EPSON
EXCEED YOUR VISION

CMOS 16-BIT SINGLE CHIP MICROCONTROLLER

S1C17624/604/622/602/621
Technical Manual

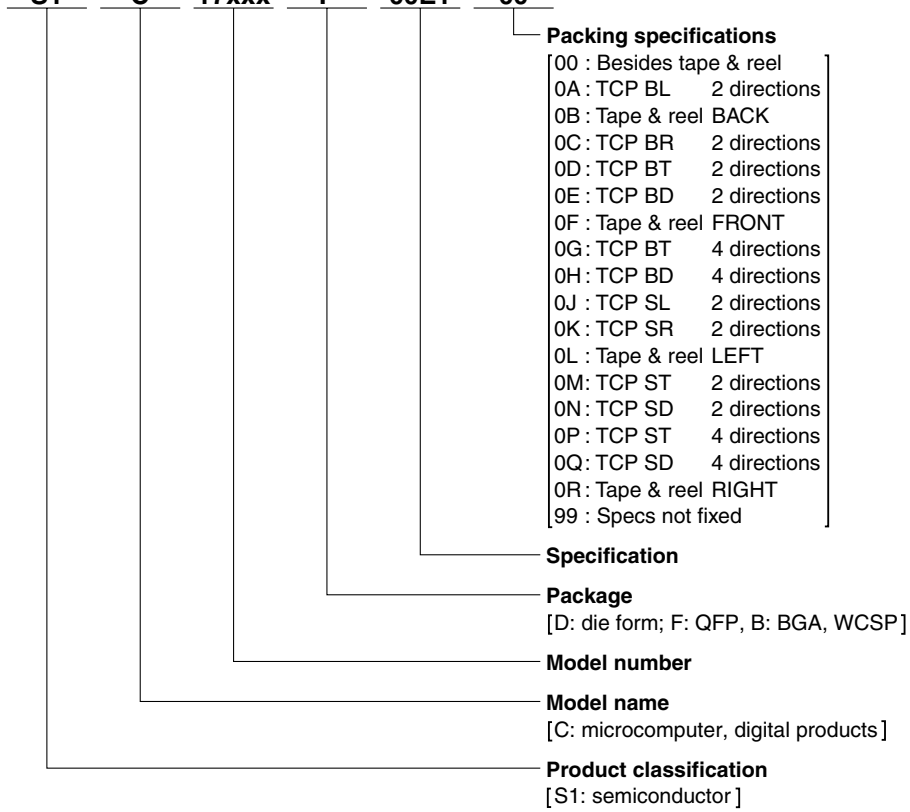
SEIKO EPSON CORPORATION

Rev. 1.3

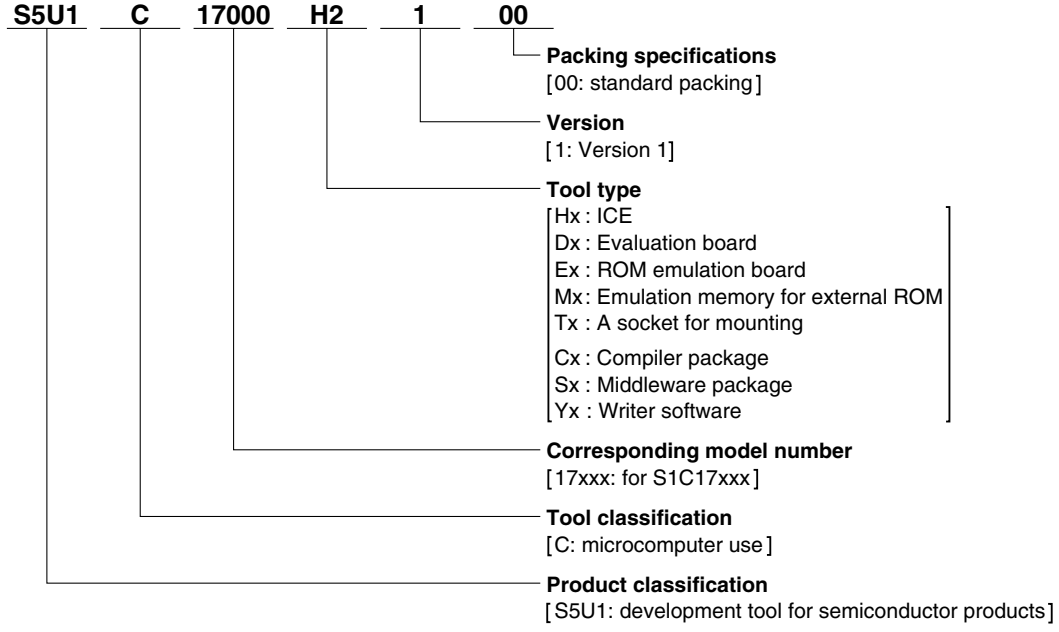
NOTICE

Configuration of product number

Devices



Development tools



– Contents –

1 Overview

The S1C17624/604/622/602/621 is a 16-bit MCU featuring high-speed low-power operations, compact dimensions, wide address space, and on-chip ICE. Based on an S1C17 CPU core, this product consists of a Flash memory, RAM, serial interface modules supporting sensors such as UART to support high-bit rate and IrDA1.0, SPI, and I²C, various timers, maximum 47 general input/output ports, maximum 52 segment × 8 common LCD driver and a power supply voltage booster circuit, A/D converter, R/F converter, supply voltage detector, and 32 kHz and maximum 8.2 MHz oscillator circuits.
It allows 8.2 MHz high-speed operation at a minimum of 1.8 V operating voltage, and executes a basic instruction in one clock cycle with 16-bit RISC processing. The S1C17624/604/622/602/621 also includes a coprocessor supporting multiplication, division, and MAC (multiply and accumulation) operations.
The on-chip ICE function allows onboard Flash programming/erasing, program debugging, and evaluations using the ICDmini (S5U1C17001H) that can be connected with three signal wires.
The S1C17624/604/622/602/621 is ideal for applications, such as health care products with sensors, sports watches, and meter modules that must be driven with battery power and require sensor interfaces and a high-definition LCD display.

1.1 Features

The main features of the S1C17624/604/622/602/621 are listed below.

Table 1.1.1 Features

Model	S1C17624	S1C17604	S1C17622	S1C17602	S1C17621
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		64K bytes		32K bytes
	(Can be used for both instructions and data.)				
Erase/program count	1,000 cycles (min.)				
Other	• Read/program protection function • Allows on-board programming using a debugging tool such as ICDmini (S5U1C17001H) and self-programming by software control.				
Embedded RAM					
Capacity	8K bytes		4K bytes		2K bytes
Embedded Display RAM					
Capacity	56 bytes	40 bytes	56 bytes	40 bytes	
Clock generator					
System clock source	3 sources (IOSC/OSC3/OSC1)				
IOSC oscillator circuit	2.7 MHz(typ.) internal oscillator circuit (oscillation start time 5 µs min.)				
OSC3 oscillator circuit	8.2 MHz (max.) crystal or ceramic oscillator circuit Supports an external clock input.				
OSC1 oscillator circuit	32.768 kHz (typ.) crystal oscillator circuit				
Other	• Core clock frequency control • Peripheral module clock supply control				
Real-time clock					
RTC module	Included (Contains second, minute, hour, day, days of week, month, and year counters.)				
I/O ports					
Number of general-purpose I/O ports	Max. 47 bits	Max. 36 bits	Max. 47 bits	Max. 36 bits	
	(Pins are shared with the peripheral I/O.)				
Serial interfaces					
SPI	1 channel				
I²C master (I2CM)	1 channel				
I²C slave (I2CS)	1 channel				
UART	2 channels (IrDA1.0 supported)				
IR remote controller (REMC)	1 channel				

Model	S1C17624	S1C17604	S1C17622	S1C17602	S1C17621
LCD driver					
LCD outputs	• 56SEG × 4COM • 52SEG × 8COM	• 40SEG × 4COM • 36SEG × 8COM	• 56SEG × 4COM • 52SEG × 8COM	• 40SEG × 4COM • 36SEG × 8COM	
Other	1/3 bias (built-in power supply voltage booster circuit)				
Timers					
8-bit timer (T8F)	2 channels (with fine mode)				
16-bit timer (T16)	3 channels				
16-bit PWM timer (T16E)	1 channel				
16-bit PWM timer (T16A2)	2 channels				
8-bit OSC1 timer (T8OSC1)	1 channel				
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	8 channels (max.)				
Resolution	10 bits				
R/F converter					
Conversion method	CR oscillation type with 24-bit counter				
Number of conversion channels	2 channels (2 sensors can be connected to each channel.)				
Sensor supported	DC-bias resistive/capacitive sensors and AC-bias resistive sensors				
Other	Supports external input for counting pulses.				
Supply voltage detector (SVD)					
Detection levels	15 programmable detection levels (1.8 V to 3.2 V)				
Interrupts					
Reset interrupt	#RESET pin				
NMI	Watchdog timer				
Programmable interrupts	20 systems (8 levels)		19 systems (8 levels)		
Power supply voltage					
Operating voltage (V_{DD})	• 1.8 V to 3.6 V (for normal operation) • 2.7 V to 3.6 V (for Flash erasing/programming) • Built-in voltage regulator (two operating voltages switchable)				
Analog voltage (AV_{DD})	$AV_{DD} = V_{DD}$				
Operating temperature					
Operating temperature range	-25°C to 70°C				
Current consumption (Typ. value)					
SLEEP state (I_{SLP})	0.75 µA	0.75 µA	0.75 µA	0.75 µA	0.75 µA
	OSC1 = OFF, IOSC = OFF, OSC3 = OFF				
HALT state (I_{HALT1})	2.3 µA	2.3 µA	2.3 µA	2.5 µA	2.5 µA
	OSC1 = 32 kHz, IOSC = OFF, OSC3 = OFF, PCKEN[1:0] = 0x0, LCD OFF				
HALT state ($I_{HALT1} + I_{LCD2}$)	4.0 µA	4.0 µA	4.0 µA	3.5 µA	3.5 µA
	OSC1 = 32 kHz, IOSC = OFF, OSC3 = OFF, PCKEN[1:0] = 0x0, LCD ON (checker pattern displayed, highest contrast, V_{C2} reference voltage)				
Run state (I_{EXE1})	14 µA	14 µA	14 µA	15 µA	15 µA
	CPU = OSC1, OSC1 = 32 kHz, IOSC = OFF, OSC3 = OFF, LCD OFF				
Run state (I_{EXE2})	400 µA	400 µA	400 µA	410 µA	410 µA
	CPU = OSC3, OSC1 = 32 kHz, IOSC = OFF, OSC3 = 1 MHz ceramic oscillation				
Shipping form					
1	TQFP15-128pin	TQFP14-100pin	TQFP15-128pin	TQFP14-100pin	
2	Die form	Die form	Die form	Die form	
3				VFBGA7H-144	
Size/pitch	TQFP15-128pin (body size: 14 mm × 14 mm, lead pitch: 0.4 mm) TQFP14-100pin (body size: 12 mm × 12 mm, lead pitch: 0.4 mm) VFBGA7H-144 (body size: 7 mm × 7 mm, ball pitch: 0.5 mm) Die form (pad pitch: 100 µm)				

1.2 Block Diagram

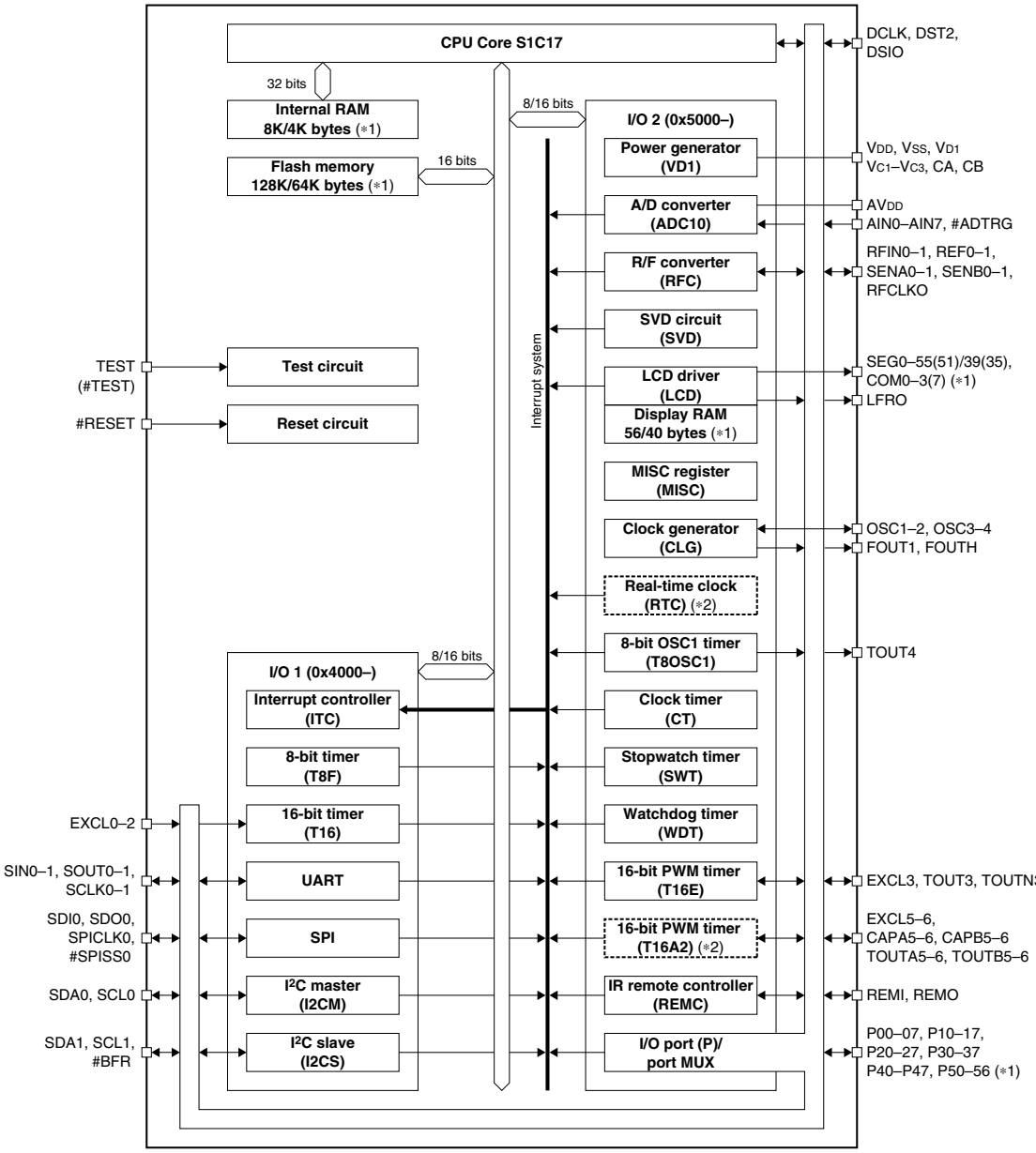


Figure 1.2.1 Block Diagram

*1: The models have a different memory size, LCD outputs and I/O port configurations.
*2: The real-time clock (RTC) and 16-bit PWM timer (T16A2) are available only in the S1C17624 and S1C17604.

Memory/function	S1C17624	S1C17604	S1C17622	S1C17602	S1C17621
Flash memory	128K bytes		64K bytes		32K bytes
Internal RAM	8K bytes		4K bytes		2K bytes
Display RAM	56 bytes	40 bytes	56 bytes	40 bytes	
SEG/COM output pins (static, 1/2–1/4 duty)	SEG0–SEG55 COM0–COM3	SEG0–SEG39 COM0–COM3	SEG0–SEG55 COM0–COM3	SEG0–SEG39 COM0–COM3	
SEG/COM output pins (1/8 duty)	SEG0–SEG51 COM0–COM7	SEG0–SEG35 COM0–COM7	SEG0–SEG51 COM0–COM7	SEG0–SEG35 COM0–COM7	
I/O port pins	47 (P00–P56)	36 (P00–P43)	47 (P00–P56)	36 (P00–P43)	
Real-time clock (RTC)	Available		Unavailable		
16-bit PWM timer (T16A2)	Available		Unavailable		

1.3 Pins/Pads

1.3.1 S1C17624 Pin Configuration Diagram

TQFP15-128pin (S1C17624)

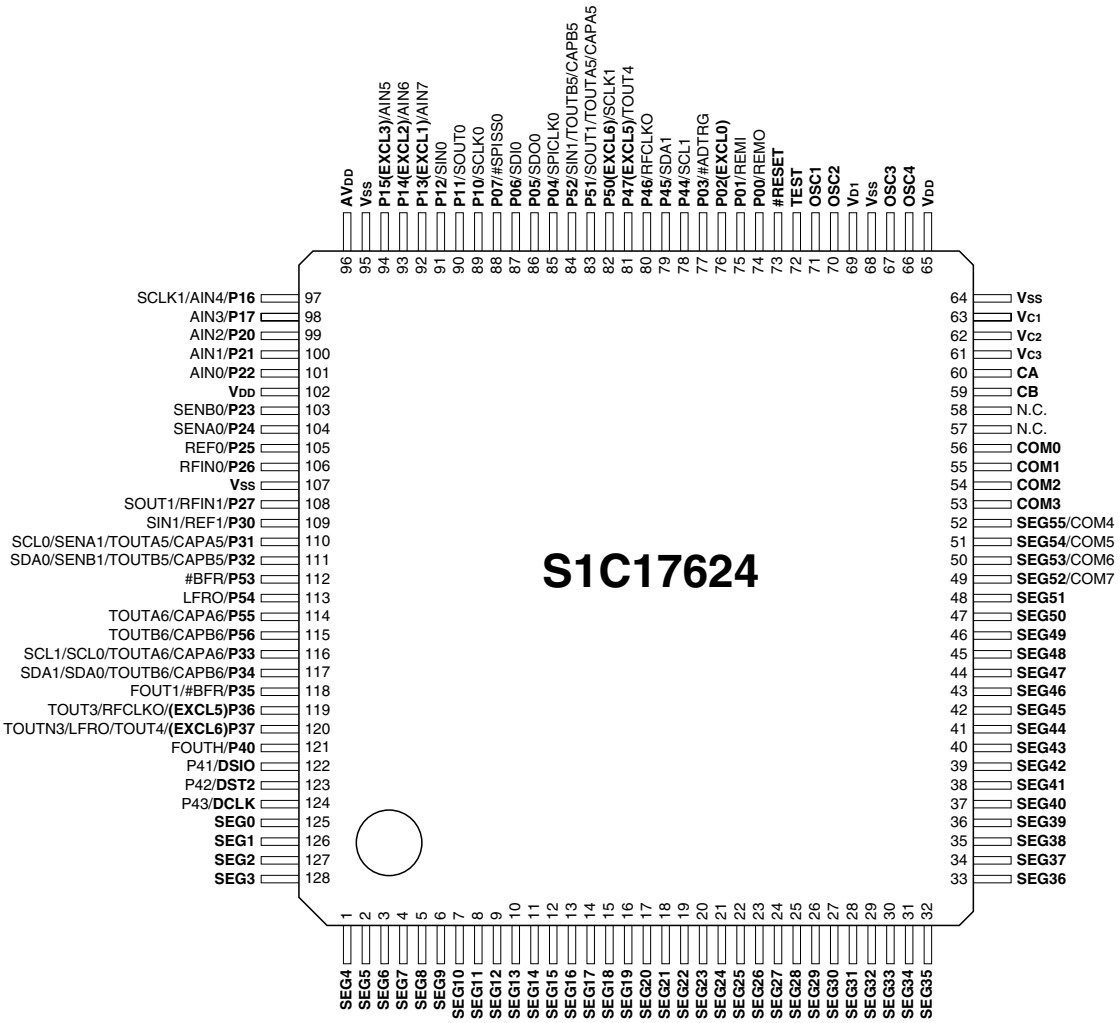


Figure 1.3.1.1 S1C17624 Pin Configuration Diagram (TQFP15-128pin)

Chip (S1C17624)

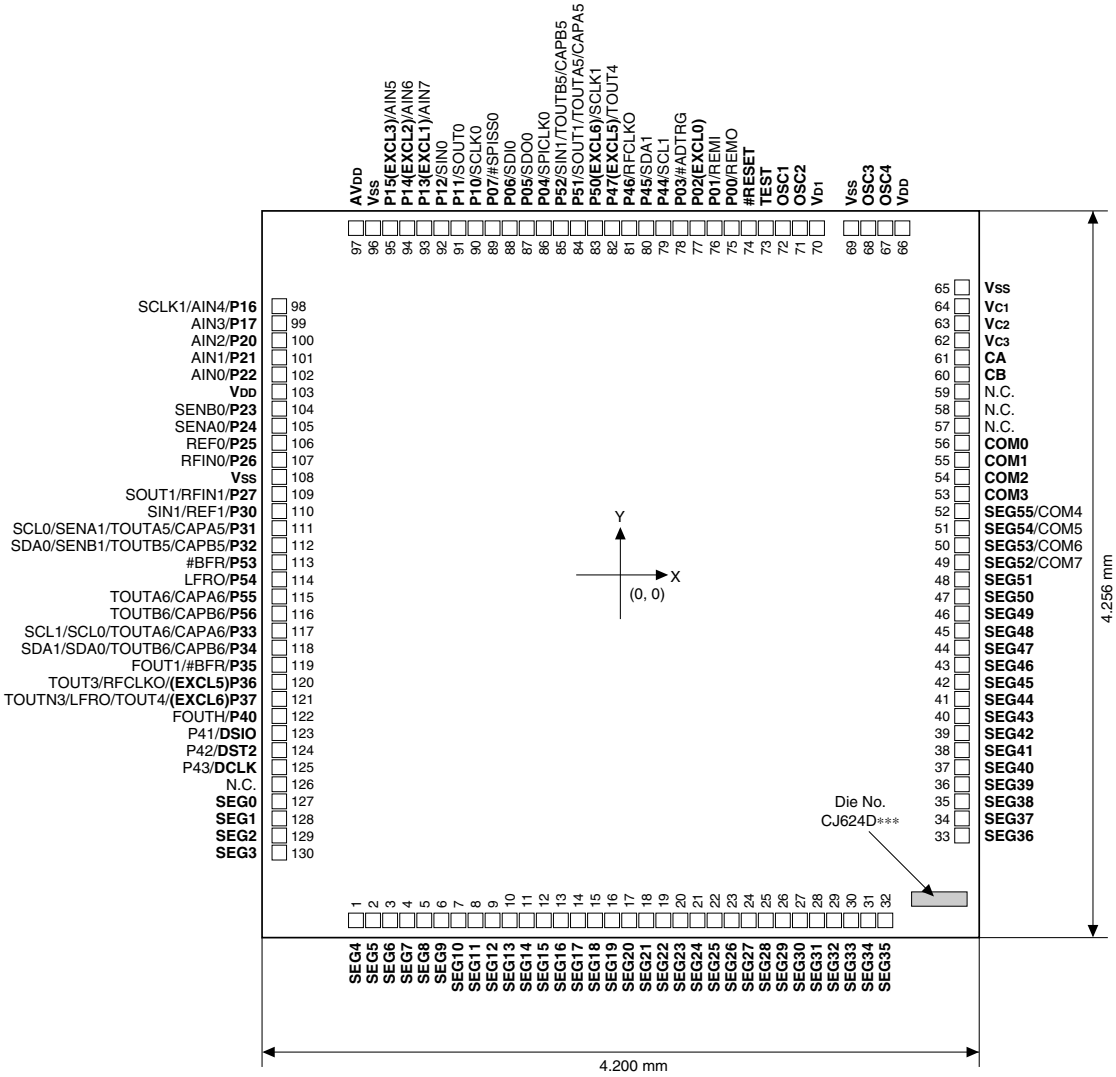


Figure 1.3.1.2 S1C17624 Pad Configuration Diagram

Chip size	X = 4.200 mm, Y = 4.256 mm
Pad opening	No. 1 to 32, 66 to 97: X = 87 µm, Y = 85 µm
	No. 33 to 65, 98 to 130: X = 85 µm, Y = 87 µm
Chip thickness	400 µm

Table 1.3.1.1 S1C17624 Pad Coordinates

No.	Name	X (mm)	Y (mm)	No.	Name	X (mm)	Y (mm)
1	**SEG4**	-1.550	-2.027	66	**VDD**	1.650	2.027
2	**SEG5**	-1.450	-2.027	67	**OSC4**	1.550	2.027
3	**SEG6**	-1.350	-2.027	68	**OSC3**	1.450	2.027
4	**SEG7**	-1.250	-2.027	69	**VSS**	1.350	2.027
5	**SEG8**	-1.150	-2.027	70	**VD1**	1.150	2.027
6	**SEG9**	-1.050	-2.027	71	**OSC2**	1.050	2.027
7	**SEG10**	-0.950	-2.027	72	**OSC1**	0.950	2.027
8	**SEG11**	-0.850	-2.027	73	**TEST**	0.850	2.027
9	**SEG12**	-0.750	-2.027	74	**#RESET**	0.750	2.027
10	**SEG13**	-0.650	-2.027	75	**P00**/REMO	0.650	2.027
11	**SEG14**	-0.550	-2.027	76	**P01**/REMI	0.550	2.027
12	**SEG15**	-0.450	-2.027	77	**P02(EXCL0)**	0.450	2.027
13	**SEG16**	-0.350	-2.027	78	**P03**/#ADTRG	0.350	2.027
14	**SEG17**	-0.250	-2.027	79	**P44**/SCL1	0.250	2.027
15	**SEG18**	-0.150	-2.027	80	**P45**/SDA1	0.150	2.027
16	**SEG19**	-0.050	-2.027	81	**P46**/RFCLKO	0.050	2.027
17	**SEG20**	0.050	-2.027	82	**P47(EXCL5)**/TOUT4	-0.050	2.027
18	**SEG21**	0.150	-2.027	83	**P50(EXCL6)**/SCLK1	-0.150	2.027
19	**SEG22**	0.250	-2.027	84	**P51**/SOUT1/TOUTA5/CAPA5	-0.250	2.027
20	**SEG23**	0.350	-2.027	85	**P52**/SIN1/TOUTB5/CAPB5	-0.350	2.027
21	**SEG24**	0.450	-2.027	86	**P04**/SPICLK0	-0.450	2.027
22	**SEG25**	0.550	-2.027	87	**P05**/SDO0	-0.550	2.027
23	**SEG26**	0.650	-2.027	88	**P06**/SDI0	-0.650	2.027
24	**SEG27**	0.750	-2.027	89	**P07**/#SPISS0	-0.750	2.027
25	**SEG28**	0.850	-2.027	90	**P10**/SCLK0	-0.850	2.027
26	**SEG29**	0.950	-2.027	91	**P11**/SOUT0	-0.950	2.027
27	**SEG30**	1.050	-2.027	92	**P12**/SIN0	-1.050	2.027
28	**SEG31**	1.150	-2.027	93	**P13(EXCL1)**/AIN7	-1.150	2.027
29	**SEG32**	1.250	-2.027	94	**P14(EXCL2)**/AIN6	-1.250	2.027
30	**SEG33**	1.350	-2.027	95	**P15(EXCL3)**/AIN5	-1.350	2.027
31	**SEG34**	1.450	-2.027	96	**VSS**	-1.450	2.027
32	**SEG35**	1.550	-2.027	97	**AVDD**	-1.550	2.027
33	**SEG36**	1.999	-1.530	98	**P16**/SCLK1/AIN4	-1.999	1.570
34	**SEG37**	1.999	-1.430	99	**P17**/AIN3	-1.999	1.470
35	**SEG38**	1.999	-1.330	100	**P20**/AIN2	-1.999	1.370
36	**SEG39**	1.999	-1.230	101	**P21**/AIN1	-1.999	1.270
37	**SEG40**	1.999	-1.130	102	**P22**/AIN0	-1.999	1.170
38	**SEG41**	1.999	-1.030	103	**VDD**	-1.999	1.070
39	**SEG42**	1.999	-0.930	104	**P23**/SENB0	-1.999	0.970
40	**SEG43**	1.999	-0.830	105	**P24**/SENA0	-1.999	0.870
41	**SEG44**	1.999	-0.730	106	**P25**/REF0	-1.999	0.770
42	**SEG45**	1.999	-0.630	107	**P26**/RFIN0	-1.999	0.670
43	**SEG46**	1.999	-0.530	108	**VSS**	-1.999	0.570
44	**SEG47**	1.999	-0.430	109	**P27**/SOUT1/RFIN1	-1.999	0.470
45	**SEG48**	1.999	-0.330	110	**P30**/SIN1/REF1	-1.999	0.370
46	**SEG49**	1.999	-0.230	111	**P31**/SCL0/SENA1/TOUTA5/CAPA5	-1.999	0.270
47	**SEG50**	1.999	-0.130	112	**P32**/SDA0/SENB1/TOUTB5/CAPB5	-1.999	0.170
48	**SEG51**	1.999	-0.030	113	**P53**/#BFR	-1.999	0.070
49	**SEG52**/COM7	1.999	0.070	114	**P54**/LFRO	-1.999	-0.030
50	**SEG53**/COM6	1.999	0.170	115	**P55**/TOUTA6/CAPA6	-1.999	-0.130
51	**SEG54**/COM5	1.999	0.270	116	**P56**/TOUTB6/CAPB6	-1.999	-0.230
52	**SEG55**/COM4	1.999	0.370	117	**P33**/SCL1/SCL0/TOUTA6/CAPA6	-1.999	-0.330
53	**COM3**	1.999	0.470	118	**P34**/SDA1/SDA0/TOUTB6/CAPB6	-1.999	-0.430
54	**COM2**	1.999	0.570	119	**P35**/FOUT1/#BFR	-1.999	-0.530
55	**COM1**	1.999	0.670	120	**P36(EXCL5)**/TOUT3/RFCLKO	-1.999	-0.630
56	**COM0**	1.999	0.770	121	**P37(EXCL6)**/TOUTN3/LFRO/TOUT4	-1.999	-0.730
57	N.C.	1.999	0.870	122	**P40**/FOUTH	-1.999	-0.830
58	N.C.	1.999	0.970	123	**DSIO**/P41	-1.999	-0.930
59	N.C.	1.999	1.070	124	**DST2**/P42	-1.999	-1.030
60	**CB**	1.999	1.170	125	**DCLK**/P43	-1.999	-1.130
61	**CA**	1.999	1.270	126	N.C.	-1.999	-1.230
62	**VC3**	1.999	1.370	127	**SEG0**	-1.999	-1.330
63	**VC2**	1.999	1.470	128	**SEG1**	-1.999	-1.430
64	**VC1**	1.999	1.570	129	**SEG2**	-1.999	-1.530
65	**VSS**	1.999	1.680	130	**SEG3**	-1.999	-1.630

1.3.2 S1C17604 Pin Configuration Diagram

TQFP14-100pin (S1C17604)

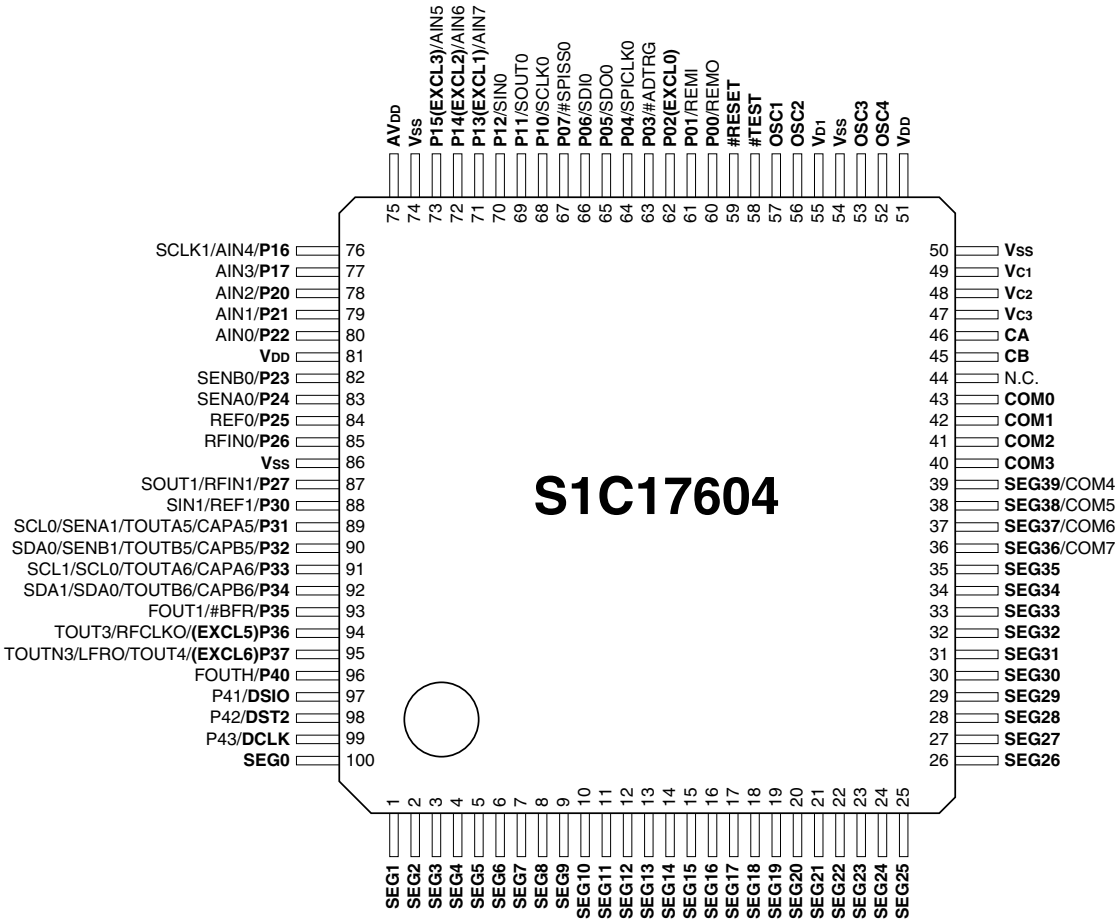


Figure 1.3.2.1 S1C17604 Pin Configuration Diagram (TQFP14-100pin)

Chip (S1C17604)

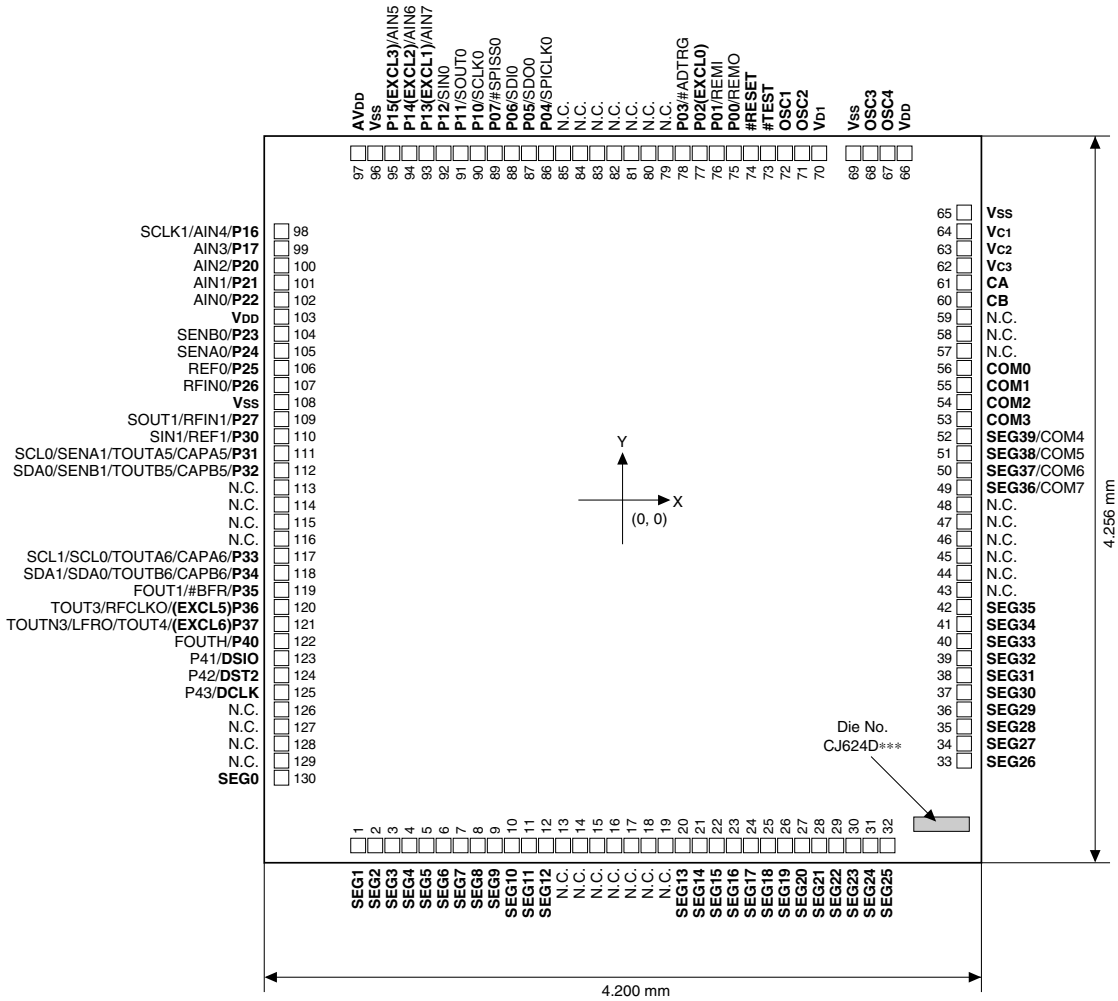


Figure 1.3.2.2 S1C17604 Pad Configuration Diagram

Chip size	X = 4.200 mm, Y = 4.256 mm
Pad opening	No. 1 to 32, 66 to 97: X = 87 µm, Y = 85 µm
	No. 33 to 65, 98 to 130: X = 85 µm, Y = 87 µm
Chip thickness	400 µm

∗ The S1C17604 chip has the same pad configuration and coordinates as those of the S1C17624. However, the P44–P47, P50–P56, and SEG40–SEG55 pads do not exist in the S1C17604 chip.

