16_CLK0)	0x4220 (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
T16 Ch.0 Reload Data Register (T16_TR0)	0x4222 (16 bits)	D15–0	TR[15:0]	Reload data TR15 = MSB TR0 = LSB	0x0 to 0xffff	0x0	R/W	
T16 Ch.0 Counter Data Register (T16_TC0)	0x4224 (16 bits)	D15–0	TC[15:0]	Counter data TC15 = MSB TC0 = LSB	0x0 to 0xffff	0xffff	R	
T16 Ch.0 Control Register (T16_CTL0)	0x4226 (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	
T16 Ch.0 Interrupt Control Register (T16_INT0)	0x4228 (16 bits)	D15–9	–	reserved	–	–	–	0 when being read.
		D8	T16IE	T16 interrupt enable	1 Enable 0 Disable	0	R/W	
		D7–1	–	reserved	–	–	–	0 when being read.
		D0	T16IF	T16 interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.

0x4240–0x4248 16-bit Timer Ch.1

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16 Ch.1 Count Clock Select Register (T16_CLK1)	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
T16 Ch.1 Reload Data Register (T16_TR1)	0x4242 (16 bits)	D15–0	TR[15:0]	Reload data TR15 = MSB TR0 = LSB	0x0 to 0xffff	0x0	R/W	
T16 Ch.1 Counter Data Register (T16_TC1)	0x4244 (16 bits)	D15–0	TC[15:0]	Counter data TC15 = MSB TC0 = LSB	0x0 to 0xffff	0xffff	R	
T16 Ch.1 Control Register (T16_CTL1)	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	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16 Ch.1 Interrupt Control Register (T16_INT1)	**0x4248** (16 bits)	D15–9	**–**	reserved	–	–	–	0 when being read.
		D8	**T16IE**	T16 interrupt enable	1 Enable 0 Disable	0	R/W	
		D7–1	**–**	reserved	–	–	–	0 when being read.
		D0	**T16IF**	T16 interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.

0x4260–0x4268 16-bit Timer Ch.2

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16 Ch.2 Count Clock Select Register (T16_CLK2)	**0x4260** (16 bits)	D15–4	**–**	reserved	–	–	–	0 when being read.
		D3–0	**DF[3:0]**	Count clock division ratio select	DF[3:0] / Division ratio: 0xf reserved; 0xe 1/16384; 0xd 1/8192; 0xc 1/4096; 0xb 1/2048; 0xa 1/1024; 0x9 1/512; 0x8 1/256; 0x7 1/128; 0x6 1/64; 0x5 1/32; 0x4 1/16; 0x3 1/8; 0x2 1/4; 0x1 1/2; 0x0 1/1	0x0	R/W	Source clock = PCLK
T16 Ch.2 Reload Data Register (T16_TR2)	**0x4262** (16 bits)	D15–0	**TR[15:0]**	Reload data TR15 = MSB TR0 = LSB	0x0 to 0xffff	0x0	R/W	
T16 Ch.2 Counter Data Register (T16_TC2)	**0x4264** (16 bits)	D15–0	**TC[15:0]**	Counter data TC15 = MSB TC0 = LSB	0x0 to 0xffff	0xffff	R	
T16 Ch.2 Control Register (T16_CTL2)	**0x4266** (16 bits)	D15–5	**–**	reserved	–	–	–	Do not write 1.
		D4	**TRMD**	Count mode select	1 One shot 0 Repeat	0	R/W	
		D3–2	**–**	reserved	–	–	–	0 when being read.
		D1	**PRESER**	Timer reset	1 Reset 0 Ignored	0	W	
		D0	**PRUN**	Timer run/stop control	1 Run 0 Stop	0	R/W	
T16 Ch.2 Interrupt Control Register (T16_INT2)	**0x4268** (16 bits)	D15–9	**–**	reserved	–	–	–	0 when being read.
		D8	**T16IE**	T16 interrupt enable	1 Enable 0 Disable	0	R/W	
		D7–1	**–**	reserved	–	–	–	0 when being read.
		D0	**T16IF**	T16 interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.

0x4280–0x4288 Fine Mode 16-bit Timer Ch.1

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16F Ch.1 Count Clock Select Register (T16F_CLK1)	**0x4280** (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
T16F Ch.1 Reload Data Register (T16F_TR1)	**0x4282** (16 bits)	D15–0	**TR[15:0]**	Reload data TR15 = MSB TR0 = LSB	0x0 to 0xffff	0x0	R/W	
T16F Ch.1 Counter Data Register (T16F_TC1)	**0x4284** (16 bits)	D15–0	**TC[15:0]**	Counter data TC15 = MSB TC0 = LSB	0x0 to 0xffff	0xffff	R	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16F Ch.1 Control Register (T16F_CTL1)	**0x4286** (16 bits)	D15–12	**–**	reserved	–	–	–	0 when being read.
		D11–8	**TFMD[3:0]**	Fine mode setup	0x0 to 0xf	0x0	R/W	Set a number of times to insert delay into a 16-underflow period.
		D7–5	**–**	reserved	–	–	–	0 when being read.
		D4	**TRMD**	Count mode select	1 One shot 0 Repeat	0	R/W	
		D3–2	**–**	reserved	–	–	–	0 when being read.
		D1	**PRESER**	Timer reset	1 Reset 0 Ignored	0	W	
		D0	**PRUN**	Timer run/stop control	1 Run 0 Stop	0	R/W	
T16F Ch.1 Interrupt Control Register (T16F_INT1)	**0x4288** (16 bits)	D15–9	**–**	reserved	–	–	–	0 when being read.
		D8	**T16FIE**	T16F interrupt enable	1 Enable 0 Disable	0	R/W	
		D7–1	**–**	reserved	–	–	–	0 when being read.
		D0	**T16FIF**	T16F interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.

0x4306–0x431c Interrupt Controller

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Interrupt Level Setup Register 0 (ITC_LV0)	**0x4306** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**ILV1[2:0]**	P1 interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**ILV0[2:0]**	P0 interrupt level	0 to 7	0x0	R/W	
Interrupt Level Setup Register 1 (ITC_LV1)	**0x4308** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**ILV3[2:0]**	CT interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**ILV2[2:0]**	SWT interrupt level	0 to 7	0x0	R/W	
Interrupt Level Setup Register 2 (ITC_LV2)	**0x430a** (16 bits)	D15–1	**–**	reserved	–	–	–	0 when being read.
		D10–8	**ILV5[2:0]**	P4 interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**ILV4[2:0]**	T16A Ch.2 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]**	T16A Ch.0 interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**ILV6[2:0]**	SPI Ch.2 interrupt level	0 to 7	0x0	R/W	
Interrupt Level Setup Register 4 (ITC_LV4)	**0x430e** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**ILV9[2:0]**	T16 Ch.0 interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**ILV8[2:0]**	T16F Ch.0 & 1/USI Ch.0 & 1 interrupt level	0 to 7	0x0	R/W	
Interrupt Level Setup Register 5 (ITC_LV5)	**0x4310** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**ILV11[2:0]**	T16 Ch.2/T16A Ch.3 interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**ILV10[2:0]**	T16 Ch.1 interrupt level	0 to 7	0x0	R/W	
Interrupt Level Setup Register 6 (ITC_LV6)	**0x4312** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**ILV13[2:0]**	UART Ch.1 interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**ILV12[2:0]**	UART Ch.0 interrupt level	0 to 7	0x0	R/W	
Interrupt Level Setup Register 7 (ITC_LV7)	**0x4314** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**ILV15[2:0]**	I2CM interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**ILV14[2:0]**	SPI Ch.0 interrupt level	0 to 7	0x0	R/W	
Interrupt Level Setup Register 8 (ITC_LV8)	**0x4316** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**ILV17[2:0]**	T16A Ch.1 interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**ILV16[2:0]**	REMC/SPI Ch.1 interrupt level	0 to 7	0x0	R/W	
Interrupt Level Setup Register 9 (ITC_LV9)	**0x4318** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**ILV19[2:0]**	P5 interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**ILV18[2:0]**	ADC10 interrupt level	0 to 7	0x0	R/W	
Interrupt Level Setup Register 10 (ITC_LV10)	**0x431a** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**ILV21[2:0]**	P3 interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**ILV20[2:0]**	P2 interrupt level	0 to 7	0x0	R/W	
Interrupt Level Setup Register 11 (ITC_LV11)	**0x431c** (16 bits)	D15–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**ILV22[2:0]**	I2CS 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) ss signal low flag (slave)	1 Busy 0 Idle 1 ss = L 0 ss = H	0	R	
		D1	**SPRBF**	Receive data buffer full flag	1 Full 0 Not full	0	R	
		D0	**SPTBE**	Transmit data buffer empty flag	1 Empty 0 Not empty	1	R	
SPI Ch.0 Transmit Data Register (SPI_TXD0)	**0x4322** (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	**SPTDB[7:0]**	SPI transmit data buffer SPTDB7 = MSB SPTDB0 = LSB	0x0 to 0xff	0x0	R/W	
SPI Ch.0 Receive Data Register (SPI_RXD0)	**0x4324** (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	**SPRDB[7:0]**	SPI receive data buffer SPRDB7 = MSB SPRDB0 = LSB	0x0 to 0xff	0x0	R	
SPI Ch.0 Control Register (SPI_CTL0)	**0x4326** (16 bits)	D15–10	–	reserved	–	–	–	0 when being read.
		D9	**MCLK**	SPI clock source select	1 T16 Ch.1 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	

0x4340–0x4346 I²C Master

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
I²C Master Enable Register (I2CM_EN)	**0x4340** (16 bits)	D15–1	–	reserved	–	–	–	0 when being read.
		D0	**I2CMEN**	I²C master enable	1 Enable 0 Disable	0	R/W	
I²C Master Control Register (I2CM_CTL)	**0x4342** (16 bits)	D15–10	–	reserved	–	–	–	0 when being read.
		D9	**RBUSY**	Receive busy flag	1 Busy 0 Idle	0	R	
		D8	**TBUSY**	Transmit busy flag	1 Busy 0 Idle	0	R	
		D7–5	–	reserved	–	–	–	0 when being read.
		D4	**NSERM**	Noise remove on/off	1 On 0 Off	0	R/W	
		D3–2	–	reserved	–	–	–	0 when being read.
		D1	**STP**	Stop control	1 Stop 0 Ignored	0	R/W	
		D0	**STRT**	Start control	1 Start 0 Ignored	0	R/W	
I²C Master Data Register (I2CM_DAT)	**0x4344** (16 bits)	D15–12	–	reserved	–	–	–	0 when being read.
		D11	**RBRDY**	Receive buffer ready flag	1 Ready 0 Empty	0	R	
		D10	**RXE**	Receive execution	1 Receive 0 Ignored	0	R/W	
		D9	**TXE**	Transmit execution	1 Transmit 0 Ignored	0	R/W	
		D8	**RTACK**	Receive/transmit ACK	1 Error 0 ACK	0	R/W	
		D7–0	**RTDT[7:0]**	Receive/transmit data RTDT7 = MSB RTDT0 = LSB	0x0 to 0xff	0x0	R/W	
I²C Master Interrupt Control Register (I2CM_ICTL)	**0x4346** (16 bits)	D15–2	–	reserved	–	–	–	0 when being read.
		D1	**RINTE**	Receive interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	**TINTE**	Transmit interrupt enable	1 Enable 0 Disable	0	R/W	

0x4360–0x436c I²C Slave

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
I²C Slave Transmit Data Register (I2CS_TRNS)	**0x4360** (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	**SDATA[7:0]**	I²C slave transmit data	0–0xff	0x0	R/W	
I²C Slave Receive Data Register (I2CS_RECV)	**0x4362** (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	**RDATA[7:0]**	I²C slave receive data	0–0xff	0x0	R	
I²C Slave Address Setup Register (I2CS_SADRS)	**0x4364** (16 bits)	D15–7	–	reserved	–	–	–	0 when being read.
		D6–0	**SADRS[6:0]**	I²C slave address	0–0x7f	0x0	R/W	

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
I²C Slave Control Register (I2CS_CTL)	**0x4366** (16 bits)	D15–9	**–**	reserved	–				–	–	0 when being read.
		D8	**TBUF_CLR**	I2CS_TRNS register clear	1	Clear state	0	Normal	0	R/W	
		D7	**I2CSEN**	I²C slave enable	1	Enable	0	Disable	0	R/W	
		D6	**SOFTRESET**	Software reset	1	Reset	0	Cancel	0	R/W	
		D5	**NAK_ANS**	NAK answer	1	NAK	0	ACK	0	R/W	
		D4	**BFREQ_EN**	Bus free request enable	1	Enable	0	Disable	0	R/W	
		D3	**CLKSTR_EN**	Clock stretch On/Off	1	On	0	Off	0	R/W	
		D2	**NF_EN**	Noise filter On/Off	1	On	0	Off	0	R/W	
		D1	**ASDET_EN**	Async.address detection On/Off	1	On	0	Off	0	R/W	
		D0	**COM_MODE**	I²C slave communication mode	1	Active	0	Standby	0	R/W	
I²C Slave Status Register (I2CS_STAT)	**0x4368** (16 bits)	D15–8	**–**	reserved	–				–	–	0 when being read.
		D7	**BSTAT**	Bus status transition	1	Changed	0	Unchanged	0	R	
		D6	**–**	reserved	–				–	–	0 when being read.
		D5	**TXUDF**	Transmit data underflow	1	Occurred	0	Not occurred	0	R/W	Reset by writing 1.
			RXOVF	Receive data overflow							
		D4	**BFREQ**	Bus free request	1	Occurred	0	Not occurred	0	R/W	
		D3	**DMS**	Output data mismatch	1	Error	0	Normal	0	R/W	
		D2	**ASDET**	Async. address detection status	1	Detected	0	Not detected	0	R/W	
		D1	**DA_NAK**	NAK receive status	1	NAK	0	ACK	0	R/W	
		D0	**DA_STOP**	STOP condition detect	1	Detected	0	Not detected	0	R/W	
I²C Slave Access Status Register (I2CS_ASTAT)	**0x436a** (16 bits)	D15–5	**–**	reserved	–				–	–	0 when being read.
		D4	**RXRDY**	Receive data ready	1	Ready	0	Not ready	0	R	
		D3	**TXEMP**	Transmit data empty	1	Empty	0	Not empty	0	R	
		D2	**BUSY**	I²C bus status	1	Busy	0	Free	0	R	
		D1	**SELECTED**	I²C slave select status	1	Selected	0	Not selected	0	R	
		D0	**R/W**	Read/write direction	1	Output	0	Input	0	R	
I²C Slave Interrupt Control Register (I2CS_ICTL)	**0x436c** (16 bits)	D15–3	**–**	reserved	–				–	–	0 when being read.
		D2	**BSTAT_IEN**	Bus status interrupt enable	1	Enable	0	Disable	0	R/W	
		D1	**RXRDY_IEN**	Receive interrupt enable	1	Enable	0	Disable	0	R/W	
		D0	**TXEMP_IEN**	Transmit interrupt enable	1	Enable	0	Disable	0	R/W	