Table 1.3.2.1 S1C17604 Pad Coordinates

No.	Name	X (mm)	Y (mm)
1	**SEG1**	-1.550	-2.027
2	**SEG2**	-1.450	-2.027
3	**SEG3**	-1.350	-2.027
4	**SEG4**	-1.250	-2.027
5	**SEG5**	-1.150	-2.027
6	**SEG6**	-1.050	-2.027
7	**SEG7**	-0.950	-2.027
8	**SEG8**	-0.850	-2.027
9	**SEG9**	-0.750	-2.027
10	**SEG10**	-0.650	-2.027
11	**SEG11**	-0.550	-2.027
12	**SEG12**	-0.450	-2.027
13	N.C.	-0.350	-2.027
14	N.C.	-0.250	-2.027
15	N.C.	-0.150	-2.027
16	N.C.	-0.050	-2.027
17	N.C.	0.050	-2.027
18	N.C.	0.150	-2.027
19	N.C.	0.250	-2.027
20	**SEG13**	0.350	-2.027
21	**SEG14**	0.450	-2.027
22	**SEG15**	0.550	-2.027
23	**SEG16**	0.650	-2.027
24	**SEG17**	0.750	-2.027
25	**SEG18**	0.850	-2.027
26	**SEG19**	0.950	-2.027
27	**SEG20**	1.050	-2.027
28	**SEG21**	1.150	-2.027
29	**SEG22**	1.250	-2.027
30	**SEG23**	1.350	-2.027
31	**SEG24**	1.450	-2.027
32	**SEG25**	1.550	-2.027
33	**SEG26**	1.999	-1.530
34	**SEG27**	1.999	-1.430
35	**SEG28**	1.999	-1.330
36	**SEG29**	1.999	-1.230
37	**SEG30**	1.999	-1.130
38	**SEG31**	1.999	-1.030
39	**SEG32**	1.999	-0.930
40	**SEG33**	1.999	-0.830
41	**SEG34**	1.999	-0.730
42	**SEG35**	1.999	-0.630
43	N.C.	1.999	-0.530
44	N.C.	1.999	-0.430
45	N.C.	1.999	-0.330
46	N.C.	1.999	-0.230
47	N.C.	1.999	-0.130
48	N.C.	1.999	-0.030
49	**SEG36**/COM7	1.999	0.070
50	**SEG37**/COM6	1.999	0.170
51	**SEG38**/COM5	1.999	0.270
52	**SEG39**/COM4	1.999	0.370
53	**COM3**	1.999	0.470
54	**COM2**	1.999	0.570
55	**COM1**	1.999	0.670
56	**COM0**	1.999	0.770
57	N.C.	1.999	0.870
58	N.C.	1.999	0.970
59	N.C.	1.999	1.070
60	**CB**	1.999	1.170
61	**CA**	1.999	1.270
62	**V_{C3}**	1.999	1.370
63	**V_{C2}**	1.999	1.470
64	**V_{C1}**	1.999	1.570
65	**V_{SS}**	1.999	1.680
66	**V_{DD}**	1.650	2.027
67	**OSC4**	1.550	2.027
68	**OSC3**	1.450	2.027
69	**V_{SS}**	1.350	2.027
70	**V_{D1}**	1.150	2.027
71	**OSC2**	1.050	2.027
72	**OSC1**	0.950	2.027
73	**#TEST**	0.850	2.027
74	**#RESET**	0.750	2.027
75	**P00**/REMO	0.650	2.027
76	**P01**/REMI	0.550	2.027
77	**P02(EXCL0)**	0.450	2.027
78	**P03**/#ADTRG	0.350	2.027
79	N.C.	0.250	2.027
80	N.C.	0.150	2.027
81	N.C.	0.050	2.027
82	N.C.	-0.050	2.027
83	N.C.	-0.150	2.027
84	N.C.	-0.250	2.027
85	N.C.	-0.350	2.027
86	**P04**/SPICLK0	-0.450	2.027
87	**P05**/SDO0	-0.550	2.027
88	**P06**/SDI0	-0.650	2.027
89	**P07**/#SPISS0	-0.750	2.027
90	**P10**/SCLK0	-0.850	2.027
91	**P11**/SOUT0	-0.950	2.027
92	**P12**/SIN0	-1.050	2.027
93	**P13(EXCL1)**/AIN7	-1.150	2.027
94	**P14(EXCL2)**/AIN6	-1.250	2.027
95	**P15(EXCL3)**/AIN5	-1.350	2.027
96	**V_{SS}**	-1.450	2.027
97	**AV_{DD}**	-1.550	2.027
98	**P16**/SCLK1/AIN4	-1.999	1.570
99	**P17**/AIN3	-1.999	1.470
100	**P20**/AIN2	-1.999	1.370
101	**P21**/AIN1	-1.999	1.270
102	**P22**/AIN0	-1.999	1.170
103	**V_{DD}**	-1.999	1.070
104	**P23**/SENB0	-1.999	0.970
105	**P24**/SENA0	-1.999	0.870
106	**P25**/REF0	-1.999	0.770
107	**P26**/RFIN0	-1.999	0.670
108	**V_{SS}**	-1.999	0.570
109	**P27**/SOUT1/RFIN1	-1.999	0.470
110	**P30**/SIN1/REF1	-1.999	0.370
111	**P31**/SCL0/SENA1/TOUTA5/CAPA5	-1.999	0.270
112	**P32**/SDA0/SENB1/TOUTB5/CAPB5	-1.999	0.170
113	N.C.	-1.999	0.070
114	N.C.	-1.999	-0.030
115	N.C.	-1.999	-0.130
116	N.C.	-1.999	-0.230
117	**P33**/SCL1/SCL0/TOUTA6/CAPA6	-1.999	-0.330
118	**P34**/SDA1/SDA0/TOUTB6/CAPB6	-1.999	-0.430
119	**P35**/FOUT1/#BFR	-1.999	-0.530
120	**P36(EXCL5)**/TOUT3/RFCLKO	-1.999	-0.630
121	**P37(EXCL6)**/TOUTN3/LFRO/TOUT4	-1.999	-0.730
122	**P40**/FOUTH	-1.999	-0.830
123	**DSIO**/P41	-1.999	-0.930
124	**DST2**/P42	-1.999	-1.030
125	**DCLK**/P43	-1.999	-1.130
126	N.C.	-1.999	-1.230
127	N.C.	-1.999	-1.330
128	N.C.	-1.999	-1.430
129	N.C.	-1.999	-1.530
130	**SEG0**	-1.999	-1.630

1.3.3 S1C17622 Pin Configuration Diagram

TQFP15-128pin (S1C17622)

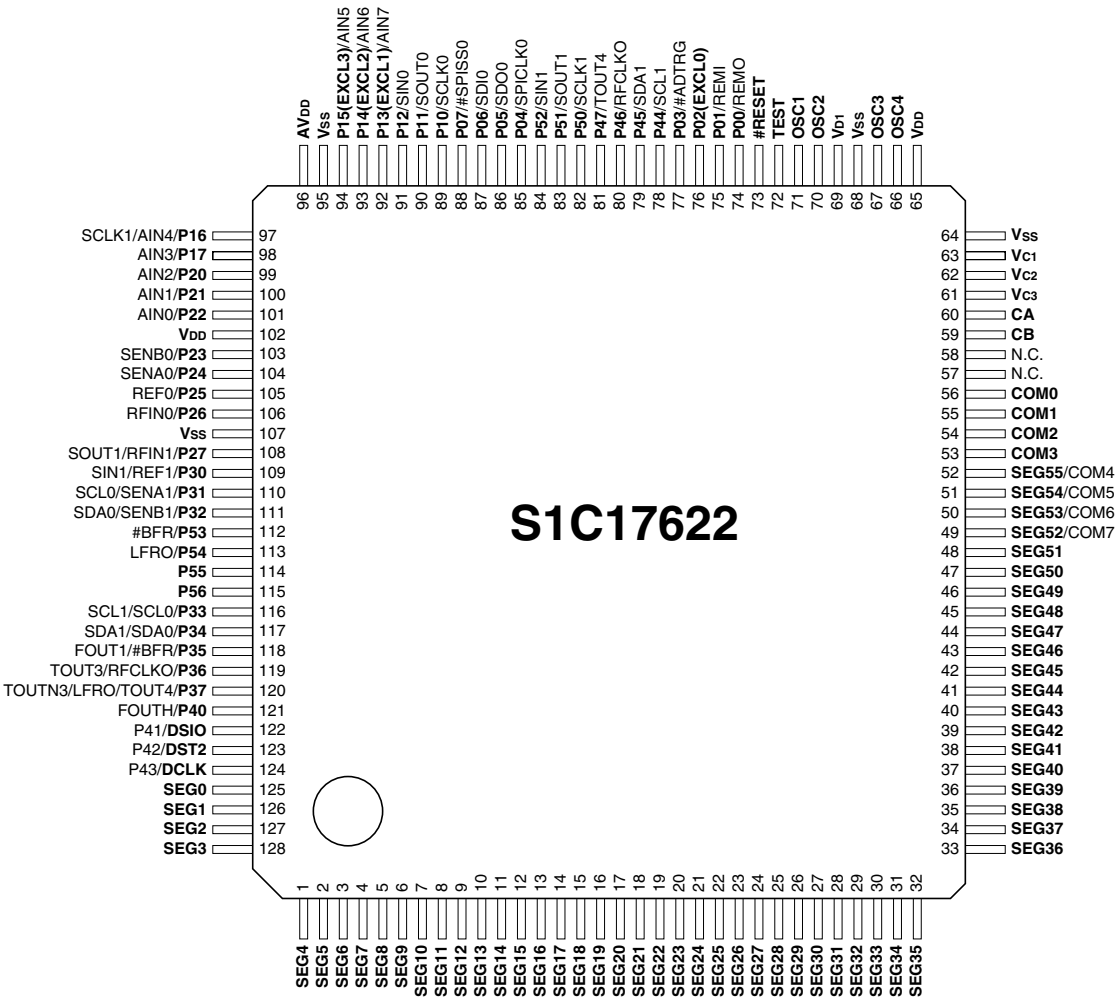


Figure 1.3.3.1 S1C17622 Pin Configuration Diagram (TQFP15-128pin)

* The S1C17622 (TQFP15-128pin) has the same pin configuration as that of the S1C17624 (TQFP15-128pin). However, 16-bit PWM timer (T16A2) input/output signals (EXCL5, TOUTA5/CAPA5, TOUTB5/CAPB5, EXCL6, TOUTA6/CAPA6, and TOUTB6/CAPB6) are not assigned to the S1C17622 pins.

Chip (S1C17622)

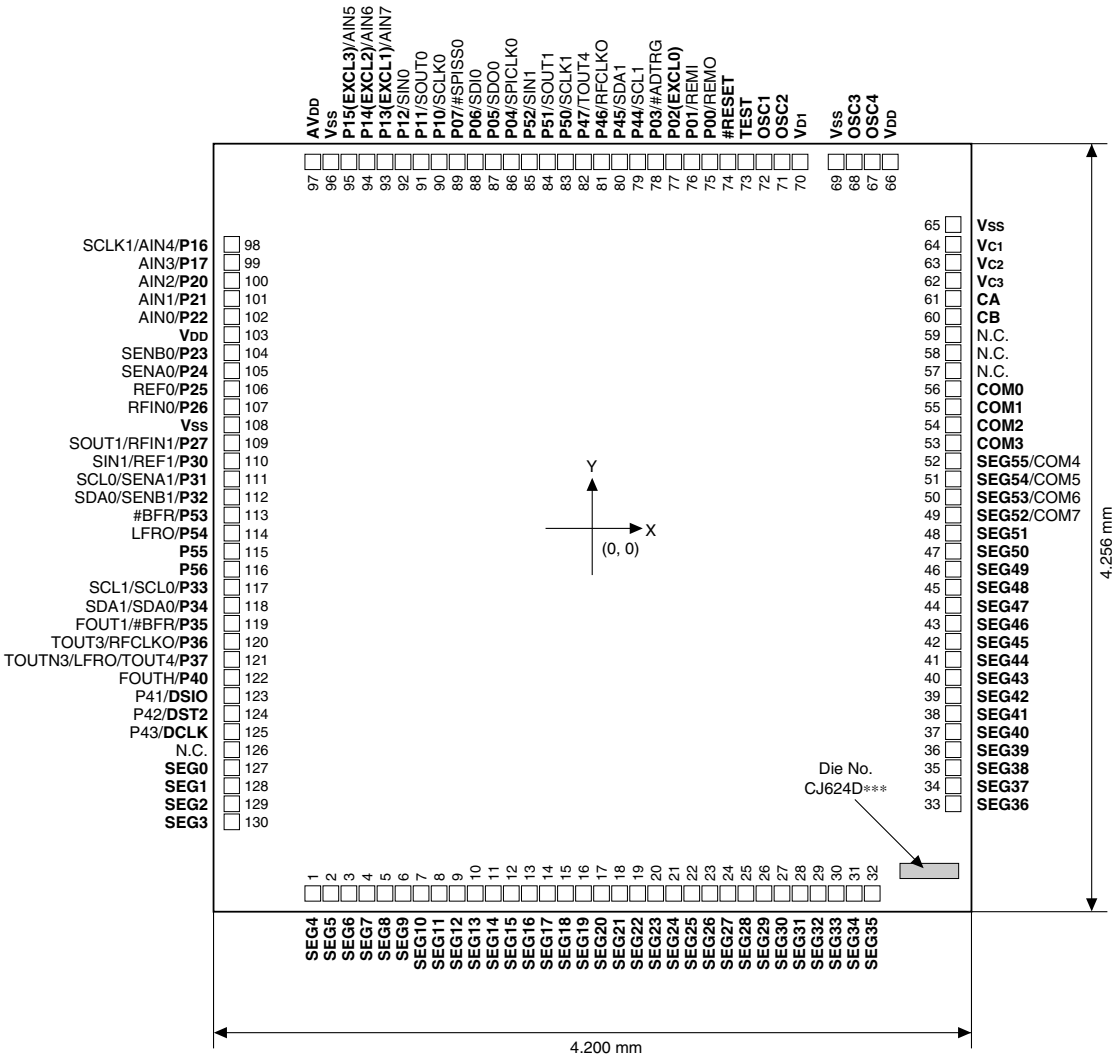


Figure 1.3.3.2 S1C17622 Pad Configuration Diagram

Chip size	X = 4.200 mm, Y = 4.256 mm
Pad opening	No. 1 to 32, 66 to 97: X = 87 µm, Y = 85 µm
	No. 33 to 65, 98 to 130: X = 85 µm, Y = 87 µm
Chip thickness	400 µm

∗ The S1C17622 chip has the same pad configuration and coordinates as those of the S1C17624. However, 16-bit PWM timer (T16A2) input/output signals (EXCL5, TOUTA5/CAPA5, TOUTB5/CAPB5, EXCL6, TOUTA6/CAPA6, and TOUTB6/CAPB6) are not assigned to the S1C17622 pads.

Table 1.3.3.1 S1C17622 Pad Coordinates

No.	Name	X (mm)	Y (mm)
1	**SEG4**	-1.550	-2.027
2	**SEG5**	-1.450	-2.027
3	**SEG6**	-1.350	-2.027
4	**SEG7**	-1.250	-2.027
5	**SEG8**	-1.150	-2.027
6	**SEG9**	-1.050	-2.027
7	**SEG10**	-0.950	-2.027
8	**SEG11**	-0.850	-2.027
9	**SEG12**	-0.750	-2.027
10	**SEG13**	-0.650	-2.027
11	**SEG14**	-0.550	-2.027
12	**SEG15**	-0.450	-2.027
13	**SEG16**	-0.350	-2.027
14	**SEG17**	-0.250	-2.027
15	**SEG18**	-0.150	-2.027
16	**SEG19**	-0.050	-2.027
17	**SEG20**	0.050	-2.027
18	**SEG21**	0.150	-2.027
19	**SEG22**	0.250	-2.027
20	**SEG23**	0.350	-2.027
21	**SEG24**	0.450	-2.027
22	**SEG25**	0.550	-2.027
23	**SEG26**	0.650	-2.027
24	**SEG27**	0.750	-2.027
25	**SEG28**	0.850	-2.027
26	**SEG29**	0.950	-2.027
27	**SEG30**	1.050	-2.027
28	**SEG31**	1.150	-2.027
29	**SEG32**	1.250	-2.027
30	**SEG33**	1.350	-2.027
31	**SEG34**	1.450	-2.027
32	**SEG35**	1.550	-2.027
33	**SEG36**	1.999	-1.530
34	**SEG37**	1.999	-1.430
35	**SEG38**	1.999	-1.330
36	**SEG39**	1.999	-1.230
37	**SEG40**	1.999	-1.130
38	**SEG41**	1.999	-1.030
39	**SEG42**	1.999	-0.930
40	**SEG43**	1.999	-0.830
41	**SEG44**	1.999	-0.730
42	**SEG45**	1.999	-0.630
43	**SEG46**	1.999	-0.530
44	**SEG47**	1.999	-0.430
45	**SEG48**	1.999	-0.330
46	**SEG49**	1.999	-0.230
47	**SEG50**	1.999	-0.130
48	**SEG51**	1.999	-0.030
49	**SEG52**/COM7	1.999	0.070
50	**SEG53**/COM6	1.999	0.170
51	**SEG54**/COM5	1.999	0.270
52	**SEG55**/COM4	1.999	0.370
53	**COM3**	1.999	0.470
54	**COM2**	1.999	0.570
55	**COM1**	1.999	0.670
56	**COM0**	1.999	0.770
57	N.C.	1.999	0.870
58	N.C.	1.999	0.970
59	N.C.	1.999	1.070
60	**CB**	1.999	1.170
61	**CA**	1.999	1.270
62	**V_{C3}**	1.999	1.370
63	**V_{C2}**	1.999	1.470
64	**V_{C1}**	1.999	1.570
65	**V_{SS}**	1.999	1.680
66	**V_{DD}**	1.650	2.027
67	**OSC4**	1.550	2.027
68	**OSC3**	1.450	2.027
69	**V_{SS}**	1.350	2.027
70	**V_{D1}**	1.150	2.027
71	**OSC2**	1.050	2.027
72	**OSC1**	0.950	2.027
73	**TEST**	0.850	2.027
74	**#RESET**	0.750	2.027
75	**P00**/REMO	0.650	2.027
76	**P01**/REMI	0.550	2.027
77	**P02(EXCL0)**	0.450	2.027
78	**P03**/#ADTRG	0.350	2.027
79	**P44**/SCL1	0.250	2.027
80	**P45**/SDA1	0.150	2.027
81	**P46**/RFCLKO	0.050	2.027
82	**P47**/TOUT4	-0.050	2.027
83	**P50**/SCLK1	-0.150	2.027
84	**P51**/SOUT1	-0.250	2.027
85	**P52**/SIN1	-0.350	2.027
86	**P04**/SPICLK0	-0.450	2.027
87	**P05**/SDO0	-0.550	2.027
88	**P06**/SDI0	-0.650	2.027
89	**P07**/#SPISS0	-0.750	2.027
90	**P10**/SCLK0	-0.850	2.027
91	**P11**/SOUT0	-0.950	2.027
92	**P12**/SIN0	-1.050	2.027
93	**P13(EXCL1)**/AIN7	-1.150	2.027
94	**P14(EXCL2)**/AIN6	-1.250	2.027
95	**P15(EXCL3)**/AIN5	-1.350	2.027
96	**V_{SS}**	-1.450	2.027
97	**AV_{DD}**	-1.550	2.027
98	**P16**/SCLK1/AIN4	-1.999	1.570
99	**P17**/AIN3	-1.999	1.470
100	**P20**/AIN2	-1.999	1.370
101	**P21**/AIN1	-1.999	1.270
102	**P22**/AIN0	-1.999	1.170
103	**V_{DD}**	-1.999	1.070
104	**P23**/SENB0	-1.999	0.970
105	**P24**/SENA0	-1.999	0.870
106	**P25**/REF0	-1.999	0.770
107	**P26**/RFIN0	-1.999	0.670
108	**V_{SS}**	-1.999	0.570
109	**P27**/SOUT1/RFIN1	-1.999	0.470
110	**P30**/SIN1/REF1	-1.999	0.370
111	**P31**/SCL0/SENA1	-1.999	0.270
112	**P32**/SDA0/SENB1	-1.999	0.170
113	**P53**/#BFR	-1.999	0.070
114	**P54**/LFRO	-1.999	-0.030
115	**P55**	-1.999	-0.130
116	**P56**	-1.999	-0.230
117	**P33**/SCL1/SCL0	-1.999	-0.330
118	**P34**/SDA1/SDA0	-1.999	-0.430
119	**P35**/FOUT1/#BFR	-1.999	-0.530
120	**P36**/TOUT3/RFCLKO	-1.999	-0.630
121	**P37**/TOUTN3/LFRO/TOUT4	-1.999	-0.730
122	**P40**/FOUTH	-1.999	-0.830
123	**DSIO**/P41	-1.999	-0.930
124	**DST2**/P42	-1.999	-1.030
125	**DCLK**/P43	-1.999	-1.130
126	N.C.	-1.999	-1.230
127	**SEG0**	-1.999	-1.330
128	**SEG1**	-1.999	-1.430
129	**SEG2**	-1.999	-1.530
130	**SEG3**	-1.999	-1.630

1.3.4 S1C17602/621 Pin Configuration Diagram

TQFP14-100pin (S1C17602/621)

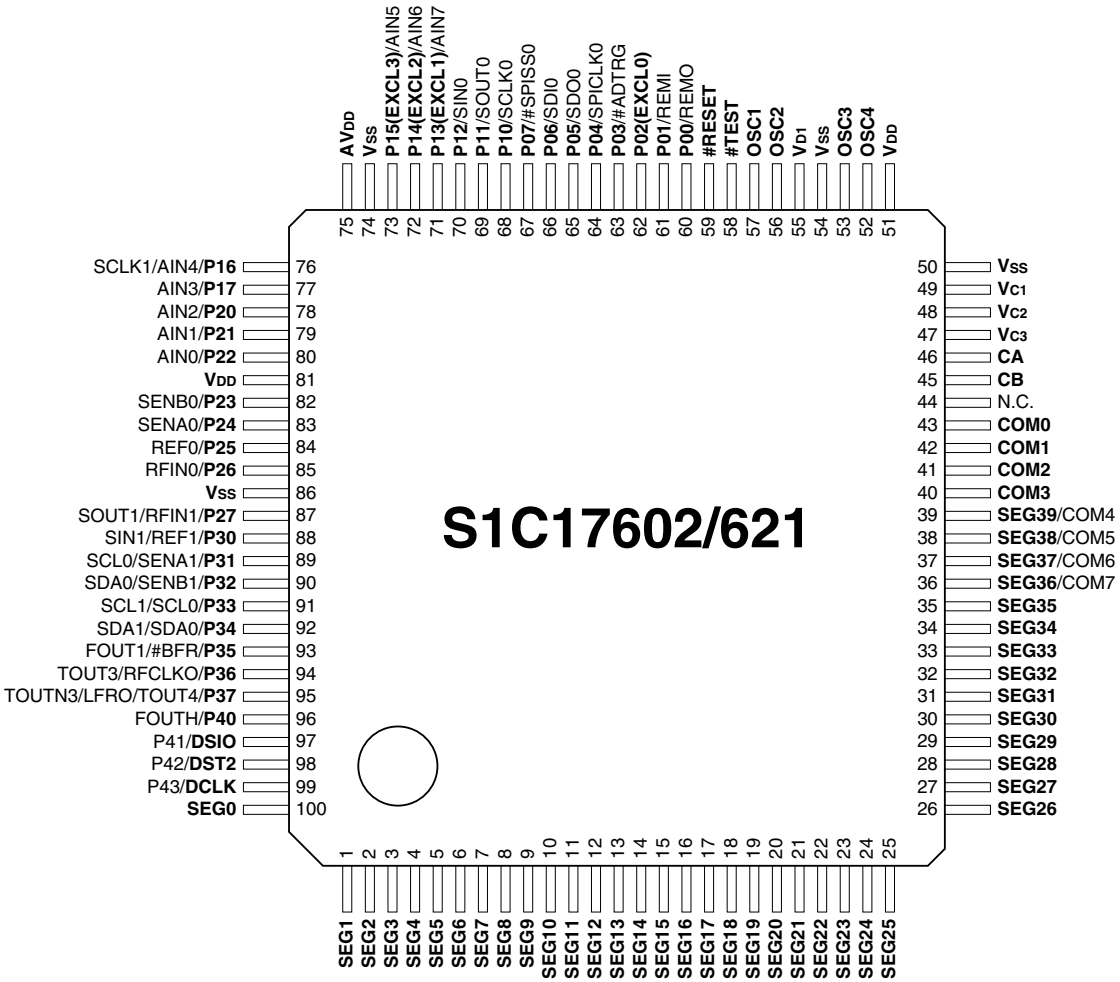
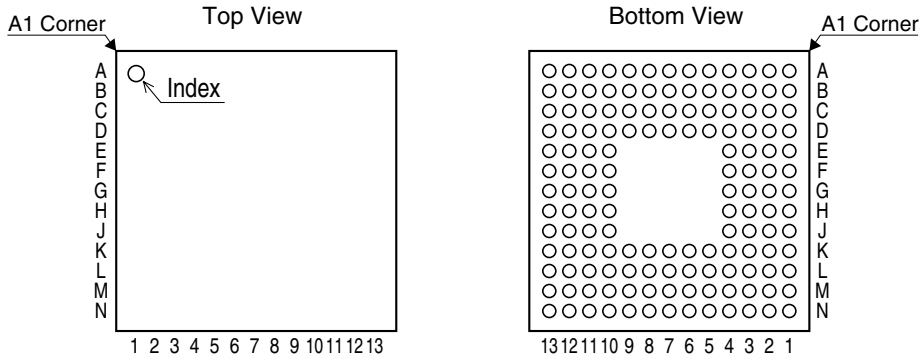


Figure 1.3.4.1 S1C17602/621 Pin Configuration Diagram (TQFP14-100pin)

∗ The S1C17602/621 (TQFP14-100pin) has the same pin configuration as that of the S1C17604 (TQFP14-100pin). However, 16-bit PWM timer (T16A2) input/output signals (EXCL5, TOUTA5/CAPA5, TOUTB5/CAPB5, EXCL6, TOUTA6/CAPA6, and TOUTB6/CAPB6) are not assigned to the S1C17602/621 pins.

VFBGA7H-144 (S1C17602/621)



	1	2	3	4	5	6	7	8	9	10	11	12	13	
A	N.C.	**DCLK** P43	**DST2** P42	**DSIO** P41	**P37** TOUTN3 LFRO TOUT4	**P34** SDA1 SDA0	**P27** SOUT1 RFIN1	**VSS**	**P26** RFIN0	**VDD**	**P21** AIN1	**P16** SCLK1 AIN4	N.C.	A
B	**SEG1**	**SEG0**	**VSS**	**VSS**	**P36** TOUT3 RFCLKO	**P33** SCL1 SCL0	**P31** SCL0 SENA1	**P30** SIN1 REF1	**P24** SENA0	**P22** AIN0	**P20** AIN2	**VSS**	**AVDD**	B
C	**SEG3**	**SEG4**	**SEG2**	**VSS**	**P40** FOUTH	**P35** FOUT1 #BFR	**P32** SDA0 SENB1	**VSS**	**P25** REF0	**P23** SENB0	**P17** AIN3	**P15 (EXCL3)** AIN5	**P14 (EXCL2)** AIN6	C
D	**SEG5**	**SEG6**	**VSS**	**VSS**	**VSS**	**VSS**	**VSS**	**VSS**	**VSS**	**VSS**	**VDD**	**P13 (EXCL1)** AIN7	**P12** SIN0	D
E	**SEG8**	**SEG7**	**VSS**	**VSS**						**VDD**	**VDD**	**P11** SOUT0	**P10** SCLK0	E
F	**SEG10**	**SEG11**	**SEG9**	**VSS**						**VDD**	**VDD**	**P07** #SPISS0	**P06** SDI0	F
G	**SEG13**	**SEG14**	**SEG12**	**VSS**			**Top View**			**VSS**	**P05** SDO0	**P04** SPICLK0	**P03** #ADTRG	G
H	**SEG16**	**SEG15**	**VSS**	**VSS**						**VDD**	**P02 (EXCL0)**	**P01** REMI	**P00** REMO	H
J	**SEG18**	**SEG17**	**VSS**	**VSS**						**VDD**	**VDD**	**#TEST**	**#RESET**	J
K	**SEG21**	**SEG20**	**SEG19**	**VSS**	**VSS**	**VSS**	**VSS**	**VSS**	**VDD**	**VSS**	**VD1**	**OSC1**	**OSC2**	K
L	**SEG23**	**SEG22**	**VSS**	**SEG29**	**SEG32**	**COM6** SEG37	**COM3**	**COM0**	N.C.	**VC3**	**VDD**	**VDD**	**VSS**	L
M	**SEG25**	**SEG24**	**SEG27**	**SEG30**	**SEG34**	**COM7** SEG36	**COM4** SEG39	**COM1**	N.C.	**CA**	**VC1**	**OSC3**	**OSC4**	M
N	N.C.	**SEG26**	**SEG28**	**SEG31**	**SEG33**	**SEG35**	**COM5** SEG38	**COM2**	N.C.	**CB**	**VC2**	**VSS**	N.C.	N
	1	2	3	4	5	6	7	8	9	10	11	12	13	

Figure 1.3.4.2 S1C17602/621 Pin Configuration Diagram (VFBGA7H-144, Top View)

Chip (S1C17602/621)

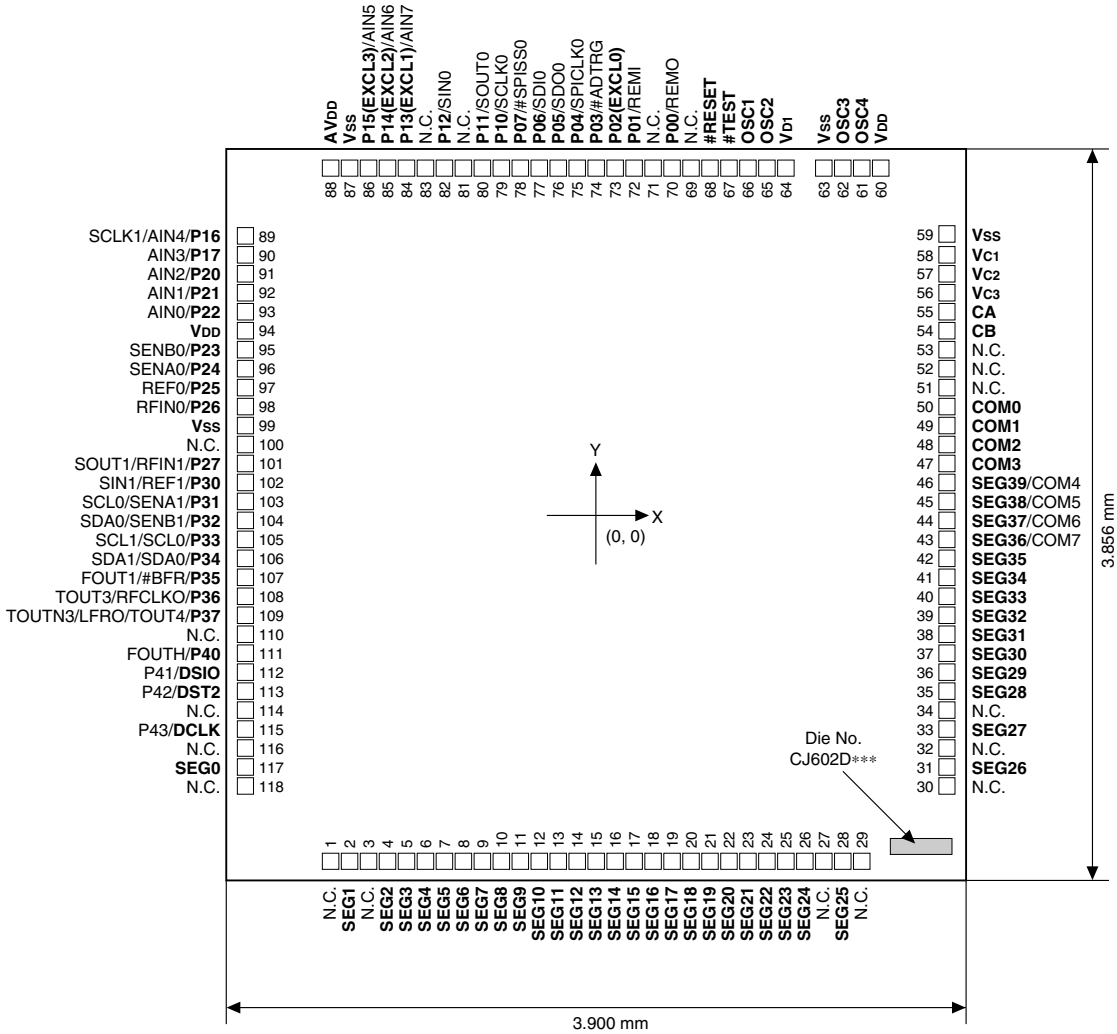


Figure 1.3.4.3 S1C17602/621 Pad Configuration Diagram

Chip size	X = 3.900 mm, Y = 3.856 mm
Pad opening	No. 1 to 29, 60 to 88: X = 87 µm, Y = 85 µm
	No. 30 to 59, 89 to 118: X = 85 µm, Y = 87 µm
Chip thickness	400 µm

Table 1.3.4.1 S1C17602/621 Pad Coordinates

No.	Name	X (mm)	Y (mm)	No.	Name	X (mm)	Y (mm)
1	N.C.	-1.400	-1.827	60	**V_{DD}**	1.500	1.827
2	**SEG1**	-1.300	-1.827	61	**OSC4**	1.400	1.827
3	N.C.	-1.200	-1.827	62	**OSC3**	1.300	1.827
4	**SEG2**	-1.100	-1.827	63	**V_{SS}**	1.200	1.827
5	**SEG3**	-1.000	-1.827	64	**V_{D1}**	1.000	1.827
6	**SEG4**	-0.900	-1.827	65	**OSC2**	0.900	1.827
7	**SEG5**	-0.800	-1.827	66	**OSC1**	0.800	1.827
8	**SEG6**	-0.700	-1.827	67	**#TEST**	0.700	1.827
9	**SEG7**	-0.600	-1.827	68	**#RESET**	0.600	1.827
10	**SEG8**	-0.500	-1.827	69	N.C.	0.500	1.827
11	**SEG9**	-0.400	-1.827	70	**P00**/REMO	0.400	1.827
12	**SEG10**	-0.300	-1.827	71	N.C.	0.300	1.827
13	**SEG11**	-0.200	-1.827	72	**P01**/REMI	0.200	1.827
14	**SEG12**	-0.100	-1.827	73	**P02(EXCL0)**	0.100	1.827
15	**SEG13**	0.000	-1.827	74	**P03**/#ADTRG	0.000	1.827
16	**SEG14**	0.100	-1.827	75	**P04**/SPICLK0	-0.100	1.827
17	**SEG15**	0.200	-1.827	76	**P05**/SDO0	-0.200	1.827
18	**SEG16**	0.300	-1.827	77	**P06**/SDI0	-0.300	1.827
19	**SEG17**	0.400	-1.827	78	**P07**/#SPISS0	-0.400	1.827
20	**SEG18**	0.500	-1.827	79	**P10**/SCLK0	-0.500	1.827
21	**SEG19**	0.600	-1.827	80	**P11**/SOUT0	-0.600	1.827
22	**SEG20**	0.700	-1.827	81	N.C.	-0.700	1.827
23	**SEG21**	0.800	-1.827	82	**P12**/SIN0	-0.800	1.827
24	**SEG22**	0.900	-1.827	83	N.C.	-0.900	1.827
25	**SEG23**	1.000	-1.827	84	**P13(EXCL1)**/AIN7	-1.000	1.827
26	**SEG24**	1.100	-1.827	85	**P14(EXCL2)**/AIN6	-1.100	1.827
27	N.C.	1.200	-1.827	86	**P15(EXCL3)**/AIN5	-1.200	1.827
28	**SEG25**	1.300	-1.827	87	**V_{SS}**	-1.300	1.827
29	N.C.	1.400	-1.827	88	**AV_{DD}**	-1.400	1.827
30	N.C.	1.849	-1.430	89	**P16**/SCLK1/AIN4	-1.849	1.470
31	**SEG26**	1.849	-1.330	90	**P17**/AIN3	-1.849	1.370
32	N.C.	1.849	-1.230	91	**P20**/AIN2	-1.849	1.270
33	**SEG27**	1.849	-1.130	92	**P21**/AIN1	-1.849	1.170
34	N.C.	1.849	-1.030	93	**P22**/AIN0	-1.849	1.070
35	**SEG28**	1.849	-0.930	94	**V_{DD}**	-1.849	0.970
36	**SEG29**	1.849	-0.830	95	**P23**/SENB0	-1.849	0.870
37	**SEG30**	1.849	-0.730	96	**P24**/SENA0	-1.849	0.770
38	**SEG31**	1.849	-0.630	97	**P25**/REF0	-1.849	0.670
39	**SEG32**	1.849	-0.530	98	**P26**/RFIN0	-1.849	0.570
40	**SEG33**	1.849	-0.430	99	**V_{SS}**	-1.849	0.470
41	**SEG34**	1.849	-0.330	100	N.C.	-1.849	0.370
42	**SEG35**	1.849	-0.230	101	**P27**/SOUT1/RFIN1	-1.849	0.270
43	**SEG36**/COM7	1.849	-0.130	102	**P30**/SIN1/REF1	-1.849	0.170
44	**SEG37**/COM6	1.849	-0.030	103	**P31**/SCL0/SENA1	-1.849	0.070
45	**SEG38**/COM5	1.849	0.070	104	**P32**/SDA0/SENB1	-1.849	-0.030
46	**SEG39**/COM4	1.849	0.170	105	**P33**/SCL1/SCL0	-1.849	-0.130
47	**COM3**	1.849	0.270	106	**P34**/SDA1/SDA0	-1.849	-0.230
48	**COM2**	1.849	0.370	107	**P35**/FOUT1/#BFR	-1.849	-0.330
49	**COM1**	1.849	0.470	108	**P36**/TOUT3/RFCLKO	-1.849	-0.430
50	**COM0**	1.849	0.570	109	**P37**/TOUTN3/LFRO/TOUT4	-1.849	-0.530
51	N.C.	1.849	0.670	110	N.C.	-1.849	-0.630
52	N.C.	1.849	0.770	111	**P40**/FOUTH	-1.849	-0.730
53	N.C.	1.849	0.870	112	**DSIO**/P41	-1.849	-0.830
54	**CB**	1.849	0.970	113	**DST2**/P42	-1.849	-0.930
55	**CA**	1.849	1.070	114	N.C.	-1.849	-1.030
56	**V_{C3}**	1.849	1.170	115	**DCLK**/P43	-1.849	-1.130
57	**V_{C2}**	1.849	1.270	116	N.C.	-1.849	-1.230
58	**V_{C1}**	1.849	1.370	117	**SEG0**	-1.849	-1.330
59	**V_{SS}**	1.849	1.480	118	N.C.	-1.849	-1.430

1.3.5 Pin Descriptions

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

Table 1.3.5.1 Pin Descriptions

Name	I/O	Default status	Function	Available (○)/unavailable (–)				
				17624	17604	17622	17602	17621
V_{DD}	–	–	Power supply pin (+)	○	○	○	○	○
AV_{DD}	–	–	Analog power supply pin (+)	○	○	○	○	○
V_{SS}	–	–	Power supply pin (GND)	○	○	○	○	○
V_{D1}	–	–	Internal operating voltage regulator output pin	○	○	○	○	○
V_{C1}–V_{C3}	–	–	LCD drive voltage output pins	○	○	○	○	○
CA, CB	–	–	LCD voltage boost capacitor connecting pin	○	○	○	○	○
OSC3	I	I	OSC3 oscillation input pin/External clock input pin	○	○	○	○	○
OSC4	O	O	OSC3 oscillation output pin	○	○	○	○	○
OSC1	I	I	OSC1 oscillation input pin	○	○	○	○	○
OSC2	O	O	OSC1 oscillation output pin	○	○	○	○	○
P00	I/O	I (Pull-up)	I/O port pin (with port input interrupt function)	○	○	○	○	○
REMO	O		REMC output pin					
P01	I/O	I (Pull-up)	I/O port pin (with port input interrupt function)	○	○	○	○	○
REMI	I		REMC input pin					
P02	I/O	I (Pull-up)	I/O port pin (with port input interrupt function)	○	○	○	○	○
(EXCL0)	I		T16 Ch.0 external clock input pin					
P03	I/O	I (Pull-up)	I/O port pin (with port input interrupt function)	○	○	○	○	○
#ADTRG	I		A/D converter external trigger input pin					
P04	I/O	I (Pull-up)	I/O port pin (with port input interrupt function)	○	○	○	○	○
SPICLK0	I/O		SPI Ch.0 clock input/output pin					
P05	I/O	I (Pull-up)	I/O port pin (with port input interrupt function)	○	○	○	○	○
SDO0	O		SPI Ch.0 data output pin					
P06	I/O	I (Pull-up)	I/O port pin (with port input interrupt function)	○	○	○	○	○
SDI0	I		SPI Ch.0 data input pin					
P07	I/O	I (Pull-up)	I/O port pin (with port input interrupt function)	○	○	○	○	○
#SPISS0	I		SPI Ch.0 slave select signal input pin					
P10	I/O	I (Pull-up)	I/O port pin (with port input interrupt function)	○	○	○	○	○
SCLK0	I		UART Ch.0 external clock input pin					
P11	I/O	I (Pull-up)	I/O port pin (with port input interrupt function)	○	○	○	○	○
SOUT0	O		UART Ch.0 data output pin					
P12	I/O	I (Pull-up)	I/O port pin (with port input interrupt function)	○	○	○	○	○
SIN0	I		UART Ch.0 data input pin					
P13	I/O	I (Pull-up)	I/O port pin (with port input interrupt function)	○	○	○	○	○
(EXCL1)	I		T16 Ch.1 external clock input pin					
AIN7	I		A/D converter Ch.7 analog signal input pin					
P14	I/O	I (Pull-up)	I/O port pin (with port input interrupt function)	○	○	○	○	○
(EXCL2)	I		T16 Ch.2 external clock input pin					
AIN6	I		A/D converter Ch.6 analog signal input pin					
P15	I/O	I (Pull-up)	I/O port pin (with port input interrupt function)	○	○	○	○	○
(EXCL3)	I		T16E Ch.0 external clock input pin					
AIN5	I		A/D converter Ch.5 analog signal input pin					
P16	I/O	I (Pull-up)	I/O port pin (with port input interrupt function)	○	○	○	○	○
SCLK1	I		UART Ch.1 external clock input pin					
AIN4	I		A/D converter Ch.4 analog signal input pin					
P17	I/O	I (Pull-up)	I/O port pin (with port input interrupt function)	○	○	○	○	○
AIN3	I		A/D converter Ch.3 analog signal input pin					
P20	I/O	I (Pull-up)	I/O port pin	○	○	○	○	○
AIN2	I		A/D converter Ch.2 analog signal input pin					
P21	I/O	I (Pull-up)	I/O port pin	○	○	○	○	○
AIN1	I		A/D converter Ch.1 analog signal input pin					
P22	I/O	I (Pull-up)	I/O port pin	○	○	○	○	○
AIN0	I		A/D converter Ch.0 analog signal input pin					
P23	I/O	I (Pull-up)	I/O port pin	○	○	○	○	○
SENB0	I/O		R/F converter Ch.0 sensor B oscillation control pin					
P24	I/O	I (Pull-up)	I/O port pin	○	○	○	○	○
SENA0	I/O		R/F converter Ch.0 sensor A oscillation control pin					
P25	I/O	I (Pull-up)	I/O port pin	○	○	○	○	○
REF0	I/O		R/F converter Ch.0 reference oscillation control pin					

Name	I/O	Default status	Function	Available (○)/unavailable (–)				
				17624	17604	17622	17602	17621
P26	I/O	I (Pull-up)	I/O port pin	○	○	○	○	○
RFIN0	I/O		R/F converter Ch.0 RFCLK input and oscillation control pin					
P27	I/O	I (Pull-up)	I/O port pin	○	○	○	○	○
SOUT1	O		UART Ch.1 data output pin					
RFIN1	I/O		R/F converter Ch.1 RFCLK input and oscillation control pin					
P30	I/O	I (Pull-up)	I/O port pin	○	○	○	○	○
SIN1	I		UART Ch.1 data input pin					
REF1	I/O		R/F converter Ch.1 reference oscillation control pin					
P31	I/O	I (Pull-up)	I/O port pin	○	○	○	○	○
SCL0	I/O		I²C master SCL input/output pin					
SENA1	I/O		R/F converter Ch.1 sensor A oscillation control pin					
TOUTA5	O		T16A2 Ch.0 TOUT A signal output pin			–	–	–
CAPA5	I		T16A2 Ch.0 capture A trigger signal input pin					
P32	I/O	I (Pull-up)	I/O port pin	○	○	○	○	○
SDA0	I/O		I²C master data input/output pin					
SENB1	I/O		R/F converter Ch.1 sensor B oscillation control pin					
TOUTB5	O		T16A2 Ch.0 TOUT B signal output pin			–	–	–
CAPB5	I		T16A2 Ch.0 capture B trigger signal input pin					
P33	I/O	I (Pull-up)	I/O port pin	○	○	○	○	○
SCL1	I/O		I²C slave SCL input/output pin					
SCL0	I/O		I²C master SCL input/output pin					
TOUTA6	O		T16A2 Ch.1 TOUT A signal output pin			–	–	–
CAPA6	I		T16A2 Ch.1 capture A trigger signal input pin					
P34	I/O	I (Pull-up)	I/O port pin	○	○	○	○	○
SDA1	I/O		I²C slave data input/output pin					
SDA0	I/O		I²C master data input/output pin					
TOUTB6	O		T16A2 Ch.1 TOUT B signal output pin			–	–	–
CAPB6	I		T16A2 Ch.1 capture B trigger signal input pin					
P35	I/O	I (Pull-up)	I/O port pin	○	○	○	○	○
FOUT1	O		OSC1 clock external output pin					
#BFR	I		I²C slave bus free request input pin					
P36	I/O	I (Pull-up)	I/O port pin	○	○	○	○	○
(EXCL5)	I		T16A2 Ch.0 external clock input pin			–	–	–
TOUT3	O		T16E Ch.0 PWM signal non-inverted output pin			○	○	○
RFCLKO	O		R/F clock monitor output pin					
P37	I/O	I (Pull-up)	I/O port pin	○	○	○	○	○
(EXCL6)	I		T16A2 Ch.1 external clock input pin			–	–	–
TOUTN3	O		T16E Ch.0 PWM signal inverted output pin			○	○	○
LFRO	O		LCD frame signal output pin					
TOUT4	O		T8OSC1 PWM signal non-inverted output pin					
P40	I/O	I (Pull-up)	I/O port pin	○	○	○	○	○
FOUTH	O		IOSC/OSC3 clock external output pin					
DSIO	I/O	I (Pull-up)	On-chip debugger data input/output pin	○	○	○	○	○
P41	I/O		I/O port pin					
DST2	O	O (L)	On-chip debugger status output pin	○	○	○	○	○
P42	I/O		I/O port pin					
DCLK	O	O (H)	On-chip debugger clock output pin	○	○	○	○	○
P43	O		Output port pin					
P44	I/O	I (Pull-up)	I/O port pin	○	–	○	–	–
SCL1	I/O		I²C slave SCL input/output pin					
P45	I/O	I (Pull-up)	I/O port pin	○	–	○	–	–
SDA1	I/O		I²C slave data input/output pin					
P46	I/O	I (Pull-up)	I/O port pin	○	–	○	–	–
RFCLKO	O		R/F clock monitor output pin					
P47	I/O	I (Pull-up)	I/O port pin	○	–	○	–	–
(EXCL5)	I		T16A2 Ch.0 external clock input pin			–		
TOUT4	O		T8OSC1 PWM signal non-inverted output pin			○		
P50	I/O	I (Pull-up)	I/O port pin	○	–	○	–	–
(EXCL6)	I		T16A2 Ch.1 external clock input pin			–		
SCLK1	I		UART Ch.1 external clock input pin			○		

Name	I/O	Default status	Function	Available (○)/unavailable (–)				
				17624	17604	17622	17602	17621
P51	I/O	I (Pull-up)	I/O port pin	○	–	○	–	–
SOUT1	O		UART Ch.1 data output pin					
TOUTA5	O		T16A2 Ch.0 TOUT A signal output pin			–		
CAPA5	I		T16A2 Ch.0 capture A trigger signal input pin					
P52	I/O	I (Pull-up)	I/O port pin	○	–	○	–	–
SIN1	I		UART Ch.1 data input pin					
TOUTB5	O		T16A2 Ch.0 TOUT B signal output pin			–		
CAPB5	I		T16A2 Ch.0 capture B trigger signal input pin					
P53	I/O	I (Pull-up)	I/O port pin	○	–	○	–	–
#BFR	I		I²C slave bus free request input pin					
P54	I/O	I (Pull-up)	I/O port pin	○	–	○	–	–
LFRO	O		LCD frame signal output pin					
P55	I/O	I (Pull-up)	I/O port pin	○	–	○	–	–
TOUTA6	O		T16A2 Ch.1 TOUT A signal output pin			–		
CAPA6	I		T16A2 Ch.1 capture A trigger signal input pin					
P56	I/O	I (Pull-up)	I/O port pin	○	–	○	–	–
TOUTB6	O		T16A2 Ch.1 TOUT B signal output pin			–		
CAPB6	I		T16A2 Ch.1 capture B trigger signal input pin					
SEG0–35	O	O (L)	LCD segment output pins	○	○	○	○	○
SEG36–39	O	O (L)	LCD segment output pins	○	○	○	○	○
COM7–4	O		LCD common output pins	–		–		
SEG40–51	O	O (L)	LCD segment output pins	○	–	○	–	–
SEG52–55	O	O (L)	LCD segment output pins	○	–	○	–	–
COM7–4	O		LCD common output pins					
COM0–3	O	O (L)	LCD common output pins	○	○	○	○	○
#RESET	I	I (Pull-up)	Initial reset input pin	○	○	○	○	○
TEST	I	I	Test input pin (Connect to V_{SS} for normal operation.)	○	–	○	–	–
#TEST	I	I	Test input pin (Connect to V_{DD} for normal operation.)	–	○	–	○	○

Note: DCLK pin initial status

The DCLK pin of the S1C17624/604/622 goes high after the #RESET pin is set to 1 (after reset status is canceled). The DCLK pin must be set to low with the pull-down resistor in the IC while the #RESET pin is set to 0 (reset status), therefore do not connect any circuit such as a pull-up resistor that may set the pin to high.

The S1C17602/621 DCLK pin goes high while the #RESET pin is set to 0 (reset status).

2 CPU

The S1C17624/604/622/602/621 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.

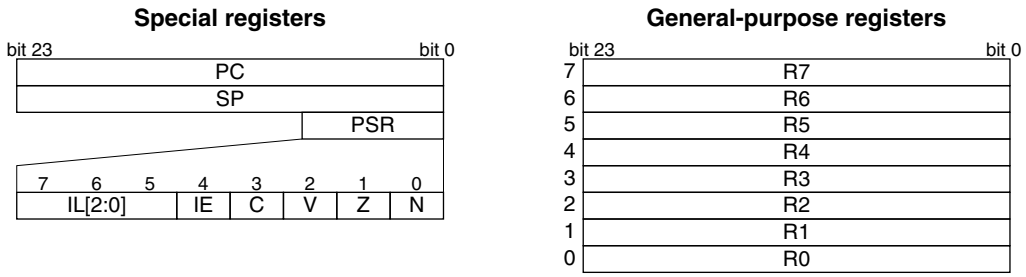


Figure 2.2.1 Registers

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 S1C17624/604/622/602/621 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 S1C17624/604/622/602/621 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

Figures 3.1, 3.2, and 3.3 show the S1C17624/604, S1C17622/602, and S1C17621 memory maps, respectively.

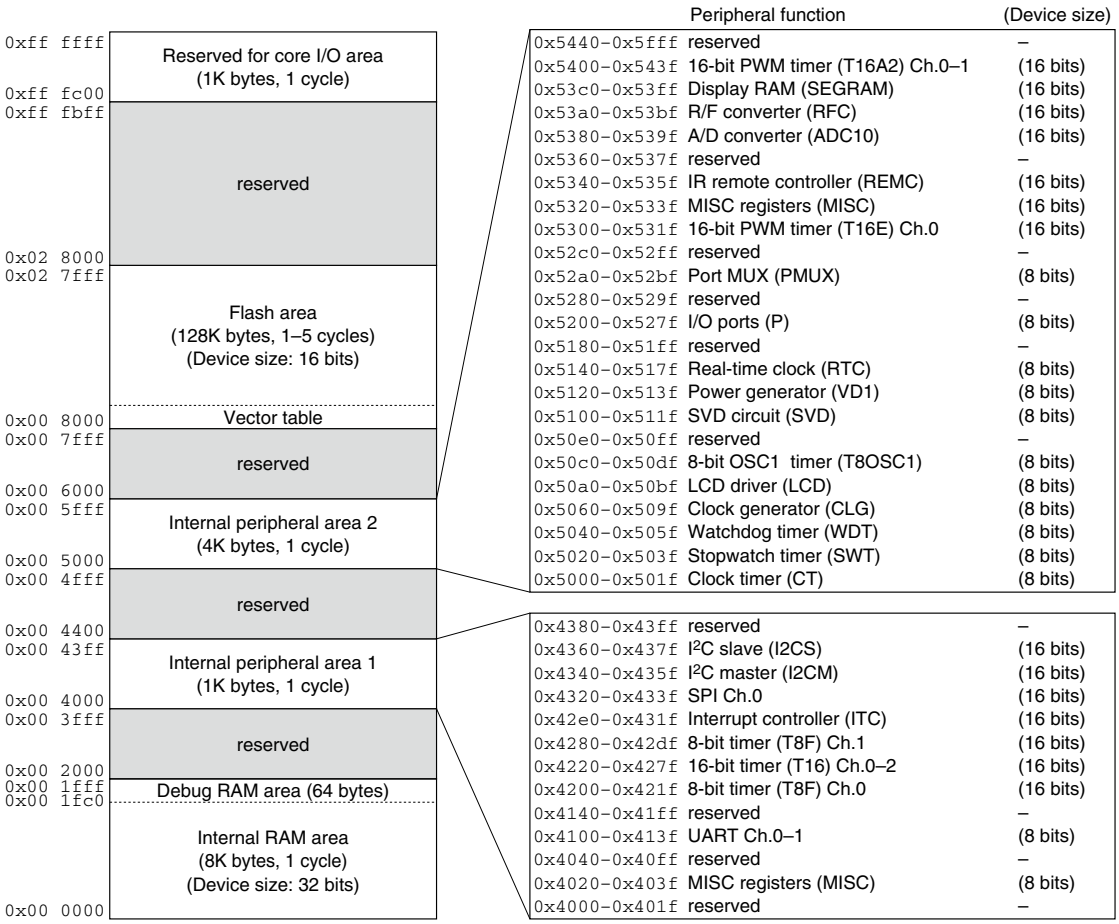
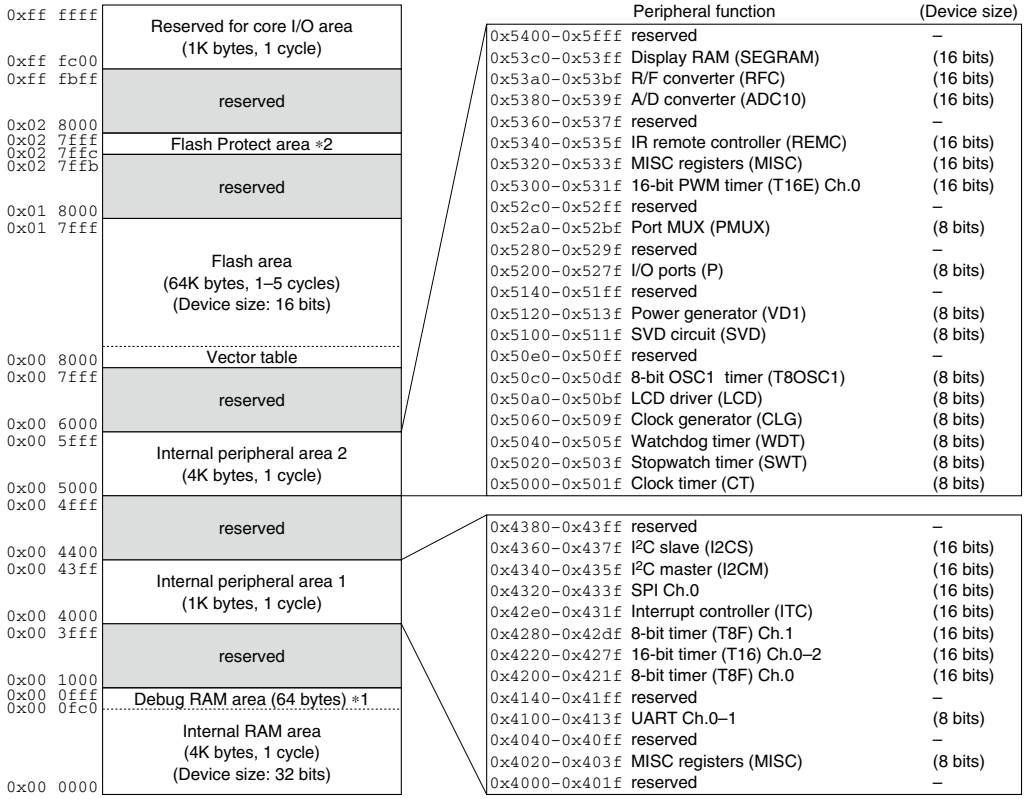


Figure 3.1 S1C17624/604 Memory Map



*1: The address range from 0x000fc0 to 0x000fff of the S1C17622 is an internal RAM area.
*2: The address range from 0x027ffc to 0x027fff of the S1C17602 is a reserved area.

Figure 3.2 S1C17622/602 Memory Map

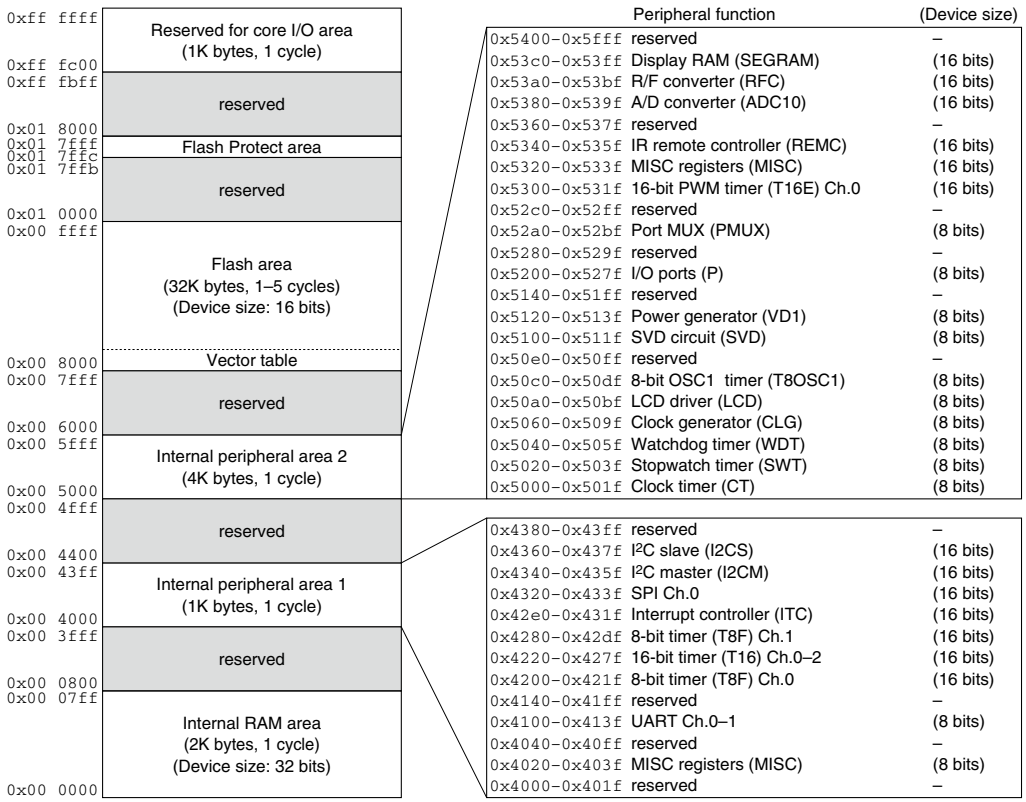


Figure 3.3 S1C17621 Memory Map

3.1 Bus Cycle

The CPU operates with CCLK as the operating clock. For CCLK, see "Controlling the CPU Core Clock (CCLK)" in the "Clock Generator (CLG)" chapter.
The period between a CCLK rising edge and the next rising edge is assumed to be one CCLK (= one bus cycle). As shown in Figures 3.1 to 3.3, the number of cycles required for one bus access depends on the peripheral or memory module. Furthermore, the number of bus accesses depends on the CPU instruction (access size) and device size.

Table 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
  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.

Number of bus cycles calculation example

Number of bus cycles when the CPU accesses the display RAM area (eight-bit device, set to two access cycles) by a 16-bit read or write instruction.

2 [cycles] × 2 [bus accesses] = 4 [CCLK cycles]

3.1.1 Restrictions on Access Size

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

Flash memory

The Flash memory allows only 16-bit write instructions for programming. Reading data from the Flash memory has no such restriction.

Other 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 access cycles.

- When the CPU executes the instruction stored in the Flash area and accesses data in the Flash area or display RAM area
- When the CPU 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 S1C17624/604 contains a 128K-byte Flash memory (4K bytes × 32 sectors), the S1C17622/602 contains a 64K-byte Flash memory (4K bytes × 16 sectors), and the S1C17621 contains a 32K-byte Flash memory (4K bytes × 8 sectors) for storing application programs and data.

Table 3.2.1.1 Embedded Flash Memory

	S1C17624/604	S1C17622/602	S1C17621
Flash memory capacity	128K bytes	64K bytes	32K bytes
Address range	0x8000 to 0x27fff	0x8000 to 0x17fff	0x8000 to 0xffff

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 Flash memory. The vector table base address can be modified with the MISC_TTBRL/MISC_TTBRH registers.
The Flash memory can be read in 1 to 5 cycles.

3.2.2 Flash Programming

The S1C17624/604/622/602/621 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 (S5U1C17001H). Furthermore, the S1C17624/604/622/602/621 supports self-programming by the application program. The Flash memory can be programmed in 16-bit units. The Flash memory supports two erase methods, chip erase and sector erase.For the Flash programming using the debugger, see the "S5U1C17001C Manual" included in the S1C17 Family C Compiler Package. For the self-programming controlled by the user program, see the "Self-Programming (FLS) Application Notes" for the S1C17624/604/622/602/621.

Note: The debugger supports chip erase only and does not allow erasing in sector units.

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. 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.

Flash Protect Bits (S1C17624/604)

Address	Bit	Function	Setting				Init.	R/W	Remarks
0x27ffc	D15–8	reserved	–				–	–	
(16 bits)	D7	Flash write-protect bit for 0x24000–0x27fff	1	Writable	0	Protected	1	R/W	
	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	

Address	Bit	Function	Setting				Init.	R/W	Remarks
0x27ffe	D15–8	reserved	–				–	–	
(16 bits)	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.

Flash Protect Bits (S1C17622)

Address	Bit	Function	Setting				Init.	R/W	Remarks
0x27ffc (16 bits)	D15–4	reserved	–				–	–	
	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	

Address	Bit	Function	Setting				Init.	R/W	Remarks
0x27ffe (16 bits)	D15–4	reserved	–				–	–	
	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.

Flash Protect Bits (S1C17602)

Address	Bit	Function	Setting				Init.	R/W	Remarks
0x17ffc (16 bits)	D15–4	reserved	–				–	–	
	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	

Address	Bit	Function	Setting				Init.	R/W	Remarks
0x17ffe (16 bits)	D15–4	reserved	–				–	–	
	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.

Flash Protect Bits (S1C17621)

Address	Bit	Function	Setting				Init.	R/W	Remarks
0x17ffc (16 bits)	D15–2	reserved	–				–	–	
	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	

Address	Bit	Function	Setting				Init.	R/W	Remarks
0x17ffe (16 bits)	D15–2	reserved	–				–	–	
	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: • 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 (S1C17624/604/622) or 0x17ffe (S1C17602/621) to 1. If it is set to 0, the program cannot be booted.

3.2.4 Access Control for the Flash Controller

The S1C17624/604/622/602/621 on-chip Flash memory is accessed via the exclusive Flash controller. A MISC register is used to set the access condition for the Flash controller.

Setting number of read access cycles for the Flash controller

In order to read data from the Flash memory properly, set the appropriate number of read access cycles according to the CCLK frequency using the FLCYC[2:0]/MISC_FL register.

FLASHC Control Register (MISC_FL)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
FLASHC Control Register (MISC_FL)	**0x5320** (16 bits)	D15–10	–	reserved	–		–	–	0 when being read.
		D9–8	–	reserved	–		0x3	–	
		D7–3	–	reserved	–		–	–	0 when being read.
		D2–0	**FLCYC[2:0]**	FLASHC read access cycle	FLCYC[2:0]	Read cycle	0x3	R/W	
					0x7–0x5	reserved			
					0x4	1 cycle			
					0x3	5 cycles			
					0x2	4 cycles			
					0x1	3 cycles			
					0x0	2 cycles			

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

Sets the number of read access cycles for the Flash controller.

Table 3.2.4.1 Setting Read Access Cycles for the Flash Controller

FLCYC[2:0]	Number of read access cycles	CCLK frequency
0x7–0x5	Reserved	–
0x4	1 cycle	8.2 MHz max.
0x3	5 cycles	
0x2	4 cycles	
0x1	3 cycles	
0x0	2 cycles	

(Default: 0x3)

Notes: • Be sure to avoid setting a number of read access cycles that exceeds the maximum allowable CCLK frequency, as it may cause a malfunction.

• For maximum performance, set FLCYC[2:0] to 0x4.

3.3 Internal RAM Area

3.3.1 Embedded RAM

The S1C17624/604 contains an 8K-byte RAM, the S1C17622/602 contains a 4K-byte RAM, and the S1C17621 contains a 2K-byte RAM.

Table 3.3.1.1 Embedded RAM

	S1C17624/604	S1C17622/602	S1C17621
RAM capacity	8K bytes	4K bytes	2K bytes
Address range	0x0 to 0x1fff	0x0 to 0xfff	0x0 to 0x7ff

The RAM is accessed in one cycle for reading/writing and 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 (S1C17624/604: 0x1fc0–0x1fff, S1C17602: 0xfc0–0xfff) 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 S1C17624/604 enables the RAM size used to apply restrictions to 8KB, 4KB, or 2KB. The S1C17622 enables the RAM size used to apply restrictions to 4KB or 2KB. For example, when using the S1C17624/604/622 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 Select Register (MISC_IRAMSZ)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
IRAM Size Select Register (MISC_IRAMSZ) (S1C17624/604)	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	–	reserved	–	–	–	0x1 when being read.
		D3	–	reserved	–	–	–	0 when being read.
		D2–0	IRAMSZ[2:0]	IRAM size select	IRAMSZ[2:0]: Size 0x3: 2KB 0x2: 4KB 0x1: 8KB Other: reserved	0x1	R/W	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
IRAM Size Select Register (MISC_IRAMSZ) (S1C17622)	**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	–	reserved	–	–	–	0x1 when being read.
		D3	–	reserved	–	–	–	0 when being read.
		D2–0	**IRAMSZ[2:0]**	IRAM size select	IRAMSZ[2:0] Size 0x3 2KB 0x2 4KB Other reserved	0x1	R/W	
IRAM Size Select Register (MISC_IRAMSZ) (S1C17602)	**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	–	reserved	–	–	–	0x2 when being read.
		D3	–	reserved	–	–	–	0 when being read.
		D2–0	**IRAMSZ[2:0]**	IRAM size select	IRAMSZ[2:0] Size 0x7–0x0 reserved	0x2	R/W	
IRAM Size Select Register (MISC_IRAMSZ) (S1C17621)	**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	–	reserved	–	–	–	0x2 when being read.
		D3	–	reserved	–	–	–	0 when being read.
		D2–0	**IRAMSZ[2:0]**	IRAM size select	IRAMSZ[2:0] Size 0x7–0x0 reserved	0x2	R/W	

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

Selects the internal RAM size used.

Table 3.3.1.2 Selecting Internal RAM Size

IRAMSZ[2:0]	Internal RAM size			
	S1C17624/604	S1C17622	S1C17602	S1C17621
0x3	2KB	2KB	Reserved	Reserved
0x2	4KB	4KB	Reserved (default)	Reserved (default)
0x1	8KB (default)	Reserved (default)	Reserved	Reserved
Other	Reserved	Reserved	Reserved	Reserved

Notes: • 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.

• In the S1C17602/621, do not alter the IRAMSZ[2:0]/MISC_IRAMSZ register setting from the default value.

3.4 Display RAM Area

The display RAM for the on-chip LCD driver is located in the 56-byte area (S1C17624/622) or 40-byte area (S1C17604/602/621) beginning with address 0x53c0 in the internal peripheral area. The display RAM is accessed in one cycle as a 16-bit device. It can be used as a general-purpose RAM when it is not used for display. See the "Display Memory" section in the "LCD Driver (LCD)" chapter for specific information on the display memory.

3.5 Internal Peripheral Area

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

3.5.1 Internal Peripheral Area 1 (0x4000–)

The internal peripheral area 1 beginning with address 0x4000 contains the I/O memory for the peripheral functions listed below and this area can be accessed in one cycle.

- MISC register (MISC, 8-bit device)
- UART (UART, 8-bit device)

- 8-bit timers (T8F, 16-bit device)
- 16-bit timers (T16, 16-bit device)
- Interrupt controller (ITC, 16-bit device)
- SPI (SPI, 16-bit device)
- I²C master (I2CM, 16-bit device)
- I²C slave (I2CS, 16-bit device)

3.5.2 Internal Peripheral Area 2 (0x5000–)

The internal peripheral area 2 beginning with address 0x5000 contains the I/O memory for the peripheral functions listed below and this area can be accessed in one cycle.

- Clock timer (CT, 8-bit device)
- Stopwatch timer (SWT, 8-bit device)
- Watchdog timer (WDT, 8-bit device)
- Clock generator (CLG, 8-bit device)
- LCD driver (LCD, 8-bit device)
- 8-bit OSC1 timer (T8OSC1, 8-bit device)
- SVD circuit (SVD, 8-bit device)
- Power generator (VD1, 8-bit device)
- Real-time clock (RTC, 8-bit device) Available only in the S1C17624/604
- I/O port & port MUX (P, 8-bit device)
- 16-bit PWM timer (T16E, 16-bit device)
- MISC register (MISC, 16-bit device)
- IR remote controller (REMC, 16-bit device)
- A/D converter (ADC10, 16-bit device)
- R/F converter (RFC, 16-bit device)
- Display RAM (SEGRAM, 16-bit device)
- 16-bit PWM timers (T16A2, 16-bit device) Available only in the S1C17624/604

3.6 S1C17 Core I/O Area

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

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

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

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

4 Power Supply

4.1 Power Supply Voltage (V_{DD})

The operating voltage range of the S1C17624/604/622/602/621 is as follows:

For normal operation: V_{DD} = 1.8 V to 3.6 V

For Flash programming: V_{DD} = 2.7 V to 3.6 V

Supply a voltage within the range to the V_{DD} pins with the V_{SS} pins as the GND level. The S1C17624/604/622/602/621 provides two or more V_{DD} and V_{SS} pins. Do not leave any power supply pins open and be sure to connect them to + power source and GND.

4.2 Analog Power Supply Voltage (AV_{DD})

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

$AV_{DD} = V_{DD}$ = 1.8 V to 3.6 V (V_{SS} = GND)

Note: Be sure to supply the same voltage as V_{DD} to the AV_{DD} pin even if the analog circuit is not used.

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.

4.3 Internal Power Supply Circuit

The S1C17624/604/622/602/621 has a built-in power supply circuit shown in Figure 4.3.1 to generate all the power voltages required for the internal circuits. The power supply module consists of two circuits.