0x4380–0x4386 SPI Ch.1

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
SPI Ch.1 Status Register (SPI_ST1)	**0x4380** (16 bits)	D15–3	**–**	reserved	–				–	–	0 when being read.
		D2	**SPBSY**	Transfer busy flag (master)	1	Busy	0	Idle	0	R	
				ss signal low flag (slave)	1	ss = L	0	ss = H			
		D1	**SPRBF**	Receive data buffer full flag	1	Full	0	Not full	0	R	
		D0	**SPTBE**	Transmit data buffer empty flag	1	Empty	0	Not empty	1	R	
SPI Ch.1 Transmit Data Register (SPI_TXD1)	**0x4382** (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.1 Receive Data Register (SPI_RXD1)	**0x4384** (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.1 Control Register (SPI_CTL1)	**0x4386** (16 bits)	D15–10	**–**	reserved	–				–	–	0 when being read.
		D9	**MCLK**	SPI clock source select	1	T16 Ch.1	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	

0x43a0–0x43a6 SPI Ch.2

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
SPI Ch.2 Status Register (SPI_ST2)	**0x43a0** (16 bits)	D15–3	**–**	reserved	–				–	–	0 when being read.
		D2	**SPBSY**	Transfer busy flag (master)	1	Busy	0	Idle	0	R	
				ss signal low flag (slave)	1	ss = L	0	ss = H			
		D1	**SPRBF**	Receive data buffer full flag	1	Full	0	Not full	0	R	
		D0	**SPTBE**	Transmit data buffer empty flag	1	Empty	0	Not empty	1	R	
SPI Ch.2 Transmit Data Register (SPI_TXD2)	**0x43a2** (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	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SPI Ch.2 Receive Data Register (SPI_RXD2)	0x43a4 (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.2 Control Register (SPI_CTL2)	0x43a6 (16 bits)	D15–10	–	reserved	–	–	–	0 when being read.
		D9	MCLK	SPI clock source select	1 T16 Ch.1 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	

0x5000–0x5003 Clock Timer

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Clock Timer Control Register (CT_CTL)	0x5000 (8 bits)	D7–5	–	reserved	–	–	–	0 when being read.
		D4	CTRST	Clock timer reset	1 Reset 0 Ignored	0	W	
		D3–1	–	reserved	–	–	–	
		D0	CTRUN	Clock timer run/stop control	1 Run 0 Stop	0	R/W	
Clock Timer Counter Register (CT_CNT)	0x5001 (8 bits)	D7–0	CTCNT[7:0]	Clock timer counter value	0x0 to 0xff	0	R	
Clock Timer Interrupt Mask Register (CT_IMSK)	0x5002 (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.
		D3	CTIE32	32 Hz interrupt enable	1 Enable 0 Disable	0	R/W	
		D2	CTIE8	8 Hz interrupt enable	1 Enable 0 Disable	0	R/W	
		D1	CTIE2	2 Hz interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	CTIE1	1 Hz interrupt enable	1 Enable 0 Disable	0	R/W	
Clock Timer Interrupt Flag Register (CT_IFLG)	0x5003 (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.
		D3	CTIF32	32 Hz interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D2	CTIF8	8 Hz interrupt flag		0	R/W	
		D1	CTIF2	2 Hz interrupt flag		0	R/W	
		D0	CTIF1	1 Hz interrupt flag		0	R/W	

0x5020–0x5023 Stopwatch Timer

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Stopwatch Timer Control Register (SWT_CTL)	0x5020 (8 bits)	D7–5	–	reserved	–	–	–	0 when being read.
		D4	SWTRST	Stopwatch timer reset	1 Reset 0 Ignored	0	W	
		D3–1	–	reserved	–	–	–	
		D0	SWTRUN	Stopwatch timer run/stop control	1 Run 0 Stop	0	R/W	
Stopwatch Timer BCD Counter Register (SWT_BCNT)	0x5021 (8 bits)	D7–4	BCD10[3:0]	1/10 sec. BCD counter value	0 to 9	0	R	
		D3–0	BCD100[3:0]	1/100 sec. BCD counter value	0 to 9	0	R	
Stopwatch Timer Interrupt Mask Register (SWT_IMSK)	0x5022 (8 bits)	D7–3	–	reserved	–	–	–	0 when being read.
		D2	SIE1	1 Hz interrupt enable	1 Enable 0 Disable	0	R/W	
		D1	SIE10	10 Hz interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	SIE100	100 Hz interrupt enable	1 Enable 0 Disable	0	R/W	
Stopwatch Timer Interrupt Flag Register (SWT_IFLG)	0x5023 (8 bits)	D7–3	–	reserved	–	–	–	0 when being read.
		D2	SIF1	1 Hz interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D1	SIF10	10 Hz interrupt flag		0	R/W	
		D0	SIF100	100 Hz interrupt flag		0	R/W	

0x5040–0x5041 Watchdog Timer

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Watchdog Timer Control Register (WDT_CTL)	0x5040 (8 bits)	D7–5	–	reserved	–	–	–	0 when being read.
		D4	WDTRST	Watchdog timer reset	1 Reset 0 Ignored	0	W	
		D3–0	WDTRUN[3:0]	Watchdog timer run/stop control	Other than 1010 Run; 1010 Stop	1010	R/W	
Watchdog Timer Status Register (WDT_ST)	0x5041 (8 bits)	D7–2	–	reserved	–	–	–	0 when being read.
		D1	WDTMD	NMI/Reset mode select	1 Reset 0 NMI	0	R/W	
		D0	WDTST	NMI status	1 NMI occurred 0 Not occurred	0	R	

0x5060–0x5081 Clock Generator

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Clock Source Select Register (CLG_SRC) **S1C17554**	**0x5060** (8 bits)	D7–2	**–**	reserved	–	–	–	0 when being read.
		D1–0	**CLKSRC[1:0]**	System clock source select	CLKSRC[1:0] Clock source 0x3 reserved 0x2 OSC3 0x1 OSC1 0x0 reserved	0x2	R/W	
Clock Source Select Register (CLG_SRC) **S1C17564**	**0x5060** (8 bits)	D7–2	**–**	reserved	–	–	–	0 when being read.
		D1–0	**CLKSRC[1:0]**	System clock source select	CLKSRC[1:0] Clock source 0x3 reserved 0x2 OSC3 0x1 OSC1 0x0 IOSC	0x0	R/W	
Oscillation Control Register (CLG_CTL) **S1C17554**	**0x5061** (8 bits)	D7–6	**–**	reserved	–	–	–	0 when being read.
		D5–4	**OSC3WT[1:0]**	OSC3 wait cycle select	OSC3WT[1:0] Wait cycle 0x3 128 cycles 0x2 256 cycles 0x1 512 cycles 0x0 1024 cycles	0x0	R/W	
		D3–2	**–**	reserved	–	–	–	0 when being read.
		D1	**OSC1EN**	OSC1 enable	1 Enable 0 Disable	0	R/W	
		D0	**OSC3EN**	OSC3 enable	1 Enable 0 Disable	1	R/W	
Oscillation Control Register (CLG_CTL) **S1C17564**	**0x5061** (8 bits)	D7–6	**IOSCWT[1:0]**	IOSC wait cycle select	IOSCWT[1:0] Wait cycle 0x3 8 cycles 0x2 16 cycles 0x1 32 cycles 0x0 64 cycles	0x0	R/W	
		D5–4	**OSC3WT[1:0]**	OSC3 wait cycle select	OSC3WT[1:0] Wait cycle 0x3 128 cycles 0x2 256 cycles 0x1 512 cycles 0x0 1024 cycles	0x0	R/W	
		D3	**–**	reserved	–	–	–	0 when being read.
		D2	**IOSCEN**	IOSC enable	1 Enable 0 Disable	1	R/W	
		D1	**OSC1EN**	OSC1 enable	1 Enable 0 Disable	0	R/W	
		D0	**OSC3EN**	OSC3 enable	1 Enable 0 Disable	0	R/W	
Noise Filter Enable Register (CLG_NFEN)	**0x5062** (8 bits)	D7–6	**–**	reserved	–	–	–	0 when being read.
		D5	**OSC1WCE**	OSC1 wait cycle enable	1 Enable 0 Disable	1	R/W	
		D4	**OSC3WCE**	OSC3 wait cycle enable	1 Enable 0 Disable	1	R/W	
		D3–0	**–**	reserved	–	–	–	0 when being read.
FOUTA Control Register (CLG_FOUTA) **S1C17554**	**0x5064** (8 bits)	D7–6	**–**	reserved	–	–	–	0 when being read.
		D5–4	**FOUTAD [1:0]**	FOUTA clock division ratio select	FOUTAD[1:0] Division ratio 0x3 reserved 0x2 1/4 0x1 1/2 0x0 1/1	0x0	R/W	When the clock source is OSC3
		D3–2	**FOUTASRC [1:0]**	FOUTA clock source select	FOUTASRC[1:0] Clock source 0x3 reserved 0x2 OSC3 0x1 OSC1 0x0 reserved	0x0	R/W	
		D1	**–**	reserved	–	–	–	0 when being read.
		D0	**FOUTAE**	FOUTA output enable	1 Enable 0 Disable	0	R/W	
FOUTA Control Register (CLG_FOUTA) **S1C17564**	**0x5064** (8 bits)	D7–6	**–**	reserved	–	–	–	0 when being read.
		D5–4	**FOUTAD [1:0]**	FOUTA clock division ratio select	FOUTAD[1:0] Division ratio 0x3 reserved 0x2 1/4 0x1 1/2 0x0 1/1	0x0	R/W	When the clock source is IOSC or OSC3
		D3–2	**FOUTASRC [1:0]**	FOUTA clock source select	FOUTASRC[1:0] Clock source 0x3 reserved 0x2 OSC3 0x1 OSC1 0x0 IOSC	0x0	R/W	
		D1	**–**	reserved	–	–	–	0 when being read.
		D0	**FOUTAE**	FOUTA output enable	1 Enable 0 Disable	0	R/W	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
FOUTB Control Register (CLG_FOUTB) **S1C17554**	**0x5065** (8 bits)	D7–6	**–**	reserved	–	–	–	0 when being read.
		D5–4	**FOUTBD [1:0]**	FOUTB clock division ratio select	FOUTBD[1:0] / Division ratio 0x3 reserved 0x2 1/4 0x1 1/2 0x0 1/1	0x0	R/W	When the clock source is OSC3
		D3–2	**FOUTBSRC [1:0]**	FOUTB clock source select	FOUTBSRC[1:0] / Clock source 0x3 reserved 0x2 OSC3 0x1 OSC1 0x0 reserved	0x0	R/W	
		D1	**–**	reserved	–	–	–	0 when being read.
		D0	**FOUTBE**	FOUTB output enable	1 Enable 0 Disable	0	R/W	
FOUTB Control Register (CLG_FOUTB) **S1C17564**	**0x5065** (8 bits)	D7–6	**–**	reserved	–	–	–	0 when being read.
		D5–4	**FOUTBD [1:0]**	FOUTB clock division ratio select	FOUTBD[1:0] / Division ratio 0x3 reserved 0x2 1/4 0x1 1/2 0x0 1/1	0x0	R/W	When the clock source is IOSC or OSC3
		D3–2	**FOUTBSRC [1:0]**	FOUTB clock source select	FOUTBSRC[1:0] / Clock source 0x3 reserved 0x2 OSC3 0x1 OSC1 0x0 IOSC	0x0	R/W	
		D1	**–**	reserved	–	–	–	0 when being read.
		D0	**FOUTBE**	FOUTB output enable	1 Enable 0 Disable	0	R/W	
IOSC Control Register (CLG_IOSC) **S1C17564**	**0x506e** (8 bits)	D7–2	**–**	reserved	–	–	–	0 when being read.
		D1–0	**IOSCSEL [1:0]**	IOSC frequency select	IOSCSEL[1:0] / Frequency 0x3 2 MHz 0x2 4 MHz 0x1 12 MHz 0x0 8 MHz	0x1	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	