Table 4.3.1 Power Supply Circuit

Circuit	Power supply circuit	Output voltage
Oscillator and internal logic circuits	Internal logic voltage regulator	V_{D1}
LCD driver	LCD system voltage regulator	V_{C1} to V_{C3}

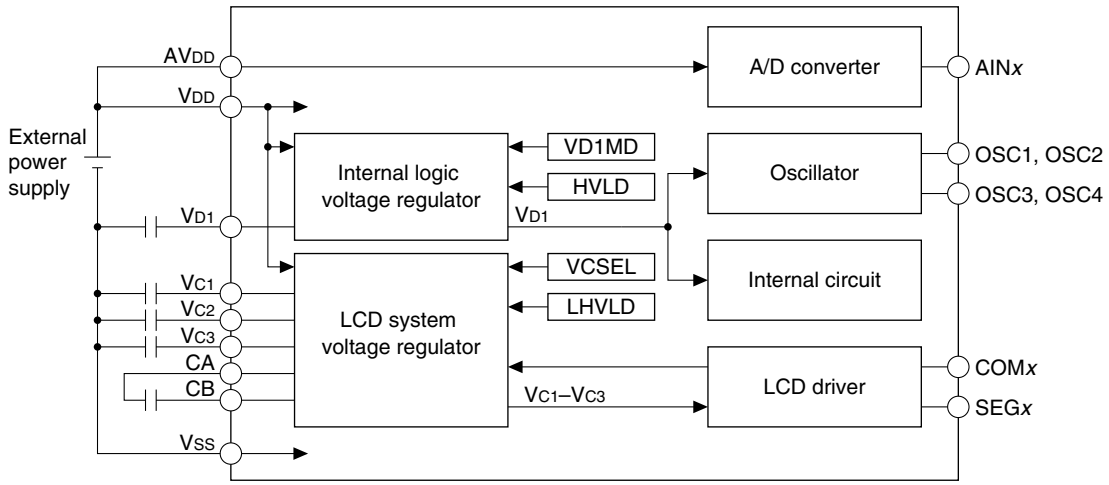


Figure 4.3.1 Configuration of Power Supply Circuit

Note: Be sure to avoid using the V_{D1} and V_{C1}–V_{C3} pin outputs to drive external circuits.

Internal logic voltage regulator

The internal logic voltage regulator generates the V_{D1} operating voltage for the internal logic circuits and oscillators. The V_{D1} voltage value can be switched in the program; set it to 1.8 V for normal operation and 2.5 V for Flash programming.

LCD system voltage regulator

The LCD system voltage regulator generates the 1/3-bias LCD drive voltages V_{C1}, V_{C2}, and V_{C3}. In the S1C17624/604/622/602/621, the LCD drive voltage is supplied to the built-in LCD driver that drives the LCD panel connected to the SEG and COM pins. The reference voltage (V_{C1} or V_{C2}) for the LCD voltage booster/reducer should be selected using VCSEL/LCD_VREG register according to the power supply voltage V_{DD}. As compared to the V_{C1} reference voltage, the V_{C2} reference voltage is lower in current consumption. For the V_{C1} to V_{C3} voltage values, see the "Electrical Characteristics" chapter.

Table 4.3.2 Power Supply Voltage V_{DD} and VCSEL Settings

Power supply voltage V_{DD}	VCSEL setting	Reference voltage
1.8 to 3.6 V	0	V_{C1}
2.5 to 3.6 V	1	V_{C2}

(Default: 0)

Note: The V_{C1} to V_{C3} voltages cannot be obtained correctly if VCSEL is set to 1 when V_{DD} is 2.5 V or less.

4.4 Controlling the Power Supply Circuit

In order to generate the internal operating voltage properly according to the power supply voltage and operating mode, or to reduce current consumption, the power supply circuit is designed to be controlled with software.

Switching the operating mode

The S1C17624/604/622/602/621 has two kinds of operating modes.

1. Normal operating mode
   This mode is provided for running the application program.
   V_{DD} = 1.8 to 3.6 V, internal operating voltage V_{D1} = 1.8 V

2. Flash erase/programming mode
   This mode is provided for erasing and programming the Flash memory.
   V_{DD} = 2.7 to 3.6 V, internal operating voltage V_{D1} = 2.5 V

The V_{D1} voltage value must be switched according to the operating mode as shown above using VD1MD/VD1_CTL register. Normally set VD1MD to 0 (V_{D1} = 1.8 V, default setting). It should be set to 1 before erasing/programming the Flash memory.

Note: When the V_{D1} voltage is switched, the V_{D1} voltage requires 5 ms (max.) to stabilize. Flash memory programming should be started after the stabilization time has elapsed.

Controlling the LCD power source

The LCD drive voltages V_{C1} to V_{C3} will be supplied to the LCD driver by setting the DSPC[1:0]/LCD_DCTL register to a value other than 0x0 (display off).
When the internal LCD driver is not used, the LCD system voltage regulator should be turned off (DSPC[1:0] = 0) to reduce current consumption.

Power control bit settings

Table 4.4.1 lists the power control bit settings in different operating conditions.

Table 4.4.1 Power Control Bit Settings

Condition			Control bits		
Operating mode	**V_{DD}**	**LCD driver**	**VD1MD**	**VCSEL**	**DSPC[1:0]**
Normal operating mode	1.8 to 3.6 V	Used	0	0	Other than 0x0
	2.5 to 3.6 V	Used	0	1	Other than 0x0
	1.8 to 3.6 V	Not used	0	0	0x0

Condition			Control bits		
Operating mode	V_{DD}	LCD driver	VD1MD	VCSEL	DSPC[1:0]
Flash erase/ programming mode 1	1.8 to 2.7 V	–	(Not supported)		
	2.7 to 3.6 V	Used	1	1	Other than 0x0
		Not used	1	0	0x0

For the DSPC[1:0] settings, see "LCD Display Control Register (LCD_DCTL)" in the "LCD Driver (LCD)" chapter.

4.5 Heavy Load Protection Function

In order to ensure a stable circuit behavior and LCD display quality even if the power supply voltage fluctuates due to driving an external load, the internal logic voltage regulator and the LCD system voltage regulator have a heavy load protection function.

The internal logic voltage regulator enters heavy load protection mode by writing 1 to the HVLD/VD1_CTL register and it ensures stable V_{D1} output.

V_{D1} may become unstable in the operations shown below and in other conditions. If the IC operations are unstable due to these conditions during evaluation, set the internal logic voltage regulator to heavy load protection mode before starting the operations.

- When driving a diode or buzzer in which a large current flows using a port output (Maintain the regulator in heavy load protection mode while the port is driving the load.)
- When switching the system clock from the high-speed clock to the low-speed clock and vice versa (Set the regulator in heavy load protection mode immediately before switching the clock and maintain it for several 10 µs after the switching has completed.)
- When turning the high-speed oscillator (OSC3, IOSC) on (Set the regulator in heavy load protection mode immediately before turning the oscillator on and maintain it until the oscillation stabilization wait time has elapsed.)
- When placing/releasing the system into/from HALT/SLEEP mode at frequent intervals (Maintain the regulator in heavy load protection mode while the processing is being repeated.)

The LCD system voltage regulator enters heavy load protection mode by writing 1 to the LHVLD/LCD_VREG register and it ensures stable V_{C1}–V_{C3} outputs. Use the heavy load protection function when the LCD display has inconsistencies in density.

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

4.6 Control Register Details

Table 4.6.1 List of Power Control Registers

Address	Register name		Function
0x5120	VD1_CTL	V_{D1} Control Register	Controls the V_{D1} voltage and heavy load protection mode.
0x50a3	LCD_VREG	LCD Voltage Regulator Control Register	Controls the LCD drive voltage regulator.

The power control registers are described in detail below. These are all 8-bit registers.

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)	0x5120 (8 bits)	D7–6	–	reserved	–	–	–	0 when being read.
		D5	HVLD	V_{D1} heavy load protection mode	1 On 0 Off	0	R/W	
		D4–1	–	reserved	–	–	–	0 when being read.
		D0	VD1MD	Flash erase/programming mode	1 Flash (2.5 V) 0 Norm.(1.8 V)	0	R/W	

D[7:6] Reserved

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

Sets the internal logic voltage regulator into heavy load protection mode.
1 (R/W): Heavy load protection On
0 (R/W): Heavy load protection Off (default)

The internal logic voltage regulator enters heavy load protection mode by writing 1 to HVLD and it ensures stable V_{D1} output. Use the heavy load protection function when a heavy load such as a lamp or buzzer is driven with a port output (see Section 4.5). Current consumption increases in heavy load protection mode, therefore do not set if unnecessary.

D[4:1] Reserved

D0 VD1MD: Flash Erase/Programming Mode Bit

Selects the V_{D1} internal operating voltage value (operating mode).
1 (R/W): V_{D1} = 2.5 V, Flash erase/programming mode
0 (R/W): V_{D1} = 1.8 V, Normal operating mode (default)

Normally set VD1MD to 0 (V_{D1} = 1.8 V, default setting). It should be set to 1 before erasing/programming the Flash memory.

Note: When the V_{D1} voltage is switched, the V_{D1} voltage requires 5 ms (max.) to stabilize. Flash memory programming should be started after the stabilization time has elapsed.

LCD Voltage Regulator Control Register (LCD_VREG)

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

D[7:5] Reserved

D4 LHVLD: LCD Heavy Load Protection Mode Bit

Sets the LCD system voltage regulator into heavy load protection mode.
1 (R/W): Heavy load protection On
0 (R/W): Heavy load protection Off (default)

The LCD system voltage regulator enters heavy load protection mode by writing 1 to LHVLD and it ensures stable V_{C1}–V_{C3} outputs. Use the heavy load protection function when the LCD display has inconsistencies in density. Current consumption increases in heavy load protection mode, therefore do not set if unnecessary.

D[3:1] Reserved

D0 VCSEL: V_C Reference Voltage Select Bit

Selects the reference voltage for the LCD voltage booster/reducer according to the power supply voltage V_{DD}.
1 (R/W): V_{C2} is used as the reference voltage
0 (R/W): V_{C1} is used as the reference voltage (default)

As compared to the V_{C1} reference voltage, the V_{C2} reference voltage is lower in current consumption. Set VCSEL to 0 when V_{DD} is less than 2.5 V.

Note: The V_{C1} to V_{C3} voltages cannot be obtained correctly if VCSEL is set to 1 when V_{DD} is less than 2.5 V.

5 Initial Reset

5.1 Initial Reset Sources

The S1C17624/604/622/602/621 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.

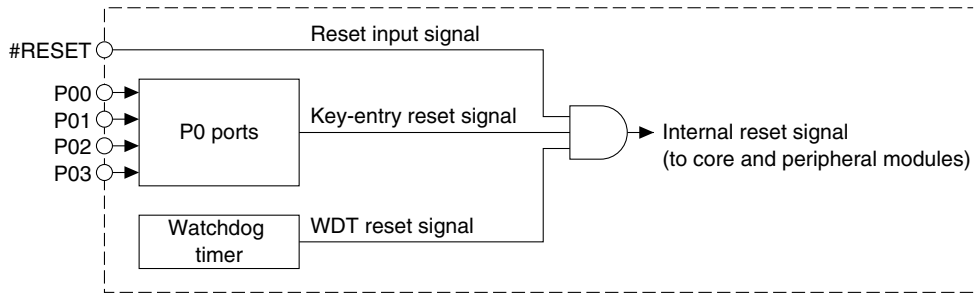


Figure 5.1.1 Configuration of Initial Reset Circuit

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 S1C17624/604/622/602/621 enters initial reset state. In order to initialize the S1C17624/604/622/602/621 for sure, the #RESET pin must be held at low for more than the prescribed time (see "Input/Output Pin Characteristics" in the "Electrical Characteristics" chapter) after the power supply voltage is supplied.
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 a CMOS Schmitt level input port with a built-in 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 S1C17624/604/622/602/621 has a built-in watchdog timer to detect runaway of the CPU. The watchdog timer overflows if it is not reset with software (due to CPU runaway) in four-second cycles. The overflow signal can generate either NMI or reset. Write 1 to the WDTMD/WDT_ST register to generate reset (NMI occurs when WDTMD = 0).
For details of the watchdog timer, see the "Watchdog Timer (WDT)" chapter.

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

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

5.2 Initial Reset Sequence

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

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

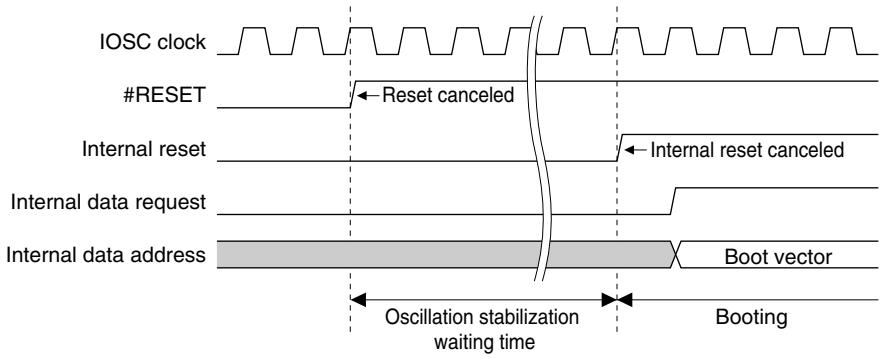


Figure 5.2.1 Operation Sequence Following Cancellation of Initial Reset

5.3 Initial Settings After an Initial Reset

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

The internal RAM 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 20 maskable interrupt systems (for 23 interrupt sources listed below).
  1. P0 port (P00–P07) interrupt (8 types)
  2. P1 port (P10–P17) interrupt (8 types)
  3. Stopwatch timer (SWT) interrupt (3 types)
  4. Clock timer (CT) interrupt (4 types)
  5. Real-time clock (RTC) interrupt (1 type) (available only for S1C17624/604)
  6. 8-bit OSC1 timer (T8OSC1) interrupt (1 type)
  7. Supply voltage detector (SVD) interrupt (1 type)
  8. LCD driver (LCD) interrupt (1 type)
  9. 16-bit PWM timer (T16E) Ch.0 interrupt (2 types)
  10. 16-bit PWM timer (T16A2) Ch.0 interrupt (6 types) (available only for S1C17624/604)
  11. 16-bit PWM timer (T16A2) Ch.1 interrupt (6 types) (available only for S1C17624/604)
  12. 8-bit timer (T8F) Ch.0 & Ch.1 interrupt (2 types)
  13. 16-bit timer (T16) Ch.0 interrupt (1 type)
  14. 16-bit timer (T16) Ch.1 interrupt (1 type)
  15. 16-bit timer (T16) Ch.2 interrupt (1 type)
  16. UART Ch.0 interrupt (3 types)
  17. UART Ch.1 interrupt (3 types)
  18. IR remote controller (REMC) interrupt (3 types)
  19. SPI Ch.0 interrupt (2 types)
  20. I^2C master (I2CM) interrupt (2 types)
  21. I^2C slave (I2CS) interrupt (3 types)
  22. A/D converter (ADC10) interrupt (2 types)
  23. R/F converter (RFC) interrupt (5 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.

Notes:
- After the S1C17622 power is turned on, write the specified values to the addresses shown below before executing the interrupt enable (`ei`) instruction.
  1. Address 0x5140 = 0x01 (in 8-bit access)
  2. Address 0x5141 = 0x06 (in 8-bit access)
- After the S1C17624/604 power is turned on, clear the RTC interrupt flag and then disable RTC interrupts as shown below before executing the interrupt enable (`ei`) instruction.
  1. RTCIRQ/RTC_INTSTAT register = 1 (clear RTC interrupt flag)
  2. RTCIEN/RTC_INTMODE register = 0 (disable RTC interrupt)

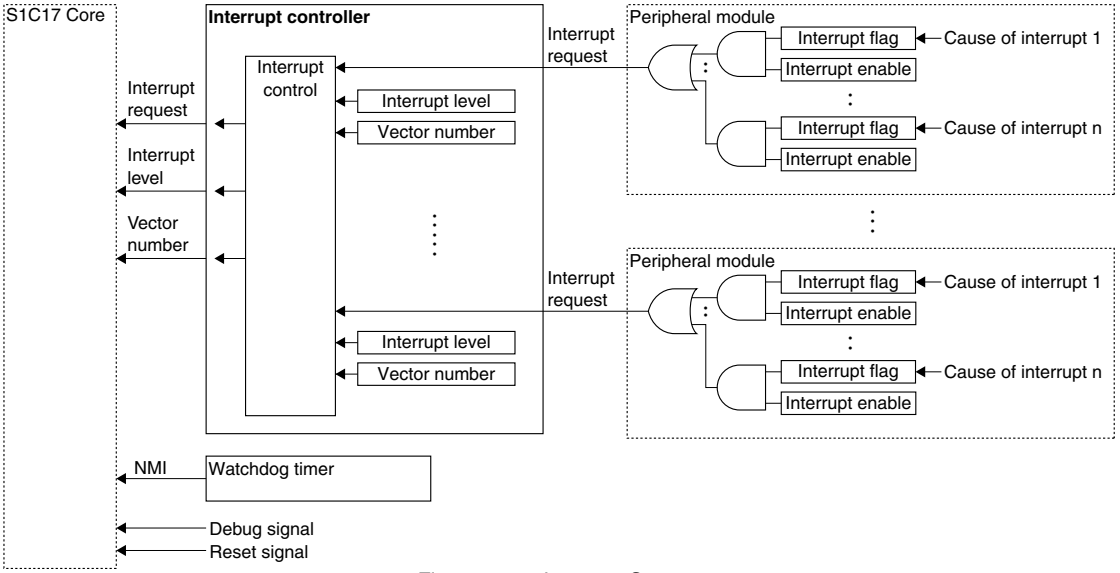


Figure 6.1.1 Interrupt System

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 S1C17624/604/622/602/621.

Table 6.2.1 Vector Table

Vector No. Software interrupt No.	Vector address	Hardware interrupt name	Cause of hardware interrupt	Priority
0 (0x00)	TTBR + 0x00	Reset	• Low input to the #RESET pin • Watchdog timer overflow *2	1
1 (0x01)	TTBR + 0x04	Address misaligned interrupt	Memory access instruction	2
–	(0xfffc00)	Debugging interrupt	`brk` instruction, etc.	3
2 (0x02)	TTBR + 0x08	NMI	Watchdog timer overflow *2	4
3 (0x03)	TTBR + 0x0c	Reserved for C compiler	–	–
4 (0x04)	TTBR + 0x10	P0 port interrupt	P00–P07 port inputs	High *1
5 (0x05)	TTBR + 0x14	P1 port interrupt	P10–P17 port inputs	↑
6 (0x06)	TTBR + 0x18	Stopwatch timer (SWT) interrupt	• 100 Hz timer signal • 10 Hz timer signal • 1 Hz timer signal	
7 (0x07)	TTBR + 0x1c	Clock timer (CT) interrupt	• 32 Hz timer signal • 8 Hz timer signal • 2 Hz timer signal • 1 Hz timer signal	
		Real-time clock (RTC) interrupt (S1C17624/604)	1/64 second, 1 second, 1 minute, or 1 hour count up	
8 (0x08)	TTBR + 0x20	8-bit OSC1 timer (T8OSC1) interrupt	Compare match	
9 (0x09)	TTBR + 0x24	Supply voltage detector (SVD) interrupt	Low supply voltage detected	
10 (0x0a)	TTBR + 0x28	LCD driver (LCD) interrupt	Frame signal	
		16-bit PWM timer (T16A2) Ch.0 interrupt (S1C17624/604)	• Compare A/B • Capture A/B • Capture A/B overwrite	
11 (0x0b)	TTBR + 0x2c	16-bit PWM timer (T16E) Ch.0 interrupt	• Compare A • Compare B	
12 (0x0c)	TTBR + 0x30	8-bit timer (T8F) Ch.0 & Ch.1 interrupt	Timer underflow	
13 (0x0d)	TTBR + 0x34	16-bit timer (T16) Ch. 0 interrupt	Timer underflow	
14 (0x0e)	TTBR + 0x38	16-bit timer (T16) Ch. 1 interrupt	Timer underflow	
15 (0x0f)	TTBR + 0x3c	16-bit timer (T16) Ch. 2 interrupt	Timer underflow	

Vector No. Software interrupt No.	Vector address	Hardware interrupt name	Cause of hardware interrupt	Priority
16 (0x10)	TTBR + 0x40	UART Ch.0 interrupt	• Transmit buffer empty • Receive buffer full • Receive error	
17 (0x11)	TTBR + 0x44	I^2C Slave (I2CS) interrupt	• Transmit buffer empty • Receive buffer full • Bus status	
		UART Ch.1 interrupt	• Transmit buffer empty • 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 (I2CM) interrupt	• Transmit buffer empty • Receive buffer full	
20 (0x14)	TTBR + 0x50	IR remote controller (REMC) interrupt	• Data length counter underflow • Input rising edge detected • Input falling edge detected	
21 (0x15)	TTBR + 0x54	16-bit PWM timer (T16A2) Ch.1 interrupt (S1C17624/604)	• Compare A/B • Capture A/B • Capture A/B overwrite	
22 (0x16)	TTBR + 0x58	A/D converter (ADC10) interrupt	• Conversion completion • Conversion result overwrite	
23 (0x17)	TTBR + 0x5c	R/F converter (RFC) interrupt	• Reference oscillation completion • Sensor A oscillation completion • Sensor B oscillation completion • Time base counter overflow error • Measurement counter overflow error	
24 (0x18) : 31 (0x1f)	TTBR + 0x60 : 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 23 are assigned to the maskable interrupts supported by the S1C17624/604/622/602/621.

Interrupts that share an interrupt vector

Interrupt vector numbers 7, 10, and 17 are shared with two different interrupt modules.
Interrupt vector 7: Clock timer (CT) and real-time clock (RTC) (S1C17624/604)
Interrupt vector 10: LCD driver (LCD) and 16-bit PWM timer (T16A2) Ch.0 (S1C17624/604)
Interrupt vector 17: I^2C slave (I2CS) and UART 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 S1C17624/604/622/602/621 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 (SWT) interrupt	ILV2[2:0] (D[2:0]/ITC_LV1 register)	0x4308
Clock timer (CT) interrupt / Real-time clock (RTC) interrupt (S1C17624/604)	ILV3[2:0] (D[10:8]/ITC_LV1 register)	0x4308
8-bit OSC1 timer (T8OSC1) interrupt	ILV4[2:0] (D[2:0]/ITC_LV2 register)	0x430a
Supply voltage detector (SVD) interrupt	ILV5[2:0] (D[10:8]/ITC_LV2 register)	0x430a
LCD driver (LCD) interrupt / 16-bit PWM timer (T16A2) Ch.0 interrupt (S1C17624/604)	ILV6[2:0] (D[2:0]/ITC_LV3 register)	0x430c
16-bit PWM timer (T16E) Ch.0 interrupt	ILV7[2:0] (D[10:8]/ITC_LV3 register)	0x430c
8-bit timer (T8F) Ch.0 & Ch.1 interrupt	ILV8[2:0] (D[2:0]/ITC_LV4 register)	0x430e
16-bit timer (T16) Ch.0 interrupt	ILV9[2:0] (D[10:8]/ITC_LV4 register)	0x430e
16-bit timer (T16) Ch.1 interrupt	ILV10[2:0] (D[2:0]/ITC_LV5 register)	0x4310
16-bit timer (T16) Ch.2 interrupt	ILV11[2:0] (D[10:8]/ITC_LV5 register)	0x4310
UART Ch.0 interrupt	ILV12[2:0] (D[2:0]/ITC_LV6 register)	0x4312
I^2C slave (I2CS) interrupt / 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 (I2CM) interrupt	ILV15[2:0] (D[10:8]/ITC_LV7 register)	0x4314
IR remote controller (REMC) interrupt	ILV16[2:0] (D[2:0]/ITC_LV8 register)	0x4316
16-bit PWM timer (T16A2) Ch.1 interrupt (S1C17624/604)	ILV17[2:0] (D[10:8]/ITC_LV8 register)	0x4316
A/D converter (ADC10) interrupt	ILV18[2:0] (D[2:0]/ITC_LV9 register)	0x4318
R/F converter (RFC) interrupt	ILV19[2:0] (D[10:8]/ITC_LV9 register)	0x4318

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 S1C17624/604/622/602/621, 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/RTC interrupt levels.
0x430a	ITC_LV2	Interrupt Level Setup Register 2	Sets the T8OSC1 and SVD interrupt levels.
0x430c	ITC_LV3	Interrupt Level Setup Register 3	Sets the LCD/T16A2 Ch.0 and T16E Ch.0 interrupt levels.
0x430e	ITC_LV4	Interrupt Level Setup Register 4	Sets the T8F Ch.0&1 and T16 Ch.0 interrupt levels.
0x4310	ITC_LV5	Interrupt Level Setup Register 5	Sets the T16 Ch.1 and T16 Ch.2 interrupt levels.
0x4312	ITC_LV6	Interrupt Level Setup Register 6	Sets the UART Ch.0 and I2CS/UART 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 and T16A2 Ch.1 interrupt levels.
0x4318	ITC_LV9	Interrupt Level Setup Register 9	Sets the ADC10 and RFC interrupt levels.

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

Notes: • After the S1C17622 power is turned on, write the specified values to the addresses shown below before executing the interrupt enable (`ei`) instruction.
1. Address 0x5140 = 0x01 (in 8-bit access)
2. Address 0x5141 = 0x06 (in 8-bit access)

• After the S1C17624/604 power is turned on, clear the RTC interrupt flag and then disable RTC interrupts as shown below before executing the interrupt enable (`ei`) instruction.
1. RTCIRQ/RTC_INTSTAT register = 1 (clear RTC interrupt flag)
2. RTCIEN/RTC_INTMODE register = 0 (disable RTC interrupt)

- When data is written to the ITC_LV0 to ITC_LV9 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 | 0x4318 (16 bits)	D15–11	–	reserved	–	–	–	0 when being read.
		D10–8	ILV*n*[2:0]	*INTn* (1, 3, ... 19) interrupt level	0 to 7	0x0	R/W	
		D7–3	–	reserved	–	–	–	0 when being read.
		D2–0	ILV*n*[2:0]	*INTn* (0, 2, ... 18) 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* = 0–19)

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 0x4318) 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 (SWT) interrupt
	ILV3[2:0] (D[10:8])	Clock timer (CT) interrupt / Real-time clock (RTC) interrupt (S1C17624/604)
ITC_LV2(0x430a)	ILV4[2:0] (D[2:0])	8-bit OSC1 timer (T8OSC1) interrupt
	ILV5[2:0] (D[10:8])	Supply voltage detector (SVD) interrupt
ITC_LV3(0x430c)	ILV6[2:0] (D[2:0])	LCD driver (LCD) interrupt / 16-bit PWM timer (T16A2) Ch.0 interrupt (S1C17624/604)
	ILV7[2:0] (D[10:8])	16-bit PWM timer (T16E) Ch.0 interrupt
ITC_LV4(0x430e)	ILV8[2:0] (D[2:0])	8-bit timer (T8F) Ch.0 & Ch.1 interrupt
	ILV9[2:0] (D[10:8])	16-bit timer (T16) Ch.0 interrupt
ITC_LV5(0x4310)	ILV10[2:0] (D[2:0])	16-bit timer (T16) Ch.1 interrupt
	ILV11[2:0] (D[10:8])	16-bit timer (T16) Ch.2 interrupt
ITC_LV6(0x4312)	ILV12[2:0] (D[2:0])	UART Ch.0 interrupt
	ILV13[2:0] (D[10:8])	I^2C slave (I2CS) interrupt / 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 (I2CM) interrupt
ITC_LV8(0x4316)	ILV16[2:0] (D[2:0])	IR remote controller (REMC) interrupt
	ILV17[2:0] (D[10:8])	16-bit PWM timer (T16A2) Ch.1 interrupt (S1C17624/604)
ITC_LV9(0x4318)	ILV18[2:0] (D[2:0])	A/D converter (ADC10) interrupt
	ILV19[2:0] (D[10:8])	R/F converter (RFC) interrupt

7 Clock Generator (CLG)

7.1 CLG Module Overview

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

- Generates the operating clocks with the built-in oscillators.
  - IOSC oscillator circuit: 2.7 MHz (typ.)
  - OSC3 oscillator circuit: 8.2 MHz (max.) crystal or ceramic oscillator circuit
    Supports an external clock input.
  - OSC1 oscillator circuit: 32.768 kHz (typ.) crystal oscillator circuit
- 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).
- Controls two clock outputs to external devices.

Figure 7.1.1 shows the clock system and CLG module configuration.

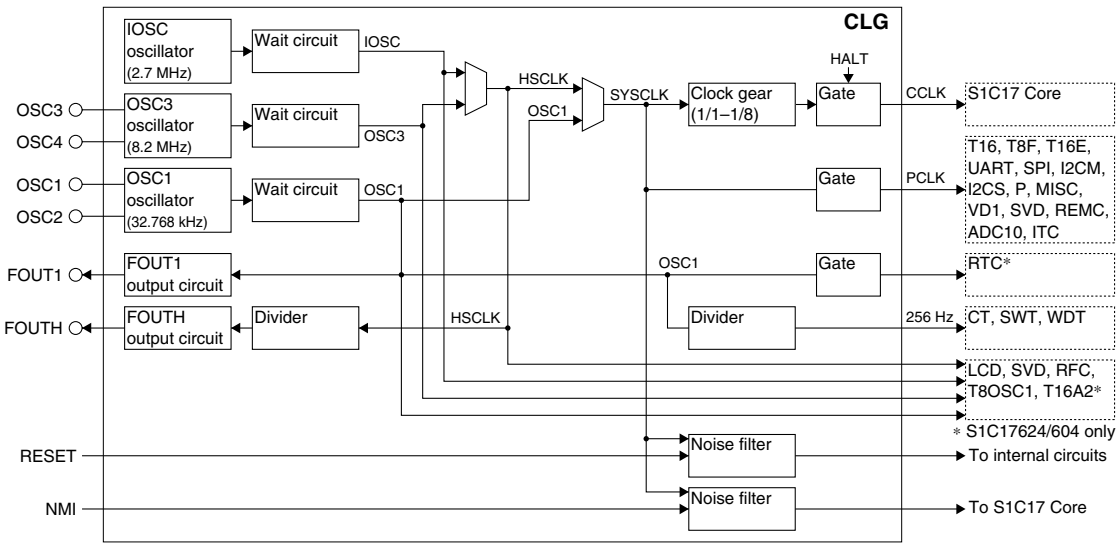


Figure 7.1.1 CLG Module Configuration

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

7.2 CLG Input/Output Pins

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

Table 7.2.1 List of CLG Pins

Pin name	I/O	Qty	Function
OSC1	I	1	OSC1 oscillator input pin Connect a crystal resonator (32.768 kHz) and a gate capacitor.
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. 8.2 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. 8.2 MHz), a feedback resistor, and a drain capacitor.
FOUT1	O	1	FOUT1 clock output pin Outputs the OSC1 clock.
FOUTH	O	1	FOUTH clock output pin Outputs a divided IOSC/OSC3 clock.

The CLG output pins (FOUT1, FOUTH) are shared with I/O ports and are initially set as general purpose I/O port pins. The pin functions must be switched using the port function select bits to use the general purpose I/O port pins as the CLG output pins. For detailed information on pin function switching, see the "I/O Ports (P)" chapter.

7.3 Oscillators

The CLG module contains three internal oscillator circuits (IOSC, OSC3, and OSC1). The IOSC and OSC3 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 IOSC clock is selected as the system clock after an initial reset. Oscillator on/off switching and system clock selection (from IOSC, OSC3 and OSC1) are controlled with software.

7.3.1 IOSC Oscillator

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

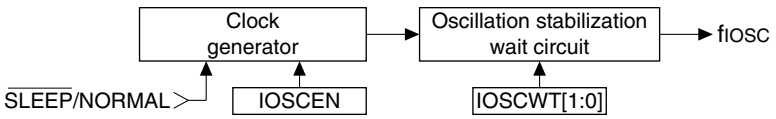


Figure 7.3.1.1 IOSC Oscillator Circuit

IOSC oscillation on/off

The IOSC oscillator stops oscillating when IOSCEN/OSC_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.

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. Figure 7.3.1.2 shows the relationship between the oscillation start time and the oscillation stabilization wait time.

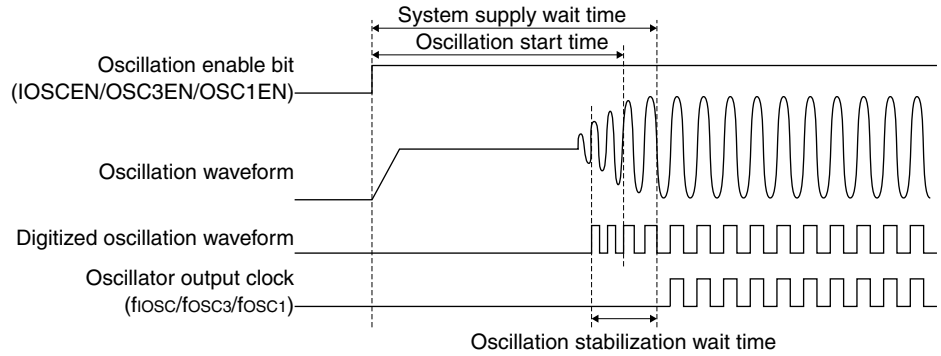


Figure 7.3.1.2 Oscillation Start Time and Oscillation Stabilization Wait Time

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

Table 7.3.1.1 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 V_{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

7.3.2 OSC3 Oscillator

The OSC3 oscillator is a high-precision, high-speed oscillator circuit that uses either a crystal resonator or a ceramic resonator. It can be switched for use with the IOSC oscillator. Figure 7.3.2.1 shows the OSC3 oscillator configuration.

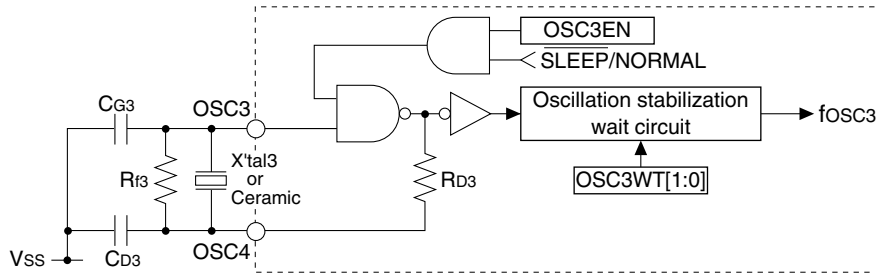


Figure 7.3.2.1 OSC3 Oscillator Circuit

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}.

OSC3 oscillation on/off

The OSC3 oscillator circuit stops oscillating when OSC3EN/OSC_CTL register is set to 0 and starts oscillating when set to 1. The OSC3 oscillator circuit stops oscillating in SLEEP mode.

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. The OSC3 clock is not supplied to the system until the time set for this circuit has elapsed. Use OSC3WT[1:0]/OSC_CTL register to select one of four oscillation stabilization wait times. For the oscillation start time, see the "Electrical Characteristics" chapter.

Table 7.3.2.1 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.

External OSC3 clock input

An external clock can be used as the OSC3 clock instead of the internal OSC3 clock described above. In this case, leave the OSC4 pin open and input a clock via the OSC3 pin. For the input clock requirements, see the "Electrical Characteristics" chapter.

7.3.3 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 8-bit OSC1 timer) and an operation clock for the LCD driver, R/F converter, and supply voltage detector. It can be used as the system clock instead of the IOSC or OSC3 clock to reduce power consumption when no high-speed processing is required. The S1C17624/604 also uses OSC1 as the clock source for the RTC.
Figures 7.3.3.1 and 7.3.3.2 show the OSC1 oscillator configurations.

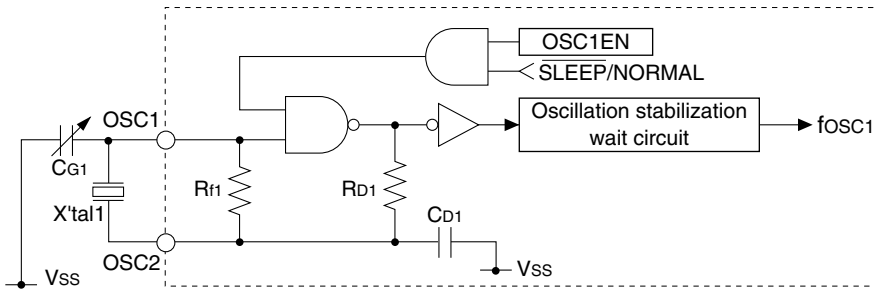


Figure 7.3.3.1 OSC1 Oscillator Circuit (S1C17622/602/621)

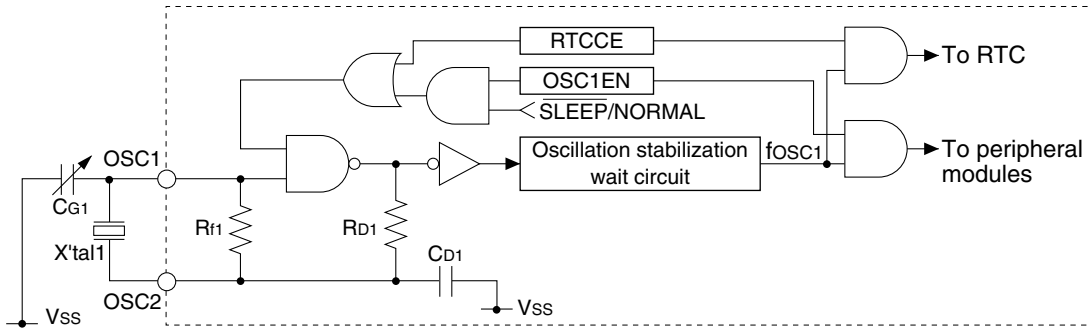


Figure 7.3.3.2 OSC1 Oscillator Circuit (S1C17624/604)

A crystal resonator (X'tal1, typ. 32.768 kHz) should be connected between the OSC1 and OSC2 pins. Additionally, trimmer capacitor C_{G1} (0 to 25 pF) should be connected between the OSC1 pin and V_{SS}.