0x50c0–0x50cf USI Ch.0

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
USI Ch.0 Global Configuration Register (USI_GCFG0)	**0x50c0** (8 bits)	D7–4	**–**	reserved	–	–	–	0 when being read.
		D3	**LSBFST**	MSB/LSB first mode select	1 MSB first 0 LSB first	0	R/W	
		D2–0	**USIMOD [2:0]**	Interface mode configuration	USIMOD[2:0] / I/F mode 0x7–0x6 reserved 0x5 I²C slave 0x4 I²C master 0x3 reserved 0x2 SPI master 0x1 UART 0x0 Software reset	0x0	R/W	
USI Ch.0 Transmit Data Buffer Register (USI_TD0)	**0x50c1** (8 bits)	D7–0	**TD[7:0]**	USI transmit data buffer TD7 = MSB TD0 = LSB	0x0 to 0xff	0x0	R/W	
USI Ch.0 Receive Data Buffer Register (USI_RD0)	**0x50c2** (8 bits)	D7–0	**RD[7:0]**	USI receive data buffer RD7 = MSB RD0 = LSB	0x0 to 0xff	0x0	R	
USI Ch.0 UART Mode Configuration Register (USI_UCFG0)	**0x50c3** (8 bits)	D7–4	**–**	reserved	–	–	–	0 when being read.
		D3	**UCHLN**	Character length select	1 8 bits 0 7 bits	0	R/W	
		D2	**USTPB**	Stop bit select	1 2 bits 0 1 bit	0	R/W	
		D1	**UPMD**	Parity mode select	1 Even 0 Odd	0	R/W	
		D0	**UPREN**	Parity enable	1 With parity 0 No parity	0	R/W	
USI Ch.0 UART Mode Interrupt Enable Register (USI_UIE0)	**0x50c4** (8 bits)	D7–3	**–**	reserved	–	–	–	0 when being read.
		D2	**UEIE**	Receive error interrupt enable	1 Enable 0 Disable	0	R/W	
		D1	**URDIE**	Receive buffer full interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	**UTDIE**	Transmit buffer empty int. enable	1 Enable 0 Disable	0	R/W	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
USI Ch.0 UART Mode Interrupt Flag Register (USI_UIF0)	**0x50c5** (8 bits)	D7	**–**	reserved	–	–	–	0 when being read.
		D6	**URBSY**	Receive busy flag	1 Busy 0 Idle	0	R	
		D5	**UTBSY**	Transmit busy flag	1 Busy 0 Idle	0	R	
		D4	**UPEIF**	Parity error flag	1 Error 0 Normal	0	R/W	Reset by writing 1.
		D3	**USEIF**	Framing error flag	1 Error 0 Normal	0	R/W	
		D2	**UOEIF**	Overrun error flag	1 Error 0 Normal	0	R/W	
		D1	**URDIF**	Receive buffer full flag	1 Full 0 Not full	0	R/W	
		D0	**UTDIF**	Transmit buffer empty flag	1 Empty 0 Not empty	0	R/W	
USI Ch.0 SPI Master Mode Configuration Register (USI_SCFG0)	**0x50c6** (8 bits)	D7–6	**–**	reserved	–	–	–	0 when being read.
		D5	**SCMD**	Command bit (for 9-bit data)	1 High 0 Low	0	R/W	
		D4	**SCHLN**	Character length select	1 9 bits 0 8 bits	0	R/W	
		D3	**SCPHA**	Clock phase select	1 Phase 1 0 Phase 0	0	R/W	
		D2	**SCPOL**	Clock polarity select	1 Active L 0 Active H	0	R/W	
		D1	**–**	reserved	–	–	–	0 when being read.
		D0	**SFSTMOD**	Fast mode select	1 Fast 0 Normal	0	R/W	
USI Ch.0 SPI Master Mode Interrupt Enable Register (USI_SIE0)	**0x50c7** (8 bits)	D7–3	**–**	reserved	–	–	–	0 when being read.
		D2	**SEIE**	Receive error interrupt enable	1 Enable 0 Disable	0	R/W	
		D1	**SRDIE**	Receive buffer full interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	**STDIE**	Transmit buffer empty int. enable	1 Enable 0 Disable	0	R/W	
USI Ch.0 SPI Master Mode Interrupt Flag Register (USI_SIF0)	**0x50c8** (8 bits)	D7–3	**–**	reserved	–	–	–	0 when being read.
		D2	**SEIF**	Overrun error flag	1 Error 0 Normal	0	R/W	Reset by writing 1.
		D1	**SRDIF**	Receive buffer full flag	1 Full 0 Not full	0	R/W	
		D0	**STDIF**	Transmit buffer empty flag	1 Empty 0 Not empty	0	R/W	
USI Ch.0 I²C Master Mode Trigger Register (USI_IMTG0)	**0x50ca** (8 bits)	D7–5	**–**	reserved	–	–	–	0 when being read.
		D4	**IMTG**	I²C master operation trigger	1 Trigger 0 Ignored	0	W	
					1 Waiting 0 Finished		R	
		D3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**IMTGMOD [2:0]**	I²C master trigger mode select	IMTGMOD[2:0] Trigger mode 0x7 reserved 0x6 Receive ACK/NAK 0x5 Transmit NAK 0x4 Transmit ACK 0x3 Receive data 0x2 Transmit data 0x1 Stop condition 0x0 Start condition	0x0	R/W	
USI Ch.0 I²C Master Mode Interrupt Enable Register (USI_IMIE0)	**0x50cb** (8 bits)	D7–2	**–**	reserved	–	–	–	0 when being read.
		D1	**IMEIE**	Receive error interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	**IMIE**	Operation completion int. enable	1 Enable 0 Disable	0	R/W	
USI Ch.0 I²C Master Mode Interrupt Flag Register (USI_IMIF0)	**0x50cc** (8 bits)	D7–6	**–**	reserved	–	–	–	0 when being read.
		D5	**IMBSY**	I²C master busy flag	1 Busy 0 Standby	0	R	
		D4–2	**IMSTA[2:0]**	I²C master status	IMSTA[2:0] Status 0x7 reserved 0x6 NAK received 0x5 ACK received 0x4 ACK/NAK sent 0x3 Rx buffer full 0x2 Tx buffer empty 0x1 Stop generated 0x0 Start generated	0x0	R	
		D1	**IMEIF**	Overrun error flag	1 Error 0 Normal	0	R/W	Reset by writing 1.
		D0	**IMIF**	Operation completion flag	1 Completed 0 Not completed	0	R/W	
USI Ch.0 I²C Slave Mode Trigger Register (USI_ISTG0)	**0x50cd** (8 bits)	D7–5	**–**	reserved	–	–	–	0 when being read.
		D4	**ISTG**	I²C slave operation trigger	1 Trigger 0 Ignored	0	W	
					1 Waiting 0 Finished		R	
		D3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**ISTGMOD [2:0]**	I²C slave trigger mode select	ISTGMOD[2:0] Trigger mode 0x7 reserved 0x6 Receive ACK/NAK 0x5 Transmit NAK 0x4 Transmit ACK 0x3 Receive data 0x2 Transmit data 0x1 reserved 0x0 Wait for start	0x0	R/W	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
USI Ch.0 I²C Slave Mode Interrupt Enable Register (USI_ISIE0)	**0x50ce** (8 bits)	D7–2	**–**	reserved	–	–	–	0 when being read.
		D1	**ISEIE**	Receive error interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	**ISIE**	Operation completion int. enable	1 Enable 0 Disable	0	R/W	
USI Ch.0 I²C Slave Mode Interrupt Flag Register (USI_ISIF0)	**0x50cf** (8 bits)	D7–6	**–**	reserved	–	–	–	0 when being read.
		D5	**ISBSY**	I²C slave busy flag	1 Busy 0 Standby	0	R	
		D4–2	**ISSTA[2:0]**	I²C slave status	ISSTA[2:0] Status 0x7 reserved 0x6 NAK received 0x5 ACK received 0x4 ACK/NAK sent 0x3 Rx buffer full 0x2 Tx buffer empty 0x1 Stop detected 0x0 Start detected	0x0	R	
		D1	**ISEIF**	Overrun error flag	1 Error 0 Normal	0	R/W	Reset by writing 1.
		D0	**ISIF**	Operation completion flag	1 Completed 0 Not completed	0	R/W	

0x50e0–0x50ef USI Ch.1

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
USI Ch.1 Global Configuration Register (USI_GCFG1)	**0x50e0** (8 bits)	D7–4	**–**	reserved	–	–	–	0 when being read.
		D3	**LSBFST**	MSB/LSB first mode select	1 MSB first 0 LSB first	0	R/W	
		D2–0	**USIMOD [2:0]**	Interface mode configuration	USIMOD[2:0] I/F mode 0x7–0x6 reserved 0x5 I²C slave 0x4 I²C master 0x3 reserved 0x2 SPI master 0x1 UART 0x0 Software reset	0x0	R/W	
USI Ch.1 Transmit Data Buffer Register (USI_TD1)	**0x50e1** (8 bits)	D7–0	**TD[7:0]**	USI transmit data buffer TD7 = MSB TD0 = LSB	0x0 to 0xff	0x0	R/W	
USI Ch.1 Receive Data Buffer Register (USI_RD1)	**0x50e2** (8 bits)	D7–0	**RD[7:0]**	USI receive data buffer RD7 = MSB RD0 = LSB	0x0 to 0xff	0x0	R	
USI Ch.1 UART Mode Configuration Register (USI_UCFG1)	**0x50e3** (8 bits)	D7–4	**–**	reserved	–	–	–	0 when being read.
		D3	**UCHLN**	Character length select	1 8 bits 0 7 bits	0	R/W	
		D2	**USTPB**	Stop bit select	1 2 bits 0 1 bit	0	R/W	
		D1	**UPMD**	Parity mode select	1 Even 0 Odd	0	R/W	
		D0	**UPREN**	Parity enable	1 With parity 0 No parity	0	R/W	
USI Ch.1 UART Mode Interrupt Enable Register (USI_UIE1)	**0x50e4** (8 bits)	D7–3	**–**	reserved	–	–	–	0 when being read.
		D2	**UEIE**	Receive error interrupt enable	1 Enable 0 Disable	0	R/W	
		D1	**URDIE**	Receive buffer full interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	**UTDIE**	Transmit buffer empty int. enable	1 Enable 0 Disable	0	R/W	
USI Ch.1 UART Mode Interrupt Flag Register (USI_UIF1)	**0x50e5** (8 bits)	D7	**–**	reserved	–	–	–	0 when being read.
		D6	**URBSY**	Receive busy flag	1 Busy 0 Idle	0	R	
		D5	**UTBSY**	Transmit busy flag	1 Busy 0 Idle	0	R	
		D4	**UPEIF**	Parity error flag	1 Error 0 Normal	0	R/W	Reset by writing 1.
		D3	**USEIF**	Framing error flag	1 Error 0 Normal	0	R/W	
		D2	**UOEIF**	Overrun error flag	1 Error 0 Normal	0	R/W	
		D1	**URDIF**	Receive buffer full flag	1 Full 0 Not full	0	R/W	
		D0	**UTDIF**	Transmit buffer empty flag	1 Empty 0 Not empty	0	R/W	
USI Ch.1 SPI Master Mode Configuration Register (USI_SCFG1)	**0x50e6** (8 bits)	D7–6	**–**	reserved	–	–	–	0 when being read.
		D5	**SCMD**	Command bit (for 9-bit data)	1 High 0 Low	0	R/W	
		D4	**SCHLN**	Character length select	1 9 bits 0 8 bits	0	R/W	
		D3	**SCPHA**	Clock phase select	1 Phase 1 0 Phase 0	0	R/W	
		D2	**SCPOL**	Clock polarity select	1 Active L 0 Active H	0	R/W	
		D1	**–**	reserved	–	–	–	0 when being read.
		D0	**SFSTMOD**	Fast mode select	1 Fast 0 Normal	0	R/W	
USI Ch.1 SPI Master Mode Interrupt Enable Register (USI_SIE1)	**0x50e7** (8 bits)	D7–3	**–**	reserved	–	–	–	0 when being read.
		D2	**SEIE**	Receive error interrupt enable	1 Enable 0 Disable	0	R/W	
		D1	**SRDIE**	Receive buffer full interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	**STDIE**	Transmit buffer empty int. enable	1 Enable 0 Disable	0	R/W	
USI Ch.1 SPI Master Mode Interrupt Flag Register (USI_SIF1)	**0x50e8** (8 bits)	D7–3	**–**	reserved	–	–	–	0 when being read.
		D2	**SEIF**	Overrun error flag	1 Error 0 Normal	0	R/W	Reset by writing 1.
		D1	**SRDIF**	Receive buffer full flag	1 Full 0 Not full	0	R/W	
		D0	**STDIF**	Transmit buffer empty flag	1 Empty 0 Not empty	0	R/W	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
USI Ch.1 I²C Master Mode Trigger Register (USI_IMTG1)	**0x50ea** (8 bits)	D7–5	**–**	reserved	–	–	–	0 when being read.
		D4	**IMTG**	I²C master operation trigger	1 Trigger 0 Ignored	0	W	
					1 Waiting 0 Finished		R	
		D3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**IMTGMOD [2:0]**	I²C master trigger mode select	IMTGMOD[2:0] Trigger mode 0x7 reserved 0x6 Receive ACK/NAK 0x5 Transmit NAK 0x4 Transmit ACK 0x3 Receive data 0x2 Transmit data 0x1 Stop condition 0x0 Start condition	0x0	R/W	
USI Ch.1 I²C Master Mode Interrupt Enable Register (USI_IMIE1)	**0x50eb** (8 bits)	D7–2	**–**	reserved	–	–	–	0 when being read.
		D1	**IMEIE**	Receive error interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	**IMIE**	Operation completion int. enable	1 Enable 0 Disable	0	R/W	
USI Ch.1 I²C Master Mode Interrupt Flag Register (USI_IMIF1)	**0x50ec** (8 bits)	D7–6	**–**	reserved	–	–	–	0 when being read.
		D5	**IMBSY**	I²C master busy flag	1 Busy 0 Standby	0	R	
		D4–2	**IMSTA[2:0]**	I²C master status	IMSTA[2:0] Status 0x7 reserved 0x6 NAK received 0x5 ACK received 0x4 ACK/NAK sent 0x3 Rx buffer full 0x2 Tx buffer empty 0x1 Stop generated 0x0 Start generated	0x0	R	
		D1	**IMEIF**	Overrun error flag	1 Error 0 Normal	0	R/W	Reset by writing 1.
		D0	**IMIF**	Operation completion flag	1 Completed 0 Not completed	0	R/W	
USI Ch.1 I²C Slave Mode Trigger Register (USI_ISTG1)	**0x50ed** (8 bits)	D7–5	**–**	reserved	–	–	–	0 when being read.
		D4	**ISTG**	I²C slave operation trigger	1 Trigger 0 Ignored	0	W	
					1 Waiting 0 Finished		R	
		D3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**ISTGMOD [2:0]**	I²C slave trigger mode select	ISTGMOD[2:0] Trigger mode 0x7 reserved 0x6 Receive ACK/NAK 0x5 Transmit NAK 0x4 Transmit ACK 0x3 Receive data 0x2 Transmit data 0x1 reserved 0x0 Wait for start	0x0	R/W	
USI Ch.1 I²C Slave Mode Interrupt Enable Register (USI_ISIE1)	**0x50ee** (8 bits)	D7–2	**–**	reserved	–	–	–	0 when being read.
		D1	**ISEIE**	Receive error interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	**ISIE**	Operation completion int. enable	1 Enable 0 Disable	0	R/W	
USI Ch.1 I²C Slave Mode Interrupt Flag Register (USI_ISIF1)	**0x50ef** (8 bits)	D7–6	**–**	reserved	–	–	–	0 when being read.
		D5	**ISBSY**	I²C slave busy flag	1 Busy 0 Standby	0	R	
		D4–2	**ISSTA[2:0]**	I²C slave status	ISSTA[2:0] Status 0x7 reserved 0x6 NAK received 0x5 ACK received 0x4 ACK/NAK sent 0x3 Rx buffer full 0x2 Tx buffer empty 0x1 Stop detected 0x0 Start detected	0x0	R	
		D1	**ISEIF**	Overrun error flag	1 Error 0 Normal	0	R/W	Reset by writing 1.
		D0	**ISIF**	Operation completion flag	1 Completed 0 Not completed	0	R/W	

0x5121 Power Generator

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
V_{D1} Control Register (VD1_CTL) S1C17564	0x5121 (8 bits)	D7–2	–	reserved	–		–	–	0 when being read.
		D1–0	VD1ECO [1:0]	Regulator operation mode select	VD1ECO[1:0]	Mode	0x0	R/W	
					0x3	reserved			
					0x2	Auto-control			
					0x1	Economy			
					0x0	Normal			