OSC1 oscillation on/off (S1C17622/602/621)

The OSC1 oscillator stops oscillating when OSC1EN/OSC_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.

OSC1 oscillation on/off (S1C17624/604)

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

The OSC1 oscillator also starts oscillating by enabling the clock supply to the RTC (RTCCE/RTC_CC register = 1) even if OSC1EN = 0.

When RTCCE = 1, the OSC1 oscillator operates even in SLEEP mode and the OSC1 clock is supplied to the RTC for running. Note that the OSC1 clock is not supplied to other peripheral modules if OSC1EN = 0.

When RTCCE = 0, the OSC1 stops in SLEEP mode regardless of how OSC1EN is set.

After an initial reset, OSC1EN and RTCCE are both set to 0, and the OSC1 oscillator circuit is halted.

Table 7.3.3.1 OSC1 Oscillator Operating Status (S1C17624/604, in normal operation)

OSC1EN	RTCCE	OSC1 oscillator	Clock supply to peripheral modules	Clock supply to RTC
1	1	On	On	On
1	0	On	On	Off
0	1	On	Off	On
0	0	Off	Off	Off

Table 7.3.3.2 OSC1 Oscillator Operating Status (S1C17624/604, in SLEEP mode)

OSC1EN	RTCCE	OSC1 oscillator	Clock supply to peripheral modules	Clock supply to RTC
1	1	On	Off	On
1	0	Off	Off	Off
0	1	On	Off	On
0	0	Off	Off	Off

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)

7.4 System Clock Switching

The figure below shows the system clock selector.

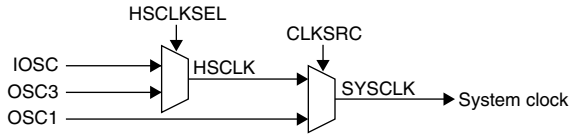


Figure 7.4.1 System Clock Selector

The S1C17624/604/622/602/621 has three system clock sources (IOSC, OSC3, and OSC1) and it start operating with the IOSC clock after an initial reset. The system clock can be switched to the OSC3 clock when a high-speed clock is required for the processing, or to the OSC1 clock for power saving. Oscillator circuits other than those selected as the system clock source and not used for running peripheral circuits can be shut down to reduce current consumption.

Use HSCLKSEL/OSC_SRC register to select the high-speed clock (HSCLK) from between IOSC and OSC3. Then select the system clock from between HSCLK and OSC1 using CLKSRC/OSC_SRC register.

The HSCLK selected may be used for some peripheral modules even if OSC1 is used as the system clock.

Table 7.4.1 System Clock Selection

HSCLKSEL	CLKSRC	HSCLK	System clock
1	1	OSC3	OSC1
1	0	OSC3	OSC3
0	1	IOSC	OSC1
0	0	IOSC	IOSC

(Default: HSCLKSEL = 0, CLKSRC = 0)

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. Turn the OSC3 oscillator on if it is off. (OSC3EN = 1)
3. Select the OSC3 clock as HSCLK. (HSCLKSEL = 1)
4. Select HSCLK (OSC3 clock) as the system clock. (CLKSRC = 0)
5. Turn the IOSC or OSC1 oscillator off if peripheral modules and FOUT1 output circuit have not used the IOSC or OSC1 clock.

Switching the system clock to OSC1 from IOSC or OSC3

1. Turn the OSC1 oscillator on. (OSC1EN = 1)
2. Select the OSC1 clock as the system clock. (CLKSRC = 1)
3. Turn the IOSC or OSC3 oscillator off if peripheral modules and FOUTH output circuit have not used the IOSC or OSC3 clock.

Switching the system clock to IOSC from OSC3 or OSC1

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 HSCLK. (HSCLKSEL = 0)
4. Select HSCLK (IOSC clock) as the system clock. (CLKSRC = 0)
5. Turn the OSC3 or OSC1 oscillator off if peripheral modules and FOUTH/1 output circuits have not used the OSC3 or OSC1 clock.

Notes: • Both the IOSC and OSC3 clocks must be active when selecting HSCLK. Otherwise, the system will fail to switch HSCLK even when HSCLKSEL is written to, and the HSCLKSEL value will remain unchanged.

• Both OSC1 and HSCLK must be operating when the system clock is selected (OSC1 or HSCLK). Otherwise, the system will not switch system clocks, even when CLKSRC is written to, and the CLKSRC value will remain unchanged.
Furthermore, the system clock switching operation takes a minimum one HSCLK cycle to maximum one OSC1 cycle.

Table 7.4.2 lists the combinations of clock operating states and register settings enabling system clock (OSC1 or HSCLK) selection.

Table 7.4.2 System Clock Switching (OSC1↔HSCLK) Conditions

IOSC	OSC3	OSC1	HSCLKSEL	System clock
On	On	On	*	IOSC/OSC3 or OSC1
On	Off	On	0	IOSC or OSC1
Off	On	On	1	OSC3 or OSC1

- When switching the HSCLK source (IOSC↔OSC3), always make sure that PCKEN[1:0]/CLG_PCLK register is set to 0x3 before writing to HSCLKSEL.
- The oscillator circuit selected as the system clock source cannot be turned off.
- Continuous write/read access to CLKSRC is prohibited. At least one instruction unrelated to CLKSRC access must be inserted between the write and read instructions.
- Canceling HALT/SLEEP mode does not change the clock status configured before the chip entered HALT/SLEEP 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.

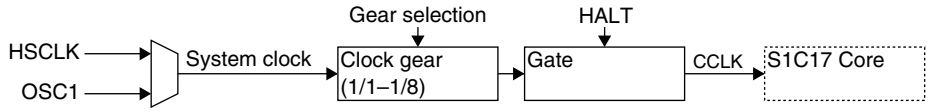


Figure 7.5.1 CCLK Supply System

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).

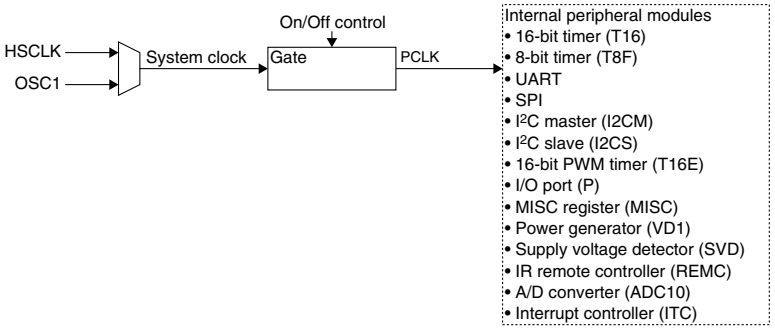


Figure 7.6.1 Peripheral Module Clock Control Circuit

Note: The interrupt controller (ITC) needs PCLK only when the register is set.

Clock supply control

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

Table 7.6.1 PCLK Control

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

(Default: PCKEN[1:0] = 0x3, PRUN = 0)

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 and PRUN/PSC_CTL register to 0, 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
16-bit timer (T16)	PCLK	The PCLK supply cannot be disabled if one or more peripheral modules in these list must be operated. The PCLK supply can be disabled if all the peripheral circuits in these list can be stopped.
8-bit timer (T8F)		
UART		
SPI		
I²C master (I2CM)		
I²C slave (I2CS)		
16-bit PWM timer (T16E)		
I/O port (P)		
MISC register (MISC)		
Power generator (VD1)		
Supply voltage detector (SVD)		
IR remote controller (REMC)		
A/D converter (ADC10)		
Interrupt controller (ITC) *		
Clock timer (CT)	OSC1 clock/ divided OSC1 clock	The OSC1 oscillator circuit cannot be disabled if one or more peripheral modules in these list must be operated. The PCLK supply can be disabled.
Stopwatch timer (SWT)		
Watchdog timer (WDT)		
8-bit OSC1 timer (T8OSC1)		
Real-time clock (RTC) (S1C17624/604)	OSC1 clock	The OSC1 oscillator and clock supply to the RTC can be controlled using RTCCE/RTC_CC register. The clock supply to the OSC1 peripheral circuits listed above and PCLK supply can be disabled.
LCD driver (LCD)	Clock selected by software (divided HSCLK clock/ OSC1 clock)	The oscillator circuit used as the clock source cannot be disabled (see each peripheral module chapter). The PCLK supply can be disabled.
R/F converter (RFC)		
FOUTH/FOUT1 outputs		
16-bit PWM timer (T16A2) (S1C17624/604)	Clock selected by software (divided IOSC/OSC3/OSC1 clock)	The oscillator circuit used as the clock source cannot be disabled (see the “16-bit PWM Timers (T16A2)” chapter). The PCLK supply can be disabled.

* The interrupt controller (ITC) needs PCLK only when the register is set.

7.7 Clock External Output (FOUTH, FOUT1)

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

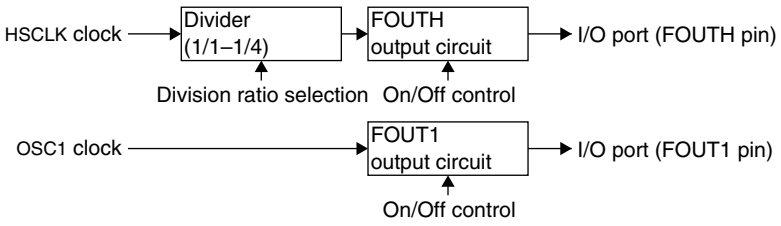


Figure 7.7.1 Clock Output Circuit

There are two output systems available: FOUTH and FOUT1.

Output pin setting

The FOUTH and FOUT1 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 FOUTH/FOUT1 pins and selecting pin functions.

FOUTH output control

FOUTH is a divided HSCLK (IOSC or OSC3) clock.

FOUTH clock frequency selection

Three different clock output frequencies can be selected. Select the division ratio for the HSCLK clock using FOUTHD[1:0]/OSC_FOUT register.

Table 7.7.1 FOUTH clock (HSCLK Division Ratio) Selection

FOUTHD[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 FOUTHE/OSC_FOUT register. Setting FOUTHE to 1 outputs the FOUTH clock from the FOUTH pin. Setting it to 0 disables output.

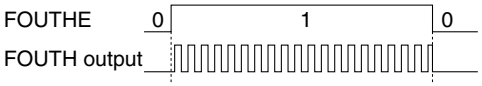


Figure 7.7.2 FOUTH Output

FOUT1 output

FOUT1 is the OSC1 clock.

Clock output control

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

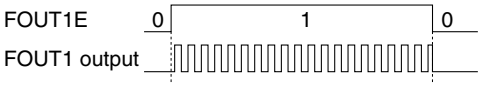


Figure 7.7.3 FOUT1 Output

Note: Since the FOUTH/FOUT1 signal is not synchronized with FOUTHE/FOUT1E writing, switching output on or off will generate certain hazards.

7.8 RESET and NMI Input Noise Filters

Since accidental activation of RESET or NMI by noise in the external input signals will cause unintended resetting or NMI processing, the CLG module incorporates noise filters operated by the system clock (SYSCLK). The filters remove noise from these signals before they reach the S1C17 Core or peripheral modules. Separate noise filters are used for each signal. You can select to use or bypass them individually.

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

Notes: • The RESET input noise filter should normally be enabled.

• The S1C17624/604/622/602/621 has no external NMI input pin, but the watchdog timer NMI request signal passes through the filter.

7.9 Control Register Details

Table 7.9.1 List of CLG Registers

Address	Register name		Function
0x4020	PSC_CTL	Prescaler Control Register	Controls prescalers.
0x5060	OSC_SRC	Clock Source Select Register	Selects the clock source.
0x5061	OSC_CTL	Oscillation Control Register	Controls oscillation.
0x5062	OSC_NFEN	Noise Filter Enable Register	Enables/disables noise filters.
0x5064	OSC_FOUT	FOUT Control Register	Controls FOUTH/FOUT1 clock outputs.
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.

Prescaler Control Register (PSC_CTL)

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

D[7:2] Reserved

D1 PRUND: 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 PRUND 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 PRUND to 1 to maintain running status for these peripheral circuits in debug mode.

D0 PRUN: Prescaler Run/Stop Control Bit

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

Write 1 to PRUN to operate the prescalers for peripheral modules. Write 0 to PRUN to stop the prescalers.

Clock Source Select Register (OSC_SRC)

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

D[7:2] Reserved

D1 HSCLKSEL: High-speed Clock Select Bit

Selects the high-speed clock (HSCLK).
1 (R/W): OSC3
0 (R/W): IOSC (default)

Note: Both the IOSC and OSC3 oscillators must be active when selecting HSCLK. Otherwise, HSCLK will not be switched, even when HSCLKSEL is written to, and the HSCLKSEL value will remain unchanged.

D0 CLKSRC: System Clock Source Select Bit

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

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

Notes:
- Both OSC1 and HSCLK must be operating when the system clock is selected (OSC1 or HSCLK). Otherwise, the system will not switch system clocks, even when CLKSRC is written to, and the CLKSRC value will remain unchanged. Furthermore, the system clock switching operation takes a minimum one HSCLK cycle to maximum one OSC1 cycle. Table 7.9.2 lists the combinations of clock operating states and register settings enabling system clock (OSC1 or HSCLK) selection.

Table 7.9.2 System Clock Switching (OSC1↔HSCLK) Conditions

IOSC	OSC3	OSC1	HSCLKSEL	System clock
On	On	On	*	IOSC/OSC3 or OSC1
On	Off	On	0	IOSC or OSC1
Off	On	On	1	OSC3 or OSC1

- When switching the HSCLK source (IOSC↔OSC3), always make sure that PCKEN[1:0]/CLG_PCLK register is set to 0x3 before writing to HSCLKSEL.
- The oscillator circuit selected as the system clock source cannot be turned off.
- Continuous write/read access to CLKSRC is prohibited. At least one instruction unrelated to CLKSRC access must be inserted between the write and read instructions.
- Canceling HALT/SLEEP mode does not change the clock status configured before the chip entered HALT/SLEEP mode.

Oscillation Control Register (OSC_CTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Oscillation Control Register (OSC_CTL)	0x5061 (8 bits)	D7–6	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] IOSCWT[1:0]: IOSC Wait Cycle Select Bits

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 7.9.3 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 V_{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 7.9.4 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.

D3 Reserved

D2 IOSCEN: IOSC Enable Bit

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

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

Noise Filter Enable Register (OSC_NFEN)

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

D[7:2] Reserved

D1 RSTFE: Reset Noise Filter Enable Bit

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

This should normally be enabled.

D0 NMIFE: NMI Noise Filter Enable Bit

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

Note: The S1C17624/604/622/602/621 has no external NMI input pin, but the watchdog timer NMI request signal passes through the filter.

FOUT Control Register (OSC_FOUT)

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

D[7:4] Reserved

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

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

Table 7.9.5 FOUTH clock (HSCLK Division Ratio) Selection

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

(Default: 0x0)

D1 FOUTHE: FOUTH Output Enable Bit

Enables or disables FOUTH clock (divided HSCLK clock) external output.
1 (R/W): Enabled (on)
0 (R/W): Disabled (off) (default)

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

D0 FOUT1E: FOUT1 Output Enable Bit

Enables or disables FOUT1 clock (OSC1 clock) external output.
1 (R/W): Enabled (on)
0 (R/W): Disabled (off) (default)

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

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.
Also set PRUN/PSC_CTL register.

Table 7.9.6 PCLK Control

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

(Default: PCKEN[1:0] = 0x3, PRUN = 0)

Peripheral modules that use PCLK

- 16-bit timer (T16)
- 8-bit timer (T8F)
- UART
- SPI
- I²C master (I2CM)
- I²C slave (I2CS)
- 16-bit PWM timer (T16E)
- I/O port (P)
- MISC register (MISC)
- Power generator (VD1)
- Supply voltage detector (SVD)
- IR remote controller (REMC)
- A/D converter (ADC10)
- Interrupt controller (ITC)

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 (CT)
- Stopwatch timer (SWT)
- Watchdog timer (WDT)
- 8-bit OSC1 timer (T8OSC1)
- LCD driver (LCD)
- R/F converter (RFC)
- 16-bit PWM timer (T16A2)
- FOUTH/FOUT1 outputs

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

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

• The interrupt controller (ITC) needs PCLK only when the register is set.

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 7.9.7 CCLK Gear Ratio Selection

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

(Default: 0x0)

8 Real-Time Clock (RTC)

Note: The RTC is available only in the S1C17624/604.

8.1 RTC Module Overview

The S1C17624/604 incorporates a real-time clock (RTC) with a perpetual calendar, and an OSC1 oscillator circuit to generate the operating clock for the RTC.
The RTC and OSC1 oscillator circuit (CLG) operate in SLEEP mode. Moreover, the RTC can periodically generate interrupt requests to the CPU.
The main features of the RTC are outlined below.

- Contains time counters (seconds, minutes, and hours) and calendar counters (days, days of the week, months, and year).
- BCD data can be read from and written to both counters.
- Includes read buffers to prevent carry over at reading.
- Capable of controlling the starting and stopping of time clocks.
- 24-hour or 12-hour mode can be selected.
- A 30-second correction function can be implemented in software.
- Periodic interrupts are possible.
- Interrupt period can be selected from 1/512 second, 1/256 second, 1/128 second, 1/64 second, 1 second, 1 minute, or 1 hour. (Level/edge interrupt mode)

Figure 8.1.1 shows a block diagram of the RTC.

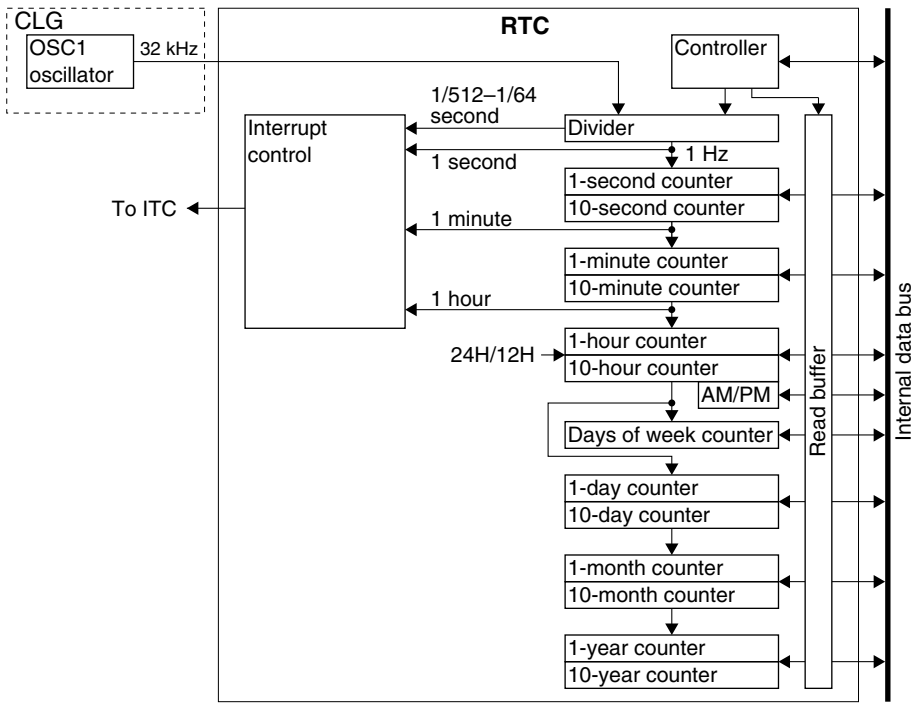


Figure 8.1.1 RTC Block Diagram

8.2 RTC Counters

The RTC contains the following 13 counters, whose count values can be read out as BCD data from the respective registers. Each counter can also be set to any desired date and time by writing data to the respective register.

1-second counter

This 4-bit BCD counter counts in units of seconds. It counts from 0 to 9 synchronously with a 1-second signal derived from the 32.768-kHz OSC1 clock by dividing the clock into smaller frequencies. This counter is reset to 0 after 9 and outputs a carry over of 1 to the 10-second counter. The count data is read out and written using RTCSL[3:0]/RTC_SEC register.

10-second counter

This 3-bit BCD counter counts tens of seconds. It counts from 0 to 5 with 1 carried over from the 1-second counter. This counter is reset to 0 after 5 and outputs a carry over of 1 to the 1-minute counter. The count data is read out and written using RTCSH[2:0]/RTC_SEC register.

1-minute counter

This 4-bit BCD counter counts in units of minutes. It counts from 0 to 9 with 1 carried over from the 10-second counter. This counter is reset to 0 after 9 and outputs a carry over of 1 to the 10-minute counter. The count data is read out and written using RTCMIL[3:0]/RTC_MIN register.

10-minute counter

This 3-bit BCD counter counts tens of minutes. It counts from 0 to 5 with 1 carried over from the 1-minute counter. This counter is reset to 0 after 5 and outputs a carry over of 1 to the 1-hour counter. The count data is read out and written using RTCMIH[2:0]/RTC_MIN register.

1-hour counter

This 4-bit BCD counter counts in units of hours. It counts from 0 to 9 with 1 carried over from the 10-minute counter. This counter is reset to 0 after 9 and outputs a carry over of 1 to the 10-hour counter. Depending whether 12-hour or 24-hour mode is selected, the counter is reset at 12 o'clock or 24 o'clock. The count data is read out and written using RTCHL[3:0]/RTC_HOUR register.

10-hour counter

This 2-bit BCD counter counts tens of hours. With a carry over of 1 from the 1-hour counter, this counter counts from 0 to 1 (when 12-hour mode is selected) or from 0 to 2 (when 24-hour mode is selected). The counter is reset at 12 o'clock or 24 o'clock, and outputs a carry over of 1 to the 1-day counter. The count data is read out and written using RTCHH[1:0]/RTC_HOUR register.
When 12-hour mode is selected, RTCAP/RTC_HOUR register that indicates A.M. or P.M. is enabled, with A.M. and P.M. represented by 0 and 1, respectively. For 24-hour mode, RTCAP is fixed to 0.

1-day counter

This 4-bit BCD counter counts in units of days. It counts from 0 to 9 with 1 carried over from the hour counter. This counter is reset to 0 after 9 and outputs a carry over of 1 to the 10-day counter. The number of days in each month and leap years are taken into account, so that the counter is reset to 1 when months change. The count data is read out and written using RTCDL[3:0]/RTC_DAY register.

10-day counter

This 2-bit BCD counter counts tens of days. It counts from 0 to 2 or 3 with 1 carried over from the 1-day counter. The number of days in each month and leap years are taken into account, so that when months change the counter is reset to 0 along with the 1-day counter, and outputs a carry over of 1 to the 1-month counter. The count data is read out and written using RTCDH[1:0]/RTC_DAY register.

1-month counter

This 4-bit BCD counter counts in units of months. It counts from 0 to 9 with 1 carried over from the day counter. This counter is reset to 0 after 9 and outputs a carry over of 1 to the 10-month counter. The counter is reset to 1 when years change. The count data is read out and written using RTCMOL[3:0]/RTC_MONTH register.

10-month counter

This counter counts in units of 10 months, and is set to 1 with 1 carried over from the 1-month counter. When years change, this counter is reset to 0 along with the 1-month counter, and outputs a carry over of 1 to the 1-year counter. The count data is read out and written using RTCMOH/RTC_MONTH register.

1-year counter

This 4-bit BCD counter counts in units of years. It counts from 0 to 9 with 1 carried over from the month counter. This counter is reset to 0 after 9 and outputs a carry over of 1 to the 10-year counter. The count data is read out and written using RTCYL[3:0]/RTC_YEAR register.

10-year counter

This 4-bit BCD counter counts tens of years. It counts from 0 to 9 with 1 carried over from the 1-year counter. The count data is read out and written using RTCYH[3:0]/RTC_YEAR register.

Days of week counter

This is a septenary counter (that counts from 0 to 6) representing the days of the week. It counts with the same timing as the 1-day counter. The count data is read out and written using RTCWK[2:0]/RTC_WEEK register.
The correspondence between the counter values and days of the week can be set in a program as desired. Table 8.2.1 lists the basic correspondence.

Table 8.2.1 Correspondence between Counter Values and Days of the Week

RTCWK[2:0]	Days of the week
0x6	Saturday
0x5	Friday
0x4	Thursday
0x3	Wednesday
0x2	Tuesday
0x1	Monday
0x0	Sunday

(Default: indeterminate)

Initial counter values

When initially reset, the counter values are not initialized. After power-on, the counter values are indeterminate. Be sure to initialize the counters by following the procedure described in Section 8.3.2, “RTC Initial Sequence.”

About detection of leap years

The algorithm used in the RTC to detect leap years is for Anno Domini (A.D.) only, and can automatically identify leap years up to the year 2399.
Years (0 to 99) without a remainder when divided by 4 are considered leap years. When the 1-year and 10-year counters both are 0, a common year is assumed.

8.3 RTC Control

8.3.1 Operating Clock Control

The RTC is clocked by the 32.768-kHz (typ.) OSC1 clock. To start running the RTC, set RTCCE/RTC_CC register to 1 to supply the OSC1 clock from the CLG to the RTC.
Setting RTCCE to 1 enables the OSC1 oscillator circuit to activate and to supply the OSC1 clock to the RTC regardless of the OSC1 oscillator setting in the CLG (even if OSC1EN/OSC_CTL register is set to 0 (OSC1 oscillation off)). Furthermore, the OSC1 clock will be supplied from the OSC1 oscillator circuit to run the RTC even in SLEEP mode if RTCCE is set to 1.
For more information on the OSC1 oscillator circuit, see the "Clock Generator (CLG)" chapter.

Note: After an initial reset, RTCCE is set to 0 and the RTC idles. The OSC1 oscillator circuit is also idle. Therefore, resetting the IC suspends the RTC operation for the period shown below.

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

8.3.2 RTC Initial Sequence

Immediately after power-on, the contents of RTC registers are indeterminate. After powering on, follow the procedure below to let the RTC start ticking the time. Later sections detail the contents of each control.

1. Power-on
2. System initialization processing
   Initialize the system. If any peripheral module other than RTC uses the OSC1 clock, turn the OSC1 oscillator circuit on using the CLG register.
3. Software reset
   Write 1 to RTCRST/RTC_CNTL0 register and then write 0 to reset the RTC.
4. Confirming accessibility status of the RTC
   See Section 8.3.5, "Counter Hold and Busy Flag."
5. Disabling the divider
   Write 1 to RTCSTP/RTC_CNTL0 register to stop the divider in the RTC module.
6. Setting the RTC interrupt
   Set the RTC_INTMODE register.
   Be sure to set RTCIMD to 1 (level sense).
7. Setting the date and time
   Set the RTC_SEC, RTC_MIN, RTC_HOUR, RTC_DAY, RTC_MONTH, RTC_YEAR, and RTC_WEEK registers. Then, write 0 to RTCHLD/RTC_CNTL1 register to release the 1-second, 10-second, 1-minute, 10-minute, 1-hour, 10-hour, 1-day, 10-day, 1-month, 10-month, 1-year, 10-year, and days of week counters from hold status.
8. Enabling the clock supply
   Write 1 to RTCCE/RTC_CC register to start supplying the OSC1 clock from the CLG to the RTC.
9. Starting the divider
   Write 0 to RTCSTP/RTC_CNTL0 register to run the divider in the RTC module.

8.3.3 12/24-hour Mode and Counter Settings

12-hour/24-hour mode selection

Whether to use the time clock in 12-hour or 24-hour mode can be selected using RTC24H/RTC_CNTL0 register.
RTC24H = 1: 24-hour mode
RTC24H = 0: 12-hour mode

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

Note: Rewriting RTC24H may corrupt count data for the hours, days, months, years or days of the week. Therefore, once RTC24H settings are changed, be sure to set data back in these counters again.

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

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

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

Counter settings

Idle counters can be accessed for read or write at any time.
However, settings like those shown below should be avoided, since such settings may cause timekeeping errors.

- Settings exceeding the effective range
  Do not set count data exceeding 60 seconds, 60 minutes, 12 or 24 hours, 31 days, 12 months, or 99 years.
- Settings nonexistent in the calendar
  Do not set such nonexistent dates as April 31 or February 29, 2006. Even if such settings are made, the counters operate normally, so that when 1 is carried over from the hour counter to the 1-day counter, the day counter counts up to the first day of the next month. (For April 31, the day counter counts up to May 1; for February 29, 2006, the day counter counts up to March 1, 2006.)

If any counter must be rewritten while operating, there is a procedure that must be followed to ensure that the counter is rewritten correctly. For details, see Section 8.3.5, “Counter Hold and Busy Flag.”

8.3.4 Start/Stop and Software Reset

Starting and stopping divider

The RTC starts counting when RTCSTP/RTC_CNTL0 register is set to 0, and stops counting when this bit is set to 1.
The RTC is started/stopped by writing data to RTCSTP at the 32-kHz input clock divide-by stage of 8,192 Hz or those stages that follow. The RTC does not stop at up to the input clock divide-by-2 stage (16,384 Hz).
If the RTC stops counting when 1 is carried over to the next-digit counter, the count value may be corrupted. Therefore, see the next section to ensure that 1 is not carried over when counters are made to stop. This is unnecessary, however, when the contents of all counters are newly set again.

Software reset

RTCRST/RTC_CNTL0 register is the software reset bit used to reset the items shown below.
• Divider
• Interrupt request signal
• Some register bits (see Section 8.5 for the control bits and their initial values.)

To perform software reset, write 1 to RTCRST and then write back to 0.
The divider bits above are all cleared 0. The interrupt request signal becomes inactive while RTCRST is set to 1 and is enabled to be output again after RTCRST is set to 0 (except when RTCCE = 0).

8.3.5 Counter Hold and Busy Flag

If 1 is carried over when writing the counters, the counter value may be corrupted. Therefore, whether counters are in a carry (busy) state should be checked before writing data to the count registers. For this purpose, control bits RTCBSY/RTC_CNTL1 register and RTCHLD/RTC_CNTL1 register are provided.
RTCBSY is a read-only flag indicating that carry is taking place. RTCBSY is set to 1 when carry is taking place; otherwise, it is 0. RTCBSY should be confirmed as being 0 before accessing the counters to ensure that the correct value will be set.
Writing 1 to RTCHLD suspends the counter operations. Note, however, that writing 1 to RTCHLD is ignored if RTCBSY is set to 1.

RTCBSY = 0 (RTC accessible)

When a value of 0 is read from RTCBSY after writing 1 to RTCHLD, it means that carry is not taking place. In this state, counter data can be written to.

After 1 is written to RTCHLD, the counters stop operating. So RTCBSY is fixed at 0, as carry will not take place. In this case, the counter hold function is also actuated, with a carry over of 1 to the 1-second counter disabled in hardware. The divider (counter for less than one second) continues operating.
Write data to the counter registers.
After writing data, reset RTCHLD to 0.
If 1 is being carried over when data is being written to a counter in the hold state, 1 second is automatically added to correct the counter values when RTCHLD is reset to 0. This correction is only effective for 1 second and no correction is conducted on the carry encountered in the second time and on. In this case, the timekeeping data gets out of order. Therefore, be sure to reset RTCHLD to 0 as soon as possible after completing the required write operation.

RTCBSY = 1 (RTC is busy)

When a value of 1 is read from RTCBSY after writing 1 to RTCHLD, it means that carry is taking place.
In this case, writing 1 to RTCHLD is ignored and RTCHLD retains 0.
A period of 4 ms per second is required for a carry over of 1 to the counters. In this case, [A] repeat writing 1 to RTCHLD and checking RTCBSY or [B] write 1 to RTCHLD and check RTCBSY after waiting for 4 ms.

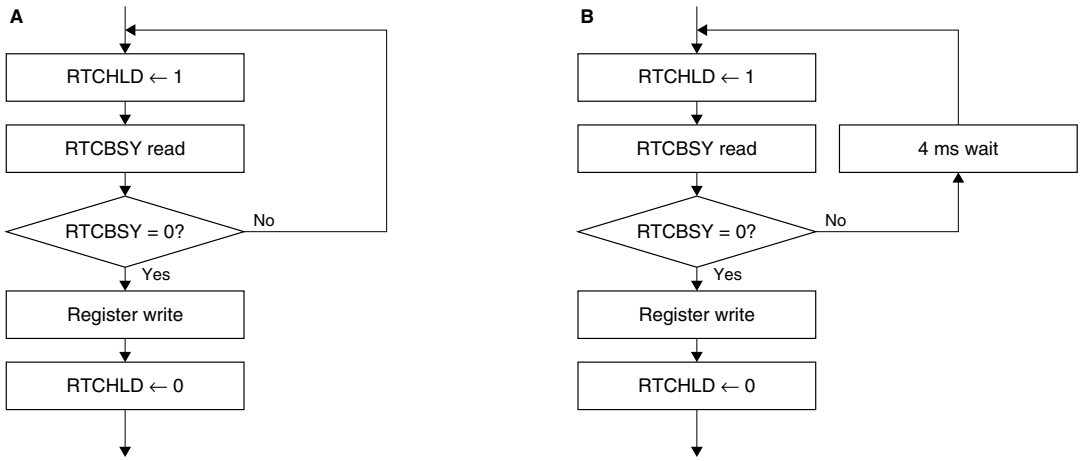


Figure 8.3.5.1 Procedure for Checking whether the RTC is Busy

8.3.6 30-second Correction

The description “30-second correction” means resetting the seconds to 0 and adding 1 to the minutes when seconds of the time clock are in the range of 30 to 59 seconds. When in the range of 0 to 29 seconds, the RTC resets the seconds to 0 but it does not change the minutes. This function may be used to round up seconds to minutes when resetting seconds in an application.
This function can be executed by writing 1 to RTCADJ/RTC_CNTL0 register.

Writing 1 to RTCADJ causes the RTC to operate as follows:

- When the 10-second counter is 3 or more, the RTC generates a carry over of 1 to start counting by the 1-minute counter.
- When the 10-second counter is 2 or less, the RTC does not generate a carry over of 1.

After RTCADJ is set to 1, it remains set for the 4-ms period required for this processing, then automatically returns to 0. To check whether the 30-second correction processing has completed or not, [A] repeat checking RTCADJ or [B] check RTCADJ after waiting for 4 ms.
Accessing the counters while RTCADJ = 1 is prohibited. Writing 0 to RTCADJ and writing 1 to RTCRST are also prohibited, because it would cause the RTC to operate erratically.
Writing 1 to RTCADJ when RTCBSY is 1 may corrupt the counter values. Always make sure that RTCBSY is set to 0 before writing 1 to RTCADJ.

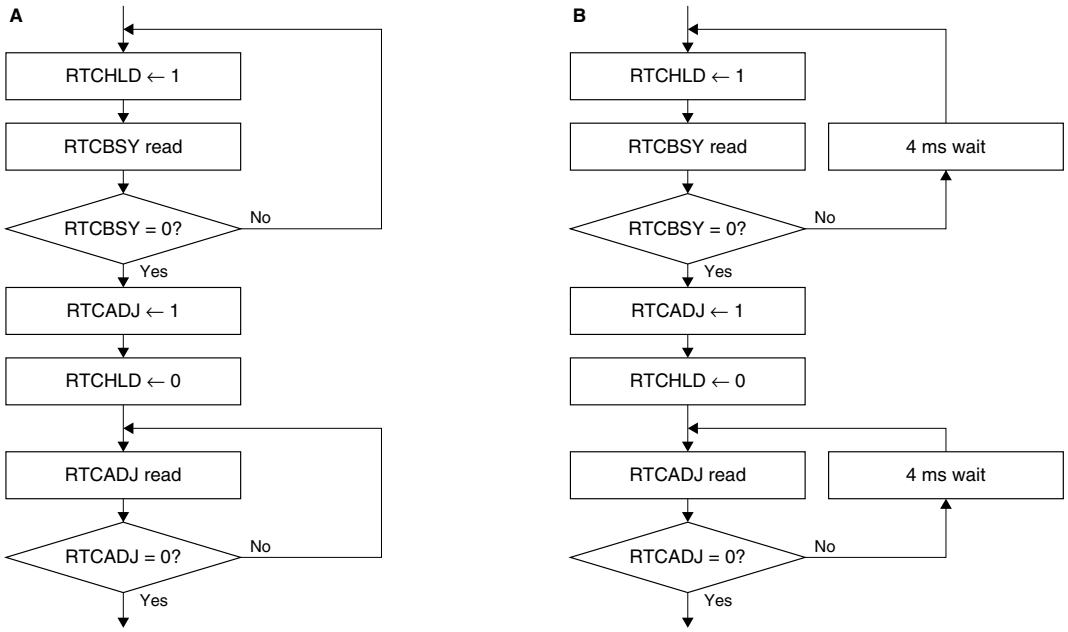


Figure 8.3.6.1 Procedure for Executing 30-second Correction

8.3.7 Counter Read

In order to prevent carry over during reading counters, the RTC includes a read buffer to hold counter data.
Before reading counter data, set RTCRDHLD/RTC_CNTL1 register to 1 to load the current counter data to the read buffer.
While RTCRDHLD is set to 1, the buffered data is read out from the counter registers. Be sure to reset RTCRDHLD to 0 after the buffered data is read out. This operation does not affect the counters. The counters keeps counting while RTCRDHLD is set to 1.