0x5200–0x52ab 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)	D7–4	–	reserved	–		–	–	0 when being read.
		D3–0	P0IN[3:0]	P0[3:0] port input data	1 1 (H)	0 0 (L)	×	R	
P0 Port Output Data Register (P0_OUT)	0x5201 (8 bits)	D7–4	–	reserved	–		–	–	0 when being read.
		D3–0	P0OUT[3:0]	P0[3:0] port output data	1 1 (H)	0 0 (L)	0	R/W	
P0 Port Output Enable Register (P0_OEN)	0x5202 (8 bits)	D7–4	–	reserved	–		–	–	0 when being read.
		D3–0	P0OEN[3:0]	P0[3:0] port output enable	1 Enable	0 Disable	0	R/W	
P0 Port Pull-up Control Register (P0_PU)	0x5203 (8 bits)	D7–4	–	reserved	–		–	–	0 when being read.
		D3–0	P0PU[3:0]	P0[3:0] port pull-up enable	1 Enable	0 Disable	1 (0xf)	R/W	
P0 Port Interrupt Mask Register (P0_IMSK)	0x5205 (8 bits)	D7–4	–	reserved	–		–	–	0 when being read.
		D3–0	P0IE[3:0]	P0[3:0] port interrupt enable	1 Enable	0 Disable	0	R/W	
P0 Port Interrupt Edge Select Register (P0_EDGE)	0x5206 (8 bits)	D7–4	–	reserved	–		–	–	0 when being read.
		D3–0	P0EDGE[3:0]	P0[3:0] port interrupt edge select	1 Falling edge	0 Rising edge	0	R/W	
P0 Port Interrupt Flag Register (P0_IFLG)	0x5207 (8 bits)	D7–4	–	reserved	–		–	–	0 when being read.
		D3–0	P0IF[3:0]	P0[3:0] 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)	D7–3	–	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)	D7–4	–	reserved	–		–	–	0 when being read.
		D3–0	P0IEN[3:0]	P0[3:0] port input enable	1 Enable	0 Disable	0xf	R/W	
P1 Port Input Data Register (P1_IN)	0x5210 (8 bits)	D7–0	P1IN[7:0]	P1[7:0] port input data	1 1 (H)	0 0 (L)	×	R	
P1 Port Output Data Register (P1_OUT)	0x5211 (8 bits)	D7–0	P1OUT[7:0]	P1[7:0] port output data	1 1 (H)	0 0 (L)	0	R/W	
P1 Port Output Enable Register (P1_OEN)	0x5212 (8 bits)	D7–0	P1OEN[7:0]	P1[7:0] port output enable	1 Enable	0 Disable	0	R/W	
P1 Port Pull-up Control Register (P1_PU)	0x5213 (8 bits)	D7–0	P1PU[7:0]	P1[7:0] port pull-up enable	1 Enable	0 Disable	1 (0xff)	R/W	
P1 Port Interrupt Mask Register (P1_IMSK)	0x5215 (8 bits)	D7–0	P1IE[7:0]	P1[7:0] port interrupt enable	1 Enable	0 Disable	0	R/W	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
P1 Port Interrupt Edge Select Register (P1_EDGE)	**0x5216** (8 bits)	D7–0	**P1EDGE[7:0]**	P1[7:0] port interrupt edge select	1 Falling edge 0 Rising edge	0	R/W	
P1 Port Interrupt Flag Register (P1_IFLG)	**0x5217** (8 bits)	D7–0	**P1IF[7:0]**	P1[7:0] port interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
P1 Port Chattering Filter Control Register (P1_CHAT)	**0x5218** (8 bits)	D7	–	reserved	–	–	–	0 when being read.
		D6–4	**P1CF2[2:0]**	P1[7:4] chattering filter time	P1CF2[2:0] Filter time 0x7 16384/fPCLK 0x6 8192/fPCLK 0x5 4096/fPCLK 0x4 2048/fPCLK 0x3 1024/fPCLK 0x2 512/fPCLK 0x1 256/fPCLK 0x0 None	0 0x0	R/W R/W	
		D3	–	reserved	–	–	–	0 when being read.
		D2–0	**P1CF1[2:0]**	P1[3:0] chattering filter time	P1CF1[2:0] Filter time 0x7 16384/fPCLK 0x6 8192/fPCLK 0x5 4096/fPCLK 0x4 2048/fPCLK 0x3 1024/fPCLK 0x2 512/fPCLK 0x1 256/fPCLK 0x0 None	0x0	R/W	
P1 Port Input Enable Register (P1_IEN)	**0x521a** (8 bits)	D7–0	**P1IEN[7:0]**	P1[7:0] port input enable	1 Enable 0 Disable	0xff	R/W	
P2 Port Input Data Register (P2_IN)	**0x5220** (8 bits)	D7–0	**P2IN[7:0]**	P2[7:0] port input data	1 1 (H) 0 0 (L)	×	R	
P2 Port Output Data Register (P2_OUT)	**0x5221** (8 bits)	D7–0	**P2OUT[7:0]**	P2[7:0] port output data	1 1 (H) 0 0 (L)	0	R/W	
P2 Port Output Enable Register (P2_OEN)	**0x5222** (8 bits)	D7–0	**P2OEN[7:0]**	P2[7:0] port output enable	1 Enable 0 Disable	0	R/W	
P2 Port Pull-up Control Register (P2_PU)	**0x5223** (8 bits)	D7–0	**P2PU[7:0]**	P2[7:0] port pull-up enable	1 Enable 0 Disable	1 (0xff)	R/W	
P2 Port Interrupt Mask Register (P2_IMSK)	**0x5225** (8 bits)	D7–0	**P2IE[7:0]**	P2[7:0] port interrupt enable	1 Enable 0 Disable	0	R/W	
P2 Port Interrupt Edge Select Register (P2_EDGE)	**0x5226** (8 bits)	D7–0	**P2EDGE[7:0]**	P2[7:0] port interrupt edge select	1 Falling edge 0 Rising edge	0	R/W	
P2 Port Interrupt Flag Register (P2_IFLG)	**0x5227** (8 bits)	D7–0	**P2IF[7:0]**	P2[7:0] port interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
P2 Port Chattering Filter Control Register (P2_CHAT)	**0x5228** (8 bits)	D7	–	reserved	–	–	–	0 when being read.
		D6–4	**P2CF2[2:0]**	P2[7:4] chattering filter time	P2CF2[2:0] Filter time 0x7 16384/fPCLK 0x6 8192/fPCLK 0x5 4096/fPCLK 0x4 2048/fPCLK 0x3 1024/fPCLK 0x2 512/fPCLK 0x1 256/fPCLK 0x0 None	0 0x0	R/W R/W	
		D3	–	reserved	–	–	–	0 when being read.
		D2–0	**P2CF1[2:0]**	P2[3:0] chattering filter time	P2CF1[2:0] Filter time 0x7 16384/fPCLK 0x6 8192/fPCLK 0x5 4096/fPCLK 0x4 2048/fPCLK 0x3 1024/fPCLK 0x2 512/fPCLK 0x1 256/fPCLK 0x0 None	0x0	R/W	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
P2 Port Input Enable Register (P2_IEN)	**0x522a** (8 bits)	D7–0	**P2IEN[7:0]**	P2[7:0] port input enable	1 Enable 0 Disable	0xff	R/W	
P3 Port Input Data Register (P3_IN)	**0x5230** (8 bits)	D7–0	**P3IN[7:0]**	P3[7:0] port input data	1 1 (H) 0 0 (L)	×	R	
P3 Port Output Data Register (P3_OUT)	**0x5231** (8 bits)	D7–0	**P3OUT[7:0]**	P3[7:0] port output data	1 1 (H) 0 0 (L)	0	R/W	
P3 Port Output Enable Register (P3_OEN)	**0x5232** (8 bits)	D7–0	**P3OEN[7:0]**	P3[7:0] port output enable	1 Enable 0 Disable	0	R/W	
P3 Port Pull-up Control Register (P3_PU)	**0x5233** (8 bits)	D7–0	**P3PU[7:0]**	P3[7:0] port pull-up enable	1 Enable 0 Disable	1 (0xff)	R/W	
P3 Port Interrupt Mask Register (P3_IMSK)	**0x5235** (8 bits)	D7–0	**P3IE[7:0]**	P3[7:0] port interrupt enable	1 Enable 0 Disable	0	R/W	
P3 Port Interrupt Edge Select Register (P3_EDGE)	**0x5236** (8 bits)	D7–0	**P3EDGE[7:0]**	P3[7:0] port interrupt edge select	1 Falling edge 0 Rising edge	0	R/W	
P3 Port Interrupt Flag Register (P3_IFLG)	**0x5237** (8 bits)	D7–0	**P3IF[7:0]**	P3[7:0] port interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
P3 Port Chattering Filter Control Register (P3_CHAT)	**0x5238** (8 bits)	D7	–	reserved	–	–	–	0 when being read.
		D6–4	**P3CF2[2:0]**	P3[7:4] chattering filter time	P3CF2[2:0] / Filter time 0x7 16384/f_{PCLK} 0x6 8192/f_{PCLK} 0x5 4096/f_{PCLK} 0x4 2048/f_{PCLK} 0x3 1024/f_{PCLK} 0x2 512/f_{PCLK} 0x1 256/f_{PCLK} 0x0 None	0 0x0	R/W R/W	
		D3	–	reserved	–	–	–	0 when being read.
		D2–0	**P3CF1[2:0]**	P3[3:0] chattering filter time	P3CF1[2:0] / Filter time 0x7 16384/f_{PCLK} 0x6 8192/f_{PCLK} 0x5 4096/f_{PCLK} 0x4 2048/f_{PCLK} 0x3 1024/f_{PCLK} 0x2 512/f_{PCLK} 0x1 256/f_{PCLK} 0x0 None	0x0	R/W	
P3 Port Input Enable Register (P3_IEN)	**0x523a** (8 bits)	D7–0	**P3IEN[7:0]**	P3[7:0] port input enable	1 Enable 0 Disable	0xff	R/W	
P4 Port Input Data Register (P4_IN)	**0x5240** (8 bits)	D7–6	–	reserved	–	–	–	0 when being read.
		D5–0	**P4IN[5:0]**	P4[5:0] port input data	1 1 (H) 0 0 (L)	×	R	
P4 Port Output Data Register (P4_OUT)	**0x5241** (8 bits)	D7–6	–	reserved	–	–	–	0 when being read.
		D5–0	**P4OUT[5:0]**	P4[5:0] port output data	1 1 (H) 0 0 (L)	0	R/W	
P4 Port Output Enable Register (P4_OEN)	**0x5242** (8 bits)	D7–6	–	reserved	–	–	–	0 when being read.
		D5–0	**P4OEN[5:0]**	P4[5:0] port output enable	1 Enable 0 Disable	0	R/W	
P4 Port Pull-up Control Register (P4_PU)	**0x5243** (8 bits)	D7–6	–	reserved	–	–	–	0 when being read.
		D5–0	**P4PU[5:0]**	P4[5:0] port pull-up enable	1 Enable 0 Disable	1 (0x2f)	R/W	
P4 Port Interrupt Mask Register (P4_IMSK)	**0x5245** (8 bits)	D7–6	–	reserved	–	–	–	0 when being read.
		D5–0	**P4IE[5:0]**	P4[5:0] port interrupt enable	1 Enable 0 Disable	0	R/W	
P4 Port Interrupt Edge Select Register (P4_EDGE)	**0x5246** (8 bits)	D7–6	–	reserved	–	–	–	0 when being read.
		D5–0	**P4EDGE[5:0]**	P4[5:0] port interrupt edge select	1 Falling edge 0 Rising edge	0	R/W	
P4 Port Interrupt Flag Register (P4_IFLG)	**0x5247** (8 bits)	D7–6	–	reserved	–	–	–	0 when being read.
		D5–0	**P4IF[5:0]**	P4[5:0] port interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
P4 Port Chattering Filter Control Register (P4_CHAT)	**0x5248** (8 bits)	D7	**–**	reserved	–	–	–	0 when being read.
		D6–4	**P4CF2[2:0]**	P4[5:4] chattering filter time	P4CF2[2:0] / Filter time: 0x7 16384/fPCLK; 0x6 8192/fPCLK; 0x5 4096/fPCLK; 0x4 2048/fPCLK; 0x3 1024/fPCLK; 0x2 512/fPCLK; 0x1 256/fPCLK; 0x0 None	0x0	R/W	
		D3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**P4CF1[2:0]**	P4[3:0] chattering filter time	P4CF1[2:0] / Filter time: 0x7 16384/fPCLK; 0x6 8192/fPCLK; 0x5 4096/fPCLK; 0x4 2048/fPCLK; 0x3 1024/fPCLK; 0x2 512/fPCLK; 0x1 256/fPCLK; 0x0 None	0x0	R/W	
P4 Port Input Enable Register (P4_IEN)	**0x524a** (8 bits)	D7–6	**–**	reserved	–	–	–	0 when being read.
		D5–0	**P4IEN[5:0]**	P4[5:0] port input enable	1 Enable; 0 Disable	0x2f	R/W	
P5 Port Input Data Register (P5_IN)	**0x5250** (8 bits)	D7–6	**–**	reserved	–	–	–	0 when being read.
		D5–0	**P5IN[5:0]**	P5[5:0] port input data	1 1 (H); 0 0 (L)	×	R	
P5 Port Output Data Register (P5_OUT)	**0x5251** (8 bits)	D7–6	**–**	reserved	–	–	–	0 when being read.
		D5–0	**P5OUT[5:0]**	P5[5:0] port output data	1 1 (H); 0 0 (L)	0	R/W	
P5 Port Output Enable Register (P5_OEN)	**0x5252** (8 bits)	D7–6	**–**	reserved	–	–	–	0 when being read.
		D5–0	**P5OEN[5:0]**	P5[5:0] port output enable	1 Enable; 0 Disable	0	R/W	
P5 Port Pull-up Control Register (P5_PU)	**0x5253** (8 bits)	D7–6	**–**	reserved	–	–	–	0 when being read.
		D5–0	**P5PU[5:0]**	P5[5:0] port pull-up enable	1 Enable; 0 Disable	1 (0x2f)	R/W	
P5 Port Interrupt Mask Register (P5_IMSK)	**0x5255** (8 bits)	D7–6	**–**	reserved	–	–	–	0 when being read.
		D5–0	**P5IE[5:0]**	P5[5:0] port interrupt enable	1 Enable; 0 Disable	0	R/W	
P5 Port Interrupt Edge Select Register (P5_EDGE)	**0x5256** (8 bits)	D7–6	**–**	reserved	–	–	–	0 when being read.
		D5–0	**P5EDGE[5:0]**	P5[5:0] port interrupt edge select	1 Falling edge; 0 Rising edge	0	R/W	
P5 Port Interrupt Flag Register (P5_IFLG)	**0x5257** (8 bits)	D7–6	**–**	reserved	–	–	–	0 when being read.
		D5–0	**P5IF[5:0]**	P5[5:0] port interrupt flag	1 Cause of interrupt occurred; 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
P5 Port Chattering Filter Control Register (P5_CHAT)	**0x5258** (8 bits)	D7	**–**	reserved	–	–	–	0 when being read.
		D6–4	**P5CF2[2:0]**	P5[5:4] chattering filter time	P5CF2[2:0] / Filter time: 0x7 16384/fPCLK; 0x6 8192/fPCLK; 0x5 4096/fPCLK; 0x4 2048/fPCLK; 0x3 1024/fPCLK; 0x2 512/fPCLK; 0x1 256/fPCLK; 0x0 None	0x0	R/W	
		D3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**P5CF1[2:0]**	P5[3:0] chattering filter time	P5CF1[2:0] / Filter time: 0x7 16384/fPCLK; 0x6 8192/fPCLK; 0x5 4096/fPCLK; 0x4 2048/fPCLK; 0x3 1024/fPCLK; 0x2 512/fPCLK; 0x1 256/fPCLK; 0x0 None	0x0	R/W	
P5 Port Input Enable Register (P5_IEN)	**0x525a** (8 bits)	D7–6	**–**	reserved	–	–	–	0 when being read.
		D5–0	**P5IEN[5:0]**	P5[5:0] port input enable	1 Enable; 0 Disable	0x2f	R/W	

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	* S1C17564 only
					0x3	reserved			
					0x2	US_SSI1*			
					0x1	AIN3			
					0x0	P03			
		D5–4	**P02MUX[1:0]**	P02 port function select	P02MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	US_SSI0*			
					0x1	AIN2			
					0x0	P02			
		D3–2	**P01MUX[1:0]**	P01 port function select	P01MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	AIN1			
					0x0	P01			
		D1–0	**P00MUX[1:0]**	P00 port function select	P00MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	AIN0			
					0x0	P00			
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	TOUT5/CAP5			
					0x1	#SPISS0			
					0x0	P13			
		D5–4	**P12MUX[1:0]**	P12 port function select	P12MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	SPICLK0			
					0x0	P12			
		D3–2	**P11MUX[1:0]**	P11 port function select	P11MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	SDO0			
					0x0	P11			
		D1–0	**P10MUX[1:0]**	P10 port function select	P10MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	SDI0			
					0x0	P10			
P1[7:4] Port Function Select Register (P14_17PMUX)	**0x52a3** (8 bits)	D7–6	**P17MUX[1:0]**	P17 port function select	P17MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	SCL0			
					0x0	P17			
		D5–4	**P16MUX[1:0]**	P16 port function select	P16MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	SPICLK1			
					0x1	SCLK1			
					0x0	P16			
		D3–2	**P15MUX[1:0]**	P15 port function select	P15MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	SDO1			
					0x1	SOUT1			
					0x0	P15			
		D1–0	**P14MUX[1:0]**	P14 port function select	P14MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	SDI1			
					0x1	SIN1			
					0x0	P14			

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
P2[3:0] Port Function Select Register (P20_23PMUX)	**0x52a4** (8 bits)	D7–6	**P23MUX[1:0]**	P23 port function select	P23MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	SDI2			
					0x0	P23/EXCL2			
		D5–4	**P22MUX[1:0]**	P22 port function select	P22MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	FOUTB			
					0x0	P22/EXCL1			
		D3–2	**P21MUX[1:0]**	P21 port function select	P21MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	TOUT3/CAP3			
					0x0	P21			
		D1–0	**P20MUX[1:0]**	P20 port function select	P20MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	TOUT2/CAP2			
					0x0	P20			
P2[7:4] Port Function Select Register (P24_27PMUX)	**0x52a5** (8 bits)	D7–6	**P27MUX[1:0]**	P27 port function select	P27MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	SCL1			
					0x0	P27			
		D5–4	**P26MUX[1:0]**	P26 port function select	P26MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	SDA1			
					0x0	P26			
		D3–2	**P25MUX[1:0]**	P25 port function select	P25MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	#SPISS2			
					0x1	#BFR			
					0x0	P25			
		D1–0	**P24MUX[1:0]**	P24 port function select	P24MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	SDO2			
					0x0	P24/EXCL3			
P3[3:0] Port Function Select Register (P30_33PMUX)	**0x52a6** (8 bits)	D7–6	**P33MUX[1:0]**	P33 port function select	P33MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	SPICLK2			
					0x1	REMI			
					0x0	P33			
		D5–4	**P32MUX[1:0]**	P32 port function select	P32MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	FOUTA			
					0x1	TOUT4/CAP4			
					0x0	P32			
		D3–2	**P31MUX[1:0]**	P31 port function select	P31MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	#ADTRG			
					0x1	#BFR			
					0x0	P31			
		D1–0	**P30MUX[1:0]**	P30 port function select	P30MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	TOUT0/CAP0			
					0x0	P30			

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
P3[7:4] Port Function Select Register (P34_37PMUX)	**0x52a7** (8 bits)	D7–6	**P37MUX[1:0]**	P37 port function select	P37MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	P37			
					0x0	DST2			
		D5–4	**P36MUX[1:0]**	P36 port function select	P36MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	P36			
					0x0	DSIO			
		D3–2	**P35MUX[1:0]**	P35 port function select	P35MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	P35			
					0x0	DCLK			
		D1–0	**P34MUX[1:0]**	P34 port function select	P34MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	#SPISS1			
					0x1	REMO			
					0x0	P34			
P4[3:0] Port Function Select Register (P40_43PMUX)	**0x52a8** (8 bits)	D7–6	**P43MUX[1:0]**	P43 port function select	P43MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	REMI			
					0x1	SDA1			
					0x0	P43			
		D5–4	**P42MUX[1:0]**	P42 port function select	P42MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	TOUT1/CAP1			
					0x1	SCLK0			
					0x0	P42			
		D3–2	**P41MUX[1:0]**	P41 port function select	P41MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	TOUT7/CAP7			
					0x1	SOUT0			
					0x0	P41			
		D1–0	**P40MUX[1:0]**	P40 port function select	P40MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	TOUT6/CAP6			
					0x1	SIN0			
					0x0	P40			
P4[5:4] Port Function Select Register (P44_45PMUX)	**0x52a9** (8 bits)	D7–4	**–**	reserved	–		–	–	0 when being read.
		D3–2	**P45MUX[1:0]**	P45 port function select	P45MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	SDA0			
					0x0	P45/EXCL0			
		D1–0	**P44MUX[1:0]**	P44 port function select	P44MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	REMO			
					0x1	SCL1			
					0x0	P44			
P5[3:0] Port Function Select Register (P50_53PMUX)	**0x52aa** (8 bits)	D7–6	**P53MUX[1:0]**	P53 port function select	P53MUX[1:0]	Function	0x0	R/W	* S1C17564 only
					0x3	reserved			
					0x2	reserved			
					0x1	US_SDI1*			
					0x0	P53			
		D5–4	**P52MUX[1:0]**	P52 port function select	P52MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	US_SCK0*			
					0x0	P52			
		D3–2	**P51MUX[1:0]**	P51 port function select	P51MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	US_SDO0*			
					0x0	P51			
		D1–0	**P50MUX[1:0]**	P50 port function select	P50MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	US_SDI0*			
					0x0	P50			

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
P5[5:4] Port Function Select Register (P54_55PMUX)	0x52ab (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.
		D3–2	P55MUX[1:0]	P55 port function select	P55MUX[1:0] Function 0x3 reserved 0x2 reserved 0x1 US_SCK1* 0x0 P55	0x0	R/W	* S1C17564 only
		D1–0	P54MUX[1:0]	P54 port function select	P54MUX[1:0] Function 0x3 reserved 0x2 reserved 0x1 US_SDO1* 0x0 P54	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.
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 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	0x6 (= 16KB)	0x6	R	
		D3	–	reserved	–	–	–	0 when being read.
		D2–0	IRAMSZ[2:0]	IRAM size select	IRAMSZ[2:0] Size 0x7 reserved 0x6 16KB 0x5 512B 0x4 1KB 0x3 2KB 0x2 4KB 0x1 8KB 0x0 12KB	0x6	R/W	
Vector Table Address Low Register (MISC_TTBRL)	0x5328 (16 bits)	D15–8	TTBR[15:8]	Vector table base address A[15:8]	0x0–0xff	0x80	R/W	
		D7–0	TTBR[7:0]	Vector table base address A[7:0] (fixed at 0)	0x0	0x0	R	
Vector Table Address High Register (MISC_TTBRH)	0x532a (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	TTBR[23:16]	Vector table base address A[23:16]	0x0–0xff	0x0	R/W	
PSR Register (MISC_PSR)	0x532c (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–5	PSRIL[2:0]	PSR interrupt level (IL) bits	0x0 to 0x7	0x0	R	
		D4	PSRIE	PSR interrupt enable (IE) bit	1 1 (enable) 0 0 (disable)	0	R	
		D3	PSRC	PSR carry (C) flag	1 1 (set) 0 0 (cleared)	0	R	
		D2	PSRV	PSR overflow (V) flag	1 1 (set) 0 0 (cleared)	0	R	
		D1	PSRZ	PSR zero (Z) flag	1 1 (set) 0 0 (cleared)	0	R	
		D0	PSRN	PSR negative (N) flag	1 1 (set) 0 0 (cleared)	0	R	

0x5340–0x5346 IR Remote Controller

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
REMC Configuration Register (REMC_CFG)	0x5340 (16 bits)	D15–12	CGCLK[3:0]	Carrier generator clock division ratio select	CGCLK[3:0] LCCLK[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
		D11–8	LCCLK[3:0]	Length counter clock division ratio select	(same as above)	0x0	R/W	
		D7–2	–	reserved	–	–	–	0 when being read.
		D1	REMMD	REMC mode select	1 Receive / 0 Transmit	0	R/W	
		D0	REMEN	REMC enable	1 Enable / 0 Disable	0	R/W	
REMC Carrier Length Setup Register (REMC_CAR)	0x5342 (16 bits)	D15–14	–	reserved	–	–	–	0 when being read.
		D13–8	REMCL[5:0]	Carrier L length setup	0x0 to 0x3f	0x0	R/W	
		D7–6	–	reserved	–	–	–	0 when being read.
		D5–0	REMCH[5:0]	Carrier H length setup	0x0 to 0x3f	0x0	R/W	
REMC Length Counter Register (REMC_LCNT)	0x5344 (16 bits)	D15–8	REMLEN[7:0]	Transmit/receive data length count (down counter)	0x0 to 0xff	0x0	R/W	
		D7–1	–	reserved	–	–	–	0 when being read.
		D0	REMDT	Transmit/receive data	1 1 (H) / 0 0 (L)	0	R/W	
REMC Interrupt Control Register (REMC_INT)	0x5346 (16 bits)	D15–11	–	reserved	–	–	–	0 when being read.
		D10	REMFIF	Falling edge interrupt flag	1 Cause of interrupt occurred / 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D9	REMRIF	Rising edge interrupt flag		0	R/W	
		D8	REMUIF	Underflow interrupt flag		0	R/W	
		D7–3	–	reserved	–	–	–	0 when being read.
		D2	REMFIE	Falling edge interrupt enable	1 Enable / 0 Disable	0	R/W	
		D1	REMRIE	Rising edge interrupt enable	1 Enable / 0 Disable	0	R/W	
		D0	REMUIE	Underflow interrupt enable	1 Enable / 0 Disable	0	R/W	

0x5380–0x5388 A/D Converter

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
A/D Conversion Result Register (ADC10_ADD)	0x5380 (16 bits)	D15–0	ADD[15:0]	A/D converted data ADD[9:0] are effective when STMD = 0 (ADD[15:10] = 0) ADD[15:6] are effective when STMD = 1 (ADD[5:0] = 0)	0x0 to 0x3ff	0x0	R	
A/D Trigger/ Channel Select Register (ADC10_TRG)	0x5382 (16 bits)	D15–14	–	reserved	–	–	–	0 when being read.
		D13–11	ADCE[2:0]	End channel select	0x0 to 0x3	0x0	R/W	
		D10–8	ADCS[2:0]	Start channel select	0x0 to 0x3	0x0	R/W	
		D7	STMD	Conversion result storing mode	1 ADD[15:6] / 0 ADD[9:0]	0	R/W	
		D6	ADMS	Conversion mode select	1 Continuous / 0 Single	0	R/W	
		D5–4	ADTS[1:0]	Conversion trigger select	ADTS[1:0] / Trigger 0x3 #ADTRG pin 0x2 reserved 0x1 T16 Ch.0 0x0 Software	0x0	R/W	
		D3	–	reserved	–	–	–	0 when being read.
		D2–0	ADST[2:0]	Sampling time setting	ADST[2:0] / Sampling time 0x7 9 cycles 0x6 8 cycles 0x5 7 cycles 0x4 6 cycles 0x3 5 cycles 0x2 4 cycles 0x1 3 cycles 0x0 2 cycles	0x7	R/W	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
A/D Control/ Status Register (ADC10_CTL)	0x5384 (16 bits)	D15	–	reserved	–	–	–	0 when being read.
		D14–12	ADICH[2:0]	Conversion channel indicator	0x0 to 0x3	0x0	R	
		D11	–	reserved	–	–	–	0 when being read.
		D10	ADIBS	ADC10 status	1 Busy 0 Idle	0	R	
		D9	ADOWE	Overwrite error flag	1 Error 0 Normal	0	R/W	Reset by writing 1.
		D8	ADCF	Conversion completion flag	1 Completed 0 Run/Stand-by	0	R	Reset when ADC10_ADD is read.
		D7–6	–	reserved	–	–	–	0 when being read.
		D5	ADOIE	Overwrite error interrupt enable	1 Enable 0 Disable	0	R/W	
		D4	ADCIE	Conversion completion int. enable	1 Enable 0 Disable	0	R/W	
		D3–2	–	reserved	–	–	–	0 when being read.
		D1	ADCTL	A/D conversion control	1 Start 0 Stop	0	R/W	
		D0	ADEN	ADC10 enable	1 Enable 0 Disable	0	R/W	
A/D Clock Control Register (ADC10_CLK)	0x5386 (16 bits)	D15–4	–	reserved	–	–	–	0 when being read.
		D3–0	ADDF[3:0]	A/D converter clock division ratio select	ADDF[3:0] / Division ratio: 0xf reserved; 0xe 1/32768; 0xd 1/16384; 0xc 1/8192; 0xb 1/4096; 0xa 1/2048; 0x9 1/1024; 0x8 1/512; 0x7 1/256; 0x6 1/128; 0x5 1/64; 0x4 1/32; 0x3 1/16; 0x2 1/8; 0x1 1/4; 0x0 1/2	0x0	R/W	Source clock = PCLK
A/D Comparator Setting Register (ADC10_COM)	0x5388 (16 bits)	D15–6	–	reserved	–	–	–	0 when being read.
		D5–4	FSEL[1:0]	A/D comparator adjustment	0x0 to 0x3	0x0	R/W	
		D3–2	–	reserved	–	–	–	0 when being read.
		D1–0	XPD[1:0]	A/D comparator adjustment	0x0 to 0x3	0x3	R/W	

0x5068, 0x5400–0x540c 16-bit PWM Timer Ch.0

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16A Clock Control Register Ch.0 (T16A_CLK0)	0x5068 (8 bits)	D7–4	CLKDIV [3:0]	Clock division ratio select	CLKDIV[3:0] / Division ratio OSC3 or IOSC / OSC1: 0xf – / –; 0xe 1/16384 / –; 0xd 1/8192 / –; 0xc 1/4096 / –; 0xb 1/2048 / –; 0xa 1/1024 / –; 0x9 1/512 / –; 0x8 1/256 / 1/256; 0x7 1/128 / 1/128; 0x6 1/64 / 1/64; 0x5 1/32 / 1/32; 0x4 1/16 / 1/16; 0x3 1/8 / 1/8; 0x2 1/4 / 1/4; 0x1 1/2 / 1/2; 0x0 1/1 / 1/1	0x0	R/W	
		D3–2	CLKSRC [1:0]	Clock source select	CLKSRC[1:0] / Clock source: 0x3 External clock; 0x2 OSC3; 0x1 OSC1; 0x0 IOSC*	0x0	R/W	* S1C17564 only
		D1	MULTIMD	Multi-comparator/capture mode select	1 Multi 0 Normal	0	R/W	
		D0	CLKEN	Count clock enable	1 Enable 0 Disable	0	R/W	
T16A Counter Ch.0 Control Register (T16A_CTL0)	0x5400 (16 bits)	D15–6	–	reserved	–	–	–	0 when being read.
		D5–4	CCABCNT [1:0]	Counter select	CCABCNT[1:0] / Counter Ch.: 0x3 Ch.3; 0x2 Ch.2; 0x1 Ch.1; 0x0 Ch.0	0x0	R/W	
		D3	CBUFEN	Compare buffer enable	1 Enable 0 Disable	0	R/W	
		D2	TRMD	Count mode select	1 One-shot 0 Repeat	0	R/W	
		D1	PRESET	Counter reset	1 Reset 0 Ignored	0	W	0 when being read.
		D0	PRUN	Counter run/stop control	1 Run 0 Stop	0	R/W	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16A Counter Ch.0 Data Register (T16A_TC0)	**0x5402** (16 bits)	D15–0	**T16ATC [15:0]**	Counter data T16ATC15 = MSB T16ATC0 = LSB	0x0 to 0xffff	0x0	R	
T16A Comparator/ Capture Ch.0 Control Register (T16A_CCCTL0)	**0x5404** (16 bits)	D15–14	**CAPBTRG [1:0]**	Capture B trigger select	CAPBTRG[1:0]: Trigger edge 0x3: ↑ and ↓ 0x2: ↓ 0x1: ↑ 0x0: None	0x0	R/W	
		D13–12	**TOUTBMD [1:0]**	TOUT B mode select	TOUTBMD[1:0]: Mode 0x3: cmp B: ↑ or ↓ 0x2: cmp A: ↑ or ↓ 0x1: cmp A: ↑, B: ↓ 0x0: Off	0x0	R/W	
		D11–10	–	reserved	–	–	–	0 when being read.
		D9	**TOUTBINV**	TOUT B invert	1: Invert, 0: Normal	0	R/W	
		D8	**CCBMD**	T16A_CCB register mode select	1: Capture, 0: Comparator	0	R/W	
		D7–6	**CAPATRG [1:0]**	Capture A trigger select	CAPATRG[1:0]: Trigger edge 0x3: ↑ and ↓ 0x2: ↓ 0x1: ↑ 0x0: None	0x0	R/W	
		D5–4	**TOUTAMD [1:0]**	TOUT A mode select	TOUTAMD[1:0]: Mode 0x3: cmp B: ↑ or ↓ 0x2: cmp A: ↑ or ↓ 0x1: cmp A: ↑, B: ↓ 0x0: Off	0x0	R/W	
		D3–2	–	reserved	–	–	–	0 when being read.
		D1	**TOUTAINV**	TOUT A invert	1: Invert, 0: Normal	0	R/W	
		D0	**CCAMD**	T16A_CCA register mode select	1: Capture, 0: Comparator	0	R/W	
T16A Comparator/ Capture Ch.0 A Data Register (T16A_CCA0)	**0x5406** (16 bits)	D15–0	**CCA[15:0]**	Compare/capture A data CCA15 = MSB CCA0 = LSB	0x0 to 0xffff	0x0	R/W	
T16A Comparator/ Capture Ch.0 B Data Register (T16A_CCB0)	**0x5408** (16 bits)	D15–0	**CCB[15:0]**	Compare/capture B data CCB15 = MSB CCB0 = LSB	0x0 to 0xffff	0x0	R/W	
T16A Comparator/ Capture Ch.0 Interrupt Enable Register (T16A_IEN0)	**0x540a** (16 bits)	D15–6	–	reserved	–	–	–	0 when being read.
		D5	**CAPBOWIE**	Capture B overwrite interrupt enable	1: Enable, 0: Disable	0	R/W	
		D4	**CAPAOWIE**	Capture A overwrite interrupt enable	1: Enable, 0: Disable	0	R/W	
		D3	**CAPBIE**	Capture B interrupt enable	1: Enable, 0: Disable	0	R/W	
		D2	**CAPAIE**	Capture A interrupt enable	1: Enable, 0: Disable	0	R/W	
		D1	**CBIE**	Compare B interrupt enable	1: Enable, 0: Disable	0	R/W	
		D0	**CAIE**	Compare A interrupt enable	1: Enable, 0: Disable	0	R/W	
T16A Comparator/ Capture Ch.0 Interrupt Flag Register (T16A_IFLG0)	**0x540c** (16 bits)	D15–6	–	reserved	–	–	–	0 when being read.
		D5	**CAPBOWIF**	Capture B overwrite interrupt flag	1: Cause of interrupt occurred, 0: Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D4	**CAPAOWIF**	Capture A overwrite interrupt flag		0	R/W	
		D3	**CAPBIF**	Capture B interrupt flag		0	R/W	
		D2	**CAPAIF**	Capture A interrupt flag		0	R/W	
		D1	**CBIF**	Compare B interrupt flag		0	R/W	
		D0	**CAIF**	Compare A interrupt flag		0	R/W	