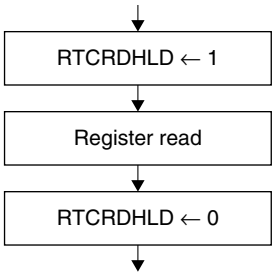


Figure 8.3.7.1 Procedure for Reading Counters

Note: At least three system clock cycles are required before the counter data can be read after data is written to the counter.

8.4 RTC Interrupts

The RTC has a function to generate interrupts at given intervals.
Since the RTC is active even in standby mode, interrupts may be used to cancel SLEEP mode.
This section describes the internal interrupt control function of the RTC. To generate interrupts to the CPU, the interrupt controller (ITC) must also be set up. For details on how to control the ITC, see the "Interrupt Controller (ITC)" chapter.

Interrupt cycle setting

The interrupt cycle (in which the RTC outputs interrupt requests at specific intervals) can be selected from seven choices listed in Table 8.4.1 by using RTCT[2:0]/RTC_INTMODE register.

Table 8.4.1 Interrupt Cycle Settings

RTCT[2:0]	Interrupt cycle
0x7	Reserved
0x6	1/128 second
0x5	1/256 second
0x4	1/512 second
0x3	1 hour
0x2	1 minute
0x1	1 second
0x0	1/64 second

RTCT[2:0] should be set while RTC interrupts are disabled. (See the procedure for enabling and disabling interrupts described below.)

Note: The system clock frequency must be set to 10 kHz or higher to use 1/512-second edge interrupts.

Setting interrupt conditions

The interrupt requests sent to the ITC can be selected as edge-triggered or level-sensed interrupts by setting a register bit. RTCIMD/RTC_INTMODE register is the bit provided for this purpose.
Setting RTCIMD to 1 selects a level-sensed interrupt; setting it to 0 selects an edge-triggered interrupt.
When an edge-triggered interrupt has been selected, the RTC outputs an interrupt pulse to the ITC using the bus clock supplied from the CLG. If a cause of interrupt occurs when the bus clock has not been supplied such as in SLEEP mode, the RTC switches the interrupt mode to level-sensed and sets the interrupt signal to the active level from occurrence of the interrupt cause until the bus clock supply is started.

Enabling and disabling interrupts

The RTC interrupt requests output to the ITC are enabled by setting RTCIEN/RTC_INTMODE register to 1 and disabled by setting it to 0.
RTC interrupts will be generated according to the divider and counter status and the time between writing 1 to RTCIEN and the first interrupt request is not fixed. Use the second and subsequent interrupts as valid.

Interrupt status

When the RTC is up and running, RTCIRQ/RTC_INTSTAT register is set at the cyclic interrupt intervals set up by RTCT[2:0]. When RTC interrupts are enabled by RTCIEN, interrupt requests are sent to the ITC.
Writing 1 to this status bit clears the bit. Because this bit is not cleared in hardware, be sure to clear it in software after an interrupt is generated. If this bit remains set while interrupts are re-enabled or control is returned from the interrupt handler routine by the reti instruction, the same interrupt may be generated again.
When RTCIEN is set to 0 (interrupt disabled), RTCIRQ is fixed at 0 (will not be set to 1).

Precautions

All RTC interrupt control bits described above are indeterminate when power is turned on. Moreover, these bits are not initialized to specific values by an initial reset.
After power-on, an RTC interrupt request is masked (not output) regardless of the RTCIEN and RTCIRQ settings until the clock supply to the RTC is enabled using RTCCE/RTC_CC register. However, be sure to set RTCIEN to 0 (interrupt disabled) to prevent the occurrence of unwanted RTC interrupts.

Also be sure to write 1 to RTCIRQ to reset it.
When a software reset is performed (RTCRST → 1 → 0), RTCIRQ and RTCIEN are reset to 0 to disable the interrupt request output. Also RTCT[2:0] is reset to 0x1.

8.5 Details of Control Registers

Table 8.5.1 RTC Register List

Address	Register name		Function
0x506e	RTC_CC	RTC Clock Control Register	Controls the RTC clock source.
0x5140	RTC_INTSTAT	RTC Interrupt Status Register	Indicates RTC interrupt status.
0x5141	RTC_INTMODE	RTC Interrupt Mode Register	Sets up RTC interrupt modes.
0x5142	RTC_CNTL0	RTC Control 0 Register	Controls the RTC.
0x5143	RTC_CNTL1	RTC Control 1 Register	
0x5144	RTC_SEC	RTC Second Register	Second counter data
0x5145	RTC_MIN	RTC Minute Register	Minute counter data
0x5146	RTC_HOUR	RTC Hour Register	Hour counter data
0x5147	RTC_DAY	RTC Day Register	Day counter data
0x5148	RTC_MONTH	RTC Month Register	Month counter data
0x5149	RTC_YEAR	RTC Year Register	Year counter data
0x514a	RTC_WEEK	RTC Days of Week Register	Days of week counter data

The following describes each RTC register. These are all 8-bit registers.

Notes:
- When data is written to the register, the “Reserved” bits must always be written as 0 and not 1.
- The contents of all RTC control registers are indeterminate when power is turned on, and are not initialized to specific values by initial reset. These registers should be initialized in software.
- If 1 is being carried over when the counters are accessed for read, the correct counter value may not be read out. Moreover, attempting to write to a counter or other control register may corrupt the counter value. Therefore, do not write to counters while 1 is being carried over. For the correct method of operation, see Section 8.3.5, “Counter Hold and Busy Flag,” and Section 8.3.7, “Counter Read.”

RTC Clock Control Register (RTC_CC)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RTC Clock Control Register (RTC_CC)	**0x506e** (8 bits)	D7–1	–	reserved	–	–	–	0 when being read.
		D0	**RTCCE**	RTC clock enable	1 Enable 0 Disable	0	R/W	

D[7:1] Reserved

D0 RTCCE: RTC Clock Enable Bit

Enables or disables the OSC1 clock supply to the RTC.
1 (R/W): Enabled (on)
0 (R/W): Disabled (off) (default)

The RTCCE default setting is 0, which disables the clock supply. Setting RTCCE to 1 sends the OSC1 clock to the RTC. When the OSC1 oscillator circuit is stopped, writing 1 to RTCCE turns it on (note, however, that the OSC1 clock is not supplied to other peripheral circuits than the RTC). When RTCCE is set to 1, the OSC1 oscillator circuit does not stop even if the IC enters SLEEP mode (the OSC1 clock will be supplied to the RTC only).
When RTCCE is set to 0, RTC interrupt requests generated by RTCIEN/RTC_INTMODE register and RTCIRQ/RTC_INTSTAT register are masked to prevent occurrence of undesired interrupts.

RTC Interrupt Status Register (RTC_INTSTAT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RTC Interrupt Status Register (RTC_INTSTAT)	0x5140 (8 bits)	D7–1	–	reserved	–	–	–	0 when being read.
		D0	RTCIRQ	Interrupt status	1 Occurred 0 Not occurred	X (0)	R/W	Reset by writing 1.

Init.: ( ) indicates the value set after a software reset (RTCRST → 1 → 0) is performed.

D[7:1] Reserved

D0 RTCIRQ: Interrupt Status Bit

This bit indicates whether a cause of RTC interrupt occurred as follows:

1 (R): Cause of interrupt occurred
0 (R): No cause of interrupt occurred (software reset value)
1 (W): Resets this bit to 0
0 (W): Has no effect

This bit is set at cyclic interrupt intervals set up by RTCT[2:0]/RTC_INTMODE register. When RTC interrupts have been enabled by RTCIEN/RTC_INTMODE register at this time, an interrupt request is sent to the ITC.

Note: Writing 1 to this status bit clears it. Because this bit is not cleared in hardware, be sure to clear it in software after an interrupt is generated. If this bit remains set while interrupts are re-enabled or control is returned from the interrupt handler routine by the reti instruction, the same interrupt may be generated again. Moreover, the value of this bit is indeterminate after power-on, and is not initialized to 0 by initial reset. To prevent the occurrence of unwanted RTC interrupts, be sure to reset this bit in software after power-on and initial reset.

RTC Interrupt Mode Register (RTC_INTMODE)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RTC Interrupt Mode Register (RTC_INTMODE)	0x5141 (8 bits)	D7–5	–	reserved	–	–	–	0 when being read.
		D4–2	RTCT[2:0]	RTC interrupt cycle setup	RTCT[2:0] Cycle 0x7 reserved 0x6 1/128 second 0x5 1/256 second 0x4 1/512 second 0x3 1 hour 0x2 1 minute 0x1 1 second 0x0 1/64 second	X (0x1)	R/W	
		D1	RTCIMD	RTC interrupt mode select	1 Level sense 0 Edge trigger	X (1)	R/W	
		D0	RTCIEN	RTC interrupt enable	1 Enable 0 Disable	X (0)	R/W	

Init.: ( ) indicates the value set after a software reset (RTCRST → 1 → 0) is performed.

D[7:5] Reserved

D[4:2] RTCT[2:0]: RTC Interrupt Cycle Setup Bits

These bits select the RTC interrupt cycle.

Table 8.5.2 Interrupt Cycle Settings

RTCT[2:0]	Interrupt cycle
0x7	Reserved
0x6	1/128 second
0x5	1/256 second
0x4	1/512 second
0x3	1 hour
0x2	1 minute
0x1	1 second
0x0	1/64 second

(Default: indeterminate, software reset: 0x1)

RTCIRQ/RTC_INTSTAT register is set by a count-up pulse of the interrupt cycle counter selected. When RTC interrupts are enabled by RTCIEN, an interrupt request is sent to the ITC.
RTCT[2:0] should be set while RTC interrupts are disabled. (These bits may also be set simultaneously when RTC interrupts are enabled.)

D1 RTCIMD: RTC Interrupt Mode Select Bit

This bit specifies whether RTC interrupts are to be generated by an edge or level of the interrupt request signal.

1 (R/W): Level sensed (software reset value)

0 (R/W): Edge triggered

When an edge-triggered interrupt is selected and used to turn off SLEEP mode via the CLG, note that no interrupts will be generated because the ITC is inactive. When an RTC interrupt handler routine must be executed after exiting SLEEP mode, select a level-sensed interrupt.

D0 RTCIEN: RTC Interrupt Enable Bit

This bit enables or disables RTC interrupt request output to the ITC.

1 (R/W): Enable interrupts

0 (R/W): Disable interrupts (software reset value)

To generate an RTC interrupt or use an RTC interrupt request signal to turn off SLEEP mode, set this bit to 1. When this bit is 0, no interrupts are generated and SLEEP mode cannot be turned off.

Note: The value of RTCIEN is indeterminate after power-on, and not initialized to 0 by initial reset. To prevent the occurrence of unwanted RTC interrupts, be sure to clear this bit in software after power-on and initial reset.

RTC Control 0 Register (RTC_CNTL0)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
RTC Control 0 Register (RTC_CNTL0)	0x5142 (8 bits)	D7–5	–	reserved	–				–	–	0 when being read.
		D4	RTC24H	24H/12H mode select	1	24H	0	12H	X (0)	R/W	
		D3	–	reserved	–				–	–	0 when being read.
		D2	RTCADJ	30-second adjustment	1	Adjust	0	–	X (0)	R/W	
		D1	RTCSTP	Divider run/stop control	1	Stop	0	Run	X (0)	R/W	
		D0	RTCRST	Software reset	1	Reset	0	–	X (0)	R/W	

Init.: ( ) indicates the value set after a software reset (RTCRST → 1 → 0) is performed.

D[7:5] Reserved

D4 RTC24H: 24H/12H Mode Select Bit

This bit selects whether to use the hour counter in 24-hour or 12-hour mode.

1 (R/W): 24-hour mode

0 (R/W): 12-hour mode (software reset value)

The count range of hour counters changes with this selection. Basically, this setting should be changed while the counters are idle. Since this register is assigned a control bit (D1) to start the counters, 12-hour or 24-hour mode may be selected when starting the counters.

Note: Rewriting RTC24H may corrupt the count data for hours, days, months, years, or days of the week. Therefore, after changing the RTC24H setting, be sure to set data back in these counters again.

D3 Reserved

D2 RTCADJ: 30-second Adjustment Bit

This bit executes 30-second correction.

1 (W): Execute 30-second correction

0 (W): Has no effect

1 (R): 30-second correction being executed

0 (R): 30-second correction completed (not being executed) (software reset value)

The description “30-second correction” means adding 1 to the minutes when seconds of the time clock are in the 30-to-59 second range, and doing nothing in the 0-to-29 second range. This function may be used to round up seconds to minutes when resetting seconds in an application.

Writing 1 to this bit causes the RTC to operate as follows:

- When the 10-second counter is 3 or more, the RTC generates a carry over of 1 to start counting by the 1-minute counter.
- When the 10-second counter is 2 or less, the RTC does not generate a carry over of 1.

After being set to 1, this bit remains set for the 4-ms period needed for the processing above, then is automatically reset to 0.

Note: Accessing the counters while RTCADJ = 1 is prohibited. Writing 0 to this bit during such time is also prohibited, because it would cause the RTC to operate erratically.

D1 RTCSTP: Divider Run/Stop Control Bit

This bit starts or stops the divider. It also indicates divider operating status.
1 (R/W): Stop divider/counters
0 (R/W): Start divider/counters (software reset value)

Setting this bit to 0 starts the divider; setting it to 1 stops the divider. The value read from this bit is 0 when the divider/counters are operating, and 1 when the counters are idle.
This bit starts/stops the divider at the 32-kHz input clock divide-by stage of 8,192 Hz or stages that follow. The counters do not stop at up to the input clock divide-by-2 stage (16,384 Hz).
If the divider stops while carry of a counter is taking place, the count value may be corrupted. Therefore, see Section 8.3.5 to ensure that carry is not taking place when the divider is stopped. This is not required when, for example, the contents of all counters are newly set again.

D0 RTCRST: Software Reset Bit

This bit resets the divider and output signals.
1 (R/W): Reset
0 (R/W): Negate reset (software reset value)

To perform software reset, write 1 to RTCRST and then write 0.
The software reset clears the 32 kHz to 2 Hz divider bits, negates the interrupt request signal, and initializes some control bits.
When setting up the RTC, first perform software reset using RTCRST.

RTC Control 1 Register (RTC_CNTL1)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
RTC Control 1 Register (RTC_CNTL1)	0x5143 (8 bits)	D7–3	–	reserved	–				–	–	0 when being read.
		D2	RTCRDHLD	Read buffer enable	1	Enable	0	Disable	X (0)	R/W	
		D1	RTCBSY	Counter busy flag	1	Busy	0	R/W possible	X (0)	R	
		D0	RTCHLD	Counter hold control	1	Hold	0	Running	X (0)	R/W	

Init.: ( ) indicates the value set after a software reset (RTCRST → 1 → 0) is performed.

D[7:3] Reserved

D2 RTCRDHLD: Read Buffer Enable Bit

This bit enables or disables the read buffer.
1 (R/W): Enabled
0 (R/W): Disabled (software reset value)

In order to prevent carry over during reading counters, the RTC includes a read buffer to hold counter data. Before reading counter data, set RTCRDHLD to 1 to load the current counter data to the read buffer. While RTCRDHLD is set to 1, the buffered data is read out from the counter registers. Be sure to reset RTCRDHLD to 0 after the buffered data is read out. This operation does not affect the counters. The counters keeps counting while RTCRDHLD is set to 1.

D1 RTCBSY: Counter Busy Flag Bit

This flag indicates whether 1 is being carried over to the next-digit counter.
1 (R): Busy (while carry is taking place)
0 (R): Accessible for read/write (software reset value)

Attempting a write or stop operation may corrupt the counter values if 1 is being carried over. Therefore, this bit should be checked to confirm that the counters are not in a carry (busy) state before writing data to the counter registers.

When a value of 0 is read from RTCBSY after writing 1 to RTCHLD, it means that carry is not taking place. In this state, counter data can be written to.
After 1 is written to RTCHLD, the counters stop operating. So RTCBSY is fixed at 0, as carry will not take place. In this case, the counter hold function is also actuated, with a carry over of 1 to the 1-second counter disabled in hardware. The divider (counter for less than one second) continues operating.
Write data to the counter registers. After writing data, reset RTCHLD to 0.
If 1 is being carried over when data is being written to a counter in the hold state, 1 second is automatically added to correct the counter values when RTCHLD is reset to 0. This correction is only effective for 1 second and no correction is conducted on the carry encountered in the second time and on. In this case, the timekeeping data gets out of order. Therefore, be sure to reset RTCHLD to 0 as soon as possible after completing the required write operation.

When a value of 1 is read from RTCBSY after writing 1 to RTCHLD, it means that carry is taking place. In this case, writing 1 to RTCHLD is ignored and RTCHLD retains 0. A period of 4 ms per second is required for a carry over of 1 to the counters. In this case, repeat writing 1 to RTCHLD and checking RTCBSY, or write 1 to RTCHLD and check RTCBSY after waiting for 4 ms.

D0 RTCHLD: Counter Hold Control Bit

This bit allows the busy state of counters to be checked and the counters held intact.
1 (R/W): Checks for busy state/Holds counters
0 (R/W): Normal operation (software reset value)

For the operation of this bit, see the description of RTCBSY above.

RTC Second Register (RTC_SEC)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RTC Second Register (RTC_SEC)	0x5144 (8 bits)	D7	–	reserved	–	–	–	0 when being read.
		D6–4	RTCSH[2:0]	RTC 10-second counter	0 to 5	X (*)	R/W	
		D3–0	RTCSL[3:0]	RTC 1-second counter	0 to 9	X (*)	R/W	

* Software reset (RTCRST → 1 → 0) does not affect the counter values. This register retains the value set before a software reset is performed.

Note: Data should not be read from or written to the counters while 1 is being carried over. (See Section 8.3.5, “Counter Hold and Busy Flag,” and Section 8.3.7, “Counter Read.”)

D7 Reserved

D[6:4] RTCSH[2:0]: RTC 10-second Counter Bits

These bits comprise a 3-bit BCD counter used to count tens of seconds.
The counter counts from 0 to 5 with a carry over of 1 from the 1-second counter. This counter is reset to 0 after 5 and outputs a carry over of 1 to the 1-minute counter.

D[3:0] RTCSL[3:0]: RTC 1-second Counter Bits

These bits comprise a 4-bit BCD counter used to count units of seconds.
The counter counts from 0 to 9 synchronously with a 1-second signal derived from the 32.768-kHz OSC1 clock. This counter is reset to 0 after 9 and outputs a carry over of 1 to the 10-second counter.

RTC Minute Register (RTC_MIN)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RTC Minute Register (RTC_MIN)	0x5145 (8 bits)	D7	–	reserved	–	–	–	0 when being read.
		D6–4	RTCMIH[2:0]	RTC 10-minute counter	0 to 5	X (*)	R/W	
		D3–0	RTCMIL[3:0]	RTC 1-minute counter	0 to 9	X (*)	R/W	

* Software reset (RTCRST → 1 → 0) does not affect the counter values. This register retains the value set before a software reset is performed.

Note: Data should not be read from or written to the counters while 1 is being carried over. (See Section 8.3.5, “Counter Hold and Busy Flag,” and Section 8.3.7, “Counter Read.”)

D7 Reserved

D[6:4] RTCMIH[2:0]: RTC 10-minute Counter Bits

These bits comprise a 3-bit BCD counter used to count tens of minutes.
The counter counts from 0 to 5 with a carry over of 1 from the 1-minute counter. This counter is reset to 0 after 5 and outputs a carry over of 1 to the 1-hour counter.

D[3:0] RTCMIL[3:0]: RTC 1-minute Counter Bits

These bits comprise a 4-bit BCD counter used to count units of minutes.
The counter counts from 0 to 9 with a carry over of 1 from the 10-second counter. This counter is reset to 0 after 9 and outputs a carry over of 1 to the 10-minute counter.

RTC Hour Register (RTC_HOUR)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RTC Hour Register (RTC_HOUR)	0x5146 (8 bits)	D7	–	reserved	–	–	–	0 when being read.
		D6	RTCAP	AM/PM indicator	1 PM 0 AM	X (*)	R/W	
		D5–4	RTCHH[1:0]	RTC 10-hour counter	0 to 2 or 0 to 1	X (*)	R/W	
		D3–0	RTCHL[3:0]	RTC 1-hour counter	0 to 9	X (*)	R/W	

* Software reset (RTCRST → 1 → 0) does not affect the counter values. This register retains the value set before a software reset is performed.

Notes: • Data should not be read from or written to the counters while 1 is being carried over. (See Section 8.3.5, "Counter Hold and Busy Flag," and Section 8.3.7, "Counter Read.")

• Rewriting RTC24H/RTC_CNTL0 register may corrupt the count data in this register. Therefore, after changing the RTC24H setting, be sure to set up this register again.

D7 Reserved

D6 RTCAP: AM/PM Indicator Bit

When 12-hour mode is selected, this bit indicates A.M. or P.M.
1 (R/W): P.M.
0 (R/W): A.M.

This bit is only effective when RTC24H/RTC_CNTL0 register is set to 0 (12-hour mode).
When 24-hour mode is selected, this bit is fixed to 0. In this case, do not write 1 to RTCAP.

Note: The RTCAP bit keeps the current set value even if RTC24H/RTC_CNTL0 register is changed from 12-hour mode to 24-hour mode, and will be fixed at 0 after the hour counter is updated (or reset in software).

D[5:4] RTCHH[1:0]: RTC 10-hour Counter Bits

These bits comprise a 2-bit BCD counter used to count tens of hours.
With a carry over of 1 from the 1-hour counter, the counter counts from 0 to 1 when 12-hour mode is selected, or from 0 to 2 when 24-hour mode is selected. The counter is reset at 12 o'clock or 24 o'clock, and outputs a carry over of 1 to the 1-day counter.

D[3:0] RTCHL[3:0]: RTC 1-hour Counter Bits

These bits comprise a 4-bit BCD counter used to count units of hours.
The counter counts from 0 to 9 with a carry over of 1 from the 10-minute counter. This counter is reset to 0 after 9 and outputs a carry over of 1 to the 10-hour counter. Depending on whether 12-hour mode or 24-hour mode is selected, the counter is reset at 12 o'clock or 24 o'clock.

RTC Day Register (RTC_DAY)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RTC Day Register (RTC_DAY)	0x5147 (8 bits)	D7–6	–	reserved	–	–	–	0 when being read.
		D5–4	RTCDH[1:0]	RTC 10-day counter	0 to 3	X (*)	R/W	
		D3–0	RTCDL[3:0]	RTC 1-day counter	0 to 9	X (*)	R/W	

* Software reset (RTCRST → 1 → 0) does not affect the counter values. This register retains the value set before a software reset is performed.

Notes: • Data should not be read from or written to the counters while 1 is being carried over. (See Section 8.3.5, "Counter Hold and Busy Flag," and Section 8.3.7, "Counter Read.")

• Rewriting RTC24H/RTC_CNTL0 register may corrupt the count data in this register. Therefore, after changing the RTC24H setting, be sure to set up this register again.

D[7:6] Reserved

D[5:4] RTCDH[1:0]: RTC 10-day Counter Bits

These bits comprise a 2-bit BCD counter used to count tens of days. The counter counts from 0 to 2 or 3 with a carry over of 1 from the 1-day counter. The number of days in each month and leap years are taken into account, so that when months change the counter is reset to 0 along with the 1-day counter, and a carry over of 1 is output to the 1-month counter.

D[3:0] RTCDL[3:0]: RTC 1-day Counter Bits

These bits comprise a 4-bit BCD counter used to count units of days.
The counter counts from 0 to 9 with a carry over of 1 from the hour counter. This counter is reset to 0 after 9 and outputs a carry over of 1 to the 10-day counter. The number of days in each month and leap years are taken into account, so that the counter is reset to 1 when months change.

RTC Month Register (RTC_MONTH)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RTC Month Register (RTC_MONTH)	0x5148 (8 bits)	D7–5	–	reserved	–	–	–	0 when being read.
		D4	RTCMOH	RTC 10-month counter	0 to 1	X (*)	R/W	
		D3–0	RTCMOL[3:0]	RTC 1-month counter	0 to 9	X (*)	R/W	

* Software reset (RTCRST → 1 → 0) does not affect the counter values. This register retains the value set before a software reset is performed.

Notes: • Data should not be read from or written to the counters while 1 is being carried over. (See Section 8.3.5, "Counter Hold and Busy Flag," and Section 8.3.7, "Counter Read.")

• Rewriting RTC24H/RTC_CNTL0 register may corrupt the count data in this register. Therefore, after changing the RTC24H setting, be sure to set up this register again.

D[7:5] Reserved

D4 RTCMOH: RTC 10-month Counter Bit

This is a tens of months count bit.
This bit is set to 1 with a carry over of 1 from the 1-month counter. When years change, this bit is reset to 0 along with the 1-month counter, and a carry over of 1 is output to the 1-year counter.

D[3:0] RTCMOL[3:0]: RTC 1-month Counter Bits

These bits comprise a 4-bit BCD counter used to count units of months.
The counter counts from 0 to 9 with a carry over of 1 from the day counter. This counter is reset to 0 after 9 and outputs a carry over of 1 to the 10-month counter. The counter is reset to 1 when years change.

RTC Year Register (RTC_YEAR)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RTC Year Register (RTC_YEAR)	0x5149 (8 bits)	D7–4	RTCYH[3:0]	RTC 10-year counter	0 to 9	X (*)	R/W	
		D3–0	RTCYL[3:0]	RTC 1-year counter	0 to 9	X (*)	R/W	

* Software reset (RTCRST → 1 → 0) does not affect the counter values. This register retains the value set before a software reset is performed.

Notes: • Data should not be read from or written to the counters while 1 is being carried over. (See Section 8.3.5, "Counter Hold and Busy Flag," and Section 8.3.7, "Counter Read.")

• Rewriting RTC24H/RTC_CNTL0 register may corrupt the count data in this register. Therefore, after changing the RTC24H setting, be sure to set up this register again.

D[7:4] RTCYH[3:0]: RTC 10-year Counter Bits

These bits comprise a 4-bit BCD counter used to count tens of years. The counter counts from 0 to 9 with a carry over of 1 from the 1-year counter.

D[3:0] RTCYL[3:0]: RTC 1-year Counter Bits

These bits comprise a 4-bit BCD counter used to count units of years.
The counter counts from 0 to 9 with a carry over of 1 from the month counter. This counter is reset to 0 after 9 and outputs a carry over of 1 to the 10-year counter.

RTC Days of Week Register (RTC_WEEK)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
RTC Days of Week Register (RTC_WEEK)	0x514a (8 bits)	D7–3	–	reserved	–		–	–	0 when being read.
		D2–0	RTCWK[2:0]	RTC days of week counter	RTCWK[2:0]	Days of week	X (*)	R/W	
					0x7	–			
					0x6	Saturday			
					0x5	Friday			
					0x4	Thursday			
					0x3	Wednesday			
					0x2	Tuesday			
					0x1	Monday			
					0x0	Sunday			

* Software reset (RTCRST → 1 → 0) does not affect the counter values. This register retains the value set before a software reset is performed.

Notes: • Data should not be read from or written to the counters while 1 is being carried over. (See Section 8.3.5, "Counter Hold and Busy Flag," and Section 8.3.7, "Counter Read.")

• Rewriting RTC24H/RTC_CNTL0 register may corrupt the count data in this register. Therefore, after changing the RTC24H setting, be sure to set up this register again.

D[7:3] Reserved

D[2:0] RTCWK[2:0]: RTC Days of Week Counter Bits

This is a septenary counter (that counts from 0 to 6) representing days of the week. This counter counts at the same timing as the 1-day counter.
The correspondence between the counter values and days of the week can be set in a program as desired. Table 8.5.3 lists the basic correspondence.

Table 8.5.3 Correspondence between Counter Values and Days of the Week

RTCWK[2:0]	Days of the week
0x6	Saturday
0x5	Friday
0x4	Thursday
0x3	Wednesday
0x2	Tuesday
0x1	Monday
0x0	Sunday

(Default: indeterminate, software reset: previous value retained)

9 I/O Ports (P)

9.1 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. These ports are shared with internal peripheral module inputs/outputs, and the pin functions can be switched by setting the registers. A number of port groups can generate interrupts caused by a transition of the input signal.
The following shows the features of the P module:

- S1C17624/622: Maximum 47 I/O ports (P0[7:0], P1[7:0], P2[7:0], P3[7:0], P4[7:0], P5[6:0]) are available.
  S1C17604/602/621: Maximum 36 I/O ports (P0[7:0], P1[7:0], P2[7:0], P3[7:0], P4[3:0]) are available.
  * The number of ports for general-purpose use depends on the peripheral functions used.
- Each port has a pull-up resistor that can be enabled with software.
- Some I/O ports support two input interface levels selectable with software: CMOS Schmitt level or CMOS level
- The P0 and P1 ports can generate input interrupts at the signal edge selected with software.
- The P0 and P1 ports include a chattering filter.
- Can generate an initial reset by entering low level simultaneously to the P0 ports selected with software.
- All port provide a port function select bit to configure the pin function (for GPIO or peripheral functions).

Figure 9.1.1 shows the I/O port configuration.

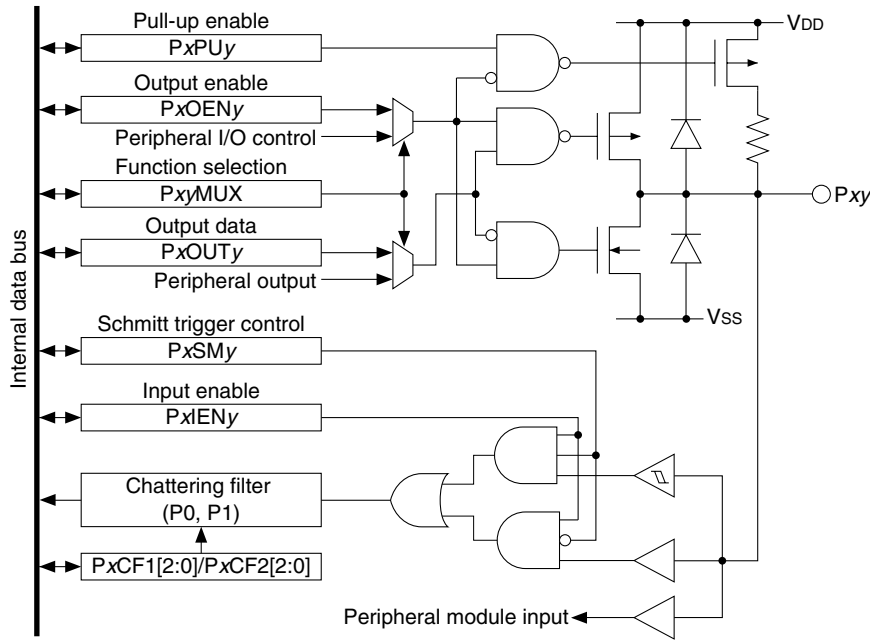


Figure 9.1.1 I/O Port Configuration

Notes: • The PCLK clock must be supplied from the clock generator to access the I/O port. The PCLK clock is also needed to operate the P0/P1 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

- The I/O ports shown below allow software to select the input interface level from two types: CMOS Schmitt level and CMOS level.
  S1C17624/622: P0[7:0], P1[5:0], P4[7:4], P5[2:0]
  S1C17604/602/621: P0[7:0], P1[5:0]

  The I/O ports shown below support only CMOS Schmitt level. (* P43 is an output-only port.)
  S1C17624/622: P1[7:6], P2[7:0], P3[7:0], P40, DSIO/P41, DST2/P42, DCLK/P43*, P5[6:3]
  S1C17604/602/621: P1[7:6], P2[7:0], P3[7:0], P40, DSIO/P41, DST2/P42, DCLK/P43*

9.2 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 9.2.1 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 (bit/register)
P00	REMO (REMC)	–	–	P00MUX[1:0]/P00_03PMUX
P01	REMI (REMC)	–	–	P01MUX[1:0]/P00_03PMUX
P02/EXCL0 (T16)*1	–	–	–	P02MUX[1:0]/P00_03PMUX
P03	#ADTRG (ADC10)	–	–	P03MUX[1:0]/P00_03PMUX
P04	SPICLK0 (SPI)	–	–	P04MUX[1:0]/P04_07PMUX
P05	SDO0 (SPI)	–	–	P05MUX[1:0]/P04_07PMUX
P06	SDI0 (SPI)	–	–	P06MUX[1:0]/P04_07PMUX
P07	#SPISS0 (SPI)	–	–	P07MUX[1:0]/P04_07PMUX
P10	SCLK0 (UART)	–	–	P10MUX[1:0]/P10_13PMUX
P11	SOUT0 (UART)	–	–	P11MUX[1:0]/P10_13PMUX
P12	SIN0 (UART)	–	–	P12MUX[1:0]/P10_13PMUX
P13/EXCL1 (T16)*1	–	AIN7 (ADC10)	–	P13MUX[1:0]/P10_13PMUX
P14/EXCL2 (T16)*1	–	AIN6 (ADC10)	–	P14MUX[1:0]/P14_17PMUX
P15/EXCL3 (T16E)*1	–	AIN5 (ADC10)	–	P15MUX[1:0]/P14_17PMUX
P16	SCLK1 (UART)	AIN4 (ADC10)	–	P16MUX[1:0]/P14_17PMUX
P17	AIN3 (ADC10)	–	–	P17MUX[1:0]/P14_17PMUX
P20	AIN2 (ADC10)	–	–	P20MUX[1:0]/P20_23PMUX
P21	AIN1 (ADC10)	–	–	P21MUX[1:0]/P20_23PMUX
P22	AIN0 (ADC10)	–	–	P22MUX[1:0]/P20_23PMUX
P23	SENB0 (RFC)	–	–	P23MUX[1:0]/P20_23PMUX
P24	SENA0 (RFC)	–	–	P24MUX[1:0]/P24_27PMUX
P25	REF0 (RFC)	–	–	P25MUX[1:0]/P24_27PMUX
P26	RFIN0 (RFC)	–	–	P26MUX[1:0]/P24_27PMUX
P27	SOUT1 (UART)	RFIN1 (RFC)	–	P27MUX[1:0]/P24_27PMUX
P30	SIN1 (UART)	REF1 (RFC)		P30MUX[1:0]/P30_33PMUX
P31	SCL0 (I2CM)	SENA1 (RFC)	TOUTA5/CAPA5 (T16A2)*3	P31MUX[1:0]/P30_33PMUX
P32	SDA0 (I2CM)	SENB1 (RFC)	TOUTB5/CAPB5 (T16A2)*3	P32MUX[1:0]/P30_33PMUX
P33	SCL1 (I2CS)	SCL0 (I2CM)	TOUTA6/CAPA6 (T16A2)*3	P33MUX[1:0]/P30_33PMUX
P34	SDA1 (I2CS)	SDA0 (I2CM)	TOUTB6/CAPB6 (T16A2)*3	P34MUX[1:0]/P34_37PMUX
P35	FOUT1 (CLG)	#BFR (I2CS)	–	P35MUX[1:0]/P34_37PMUX
P36/EXCL5 (T16A2)*1	TOUT3 (T16E)	RFCLKO (RFC)	–	P36MUX[1:0]/P34_37PMUX
P37/EXCL6 (T16A2)*1	TOUTN3 (T16E)	LFRO (LCD)	TOUT4 (T8OSC1)	P37MUX[1:0]/P34_37PMUX
P40	FOUTH (CLG)	–	–	P40MUX[1:0]/P40_43PMUX
DSIO (DBG)	P41	–	–	P41MUX[1:0]/P40_43PMUX
DST2 (DBG)	P42	–	–	P42MUX[1:0]/P40_43PMUX
DCLK (DBG)	P43*5	–	–	P43MUX[1:0]/P40_43PMUX
P44	SCL1 (I2CS)	–	–	P44MUX[1:0]/P44_47PMUX*4
P45	SDA1 (I2CS)	–	–	P45MUX[1:0]/P44_47PMUX*4
P46	RFCLKO (RFC)	–	–	P46MUX[1:0]/P44_47PMUX*4
P47/EXCL5 (T16A2)*2	TOUT4 (T8OSC1)	–	–	P47MUX[1:0]/P44_47PMUX*4
P50/EXCL6 (T16A2)*2	SCLK1 (UART)	–	–	P50MUX[1:0]/P50_53PMUX*4
P51	SOUT1 (UART)	TOUTA5/CAPA5 (T16A2)*2	–	P51MUX[1:0]/P50_53PMUX*4
P52	SIN1 (UART)	TOUTB5/CAPB5 (T16A2)*2	–	P52MUX[1:0]/P50_53PMUX*4
P53	#BFR (I2CS)	–	–	P53MUX[1:0]/P50_53PMUX*4
P54	LFRO (LCD)	–	–	P54MUX[1:0]/P54_56PMUX*4
P55	–	TOUTA6/CAPA6 (T16A2)*2	–	P55MUX[1:0]/P54_56PMUX*4
P56	–	TOUTB6/CAPB6 (T16A2)*2	–	P56MUX[1:0]/P54_56PMUX*4

*1: The P02, P13–P15, P36, P37, P47, and P50 pins can also be used as an external clock input pin for the timer module by setting them to input mode. However, general-purpose input port function is also effective in this case. In the S1C17624, either P36 or P47 can be selected as the EXCL5 input port and either P37 or P50 can be selected as the EXCL6 input port using EXCL5S/P54_56PMUX register and EXCL6S/P54_56PMUX register. P36 and P37 are only available in the S1C17604.