0x5069, 0x5420–0x542c 16-bit PWM Timer Ch.1

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16A Clock Control Register Ch.1 (T16A_CLK1)	**0x5069** (8 bits)	D7–4	**CLKDIV [3:0]**	Clock division ratio select	CLKDIV[3:0]: Division ratio — OSC3 or IOSC / OSC1 0xf: – / – 0xe: 1/16384 / – 0xd: 1/8192 / – 0xc: 1/4096 / – 0xb: 1/2048 / – 0xa: 1/1024 / – 0x9: 1/512 / – 0x8: 1/256 / 1/256 0x7: 1/128 / 1/128 0x6: 1/64 / 1/64 0x5: 1/32 / 1/32 0x4: 1/16 / 1/16 0x3: 1/8 / 1/8 0x2: 1/4 / 1/4 0x1: 1/2 / 1/2 0x0: 1/1 / 1/1	0x0	R/W	
		D3–2	**CLKSRC [1:0]**	Clock source select	CLKSRC[1:0]: Clock source 0x3: External clock 0x2: OSC3 0x1: OSC1 0x0: IOSC*	0x0	R/W	* S1C17564 only
		D1	**–**	reserved	–	–	–	
		D0	**CLKEN**	Count clock enable	1: Enable / 0: Disable	0	R/W	
T16A Counter Ch.1 Control Register (T16A_CTL1)	**0x5420** (16 bits)	D15–6	**–**	reserved	–	–	–	0 when being read.
		D5–4	**CCABCNT [1:0]**	Counter select	CCABCNT[1:0]: Counter Ch. 0x3: Ch.3 0x2: Ch.2 0x1: Ch.1 0x0: Ch.0	0x0	R/W	
		D3	**CBUFEN**	Compare buffer enable	1: Enable / 0: Disable	0	R/W	
		D2	**TRMD**	Count mode select	1: One-shot / 0: Repeat	0	R/W	
		D1	**PRESET**	Counter reset	1: Reset / 0: Ignored	0	W	0 when being read.
		D0	**PRUN**	Counter run/stop control	1: Run / 0: Stop	0	R/W	
T16A Counter Ch.1 Data Register (T16A_TC1)	**0x5422** (16 bits)	D15–0	**T16ATC [15:0]**	Counter data T16ATC15 = MSB T16ATC0 = LSB	0x0 to 0xffff	0x0	R	
T16A Comparator/ Capture Ch.1 Control Register (T16A_CCCTL1)	**0x5424** (16 bits)	D15–14	**CAPBTRG [1:0]**	Capture B trigger select	CAPBTRG[1:0]: Trigger edge 0x3: ↑ and ↓ 0x2: ↓ 0x1: ↑ 0x0: None	0x0	R/W	
		D13–12	**TOUTBMD [1:0]**	TOUT B mode select	TOUTBMD[1:0]: Mode 0x3: cmp B: ↑ or ↓ 0x2: cmp A: ↑ or ↓ 0x1: cmp A: ↑, B: ↓ 0x0: Off	0x0	R/W	
		D11–10	**–**	reserved	–	–	–	0 when being read.
		D9	**TOUTBINV**	TOUT B invert	1: Invert / 0: Normal	0	R/W	
		D8	**CCBMD**	T16A_CCB register mode select	1: Capture / 0: Comparator	0	R/W	
		D7–6	**CAPATRG [1:0]**	Capture A trigger select	CAPATRG[1:0]: Trigger edge 0x3: ↑ and ↓ 0x2: ↓ 0x1: ↑ 0x0: None	0x0	R/W	
		D5–4	**TOUTAMD [1:0]**	TOUT A mode select	TOUTAMD[1:0]: Mode 0x3: cmp B: ↑ or ↓ 0x2: cmp A: ↑ or ↓ 0x1: cmp A: ↑, B: ↓ 0x0: Off	0x0	R/W	
		D3–2	**–**	reserved	–	–	–	0 when being read.
		D1	**TOUTAINV**	TOUT A invert	1: Invert / 0: Normal	0	R/W	
		D0	**CCAMD**	T16A_CCA register mode select	1: Capture / 0: Comparator	0	R/W	
T16A Comparator/ Capture Ch.1 A Data Register (T16A_CCA1)	**0x5426** (16 bits)	D15–0	**CCA[15:0]**	Compare/capture A data CCA15 = MSB CCA0 = LSB	0x0 to 0xffff	0x0	R/W	
T16A Comparator/ Capture Ch.1 B Data Register (T16A_CCB1)	**0x5428** (16 bits)	D15–0	**CCB[15:0]**	Compare/capture B data CCB15 = MSB CCB0 = LSB	0x0 to 0xffff	0x0	R/W	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16A Comparator/ Capture Ch.1 Interrupt Enable Register (T16A_IEN1)	**0x542a** (16 bits)	D15–6	**–**	reserved	–	–	–	0 when being read.
		D5	**CAPBOWIE**	Capture B overwrite interrupt enable	1 Enable 0 Disable	0	R/W	
		D4	**CAPAOWIE**	Capture A overwrite interrupt enable	1 Enable 0 Disable	0	R/W	
		D3	**CAPBIE**	Capture B interrupt enable	1 Enable 0 Disable	0	R/W	
		D2	**CAPAIE**	Capture A interrupt enable	1 Enable 0 Disable	0	R/W	
		D1	**CBIE**	Compare B interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	**CAIE**	Compare A interrupt enable	1 Enable 0 Disable	0	R/W	
T16A Comparator/ Capture Ch.1 Interrupt Flag Register (T16A_IFLG1)	**0x542c** (16 bits)	D15–6	**–**	reserved	–	–	–	0 when being read.
		D5	**CAPBOWIF**	Capture B overwrite interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D4	**CAPAOWIF**	Capture A overwrite interrupt flag		0	R/W	
		D3	**CAPBIF**	Capture B interrupt flag		0	R/W	
		D2	**CAPAIF**	Capture A interrupt flag		0	R/W	
		D1	**CBIF**	Compare B interrupt flag		0	R/W	
		D0	**CAIF**	Compare A interrupt flag		0	R/W	

0x506a, 0x5440–0x544c 16-bit PWM Timer Ch.2

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16A Clock Control Register Ch.2 (T16A_CLK2)	**0x506a** (8 bits)	D7–4	**CLKDIV [3:0]**	Clock division ratio select	CLKDIV[3:0]: 0xf – / – ; 0xe 1/16384 / – ; 0xd 1/8192 / – ; 0xc 1/4096 / – ; 0xb 1/2048 / – ; 0xa 1/1024 / – ; 0x9 1/512 / – ; 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: OSC3 or IOSC / OSC1)	0x0	R/W	
		D3–2	**CLKSRC [1:0]**	Clock source select	CLKSRC[1:0]: 0x3 External clock ; 0x2 OSC3 ; 0x1 OSC1 ; 0x0 IOSC*	0x0	R/W	* S1C17564 only
		D1	**–**	reserved	–	–	–	
		D0	**CLKEN**	Count clock enable	1 Enable 0 Disable	0	R/W	
T16A Counter Ch.2 Control Register (T16A_CTL2)	**0x5440** (16 bits)	D15–6	**–**	reserved	–	–	–	0 when being read.
		D5–4	**CCABCNT [1:0]**	Counter select	CCABCNT[1:0]: 0x3 Ch.3 ; 0x2 Ch.2 ; 0x1 Ch.1 ; 0x0 Ch.0	0x0	R/W	
		D3	**CBUFEN**	Compare buffer enable	1 Enable 0 Disable	0	R/W	
		D2	**TRMD**	Count mode select	1 One-shot 0 Repeat	0	R/W	
		D1	**PRESET**	Counter reset	1 Reset 0 Ignored	0	W	0 when being read.
		D0	**PRUN**	Counter run/stop control	1 Run 0 Stop	0	R/W	
T16A Counter Ch.2 Data Register (T16A_TC2)	**0x5442** (16 bits)	D15–0	**T16ATC [15:0]**	Counter data T16ATC15 = MSB T16ATC0 = LSB	0x0 to 0xffff	0x0	R	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16A Comparator/ Capture Ch.2 Control Register (T16A_CCCTL2)	**0x5444** (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.2 A Data Register (T16A_CCA2)	**0x5446** (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.2 B Data Register (T16A_CCB2)	**0x5448** (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.2 Interrupt Enable Register (T16A_IEN2)	**0x544a** (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.2 Interrupt Flag Register (T16A_IFLG2)	**0x544c** (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	

0x506b, 0x5460–0x546c 16-bit PWM Timer Ch.3

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16A Clock Control Register Ch.3 (T16A_CLK3)	**0x506b** (8 bits)	D7–4	**CLKDIV [3:0]**	Clock division ratio select	CLKDIV[3:0] / Division ratio: OSC3 or IOSC / OSC1 0xf / – / – 0xe / 1/16384 / – 0xd / 1/8192 / – 0xc / 1/4096 / – 0xb / 1/2048 / – 0xa / 1/1024 / – 0x9 / 1/512 / – 0x8 / 1/256 / 1/256 0x7 / 1/128 / 1/128 0x6 / 1/64 / 1/64 0x5 / 1/32 / 1/32 0x4 / 1/16 / 1/16 0x3 / 1/8 / 1/8 0x2 / 1/4 / 1/4 0x1 / 1/2 / 1/2 0x0 / 1/1 / 1/1	0x0	R/W	
		D3–2	**CLKSRC [1:0]**	Clock source select	CLKSRC[1:0] / Clock source 0x3 / External clock 0x2 / OSC3 0x1 / OSC1 0x0 / IOSC*	0x0	R/W	* S1C17564 only
		D1	**–**	reserved	–	–	–	
		D0	**CLKEN**	Count clock enable	1 Enable, 0 Disable	0	R/W	
T16A Counter Ch.3 Control Register (T16A_CTL3)	**0x5460** (16 bits)	D15–6	**–**	reserved	–	–	–	0 when being read.
		D5–4	**CCABCNT [1:0]**	Counter select	CCABCNT[1:0] / Counter Ch. 0x3 / Ch.3 0x2 / Ch.2 0x1 / Ch.1 0x0 / Ch.0	0x0	R/W	
		D3	**CBUFEN**	Compare buffer enable	1 Enable, 0 Disable	0	R/W	
		D2	**TRMD**	Count mode select	1 One-shot, 0 Repeat	0	R/W	
		D1	**PRESET**	Counter reset	1 Reset, 0 Ignored	0	W	0 when being read.
		D0	**PRUN**	Counter run/stop control	1 Run, 0 Stop	0	R/W	
T16A Counter Ch.3 Data Register (T16A_TC3)	**0x5462** (16 bits)	D15–0	**T16ATC [15:0]**	Counter data T16ATC15 = MSB T16ATC0 = LSB	0x0 to 0xffff	0x0	R	
T16A Comparator/ Capture Ch.3 Control Register (T16A_CCCTL3)	**0x5464** (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.3 A Data Register (T16A_CCA3)	**0x5466** (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.3 B Data Register (T16A_CCB3)	**0x5468** (16 bits)	D15–0	**CCB[15:0]**	Compare/capture B data CCB15 = MSB CCB0 = LSB	0x0 to 0xffff	0x0	R/W	

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
T16A Comparator/ Capture Ch.3 Interrupt Enable Register (T16A_IEN3)	**0x546a** (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.3 Interrupt Flag Register (T16A_IFLG3)	**0x546c** (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)	D15–2	**–**	reserved	–		–	–	0 when being read.
		D1–0	**RDWAIT [1:0]**	Flash read wait cycle	RDWAIT[1:0]	Wait	0x3	R/W	
					0x3	2 wait			
					0x2	1 wait			
					0x1	No wait			
					0x0	reserved			

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	0x2fc0				0x2fc0	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 (All clocks stopped)

- Execute the `slp` instruction
  Execute the `slp` instruction when the entire system can be stopped. The CPU enters SLEEP mode and the system clocks stop. This also stops all peripheral circuits using clocks. Starting up the CPU from SLEEP mode is therefore limited to startup using a port (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 IOSC and OSC3 oscillators.

CPU clock (CCLK)

- Execute the `halt` instruction
  Execute the `halt` instruction when program execution by the CPU is not required—for example, when only the display is required or for interrupt standby. The CPU 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 timers and interrupts. 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.

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
  • UART Ch.0 and Ch.1
  • Fine mode 16-bit timer Ch.0 and Ch.1
  • 16-bit timer Ch.0 to Ch.2
  • SPI Ch.0 to Ch.2
  • USI Ch.0 and Ch.1 (S1C17564)
  • I^2C master
  • I^2C slave
  • Power generator (S1C17564)
  • P ports and port MUX (control registers, chattering filters)
  • MISC registers
  • IR remote controller
  • A/D converter

PCLK is not required for the peripheral modules/functions shown below.