∗2: Available only for S1C17624, ∗3: Available only for S1C17624/604, ∗4: Available only for S1C17624/622
∗5: P43 can only be used as an output port. For how to configure P43, see the P43MUX/P40_43PMUX register description.

At initial reset, each I/O port pin (P*xy*) is initialized for the default function ("Pin function 1" in Table 9.2.1).
For information on functions other than the I/O ports, see the descriptions of the peripheral modules indicated in parentheses. The sections below describe port functions with the pins set as general-purpose I/O ports.

9.3 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 9.3.1 Data Input/Output Status

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

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

Data input

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

Data output

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

Note: P43 can only be used as an output port. For how to configure P43, see the P43MUX/P40_43PMUX register description.

9.4 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

9.5 Input Interface Level

Some I/O ports allow software to select the input interface level from two types: CMOS Schmitt level and CMOS level.

Table 9.5.1 Input Interface Level

I/O port	S1C17624/622		S1C17604/602/621	
	CMOS Schmitt level (P*x*SM*y* = 1)	CMOS level (P*x*SM*y* = 0)	CMOS Schmitt level (P*x*SM*y* = 1)	CMOS level (P*x*SM*y* = 0)
P00–P07	○ (Selectable)	○ (Selectable)	○ (Selectable)	○ (Selectable)
P10–P15	○ (Selectable)	○ (Selectable)	○ (Selectable)	○ (Selectable)
P16–P17	○ (Fixed)	×	○ (Fixed)	×
P20–P27	○ (Fixed)	×	○ (Fixed)	×
P30–P37	○ (Fixed)	×	○ (Fixed)	×
P40	○ (Fixed)	×	○ (Fixed)	×
DSIO (P41)	○ (Fixed)	×	○ (Fixed)	×
DST2 (P42)	○ (Fixed)	×	○ (Fixed)	×
DCLK (P43)*	–	–	–	–
P44–P47	○ (Selectable)	○ (Selectable)		
P50–P52	○ (Selectable)	○ (Selectable)		
P53–P56	○ (Fixed)	×		

* DCLK (P43) is an output-only port.
○ (Selectable) The input interface can be selected using the P*x*SM*y* bit.
○ (Fixed) The input interface level is fixed at CMOS Schmitt level.

The input interface level for the I/O ports listed with "○ (Selectable)" can be selected individually for each bit using P*x*SM*y*/P*x*_SM register. Setting P*x*SM*y* to 1 (default) selects CMOS Schmitt level; setting to 0 selects CMOS level.

The input interface level for the I/O ports listed with "○ (Fixed)" is fixed at CMOS Schmitt level and cannot be switched to CMOS level. In the S1C17624/622/604, the P*x*SM*y* bits for these ports are read-only bits (always read as 1) and cannot be altered. In the S1C17602/621, both 1 and 0 can be written to and read from these bits. However, the input interface level cannot be switched.

9.6 P0 and P1 Port Chattering Filter Function

The P0 and P1 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 9.6.1 Chattering Filter Function Settings

P*x*CF1[2:0]/P*x*CF2[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 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.

9.7 Port Input Interrupt

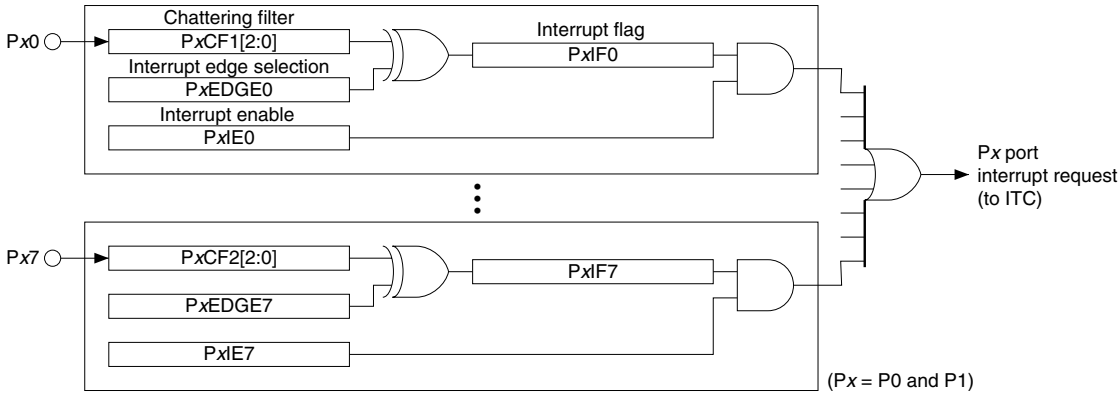


Figure 9.7.1 Port Input Interrupt Circuit Configuration

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

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 two interrupt requests from the P0 and P1 ports, and the P port module contains interrupt flags P*x*IF*y*/P*x*_IFLG register corresponding to the individual 16 ports to enable individual control of the 16 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*.

9.8 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 9.8.1 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.

9.9 Control Register Details

Table 9.9.1 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.
0x5204	P0_SM	P0 Port Schmitt Trigger Control Register	Controls the P0 port Schmitt trigger input.
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.
0x5214	P1_SM	P1 Port Schmitt Trigger Control Register	Controls the P1 port Schmitt trigger input.
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

Address	Register name		Function
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.
0x5224	P2_SM	P2 Port Schmitt Trigger Control Register	Controls the P2 port Schmitt trigger input.
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.
0x5234	P3_SM	P3 Port Schmitt Trigger Control Register	Controls the P3 port Schmitt trigger input.
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.
0x5244	P4_SM	P4 Port Schmitt Trigger Control Register	Controls the P4 port Schmitt trigger input.
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. *
0x5254	P5_SM	P5 Port Schmitt Trigger Control Register	Controls the P5 port Schmitt trigger input. *
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.
0x52a1	P04_07PMUX	P0[7:4] Port Function Select Register	Selects the P0[7:4] port functions.
0x52a2	P10_13PMUX	P1[3:0] Port Function Select Register	Selects the P1[3:0] port functions.
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_47PMUX	P4[7:4] Port Function Select Register	Selects the P4[7:4] port functions. *
0x52aa	P50_53PMUX	P5[3:0] Port Function Select Register	Selects the P5[3:0] port functions. *
0x52ab	P54_56PMUX	P5[6:4] Port Function Select Register	Selects the P5[6:4] port functions. *

* Available only for S1C17624/622

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: The P*x*IN*y* bits for unavailable ports 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: The P*x*OUT*y* bits for unavailable ports 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	

Notes: • The P*x*OEN*y* bits for unavailable ports are reserved and always read as 0.

• P43 can only be used as an output port. For how to configure P43, see the P43MUX/P40_43PMUX register description.

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 9.3.1 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: The P*x*PU*y* bits for unavailable ports 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 Schmitt Trigger Control Registers (P*x*_SM)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
P0 Port Schmitt Trigger Control Register (P0_SM)	0x5204 (8 bits)	D7–0	P0SM[7:0]	P0[7:0] port Schmitt trigger input enable	1	Enable (Schmitt)	0	Disable (CMOS)	1	R/W	
P1 Port Schmitt Trigger Control Register (P1_SM)	0x5214 (8 bits)	D7–6	P1SM[7:6]	P1[7:6] port Schmitt trigger input enable	1	Enable (Schmitt)	0	–	1	R	Always enabled
		D5–0	P1SM[5:0]	P1[5:0] port Schmitt trigger input enable	1	Enable (Schmitt)	0	Disable (CMOS)	1	R/W	
P2 Port Schmitt Trigger Control Register (P2_SM)	0x5224 (8 bits)	D7–0	P2SM[7:0]	P2[7:0] port Schmitt trigger input enable	1	Enable (Schmitt)	0	–	1	R	Always enabled
P3 Port Schmitt Trigger Control Register (P3_SM)	0x5234 (8 bits)	D7–0	P3SM[7:0]	P3[7:0] port Schmitt trigger input enable	1	Enable (Schmitt)	0	–	1	R	Always enabled
P4 Port Schmitt Trigger Control Register (P4_SM)	0x5244 (8 bits)	D7–4	P4SM[7:4]	P4[7:4] port Schmitt trigger input enable	1	Enable (Schmitt)	0	Disable (CMOS)	1	R/W	D[7:4] = reserved in S1C17604/602/621
		D3–0	P4SM[3:0]	P4[3:0] port Schmitt trigger input enable	1	Enable (Schmitt)	0	–	1	R	Always enabled
P5 Port Schmitt Trigger Control Register (P5_SM) (S1C17624/622)	0x5254 (8 bits)	D7	–	reserved	–				–	–	0 when being read.
		D6–3	P5SM[6:3]	P5[6:3] port Schmitt trigger input enable	1	Enable (Schmitt)	0	–	1	R	Always enabled
		D2–0	P5SM[2:0]	P5[2:0] port Schmitt trigger input enable	1	Enable (Schmitt)	0	Disable (CMOS)	1	R/W	

Note: The P*x*SM*y* bits for unavailable ports are reserved and always read as 0.

D[7:0] P*x*SM[7:0]: P*x*[7:0] Port Schmitt Trigger Input Enable Bits

Enables or disables the Schmitt trigger input buffer for each port.
1 (R/W): Enable (Schmitt) (default)
0 (R/W): Disable (CMOS level)

P*x*SM*y* is the Schmitt input control bit that corresponds directly to the P*xy* port. Setting to 1 enables the Schmitt input buffer, and setting to 0 uses the CMOS level input buffer.

The I/O ports shown below support only CMOS Schmitt input.
S1C17624/622: P1[7:6], P2[7:0], P3[7:0], P40, DSIO/P41, DST2/P42, DCLK/P43*, P5[6:3]
S1C17604/602/621: P1[7:6], P2[7:0], P3[7:0], P40, DSIO/P41, DST2/P42, DCLK/P43*
(* P43 is an output-only port.)

In the S1C17624/622/604, the P*x*SM*y* bits for these ports are read-only bits that are always read as 1. In the S1C17602/621, both 1 and 0 can be written to and read from these bits. However, the input interface level cannot be switched.

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 (8 bits)	D7–0	P*x*IE[7:0]	P*x*[7:0] port interrupt enable	1	Enable	0	Disable	0	R/W	

Note: The P*x*IMSK registers are available only for P0 and P1 ports.

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** (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: The P*x*EDGE registers are available only for P0 and P1 ports.

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** (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: The P*x*IFLG registers are available only for P0 and P1 ports.

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 corresponding to the individual 16 ports of P0 and P1 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.

P*x* Port Chattering Filter Control Registers (P*x*_CHAT)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
P*x* Port Chattering Filter Control Register (P*x*_CHAT)	0x5208 0x5218 (8 bits)	D7	–	reserved	–		–	–	0 when being read.
		D6–4	P*x*CF2[2:0]	P*x*[7:4] chattering filter time select	P*x*CF2[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	P*x*CF1[2:0]	P*x*[3:0] chattering filter time select	P*x*CF1[2:0]	Filter time	0x0	R/W	
					0x7	16384/fPCLK			
					0x6	8192/fPCLK			
					0x5	4096/fPCLK			
					0x4	2048/fPCLK			
					0x3	1024/fPCLK			
					0x2	512/fPCLK			
					0x1	256/fPCLK			
					0x0	None			

Note: The P*x*CHAT registers are available only for P0 and P1 ports.

D7 Reserved

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

Configures the chattering filter circuit for the P*x*[7:4] ports.

D3 Reserved

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

Configures the chattering filter circuit for the P*x*[3:0] ports.

The P0 and P1 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] and P*x*CF2[2:0], respectively.

Table 9.9.2 Chattering Filter Function Settings

P*x*CF1[2:0]/P*x*CF2[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 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* 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 9.9.3 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 0x525a (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: The P*x*IEN*y* bits for unavailable ports 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 9.3.1 for more information on port input/output status, including settings other than for the P*x*_IEN register.

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

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
P0[3:0] Port Function Select Register (P00_03PMUX)	0x52a0 (8 bits)	D7–6	P03MUX[1:0]	P03 port function select	P03MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	#ADTRG			
					0x0	P03			
		D5–4	P02MUX[1:0]	P02 port function select	P02MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	reserved			
					0x0	P02/EXCL0			
		D3–2	P01MUX[1:0]	P01 port function select	P01MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	REMI			
					0x0	P01			
		D1–0	P00MUX[1:0]	P00 port function select	P00MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	REMO			
					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): Reserved
0x1 (R/W): #ADTRG (ADC10)
0x0 (R/W): P03 (default)

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

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): Reserved
0x0 (R/W): P02/EXCL0 (T16 Ch.0) (default)

To use the P02 pin for EXCL0 input, P0OEN2/P0_OEN register must be set to 0 and P0IEN2/P0_IEN register must be set to 1.

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

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): REMI (REMC)
0x0 (R/W): P01 (default)

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

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): REMO (REMC)
0x0 (R/W): P00 (default)

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

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
P0[7:4] Port Function Select Register (P04_07PMUX)	0x52a1 (8 bits)	D7–6	P07MUX[1:0]	P07 port function select	P07MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	#SPISS0			
					0x0	P07			
		D5–4	P06MUX[1:0]	P06 port function select	P06MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	SDI0			
					0x0	P06			
		D3–2	P05MUX[1:0]	P05 port function select	P05MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	SDO0			
					0x0	P05			
		D1–0	P04MUX[1:0]	P04 port function select	P04MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	SPICLK0			
					0x0	P04			

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

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

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): #SPISS0 (SPI Ch.0)
0x0 (R/W): P07 (default)

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

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

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

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): SDO0 (SPI Ch.0)
0x0 (R/W): P05 (default)

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

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): SPICLK0 (SPI Ch.0)
0x0 (R/W): P04 (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	AIN7			
					0x1	reserved			
					0x0	P13/EXCL1			
		D5–4	**P12MUX[1:0]**	P12 port function select	P12MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	SIN0			
					0x0	P12			
		D3–2	**P11MUX[1:0]**	P11 port function select	P11MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	SOUT0			
					0x0	P11			
		D1–0	**P10MUX[1:0]**	P10 port function select	P10MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	SCLK0			
					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): AIN7 (ADC10)
0x1 (R/W): Reserved
0x0 (R/W): P13/EXCL1 (T16 Ch.1) (default)

To use the P13 pin for EXCL1 input, P1OEN3/P1_OEN register must be set to 0 and P1IEN3/P1_IEN register must be set to 1.

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

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): SIN0 (UART Ch.0)
0x0 (R/W): P12 (default)

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

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): SOUT0 (UART Ch.0)
0x0 (R/W): P11 (default)

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

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): SCLK0 (UART Ch.0)
0x0 (R/W): P10 (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	AIN3			
					0x0	P17			
		D5–4	P16MUX[1:0]	P16 port function select	P16MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	AIN4			
					0x1	SCLK1			
					0x0	P16			
		D3–2	P15MUX[1:0]	P15 port function select	P15MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	AIN5			
					0x1	reserved			
					0x0	P15/EXCL3			
		D1–0	P14MUX[1:0]	P14 port function select	P14MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	AIN6			
					0x1	reserved			
					0x0	P14/EXCL2			

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): AIN3 (ADC10)
0x0 (R/W): P17 (default)

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

0x3 (R/W): Reserved
0x2 (R/W): AIN4 (ADC10)
0x1 (R/W): SCLK1 (UART Ch.1)
0x0 (R/W): P16 (default)

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

0x3 (R/W): Reserved
0x2 (R/W): AIN5 (ADC10)
0x1 (R/W): Reserved
0x0 (R/W): P15/EXCL3 (T16E Ch.0) (default)

To use the P15 pin for EXCL3 input, P1OEN5/P1_OEN register must be set to 0 and P1IEN5/P1_IEN register must be set to 1.

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

0x3 (R/W): Reserved
0x2 (R/W): AIN6 (ADC10)
0x1 (R/W): Reserved
0x0 (R/W): P14/EXCL2 (T16 Ch.2) (default)

To use the P14 pin for EXCL2 input, P1OEN4/P1_OEN register must be set to 0 and P1IEN4/P1_IEN register must be set to 1.

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	SENB0			
					0x0	P23			
		D5–4	P22MUX[1:0]	P22 port function select	P22MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	AIN0			
					0x0	P22			
		D3–2	P21MUX[1:0]	P21 port function select	P21MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	AIN1			
					0x0	P21			
		D1–0	P20MUX[1:0]	P20 port function select	P20MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	AIN2			
					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): SENB0 (RFC)
0x0 (R/W): P23 (default)

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

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): AIN0 (ADC10)
0x0 (R/W): P22 (default)

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

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): AIN1 (ADC10)
0x0 (R/W): P21 (default)

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

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): AIN2 (ADC10)
0x0 (R/W): P20 (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	RFIN1			
					0x1	SOUT1			
					0x0	P27			
		D5–4	P26MUX[1:0]	P26 port function select	P26MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	RFIN0			
					0x0	P26			
		D3–2	P25MUX[1:0]	P25 port function select	P25MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	REF0			
					0x0	P25			
		D1–0	P24MUX[1:0]	P24 port function select	P24MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	SENA0			
					0x0	P24			

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): RFIN1 (RFC)
0x1 (R/W): SOUT1 (UART Ch.1)
0x0 (R/W): P27 (default)

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

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): RFIN0 (RFC)
0x0 (R/W): P26 (default)

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

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): REF0 (RFC)
0x0 (R/W): P25 (default)

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

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): SENA0 (RFC)
0x0 (R/W): P24 (default)

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	TOUTA6/CAPA6: S1C17624/604 only
					0x3	TOUTA6/CAPA6			
					0x2	SCL0			
					0x1	SCL1			
					0x0	P33			
		D5–4	P32MUX[1:0]	P32 port function select	P32MUX[1:0]	Function	0x0	R/W	TOUTB5/CAPB5: S1C17624/604 only
					0x3	TOUTB5/CAPB5			
					0x2	SENB1			
					0x1	SDA0			
					0x0	P32			
		D3–2	P31MUX[1:0]	P31 port function select	P31MUX[1:0]	Function	0x0	R/W	TOUTA5/CAPA5: S1C17624/604 only
					0x3	TOUTA5/CAPA5			
					0x2	SENA1			
					0x1	SCL0			
					0x0	P31			
		D1–0	P30MUX[1:0]	P30 port function select	P30MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	REF1			
					0x1	SIN1			
					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): TOUTA6 (T16A2 Ch.1 comparator mode)
or CAPA6 (T16A2 Ch.1 capture mode) (Selectable only in the S1C17624/604)
0x2 (R/W): SCL0 (I2CM)
0x1 (R/W): SCL1 (I2CS)
0x0 (R/W): P33 (default)

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

0x3 (R/W): TOUTB5 (T16A2 Ch.0 comparator mode)
or CAPB5 (T16A2 Ch.0 capture mode) (Selectable only in the S1C17624/604)
0x2 (R/W): SENB1 (RFC)
0x1 (R/W): SDA0 (I2CM)
0x0 (R/W): P32 (default)

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

0x3 (R/W): TOUTA5 (T16A2 Ch.0 comparator mode)
or CAPA5 (T16A2 Ch.0 capture mode) (Selectable only in the S1C17624/604)
0x2 (R/W): SENA1 (RFC)
0x1 (R/W): SCL0 (I2CM)
0x0 (R/W): P31 (default)

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

0x3 (R/W): Reserved
0x2 (R/W): REF1 (RFC)
0x1 (R/W): SIN1 (UART Ch.1)
0x0 (R/W): P30 (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	EXCL6: S1C17624/604 only (Set EXCL6S to 0 in S1C17624.)
					0x3	TOUT4			
					0x2	LFRO			
					0x1	TOUTN3			
					0x0	P37/EXCL6			
		D5–4	P36MUX[1:0]	P36 port function select	P36MUX[1:0]	Function	0x0	R/W	EXCL5: S1C17624/604 only (Set EXCL5S to 0 in S1C17624.)
					0x3	reserved			
					0x2	RFCLKO			
					0x1	TOUT3			
					0x0	P36/EXCL5			
		D3–2	P35MUX[1:0]	P35 port function select	P35MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	#BFR			
					0x1	FOUT1			
					0x0	P35			
		D1–0	P34MUX[1:0]	P34 port function select	P34MUX[1:0]	Function	0x0	R/W	TOUTB6/CAPB6: S1C17624/604 only
					0x3	TOUTB6/CAPB6			
					0x2	SDA0			
					0x1	SDA1			
					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): TOUT4 (T8OSC1)
0x2 (R/W): LFRO (LCD)
0x1 (R/W): TOUTN3 (T16E Ch.0)
0x0 (R/W): P37/EXCL6 (T16A2 Ch.1) (default) (EXCL6 is available only in the S1C17624/604.)

To use the P37 pin for EXCL6 input, P3OEN7/P3_OEN register must be set to 0 and P3IEN7/P3_IEN register must be set to 1. In addition to these settings, EXCL6S/P54_56PMUX register must be set to 0 in the S1C17624.

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

0x3 (R/W): Reserved
0x2 (R/W): RFCLKO (RFC)
0x1 (R/W): TOUT3 (T16E Ch.0)
0x0 (R/W): P36/EXCL5 (T16A2 Ch.0) (default) (EXCL5 is available only in the S1C17624/604.)

To use the P36 pin for EXCL5 input, P3OEN6/P3_OEN register must be set to 0 and P3IEN6/P3_IEN register must be set to 1. In addition to these settings, EXCL5S/P54_56PMUX register must be set to 0 in the S1C17624.

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

0x3 (R/W): Reserved
0x2 (R/W): #BFR (I2CS)
0x1 (R/W): FOUT1 (CLG)
0x0 (R/W): P35 (default)

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

0x3 (R/W): TOUTB6 (T16A2 Ch.1 comparator mode)
or CAPB6 (T16A2 Ch.1 capture mode) (Selectable only in the S1C17624/604)
0x2 (R/W): SDA0 (I2CM)
0x1 (R/W): SDA1 (I2CS)
0x0 (R/W): P34 (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	reserved			
					0x1	P43			
					0x0	DCLK			
		D5–4	P42MUX[1:0]	P42 port function select	P42MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	P42			
					0x0	DST2			
		D3–2	P41MUX[1:0]	P41 port function select	P41MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	P41			
					0x0	DSIO			
		D1–0	P40MUX[1:0]	P40 port function select	P40MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	FOUTH			
					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): Reserved
0x1 (R/W): P43
0x0 (R/W): DCLK (DBG) (default)

P43 is an output-only port and no external signal cannot be input.
To use P43 as a general-purpose output port, make the following settings:
1. Set P4OEN3/P4_OEN register to 1 (output).
2. Set P43MUX[1:0] to 0x1 (P43).

When the P43 output port is not used (or used as the DCLK port), make the following settings:
1. Set P43MUX[1:0] to 0x0 (DCLK).
2. Set P4OEN3 to 0 (input).

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

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): P42
0x0 (R/W): DST2 (DBG) (default)

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

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): P41
0x0 (R/W): DSIO (DBG) (default)

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

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): FOUTH (CLG)
0x0 (R/W): P40 (default)

P4[7:4] Port Function Select Register (P44_47PMUX)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
P4[7:4] Port Function Select Register (P44_47PMUX)	**0x52a9** (8 bits)	D7–6	**P47MUX[1:0]**	P47 port function select	P47MUX[1:0]	Function	0x0	R/W	EXCL5: S1C17624 only (Set EXCL5S to 1.)
					0x3	reserved			
					0x2	reserved			
					0x1	TOUT4			
					0x0	P47/EXCL5			
S1C17624/622		D5–4	**P46MUX[1:0]**	P46 port function select	P46MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	RFCLKO			
					0x0	P46			
		D3–2	**P45MUX[1:0]**	P45 port function select	P45MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	SDA1			
					0x0	P45			
		D1–0	**P44MUX[1:0]**	P44 port function select	P44MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	SCL1			
					0x0	P44			

Note: This register is effective only in the S1C17624/622.

The P44 to P47 port pins are shared with the peripheral module pins. This register is used to select how the pins are used.

D[7:6] P47MUX[1:0]: P47 Port Function Select Bits

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): TOUT4 (T8OSC1)
0x0 (R/W): P47/EXCL5 (T16A2 Ch.0) (default) (EXCL5 is available only in the S1C17624.)

To use the P47 pin for EXCL5 input, set P4OEN7/P4_OEN register to 0, P4IEN7/P4_IEN register to 1 and EXCL5S/P54_56PMUX register to 1.

D[5:4] P46MUX[1:0]: P46 Port Function Select Bits

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): RFCLKO (RFC)
0x0 (R/W): P46 (default)

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

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): SDA1 (I2CS)
0x0 (R/W): P45 (default)

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

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): SCL1 (I2CS)
0x0 (R/W): P44 (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	
					0x3	reserved			
					0x2	reserved			
					0x1	#BFR			
					0x0	P53			
S1C17624/622		D5–4	**P52MUX[1:0]**	P52 port function select	P52MUX[1:0]	Function	0x0	R/W	TOUTB5/CAPB5: S1C17624 only
					0x3	reserved			
					0x2	TOUTB5/CAPB5			
					0x1	SIN1			
					0x0	P52			
		D3–2	**P51MUX[1:0]**	P51 port function select	P51MUX[1:0]	Function	0x0	R/W	TOUTA5/CAPA5: S1C17624 only
					0x3	reserved			
					0x2	TOUTA5/CAPA5			
					0x1	SOUT1			
					0x0	P51			
		D1–0	**P50MUX[1:0]**	P50 port function select	P50MUX[1:0]	Function	0x0	R/W	EXCL6: S1C17624 only (Set EXCL6S to 1.)
					0x3	reserved			
					0x2	reserved			
					0x1	SCLK1			
					0x0	P50/EXCL6			

Note: This register is effective only in the S1C17624/622.

The P50 to P53 port pins 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): #BFR (I2CS)
0x0 (R/W): P53 (default)

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

0x3 (R/W): Reserved
0x2 (R/W): TOUTB5 (T16A2 Ch.0 comparator mode)
or CAPB5 (T16A2 Ch.0 capture mode) (Selectable only in the S1C17624)
0x1 (R/W): SIN1 (UART Ch.1)
0x0 (R/W): P52 (default)

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

0x3 (R/W): Reserved
0x2 (R/W): TOUTA5 (T16A2 Ch.0 comparator mode)
or CAPA5 (T16A2 Ch.0 capture mode) (Selectable only in the S1C17624)
0x1 (R/W): SOUT1 (UART Ch.1)
0x0 (R/W): P51 (default)

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

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): SCLK1 (UART Ch.1)
0x0 (R/W): P50/EXCL6 (T16A Ch.1) (default) (EXCL6 is available only in the S1C17624.)

To use the P50 pin for EXCL6 input, set P5OEN0/P5_OEN register to 0, P5IEN0/P5_IEN register to 1 and EXCL6S/P54_56PMUX register to 1.

P5[6:4] Port Function Select Register (P54_56PMUX)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
P5[6:4] Port Function Select Register (P54_56PMUX) S1C17624/622	0x52ab (8 bits)	D7	EXCL6S	EXCL6 input select	1 P50/EXCL6 0 P37/EXCL6	0	R/W	S1C17624 only
		D6	EXCL5S	EXCL5 input select	1 P47/EXCL5 0 P36/EXCL5	0	R/W	
		D5–4	P56MUX[1:0]	P56 port function select	P56MUX[1:0] Function 0x3 reserved 0x2 TOUTB6/CAPB6 0x1 reserved 0x0 P56	0x0	R/W	
		D3–2	P55MUX[1:0]	P55 port function select	P55MUX[1:0] Function 0x3 reserved 0x2 TOUTA6/CAPA6 0x1 reserved 0x0 P55	0x0	R/W	
		D1–0	P54MUX[1:0]	P54 port function select	P54MUX[1:0] Function 0x3 reserved 0x2 reserved 0x1 LFRO 0x0 P54	0x0	R/W	

Note: This register is effective only in the S1C17624/622.

The P54 to P56 port pins are shared with the peripheral module pins. This register is used to select how the pins are used.

D7 EXCL6S: EXCL6 Input Select Bit

Selects a port to be used as the EXCL6 input. (Selectable only in the S1C17624)
1 (R/W): P50/EXCL6
0 (R/W): P37/EXCL6 (default)

D6 EXCL5S: EXCL5 Input Select Bit

Selects a port to be used as the EXCL5 input. (Selectable only in the S1C17624)
1 (R/W): P47/EXCL5
0 (R/W): P36/EXCL5 (default)

D[5:4] P56MUX[1:0]: P56 Port Function Select Bits

0x3 (R/W): Reserved
0x2 (R/W): TOUTB6 (T16A2 Ch.1 comparator mode)
or CAPB6 (T16A2 Ch.1 capture mode) (Selectable only in the S1C17624)
0x1 (R/W): Reserved
0x0 (R/W): P56 (default)

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

0x3 (R/W): Reserved
0x2 (R/W): TOUTA6 (T16A2 Ch.1 comparator mode)
or CAPA6 (T16A2 Ch.1 capture mode) (Selectable only in the S1C17624)
0x1 (R/W): Reserved
0x0 (R/W): P55 (default)

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

0x3 (R/W): Reserved
0x2 (R/W): Reserved
0x1 (R/W): LFRO (LCD)
0x0 (R/W): P54 (default)

10 Fine Mode 8-bit Timers (T8F)

10.1 T8F Module Overview

The S1C17624/604/622/602/621 includes two-channel fine mode 8-bit timer module (T8F).
The features of the T8F module are listed below.

- 8-bit presettable down counter with an 8-bit reload data register for setting the preset value
- Generates the UART operating clock from the counter underflow signals.
- Generates underflow interrupt signals to the interrupt controller (ITC).
- Any desired time intervals and serial transfer rates can be programmed by selecting an appropriate count clock and preset value.
- Fine mode is provided to minimize transfer rate errors.

Figure 10.1.1 shows the T8F configuration.

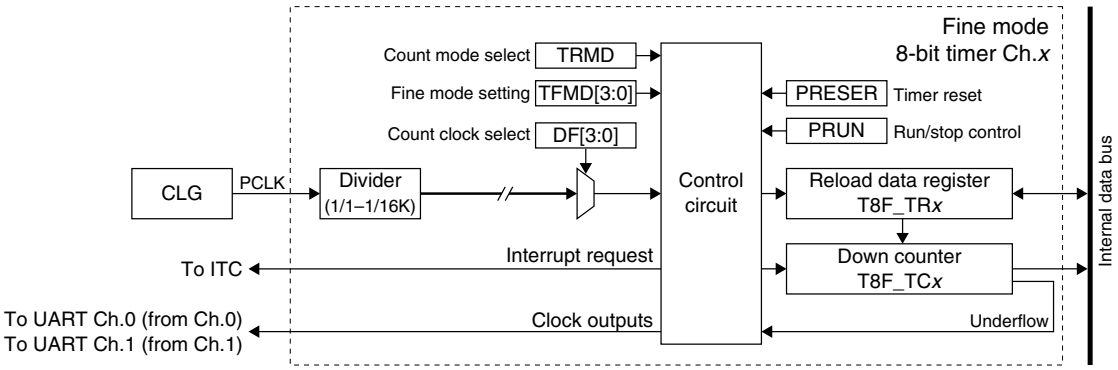


Figure 10.1.1 T8F Configuration (1 Channel)

Each channel of the T8F module consists of an 8-bit presettable down counter and an 8-bit reload data register holding the preset value. The timer counts down from the initial value set in the reload data register and outputs an underflow signal when the counter underflows. The underflow signal is used to generate an interrupt and an internal serial interface clock. The underflow cycle can be programmed by selecting the count clock and reload data, enabling the application program to obtain time intervals and serial transfer rates as required.

Note: Both T8F 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: T8F_CTL*x* register
Ch.0: T8F_CTL0 register
Ch.1: T8F_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]/T8F_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 T8F module features two count modes: repeat mode and one-shot mode. These modes are selected using TRMD/T8F_CTL*x* register.

Repeat mode (TRMD = 0, default)

Setting TRMD to 0 sets T8F 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. T8F 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 T8F 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. T8F should be set to this mode to set a specific wait time.

10.4 Reload Data Register and Underflow Cycle

The reload data register T8F_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.

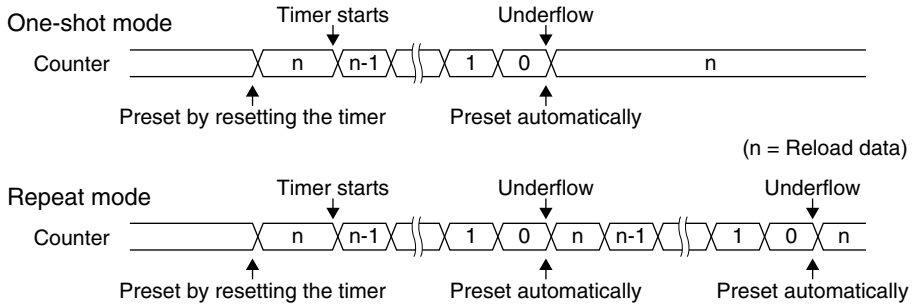


Figure 10.4.1 Preset Timing

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 (0–255)

10.5 Timer Reset

The timer is reset by writing 1 to PRESER/T8F_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/T8F_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.

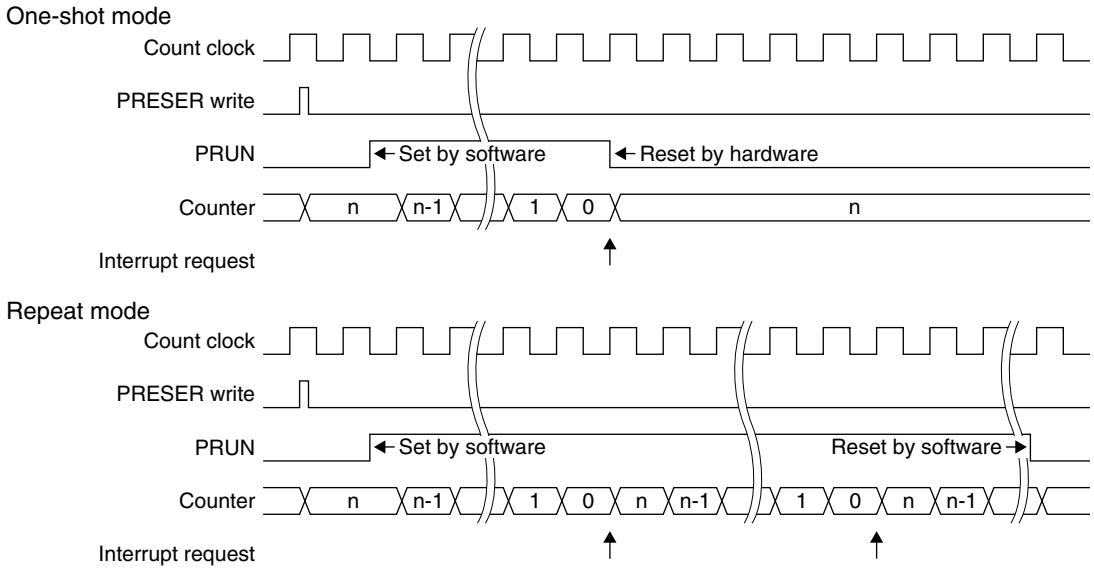


Figure 10.6.1 Count Operation

10.7 T8F Output Signals

The T8F module outputs underflow pulses when the counter underflows.
These pulses are used for timer interrupt requests.
These pulses are also used to generate a UART clock. The clock generated is sent to the internal peripheral module, as shown below.
T8F Ch.0 output clock → UART Ch.0
T8F Ch.1 output clock → UART Ch.1

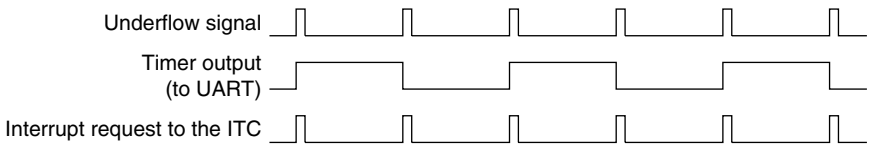


Figure 10.7.1 T8F Output Clock

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

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

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

bps: Transfer rate (bits/second)
clk_in: Count clock (PCLK/1 to PCLK/16384) frequency [Hz]
T8F_TR: Reload data (0 to 255)
TFMD: Fine mode setting (0 to 15)

10.8 Fine Mode

Fine mode provides a function that minimizes transfer rate errors.
T8F can output a programmable clock signal for use as the UART serial transfer clock. The timer output clock can be set to the required frequency by selecting the appropriate 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]/T8F_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.