Peripheral circuits/functions that do not use PCLK
- Clock timer
- Stopwatch timer
- Watchdog timer
- 16-bit PWM timer Ch.0 to Ch.3
- 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	IOSC/OSC3	CPU (CCLK)	PCLK peripheral	OSC1 peripheral	CPU stop method	CPU startup method
↑ Low	Stop	Stop	Stop	Stop	Stop	Execute `slp` instruction	1
	Oscillation (system CLK)	Stop	Stop	Stop	Run	Execute `halt` instruction	1, 2
	Oscillation (system CLK)	Stop	Stop	Run	Run	Execute `halt` instruction	1, 2, 3
	Oscillation (system CLK)	Stop	Run (1/1)	Run	Run		
	Oscillation	Oscillation (system CLK)	Stop	Run	Run	Execute `halt` instruction	1, 2, 3
	Oscillation	Oscillation (system CLK)	Run (low gear)	Run	Run		
High ↓	Oscillation	Oscillation (system CLK)	Run (1/1)	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 OSC1 peripheral circuit
   Started up by a clock timer, stopwatch timer, or watchdog timer interrupt.
3. 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.

Regulator operating mode (S1C17564)

When the embedded regulator is used in the S1C17564, it should be placed into economy mode (power saving mode) to reduce current consumption during low-speed (32 kHz) operation or standby mode (HALT or SLEEP). For controlling economy mode, see the “Power Supply” chapter.

Appendix C Mounting Precautions

This section describes various precautions for circuit board design and IC mounting.

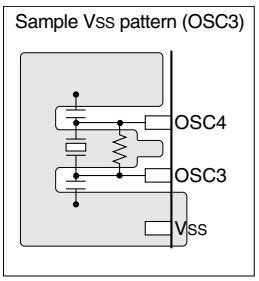
Oscillator circuit

- Oscillation characteristics depend on factors such as components used (resonator, Rf, CG, CD) and circuit board patterns. In particular, with ceramic or crystal resonators, select the appropriate external resistor (Rf) and capacitors (CG, CD) only after fully evaluating components actually mounted on the circuit board.
- Oscillator clock disturbances caused by noise may cause malfunctions. To prevent such disturbances, consider the following points. The latest devices, in particular, are manufactured by microscopic processes, making them especially susceptible to noise.
  Areas in which noise countermeasures are especially important include the OSC2 pin and related circuit components and wiring. OSC1 pin handling is equally important. The noise precautions required for the OSC1 and OSC2 pins are described below. We also recommend applying similar noise countermeasures to 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

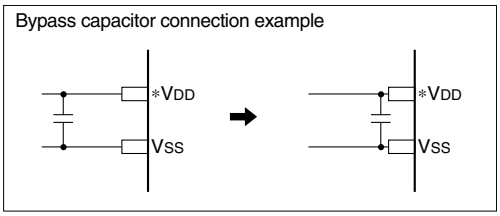
- The reset signal input to the #RESET pin when power is turned on will vary, depending on various factors, such as power supply start-up time, components used, and circuit board patterns. Constants such as capacitance and resistance should be determined through testing with real-world products.
- Components 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.

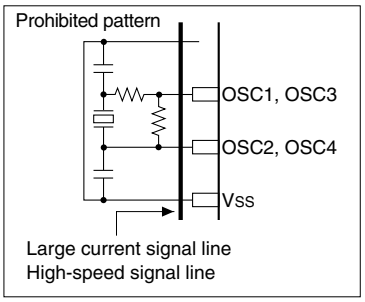
(1) Connections from the power supply to the ∗VDD (LVDD, HVDD, AVDD) and VSS pins should be implemented via the shortest, thickest patterns possible.

(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.



Noise-induced malfunctions

Check the following three points if you suspect the presence of noise-induced IC malfunctions.

(1) DSIO pin

Low-level noise to this pin will cause a switch to debug mode. The switch to debug mode can be confirmed by the clock output from DCLK and a High signal from the DST2 pin.

For the product version, we recommend connecting the DSIO pin directly to HVDD or pulling up the DISO pin using a resistor not exceeding 10 kΩ. The IC includes an internal pull-up resistor. The resistor has a relatively high impedance of 100 kΩ to 500 kΩ and is not noise-resistant.

(2) #RESET pin

Low-level noise to this pin will reset the IC. Depending on the input waveform, the reset may not proceed correctly. This is more likely to occur if, due to circuit design choices, the impedance is high when the reset input is High.

(3) LVDD, HVDD, AVDD, and VSS power supply

The IC will malfunction at the instant when noise falling below the rated voltage is input. Incorporate countermeasures on the circuit board, including close patterns for circuit board power supply circuits, noise-filtering decoupling capacitors, and surge/noise prevention components on the power supply line.

Perform the inspections described above using an oscilloscope capable of observing waveforms of at least 200 MHz. It may not be possible to observe high-speed noise events with a low-speed oscilloscope.

If you detect potential noise-induced malfunctions while observing the waveform with an oscilloscope, recheck with a low-impedance (less than 1 kΩ) resistor connecting the relevant pin to GND or to the power supply. Malfunctions at that pin are likely if changes are visible, such as the malfunction disappearing, becoming less frequent, or the phenomena changing.

The DSIO and #RESET input circuits described above detect input signal edges and are susceptible to malfunctions induced by spike noise. This makes these digital signal pins the most susceptible to noise.
To reduce potential noise, keep the following two points in mind when designing circuit boards:

(A) It is important to lower the signal-driving impedance, as described above. Connect pins to the power supply or GND, with impedance of 1 kΩ or less, preferably 0 Ω. The signal lines connected should be no longer than approximately 5 mm.

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

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 an oscillator circuit is not used, the oscillator pins should be left open. The control registers should be fixed at the initial status (oscillation disabled).

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 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 p1_handler                ; 0x05       0x14    P1 port
.long swt_handler               ; 0x06       0x18    SWT
.long ct_handler                ; 0x07       0x1c    CT
.long t16a_2_handler            ; 0x08       0x20    T16A ch2
.long P4_handler                ; 0x09       0x24    P4 port
.long spi_2_handler             ; 0x0a       0x28    SPI ch2
.long t16a_0_handler            ; 0x0b       0x2c    T16A ch0
.long t16f_0_1_usi_0_1_handler  ; 0x0c       0x30    T16F ch0,ch1/USI ch0,ch1
.long t16_0_handler             ; 0x0d       0x34    T16 ch0
.long t16_1_handler             ; 0x0e       0x38    T16 ch1
.long t16_2_t16a_3_handler      ; 0x0f       0x3c    T16 ch2/T16A ch3
.long uart_0_handler            ; 0x10       0x40    UART ch0
.long uart_1_handler            ; 0x11       0x44    UART ch1
.long spi_0_handler             ; 0x12       0x48    SPI ch0
.long i2cm_handler              ; 0x13       0x4c    I2C master
.long remc_spi_1_handler        ; 0x14       0x50    REMC/SPI ch1
.long t16a_1_handler            ; 0x15       0x54    T16A ch1
.long adc10_handler             ; 0x16       0x58    ADC10
.long p5_handler                ; 0x17       0x5c    P5 port
.long p2_handler                ; 0x18       0x60    P2 port
.long p3_handler                ; 0x19       0x64    P3 port
.long i2cs_handler              ; 0x1a       0x68    I2C slave
.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, 0x0f00                                               ...(4)

        ; ----- Memory controller ----------------
        Xld.a   %r1, 0x54b0     ; FLASHC register address

        ; Flash read wait cycle
        Xld.a   %r0, 0x01       ; No wait  under 14.75 MHz
        ld.b    [%r1], %r0      ; [0x54b0] <= 0x01                        ...(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 when the system clock is 14.75 MHz or lower.
(See the "Memory Map, Bus Control" chapter.)

Appendix E Recommended Resonators

Optimum oscillator component values vary depending on operating conditions such as a printed circuit board and power voltage. Please ask the manufacturer to evaluate the resonator mounted on the circuit board.

(1) OSC1 crystal resonator

Oscillation frequency [kHz]	Manufacturer	Product number
32.768	Epson Toyocom Corporation	MC-146 (SMD)

(2) OSC3 crystal resonator

Oscillation frequency [MHz]	Manufacturer	Product number
4	Epson Toyocom Corporation	MA-406 (SMD)
16	Epson Toyocom Corporation	FA-238 (SMD)
20	Epson Toyocom Corporation	FA-238 (SMD)

(3) OSC3 ceramic resonators

Oscillation frequency [MHz]	Manufacturer	Product number
2	Murata Manufacturing Co., Ltd.	CSTCC2M00G56 (SMD)
8	Murata Manufacturing Co., Ltd.	CSTCE8M00G55 (SMD)
14.75	Murata Manufacturing Co., Ltd.	CSTCE14M7V51 (SMD)
24	Murata Manufacturing Co., Ltd.	CSTCG24M0V51 (SMD)

Revision History

Code No.	Page	Contents
411914400	All	New establishment
411914401	1-12	Pin Descriptions: P02/P03 (US_SSI*x*) pins (Old) USI Ch.*x* slave select signal input or data input/output pin (S1C17564) (New) USI Ch.*x* data input/output pin (S1C17564)
	3-2	Flash programming (Old) The Flash memory supports sector erase method. ... see the “Self-Programming (FLS) Application Notes” for the S1C17554/564. (New) Deleted
	4-1	Power supply: I/O power supply voltage (HV_{DD}) (Old) No description (New) Notes: ... • AV_{DD} is supplied to the P00 to P03 (AIN0 to AIN3) ports as their power source (see Section 4.3).
		Power supply: Analog power supply voltage (AV_{DD}) (Old) AV_{DD} = 2.7 V to 5.5 V (V_{SS} = GND) Note: Be sure to supply the same voltage as HV_{DD} ... when the analog circuit is not used. (New) AV_{DD} = 2.7 V to 5.5 V (V_{SS} = GND) when A/D converter is used ... Notes: ... so that the digital circuit will not affect A/D conversion results.
	7-7, 7-11	CLG: System clock switching conditions (S1C17564) Tables 7.4.3 and 7.8.4 were modified.
	7-10	CLG: FOUTA/FOUTB output Modified Figure 7.7.2
	8-4, 8-10	P: I/O port chattering filter function (Old) No description (New) Notes: • An unexpected interrupt may occur ... disabled before placing the CPU into SLEEP status.
	11-5	T16A: Compare buffers (Old) Although the T16A_CCA*x* and T16A_CCB*x* registers ... compare buffers will be accessed. (New) Although the T16A_CCA*x* ... regardless of the CBUFEN setting, compare registers will be accessed.
	15-6	UART: Data reception control (Old) (2) RDRY = 1, RD2B = 0 ... This clears the data inside the buffer and resets the RDRY flag. ... (3) RDRY = 1, RD2B = 1 ... The receive data buffer ... and resetting the RD2B flag. ... Even when the receive data buffer is full, ... and the new data will overwrite the shift register data. (New) (2) RDRY = 1, RD2B = 0 ... This resets the RDRY flag. ... (3) RDRY = 1, RD2B = 1 ... The receive data buffer outputs the oldest data first. This resets the RD2B flag. ... Even when the receive data buffer is full, ... In this case, the last received data cannot be read.
	15-7	UART: Overrun error (Old) However, if the receive data buffer is not emptied ... and generate an overrun error. (New) However, if the receive data buffer is not emptied ... sent to the buffer and an overrun error will occur.
	16-2	SPI: SPI clock (Old) In slave mode, the SPI clock is input ... differentiated and used to sync with the PCLK clock. (New) In slave mode, the SPI clock is input via the SPICLK*x* pin.
	16-4	SPI: Data transmission timing chart Deleted Figure 16.5.1
		SPI: Data transmission control (Old) No description (New) Note: When the SPI module is used in master mode with CPHA set to 0, ... (Added Figure 16.5.1) ... transmit data bits and the second and following bytes during continuous transfer.
	16-5	SPI: Data transmission/receiving timing chart Modified Figure 16.5.2
		SPI: Disabling data transfers (Old) After a data transfer is completed ... the SPRBF flag is 0 before disabling data transfer. Setting SPEN to 0 empties ... if SPEN is set to 0 while data is being sent or received. (New) After a data transfer is completed ... the SPBSY flag is 0 before disabling data transfer. The data being transferred cannot be ... if SPEN is set to 0 while data is being sent or received.
	16-7	SPI: SPI Ch.*x* Transmit Data Registers (SPI_TXD*x*) (Old) No description (New) Note: Make sure that SPEN is set to 1 before writing data ... to start data transmission/reception.
	17-2	I2CM: I^2C connection example Added Figure 17.2.1
		I2CM: Synchronization clock (upper limit of transfer rate) (Old) No description (New) When the I2CM module is used to ... 50 kbps in standard mode or 200 kbps in fast mode.

REVISION HISTORY

Code No.	Page	Contents
411914401	17-3	I2CM: Slave address transmission (Old) ... In 10-bit mode, data is sent twice under software control. ... (New) ... In 10-bit mode, data is sent twice or three times under software control. ...
		I2CM: Transmit data specifying slave address and transfer direction Modified Figure 17.5.2
	17-4	I2CM: Data reception control (Old) The data is loaded to the shift register in sequence at the clock rising edge, with the MSB leading. RXE is reset to 0 when D6 is loaded. (New) The data is shifted into the shift register from the MSB first in sync with the clock. RXE is reset to 0 when D7 is loaded.
	17-5	I2CM: End of data transfers (Generating stop condition) (Old) Stop condition generation can be reserved. ... and then set STP to 1. (New) Before STP can be set to 1, ... make the wait time longer than 1/4 of the I^2C clock cycle set.
		I2CM: Disabling data transfer (Old) After data transfer has completed ... if I2CMEN is set to 0 during the transfer. (New) After the stop condition has been generated, ... transfer data at that point cannot be guaranteed.
	17-5, 17-6	I2CM: Timing chart Modified Figures 17.5.6 to 17.5.9
	17-7	I2CM: Checking whether a transmit buffer empty interrupt has occurred or not (Old) A transmit buffer empty interrupt has occurred if TBUSY/I2CM_CTL register is read as 0 ... (4) Read TBUSY/I2CM_CTL register. (New) A transmit buffer empty interrupt has occurred if TXE/I2CM_DAT register is read as 0 ... (5) Read TXE/I2CM_DAT register.
		I2CM: Checking whether a receive buffer full interrupt has occurred or not (Old) A receive buffer full interrupt has occurred if RBUSY/I2CM_CTL register is read as 1 ... (3) Read RBUSY/I2CM_CTL register. (New) A receive buffer full interrupt has occurred if RBRDY/I2CM_DAT register is read as 1 ... (3) Read RBRDY/I2CM_DAT register.
	17-9	I2CM: I^2C Master Control Register (I2CM_CTL) - (D1) STP: Stop Control Bit (Old) STP is disabled if any of TXE, RXE, or STRT is 1. (New) Deleted
		I2CM: I^2C Master Data Register (I2CM_DAT) - (D10) RXE: Receive Execution Bit (Old) RXE is reset to 0 as soon as D6 is loaded to the shift register. (New) RXE is reset to 0 as soon as D7 is loaded to the shift register.
	18-1	I2CS: I2CS Module overview Modified Figure 18.1.1
		I2CS: List of I2CS Pins - SCL1 Modified Table 18.2.1
	18-2	I2CS: I^2C connection example Added Figure 18.2.1
		I2CS: Bus free request with an input from the #BFR pin (Old) When this function is enabled, a Low pulse (five peripheral module clock (PCLK) cycles or more pulse width is required) input to the #BFR pin sets BFREQ/I2CS_STAT register to 1. (New) When this function is enabled, a low pulse (One peripheral module clock (PCLK) cycle or more pulse width is required. Two PCLK clock cycles or more pulse width is recommended.) input to the #BFR pin sets BFREQ/I2CS_STAT register to 1.
	18-3	I2CS: Clock stretch function (Old) No description (New) Note that the data setup time ... depending on the I2CS module operating clock (PCLK) frequency.
	18-4	I2CS: Starting data transfer (Old) Both BUSY and SELECTED are maintained at 1 until a stop condition is detected. (New) BUSY is maintained at 1 until a stop condition is detected. SELECTED is maintained at 1 until a stop condition or repeated start condition is detected.
	18-4, 18-5, 18-10	I2CS: Data transmission (Old) No description (New) When the asynchronous address detection function is used, ... after TXEMP has been set to 1.
	18-6	I2CS: Note on data transmission (Old) Note: If the I2CS module has sent back a NAK as the response to the address sent by the master ... 1. More than one slave device is connected to the I^2C bus. ... 4. The I2CS module is placed into transfer standby state ... used as the operating clock (PCLK). (New) Note: If the I2CS module has sent back a NAK as the response to the address sent by the master ... 1. The transfer rate is set to 320 kbps or higher. ... 3. The I2CS module is placed into transfer standby state ... used as the operating clock (PCLK).
	18-7, 18-8	I2CS: Timing charts Modified Figures 18.5.5 to 18.5.8
	18-9	I2CS: Bus status interrupt (Old) 7. DA_STOP/I2CS_STAT ... if a stop condition is detected ... as the slave device. (New) 7. DA_STOP/I2CS_STAT register: This bit is set to 1 if a stop condition or a repeated start condition is detected while this module is selected as the slave device.

Code No.	Page	Contents
411914401	18-15	I2CS: I²C Slave Status Register (I2CS_STAT) - (D0) DA_STOP: Stop Condition Detect Bit (Old) Indicates that a stop condition is detected. ... I²C communication process to enter standby state that is ready to detect the next start condition. (New) Indicates that a stop condition or a repeated start condition is detected. ... I2CS module sets DA_STOP to 1. At the same time, it initializes the I²C communication process.
		I2CS: I²C Slave Access Status Register (I2CS_ASTAT) - (D1) SELECTED: I²C Slave Select Status Bit (Old) After SELECTED is set to 1, it is reset to 0 when a stop condition is detected. (New) After SELECTED is set to 1, it is reset to 0 ... or a repeated start condition is detected.
	19-1	USI: USI module overview - SPI master mode (Old) - Receive data mask function is available. (New) Deleted
	19-3	USI: Note on clock source (Old) Note: When the USI is set to I²C slave mode, i2c_scl (I²C clock) is supplied from ... Figure 19.3.2 Example of Delayed I2C Clock (New) Note: When the USI module is set to I²C slave mode, ... after one clock cycle from writing 1 to ISTG.
		USI: USI module settings (Old) (1) Configure the pins to be used for USI according to the interface mode. (See Section 19.2.) ... (6) Set interrupt conditions if necessary. (See Section 19.7.) (New) SPI master and I²C master/slave modes ... The USI pins pulled down to low in the initial status. When using USI in UART mode, ... the pin functions to prevent undesired start bit to be generated.
	19-5	USI: Settings for SPI master mode (Old) When the USI is used in SPI master mode, ... Also enable/disable the receive data mask function. (New) When the USI is used in SPI master mode, ... polarity/phase, clock mode, and data length.
	19-6	USI: Receive data mask function (Old) Receive data mask function The USI in SPI master mode provides a receive data mask (data retransmission) function. ... ... For normal data transfer, set SMSKEN to 0 (default) to disable the receive data mask function. (New) Deleted
	19-7	USI: Data receiving timing chart (UART mode) Modified Figure 19.5.1.2
		USI: Data transmission (Old) The SPI controller includes two status flags for transfer control: ... and SSIF/USI_SIF*x* register. (New) The SPI controller includes STDIF/USI_SIF*x* register for transfer control.
		USI: Data transmission (Old) In SPI master mode, ... Read this flag to check whether the SPI controller is operating or at standby. (New) Deleted
	19-8	USI: Data transmission timing chart (SPI master mode) Modified Figure 19.5.2.1
		USI: Data reception (Old) The SPI controller includes two status flags for transfer control: ... and SSIF/USI_SIF*x* register. (New) The SPI controller includes SRDIF/USI_SIF*x* register for transfer control.
		USI: Data reception (Old) The SSIF flag indicates the shift register status. ... the SPI controller is operating or at standby. (New) Deleted
		USI: Data receiving timing chart (SPI master mode) Modified Figure 19.5.2.2
	19-9	USI: I²C mode connection example Added Figure 19.5.3.1
	19-10	USI: I²C master data transmission timing chart Modified Figure 19.5.3.3
	19-11	USI: Transmit data specifying slave address and transfer direction Modified Figure 19.5.3.5
		USI: Transmit data specifying slave address and transfer direction (Old) ... In 10-bit mode, data is sent twice under software control. ... To send a 10-bit address, execute this procedure twice as shown in Figure 19.5.3.5. ... (New) ... In 10-bit mode, data is sent twice or three times under software control. ... To send a 10-bit address, execute this procedure twice or three times as shown in Figure 19.5.3.5. ...
	19-13	USI: I²C master data receiving timing chart Modified Figure 19.5.3.10
	19-14	USI: Data reception in I²C master mode (Old) (3) Data reception ... It is necessary to send back an ACK or NAK ... after an 8-bit data has been received. ... (New) ... It is necessary to send back an ACK or NAK ... this operation after the received data is read). ...
	19-15	USI: Control method in I²C slave mode (Old) ... After an interrupt occurs, ... (ISSTA[2:0]/USI_ISIF*x* register) to check the operation finished. (New) ... After an interrupt occurs, ... This also automatically reset ISSTA[2:0] to 0x0.

REVISION HISTORY

Code No.	Page	Contents
411914401	19-16	USI: I²C slave data transmission timing chart Modified Figure 19.5.3.13
	19-17	USI: Data transmission in I²C slave mode (Old) (4) When a stop condition is received If the ISSTA[2:0] value read during data transmission is 0x1, ... abort data transmission. (New) ... abort data transmission. The stop condition can be received only when ISTGMOD[2:0] is set to an operation mode ... ... 0x6 (ACK/NAK reception)
	19-18	USI: I²C slave data receiving timing chart Modified Figure 19.5.3.15
	19-19	USI: Data reception in I²C slave mode (Old) (3) Data reception ... It is necessary to send back an ACK or NAK ... after an 8-bit data has been received. ... (New) ... It is necessary to send back an ACK or NAK ... this operation after the received data is read). ...
		USI: Data reception in I²C slave mode (Old) (4) When a stop condition is received If the ISSTA[2:0] value read during data reception is 0x1, ... abort data reception. (New) ... abort data reception. The stop condition can be received only when ISTGMOD[2:0] is set to an operation mode ... ... 0x6 (ACK/NAK reception)
	19-21	USI: Interrupts in UART mode - Receive error interrupt (Old) ... If any of the error flags has the value 1, ... proceed with error recovery. (New) ... If any of the error flags has the value 1, ... proceed with error recovery. To reset an overrun error, ... USIMOD[2:0]/USI_GCFG*x* register) to initialize USI.
		USI: Interrupts in SPI master mode - Receive error interrupt (Old) ... If SEIF is 1, the interrupt handler routine will proceed with error recovery. (New) ... If SEIF is 1, the interrupt handler routine will proceed with error recovery. To reset an overrun error, clear SEIF ... then read the receive data buffer (USI_RD*x* register) twice.
	19-22, 19-23	USI: Interrupts in I²C master/slave mode - Receive error interrupt (Old) To use this interrupt, ... interrupt requests for this cause will not be sent to the ITC. ... ... If IMEIF is 1, the interrupt handler routine will proceed with error recovery. (New) To use this interrupt, ... interrupt requests for this cause will not be sent to the ITC. An overrun error occurs ... two-byte data has been received without reading the receive data buffer. The USI module sets ... the interrupt handler routine will proceed with error recovery. To reset an overrun error, ... read the receive data buffer (USI_RD*x* register) twice.
	19-23, 19-24, 19-31, AP-A-2, AP-A-3, AP-A-16, AP-A-18	USI: 0x50c9/0x50e9 USI Ch.*x* SPI Master Mode Receive Data Mask Registers (USI_SMSK*x*) (New) Deleted
	19-27	USI: USI Ch.*x* UART Mode Interrupt Flag Registers (USI_UIF*x*) - (D2) UOEIF: Overrun Error Flag Bit (Old) UOEIF is reset by writing 1. (New) To reset UOEIF, ... (write 0x0 to USIMOD[2:0]/USI_GCFG*x* register) to initialize USI.
	19-28, 19-29, AP-A-16, AP-A-17	USI: USI Ch.*x* SPI Master Mode Configuration Registers (USI_SCFG*x*) (Old) D1 SMSKEN: Receive Data Mask Enable Bit (New) D1 Reserved
	19-30, AP-A-16, AP-A-17	USI: USI Ch.*x* SPI Master Mode Interrupt Flag Registers (USI_SIF*x*) (Old) D3 SSIF: Transfer Busy Flag Bit (New) D3 Reserved
	19-30	USI: USI Ch.*x* SPI Master Mode Interrupt Flag Registers (USI_SIF*x*) - (D2) SEIF: Overrun Error Flag Bit (Old) ... SEIF is reset by writing 1. (New) ... When data reception for one byte has completed, ... USI_RD*x* register twice can be reversed.
	19-32	USI: USI Ch.*x* I²C Master Mode Interrupt Flag Registers (USI_IMIF*x*) - (D[4:2]) IMSTA[2:0]: I²C Master Status Bits (Old) When an operation completion interrupt occurs, ... the operation that has been finished. (New)... the operation that has been finished. IMSTA[2:0] is automatically reset to 0x0 by writing 1 to IMIF.
		USI: USI Ch.*x* I²C Master Mode Interrupt Flag Registers (USI_IMIF*x*) - (D1) IMEIF: Overrun Error Flag Bit (Old) An overrun error occurs when the previous received data ... IMEIF is reset by writing 1. (New) An overrun error occurs ... and then read the receive data buffer (USI_RD*x* register) twice.
	19-35	USI: USI Ch.*x* I²C Slave Mode Interrupt Flag Registers (USI_ISIF*x*) - (D[4:2]) ISSTA[2:0]: I²C Slave Status Bits (Old) When an operation completion interrupt occurs, ... the operation that has been finished. (New)... the operation that has been finished. ISSTA[2:0] is automatically reset to 0x0 by writing 1 to ISIF.
		USI: USI Ch.*x* I²C Slave Mode Interrupt Flag Registers (USI_ISIF*x*) - (D1) ISEIF: Overrun Error Flag Bit (Old) An overrun error occurs when the previous received data ... ISEIF is reset by writing 1. (New) An overrun error occurs ... and then read the receive data buffer (USI_RD*x* register) twice.

Code No.	Page	Contents
411914401	19-35	USI: Busy flags and delay conditions Mofified Table 19.9.1
	21-1	ADC: ADC10 module overview (Old) Sampling rate: Max. 100 ksps (New) Sampling rate: f_{ADCLK}/13 to f_{ADCLK}/20 [sps] (f_{ADCLK}: A/D conversion clock frequency)
	21-2	ADC: A/D converter settings (Old) (1) Set the analog input pins. See Section 21.2. ... (8) When using A/D converter interrupts, set interrupt conditions. See Section 21.5. (New) (1) Set the analog input pins. See Section 21.2. (2) Adjust the comparator according to the A/D conversion voltage range used. ... (9) When using A/D converter interrupts, set interrupt conditions. See Section 21.5.
	21-5	ADC: Expression for calculating sampling rate (Old) No description (New) The following shows the relation between sampling time and sampling rate. Sampling rate [sps] = ...
	21-12	ADC: A/D Control/Status Register (ADC10_CTL) - (D1) ADCTL: A/D Conversion Control Bit (Old) When ADEN is 0 (A/D conversion disabled), ADCTL is fixed to 0, with no trigger accepted. (New) When ADEN is 0, no trigger will be accepted.
	24-1	Absolute maximum ratings: Storage temperature (Tstg) (Old) -65 to 150°C (New) -65 to 125°C
		Recommended operating conditions: Analog power supply voltage (AV_{DD}) (Old) Min. 2.7V Max. 5.5V (New) When ADC is used: Min. 2.7V Max. 5.5V When ADC is not used: Min. 1.65V Max. 5.5V
		Recommended operating conditions: Operating temperature (Ta) (Old) Min. -40°C Max. 85°C (New) During normal operation (Flash read only): Min. -40°C Max. 85°C During Flash programming: Min. 10°C Max. 40°C
	24-6	Input/output pin characteristics: Conditions unless otherwise specified (Old) HV_{DD} = xxV (New) HV_{DD} = AV_{DD} = xxV
	24-10	Flash memory characteristics (Old) Ta = -40 to 85°C, Erase/program count *1, *1 The erase/program ... 10 years. (New) Ta = 10 to 40°C, Program count *1, *1 The program count assumes ... 10 years.
	26-2	Package: WCSP-48 package Modified table (dimensions)
411914402	1-1	Features: Embedded Flash memory - Erase/program count, Flash programming/erasing voltage Modified Table 1.1.1
	1-12	Pin descriptions: V_{PP} Modified Table 1.3.3.1
	4-1	Power supply: Flash programming power supply voltage (V_{PP}) (Old) Supply a voltage shown below to the V_{PP} pin ... to program the Flash memory. V_{PP} = 7 V (V_{SS} = GND) (New) Supply a voltage shown below to the V_{PP} pin ... to program/erase the Flash memory. V_{PP} = 7 V (V_{SS} = GND) for programming V_{PP} = 7.5 V (V_{SS} = GND) for erasing
	16-5	SPI: Transmit buffer empty interrupt (Old) If SPTBE is 0, the next transmit data can be written ... by the interrupt handler routine. (New) If SPTBE is 1, the next transmit data can be written ... by the interrupt handler routine.
	17-5	I2CM: End of data transfers (Generating stop condition) (Old) The stop condition is generated as soon as data transfer (including ACK transfer) ends. (New) When generating a stop condition ... slave device to finish clock stretching has elapsed.
	17-6	I2CM: Timing chart Modified Figure 17.5.7
	24-1	Recommended operating conditions: Operating temperature (Old) During Flash programming (New) During Flash programming and erasing
	24-10	Flash memory characteristics (Old) Unless otherwise specified: LV_{DD} = 1.65 to 1.95V, V_{PP} = 7.0V, ... *1 The program count assumes that one writing is one count including factory programming. The programmed data is guaranteed to be retained for 10 years. (New) Unless otherwise specified: LV_{DD} = 1.65 to 1.95V, V_{PP} = 7.0V (for programming)/7.5V (for erasing), ... (Modified the table) *1 Assumed that Erasing + Programming as count of 1. The count includes programming in the factory. *2 Applied only when FLS V1.0 or later is used.

Code No.	Page	Contents
411914403	9-7	9.7 T16 Output Signals (Old) A/D converter TR = (ct_clk × adi) / 2 - 1 (New) A/D converter TR = (ct_clk × adi) - 1
	24-5	24.6 External Clock Input Characteristics (Old) OSC1 input High pulse width tOSC1H = 14 µs (Min.) OSC1 input Low pulse width tOSC1H = 14 µs (Min.) (New) OSC1 input High pulse width tOSC1H = 9 µs (Min.) OSC1 input Low pulse width tOSC1H = 9 µs (Min.)

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: 411914403
First Issue April 2010 Ⓛ
Revised October 2011 in JAPAN