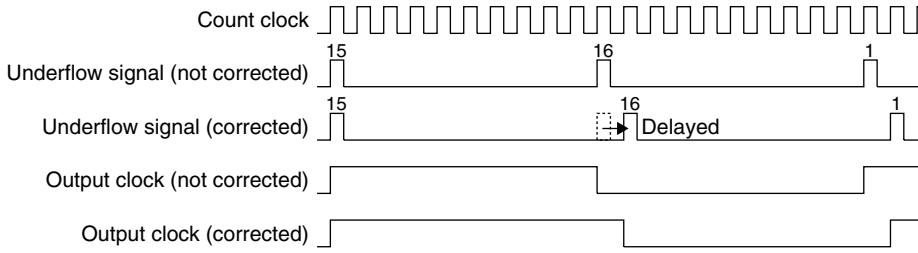


Figure 10.8.1 Delay Cycle Insertion in Fine Mode

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

10.9 T8F Interrupts

Each channel of the T8F module outputs an interrupt request to the interrupt controller (ITC) when the counter underflows.

Underflow interrupt

When the counter underflows, the interrupt flag T8FIF/T8F_INT*x* register, which is provided for each channel in the T8F module, is set to 1. At the same time, an interrupt request is sent to the ITC if T8FIE/T8F_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 T8FIE 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 T8F module interrupt flag T8FIF must be reset in the interrupt handler routine after a T8F interrupt has occurred to prevent recurring interrupts.

• Reset T8FIF before enabling T8F interrupts with T8FIE to prevent occurrence of unwanted interrupt. T8FIF is reset by writing 1.

10.10 Control Register Details

Table 10.10.1 List of T8F Registers

Address	Register name		Function
0x4200	T8F_CLK0	T8F Ch.0 Count Clock Select Register	Selects a count clock.
0x4202	T8F_TR0	T8F Ch.0 Reload Data Register	Sets reload data.
0x4204	T8F_TC0	T8F Ch.0 Counter Data Register	Counter data
0x4206	T8F_CTL0	T8F Ch.0 Control Register	Sets the timer mode and starts/stops the timer.
0x4208	T8F_INT0	T8F Ch.0 Interrupt Control Register	Controls the interrupt.
0x4280	T8F_CLK1	T8F Ch.1 Count Clock Select Register	Selects a count clock.
0x4282	T8F_TR1	T8F Ch.1 Reload Data Register	Sets reload data.
0x4284	T8F_TC1	T8F Ch.1 Counter Data Register	Counter data
0x4286	T8F_CTL1	T8F Ch.1 Control Register	Sets the timer mode and starts/stops the timer.
0x4288	T8F_INT1	T8F Ch.1 Interrupt Control Register	Controls the interrupt.

The T8F 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.

T8F Ch.*x* Count Clock Select Registers (T8F_CLK*x*)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
T8F Ch.*x* Count	**0x4200**	D15–4	**–**	reserved	–		–	–	0 when being read.
Clock Select	**0x4280**	D3–0	**DF[3:0]**	Count clock division ratio select	DF[3:0]	Division ratio	0x0	R/W	Source clock = PCLK
Register	(16 bits)				0xf	reserved			
(T8F_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.

T8F Ch.*x* Reload Data Registers (T8F_TR*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T8F Ch.*x*	**0x4202**	D15–8	**–**	reserved	–	–	–	0 when being read.
Reload Data	**0x4282**	D7–0	**TR[7:0]**	Reload data	0x0 to 0xff	0x0	R/W	
Register	(16 bits)			TR7 = MSB				
(T8F_TR*x*)				TR0 = LSB				

D[15:8] Reserved

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

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

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

T8F Ch.*x* Counter Data Registers (T8F_TC*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T8F Ch.*x* Counter Data Register (T8F_TC*x*)	0x4204	D15–8	–	reserved	–	–	–	0 when being read.
	0x4284 (16 bits)	D7–0	TC[7:0]	Counter data TC7 = MSB TC0 = LSB	0x0 to 0xff	0xff	R	

D[15:8] Reserved

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

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

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

T8F Ch.*x* Control Registers (T8F_CTL*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T8F Ch.*x* Control Register (T8F_CTL*x*)	0x4206	D15–12	–	reserved	–	–	–	0 when being read.
	0x4286 (16 bits)	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.

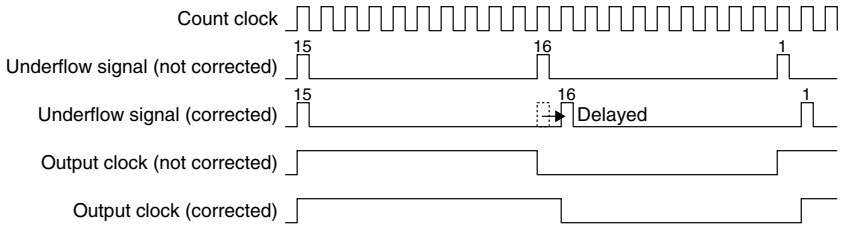


Figure 10.10.1 Delay Cycle Insertion in Fine Mode

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

D[3:2] Reserved

D1 PRESER: Timer Reset Bit

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

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

D0 PRUN: Timer Run/Stop Control Bit

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

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

T8F Ch.*x* Interrupt Control Registers (T8F_INT*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T8F Ch.*x* Interrupt Control Register (T8F_INT*x*)	0x4208 0x4288 (16 bits)	D15–9	–	reserved	–	–	–	0 when being read.
		D8	T8FIE	T8F interrupt enable	1 Enable 0 Disable	0	R/W	
		D7–1	–	reserved	–	–	–	0 when being read.
		D0	T8FIF	T8F 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 T8FIE: T8F 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 T8FIE to 1 enables T8F interrupt requests to the ITC; setting to 0 disables interrupts.

D[7:1] Reserved

D0 T8FIF: T8F 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

T8FIF is the T8F module interrupt flag that is set to 1 when the counter underflows.
T8FIF is reset by writing 1.

11 16-bit Timers (T16)

11.1 T16 Module Overview

The S1C17624/604/622/602/621 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.
- Includes an event counter function and a pulse width measurement function using external signal input from the I/O port pins.
- Generates underflow interrupt signals to the interrupt controller (ITC).
- Any desired time intervals and serial transfer rates can be programmed by selecting an appropriate count clock and preset value.

Figure 11.1.1 shows the T16 configuration.

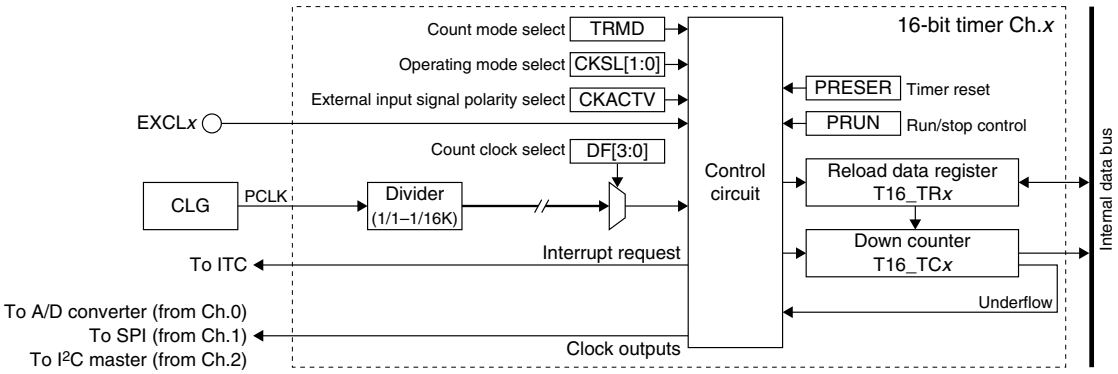


Figure 11.1.1 T16 Configuration (1 Channel)

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

11.2 T16 Input Pins

Table 11.2.1 lists the input pins for the T16 module.

Table 11.2.1 List of T16 Pins

Pin name	I/O	Qty	Function
EXCL0	I	1	Ch.0 external clock input pin Inputs an external clock for the event counter function or an external signal for measuring the pulse width.
EXCL1	I	1	Ch.1 external clock input pin Inputs an external clock for the event counter function or an external signal for measuring the pulse width.
EXCL2	I	1	Ch.2 external clock input pin Inputs an external clock for the event counter function or an external signal for measuring the pulse width.

The T16 input pins (EXCL*x*) are shared with I/O ports. Setting the port to input mode enables it to be used as the T16 input pin with a general-purpose input function. For detailed information on the port control, see the "I/O Ports (P)" chapter.

11.3 Operating Modes

The T16 module has the following three operating modes:
1. Internal clock mode (normal timer for counting an internal clock)
2. External clock mode (functions as an event counter)
3. Pulse width measurement mode (measures the external input pulse width using an internal clock)

The operating mode is selected using CKSL[1:0]/T16_CTL*x* register.

Table 11.3.1 Operating Mode Selection

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

(Default: 0x0)

11.3.1 Internal Clock Mode

Internal clock mode uses a divided PCLK clock as the count clock.
The timer counts down from the initial value set in the reload data register and outputs an underflow signal when the counter underflows. The underflow signal is used to generate an interrupt and an internal serial interface clock.
The time until underflow occurs can be finely programmed by selecting the clock division ratio and initial counter value, making it useful for serial transfer clock generation and sporadic time measurement.

Count clock selection

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 11.3.1.1 PCLK Division Ratio Selection

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

(Default: 0x0)

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

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

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

11.3.2 External Clock Mode

External clock mode uses the clock and pulses input via the I/O port as the count clock. This enables T16 to be used as an event counter. Timer operations other than the input clock are the same as for internal clock mode.

External clock input port

To input an external clock to T16, the I/O port shared with an EXCL*x* input must be set to input mode in advance. For detailed information on the port control, see the "I/O Ports (P)" chapter.

Signal polarity selection

CKACTV/T16_CTL*x* register is used in external clock mode to select either the falling edge or rising edge of the input signal for counting.

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

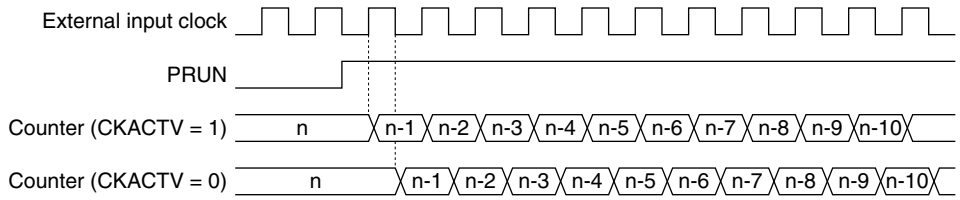


Figure 11.3.2.1 Counting in External Clock Mode

11.3.3 Pulse Width Measurement Mode

In pulse width measurement mode, when pulses with the specified polarity are input from the external clock port, the internal clock is fed only while the input pulse is active, enabling counting. This enables T16 to generate an interrupt when a pulse with the specified width or greater, or to measure the input pulse width.

Pulse input port

The I/O port (EXCL*x*) used for external pulse input is the same as for external clock mode.

Input pulses by enabling the EXCL*x* port function corresponding to the timer channel to be used.

Count clock selection

Counting uses the divided PCLK clock selected by DF[3:0]/T16_CLK*x* register in the same way as for internal clock mode. Select the clock to suit approximate input pulse widths and counting accuracy (see Table 11.3.1.1).

Signal polarity selection

CKACTV/T16_CTL*x* register is used to select the active level for the pulses counted. Setting CKACTV to 1 (default) will measure the high period and setting it to 0 will measure the low period.

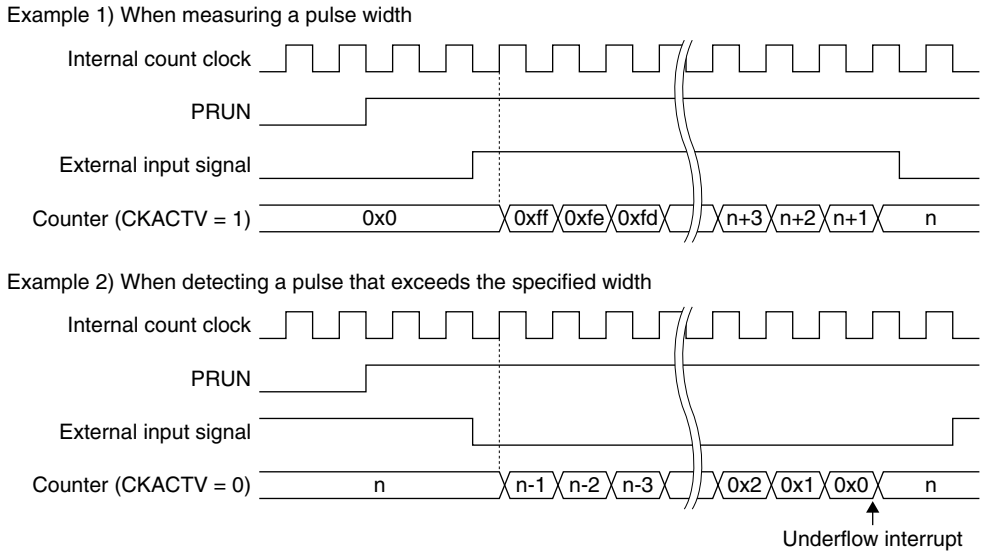


Figure 11.3.3.1 Count Operation in Pulse Width Measurement Mode

11.4 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 or for pulse width measurement.

11.5 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.

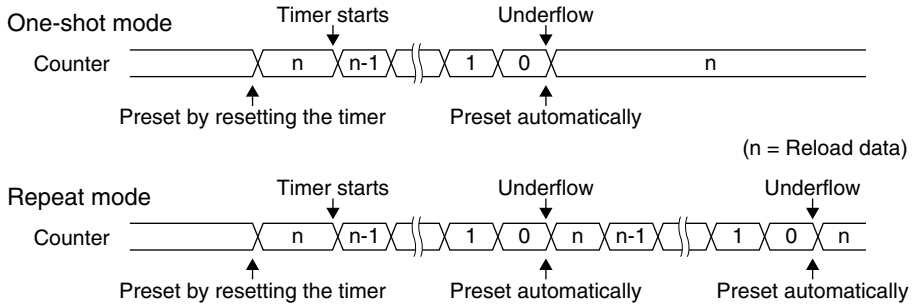


Figure 11.5.1 Preset Timing

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)

11.6 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.

11.7 Timer RUN/STOP Control

Make the following settings before starting the timer.

(1) Select the operating mode (internal clock mode, external clock mode, or pulse width measurement mode). See Section 11.3.
(2) For internal clock or pulse width measurement mode, select the count clock (divided PCLK clock). See Section 11.3.1.
(3) Set the count mode (one-shot or repeat). See Section 11.4.
(4) Calculate the initial counter value and set it to the reload data register. See Section 11.5.
(5) Reset the timer to preset the counter to the initial value. See Section 11.6.
(6) When using timer interrupts, set the interrupt level and enable interrupts for the relevant timer channel. See Section 11.9.

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.

In pulse width measurement mode, the timer counts only while PRUN is set to 1 and the external input signal is at the specified active level. When the external input signal becomes inactive, the timer stops counting and retains the counter value until the next active level input. (See Figure 11.3.3.1.)

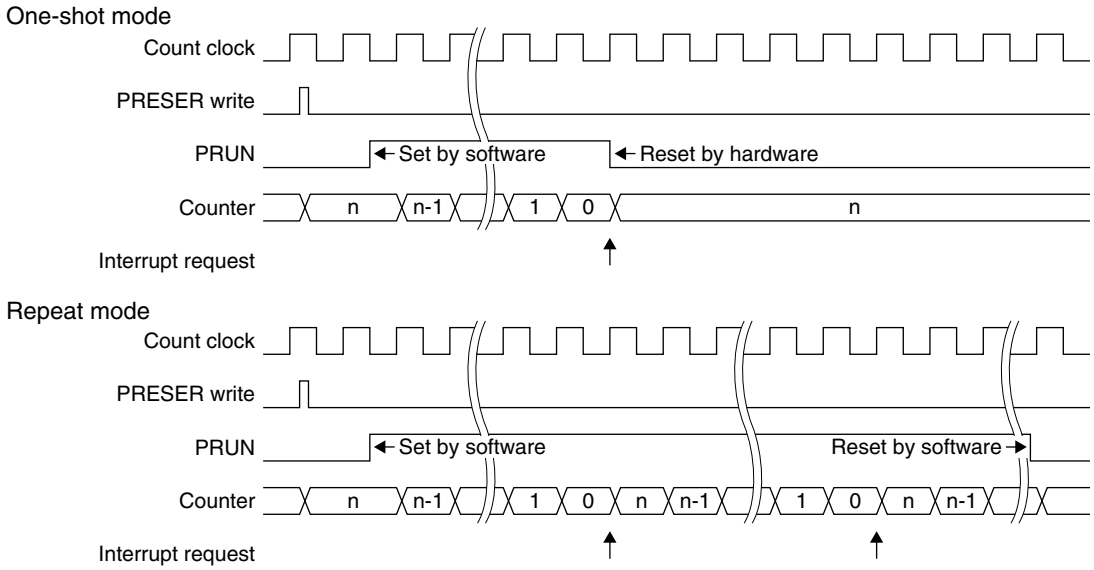


Figure 11.7.1 Count Operation

11.8 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 $TR = \frac{ct\_clk \times adi}{2} - 1$

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

11.9 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.

11.10 Control Register Details

Table 11.10.1 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 11.10.2 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.

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**	D15–11	**–**	reserved	–	–	–	0 when being read.
Control Register	**0x4246**	D10	**CKACTV**	External clock active level select	1 High 0 Low	1	R/W	
(T16_CTL*x*)	**0x4266**	D9–8	**CKSL[1:0]**	Operating mode select	CKSL[1:0] Mode: 0x3 reserved; 0x2 Pulse width; 0x1 External clock; 0x0 Internal clock	0x0	R/W	
	(16 bits)	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[15:11] Reserved

D10 CKACTV: External Clock Active Level Select Bit

Selects the external input pulse polarity or external clock counting edge.
1 (R/W): Active high/rising edge (default)
0 (R/W): Active low/falling edge

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

D[9:8] CKSL[1:0]: Operating Mode Select Bits

Selects the 16-bit timer operating mode.

Table 11.10.3 Operating Mode Selection

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

(Default: 0x0)

Internal clock mode uses a divided PCLK clock as the count clock. The timer counts down from the initial value set in the reload data register and outputs an underflow signal when the counter underflows. The underflow signal is used to generate an interrupt and an internal serial interface clock. The time until underflow occurs can be finely programmed by selecting the clock division ratio and initial counter value, making it useful for serial transfer clock generation and sporadic time measurement.

External clock mode uses the clock and pulses input via the EXCL*x* port as the count clock. This enables T16 to be used as an event counter. Timer operations other than the input clock are the same as for internal clock mode.

In pulse width measurement mode, when pulses with the specified polarity are input from the external clock port, the internal clock is fed only while the input pulse is active, enabling counting. This enables T16 to generate an interrupt when a pulse with the specified width or greater, or to measure the input pulse width.

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 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 or for pulse width measurement.

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.

12 16-bit PWM Timer (T16E)

12.1 T16E Module Overview

The S1C17624/604/622/602/621 includes a 16-bit PWM timer module (T16E) with one timer channel (Ch.0). The features of T16E are listed below.

- 16-bit up counter with a comparator
- The counter value can be compared with two specified comparison values by the comparator.
- The comparison results generate two different types of interrupts.
- Can generate inverted and non-inverted PWM signals from the comparison results and output them outside the IC.
- Supports event counter function using an external clock.

Figure 12.1.1 shows the T16E module configuration.

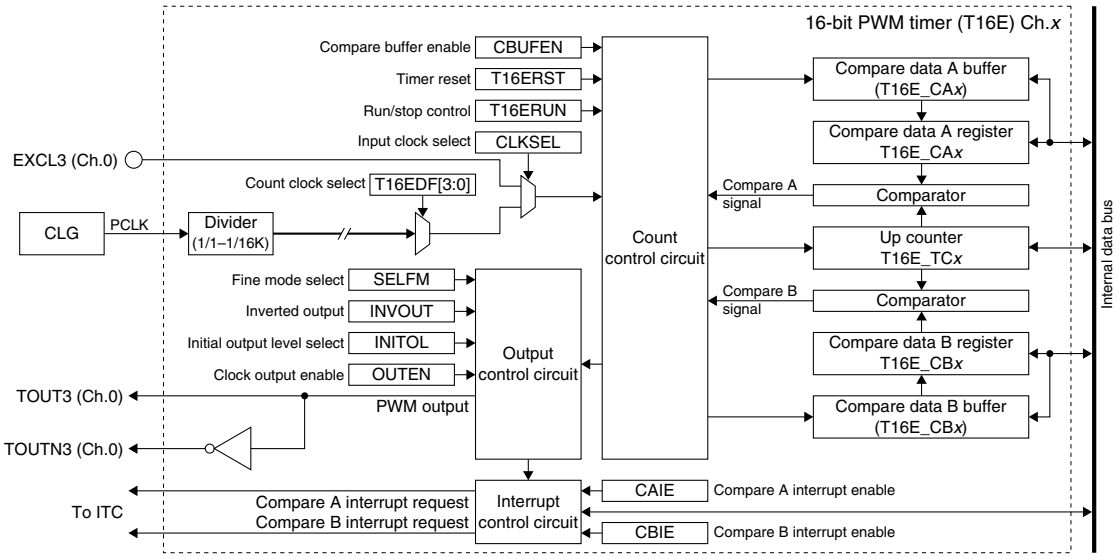


Figure 12.1.1 T16E Module Configuration

The T16E module includes a 16-bit up-counter (T16E_TC*x* register), two 16-bit compare data registers (T16E_CA*x* and T16E_CB*x* registers), and the corresponding buffers.

The 16-bit counter value can be configured, reset to 0, and read via software. The counter counts up with a divided PCLK clock or an external signal input from the EXCL*x* pin.

The compare data A and B registers hold data for comparison against the up-counter contents. Data can be read or written directly from/to the compare data registers. The compare data buffers enable loading to the compare data registers of comparison values set when the counter is reset via software or by the compare B match signal.
Software can select either the compare data register or the buffer for writing comparison values.

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

Notes: • The letter '*x*' in register names refers to a channel number (0).
Example: T16E_CTL*x* register
Ch.0: T16E_CTL0 register

• The letter '*x*' in EXCL*x*, TOUT*x*, and TOUTN*x* pins refers to a signal number (Ch.0 = 3).
Ch.0: EXCL3, TOUT3, TOUTN3

12.2 T16E Input/Output Pins

Table 12.2.1 lists the input/output pins for the T16E module.

Table 12.2.1 List of T16E Pins

Pin name		I/O	Qty	Function
EXCL3	(Ch.0)	I	1	External clock input pin Inputs an external clock for the event counter function.
TOUT3	(Ch.0)	O	1	Non-inverted PWM signal output pin Outputs the PWM signal generated by the timer.
TOUTN3	(Ch.0)	O	1	Inverted PWM signal output pin Outputs the inverted PWM signal generated by the timer.

The T16E output pins (TOUT*x*, and TOUTN*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 T16E output pins. Also the T16E input pin (EXCL*x*) is shared with an I/O port. Setting the port to input mode enables it to be used as the T16E input pin with a general-purpose input function. For detailed information on pin function switching and port control, see the "I/O Ports (P)" chapter.

12.3 Operating Modes

The T16E module has the following two operating modes:
1. Internal clock mode (timer for counting an internal clock)
2. External clock mode (functions as an event counter)

The operating mode is selected using CLKSEL/T16E_CTL*x* register.
Setting CLKSEL to 0 (default) selects internal clock mode, while setting to 1 selects external clock mode.

12.3.1 Internal Clock Mode

Internal clock mode uses a divided PCLK clock as the 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 T16EDF[3:0]/T16E_DF*x* register.

Table 12.3.1.1 PCLK Division Ratio Selection

T16EDF[3:0]	Division ratio	T16EDF[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.

12.3.2 External Clock Mode

External clock mode uses the clock and pulses input via the EXCL*x* port as the count clock. This enables T16E to be used as an event counter. Timer operations other than the input clock are the same as for internal clock mode.
To input an external clock to T16E, the I/O port shared with an EXCL*x* input must be set to input mode in advance. For detailed information on the port control, see the "I/O Ports (P)" chapter.
The T16E counter counts up at the rising edge of the input signal.

12.4 Setting and Resetting Counter Value

The T16E counter can be reset to 0 by writing 1 to the T16ERST/T16E_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 matches compare data B after the count starts.

The counter can also be set to any desired value by writing data to TC[15:0]/T16E_TC*x* register.

12.5 Compare Data Settings

Compare data register/buffer selection

The T16E module includes a data comparator allowing comparison of counter data against any desired value. The compare data A and B registers are used for storing comparison data. Data can be read or written directly from/to the compare data registers.
The compare data buffers enable automatic loading of the comparison values set in the buffers to the compare data registers when the counter is reset via software (writing 1 to T16ERST) or by the compare B match signal. CBUFEN/T16E_CTL*x* resister is used to set which of the compare data register and buffer the comparison values are written to.
Writing 1 to CBUFEN selects the compare data buffer. Writing 0 to it selects the compare data register. The compare data register is selected after an initial reset.

Writing compare data

Compare data A is written to CA[15:0]/T16E_CA*x* register. Compare data B is written to CB[15:0]/T16E_CB*x* register.
When CBUFEN is set to 0, the compare data register values can be read or written directly by these registers.
When CBUFEN is set to 1, data is read from and written to these registers via the compare data buffers. The buffer contents are loaded into the compare data registers when the counter is reset.
The compare data registers and buffers are set to 0x0 after an initial reset.

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

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

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

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

CB: Compare data B (T16E_CB*x* register value)
ct_clk: Count clock frequency

12.6 Timer RUN/STOP Control

Make the following settings before starting T16E.
(1) Set the operating mode (input clock). See Section 12.3.
(2) Set the clock output conditions. See Section 12.7.
(3) If using interrupts, set the interrupt level and enable interrupts for T16E. See Section 12.8.
(4) Set the counter value or reset to 0. See Section 12.4.
(5) Set compare data. See Section 12.5.

The T16E module includes T16ERUN/T16E_CTL*x* register to control run/stop of the timer.
The timer starts counting when 1 is written to T16ERUN. Writing 0 to T16ERUN 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 T16ERUN and T16ERST are written as 1 simultaneously, the timer starts counting after reset.

If the counter matches the compare data A register value during counting, the compare A match signal is output and the cause of compare A interrupt occurs.
Likewise, if the counter matches the compare data B register value, the compare B match signal is output and the cause of compare B interrupt occurs. The counter is reset to 0 at the same time if CBUFEN is set to 0. If CBUFEN is set to 1, the values set in the compare data buffers are loaded into the compare data registers.
If interrupts are enabled, an interrupt request is sent to the interrupt controller (ITC).
In either case, counting continues unaffected.

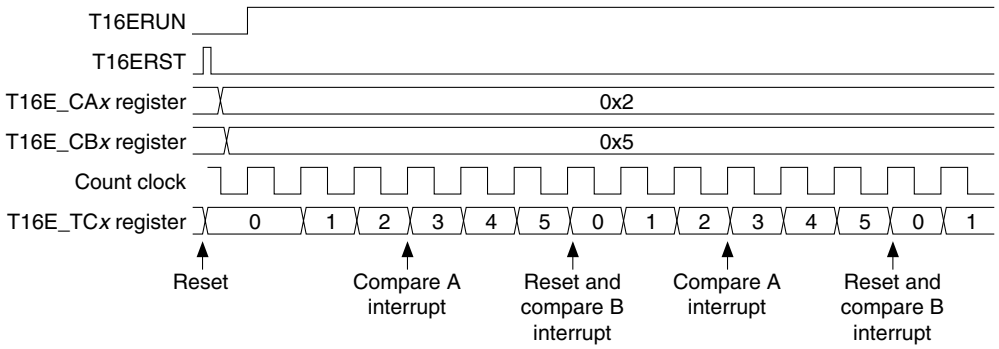


Figure 12.6.1 Basic Counter Operation Timing

12.7 Clock Output Control

The T16E module can generate TOUT*x* and TOUTN*x* signals using the compare match signals.
Figure 12.7.1 shows the T16E clock output circuit.

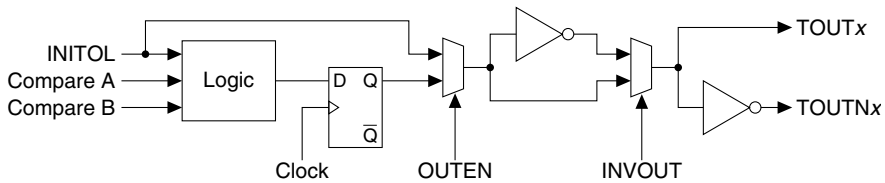


Figure 12.7.1 T16E Clock Output Circuit

Initial output level setting

The default output level is 0 (low level) while the TOUT*x* clock output is Off (TOUTN*x* output level is high). This can be changed to 1 (TOUT*x* = high level, TOUTN*x* = low level) using INITOL/T16E_CTL*x* register.
When INITOL is 0 (default), TOUT*x* initial output level is low (TOUTN*x* output level is high). When INITOL is set to 1, the initial output level is set to high (TOUTN*x* output level is low).

Output signal polarity selection

By default, an active high (normal low) TOUT*x* output signal is generated (active low TOUTN*x* output signal is generated). This logic can be inverted by INVOUT/T16E_CTL*x* register. Writing 1 to INVOUT causes the timer to generate an active low (normal high) TOUT*x* signal (active high TOUTN*x* signal).
Setting INVOUT to 1 also inverts the initial output level set using INITOL.
See Figure 12.7.2 for output waveforms.

Output pin initial status

The TOUT*x* and TOUTN*x* pins used for output are configured as general-purpose I/O ports after an initial reset and the ports enter input mode. The pins then become high-impedance.
Switching the pin function to TOUT*x*/TOUTN*x* output causes the pin to output the level set by INITOL and INVOUT. After the timer output starts, the output is maintained at this level until changed by the counter value.

Table 12.7.1 Initial Output Level

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

Starting clock output

To output the TOUT*x* and TOUTN*x* clocks, write 1 to OUTEN/T16E_CTL*x* register. Writing 0 to OUTEN switches the output to the initial output level as set by INITOL and INVOUT.
Figure 12.7.2 shows the output waveform.

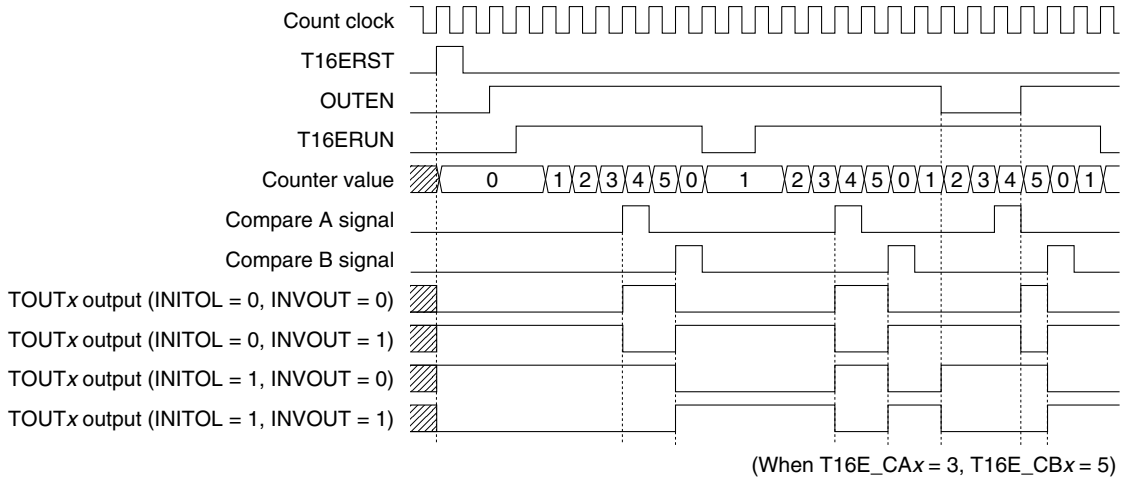


Figure 12.7.2 T16E Output Waveform

TOUT*x* output when INVOUT = 0 (active high)

The TOUT*x* pin outputs low level (initial output level at output start) until the counter matches the compare data A set in the T16E_CA*x* register. When the counter reaches the next compare data A value, the output pin goes to high level, and a cause of compare A interrupt occurs. If the counter subsequently counts up to compare data B set in the T16E_CB*x* register, the counter is reset and the output pin is returned to low level. A cause of compare B interrupt is also occurred at the same time.
The TOUTN*x* pin outputs the inverted signals described above.

TOUT*x* output when INVOUT = 1 (active low)

The TOUT*x* pin outputs high level (inverted value of the initial output level at output start) until the counter matches the compare data A set in the T16E_CA*x* register. When the counter reaches the next compare data A value, the output pin goes to low level, and a cause of compare A interrupt occurs. If the counter subsequently counts up to compare data B set in the T16E_CB*x* register, the counter is reset and the output pin is returned to high level. A cause of compare B interrupt is also occurred at the same time.
The TOUTN*x* pin outputs the inverted signals described above.

Fine mode clock output setting

By default, the clock output changes at the rising edge of the count clock when the counter value matches the compare data A.

In fine mode, the clock output changes in accordance with the compare data A bit 0 (CA0) value when the counter data register TC[14:0] matches the compare data A register CA[15:1].
When CA0 is 0: Changes at the rising edge of the count clock.
When CA0 is 1: Changes at the half-cycle delayed falling edge of the count clock.

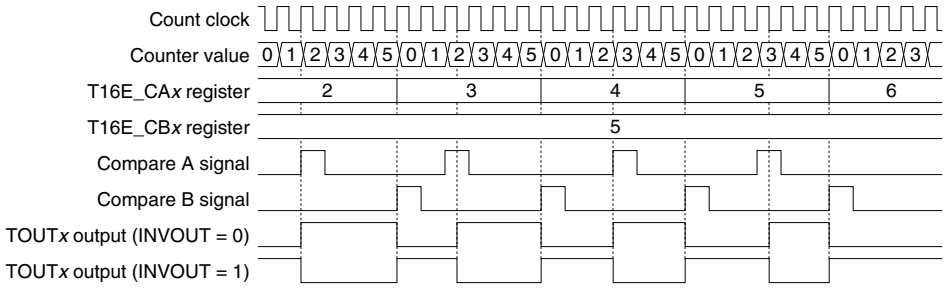


Figure 12.7.3 Fine Mode Clock Output

The output duty can be adjusted in fine mode in count clock half-cycle steps. Note that a pulse will be output with one count clock cycle width when compare data A = 0 (same as for default). The maximum value for compare data B in fine mode is 2^{15} - 1 = 32,767, and the compare data A range will be 0 to (2 × compare data B - 1).
Fine mode is set using SELFM/T16E_CTL*x* register.
Writing 1 to SELFM sets T16E into fine mode. Fine mode is disabled after an initial reset.

Precautions

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

(2) Setting compare data with A > B (A > B × 2 for fine mode) generates a compare B match signal only. It does not generate a compare A match signal. In this case, the TOUT*x* output is fixed at low (high when INVOUT = 1), and the TOUTN*x* output is fixed at high (low when INVOUT = 1).

(3) When fine mode is used, set compare data with B < A / 2 + 0x8000.

(4) Be sure to set T16EDF[3:0]/T16E_DF*x* register to 0x0 (PCLK•1/1) when using fine mode.

12.8 T16E Interrupts

The T16E module can generate the following two kinds of interrupts:
• Compare A interrupt
• Compare B interrupt

A T16E timer channel outputs a single interrupt signal shared by the above interrupt causes to the interrupt controller (ITC). Read the interrupt flags in the T16E module to identify the interrupt cause that has been occurred.

Compare A interrupt

This interrupt request is generated when the counter matches the compare data A register value during counting. It sets the interrupt flag CAIF/T16E_IFLG*x* register within the T16E module to 1.
To use this interrupt, set CAIE/T16E_IMSK*x* register to 1. If CAIE is set to 0 (default), CAIF will not be set to 1, and the interrupt request for this cause will not be sent to the ITC.

Compare B interrupt

This interrupt request is generated when the counter matches the compare data B register value during counting. It sets the interrupt flag CBIF/T16E_IFLG*x* register within the T16E module to 1.
To use this interrupt, set CBIE/T16E_IMSK*x* register to 1. If CBIE is set to 0 (default), CBIF will not be set to 1, and the interrupt request for this cause will not be sent to the ITC.

If the interrupt flag is set to 1 when the interrupt has been enabled, the T16E 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 T16E module must be reset in the interrupt handler routine.

12.9 Control Register Details

Table 12.9.1 List of T16E Registers

Address	Register name		Function
0x5300	T16E_CA0	T16E Ch.0 Compare Data A Register	Sets compare data A.
0x5302	T16E_CB0	T16E Ch.0 Compare Data B Register	Sets compare data B.
0x5304	T16E_TC0	T16E Ch.0 Counter Data Register	Counter data
0x5306	T16E_CTL0	T16E Ch.0 Control Register	Sets the timer mode and starts/stops the timer.
0x5308	T16E_DF0	T16E Ch.0 Clock Division Ratio Select Register	Selects the count clock.
0x530a	T16E_IMSK0	T16E Ch.0 Interrupt Mask Register	Sets the interrupt mask.
0x530c	T16E_IFLG0	T16E Ch.0 Interrupt Flag Register	Indicates and reset interrupt occurrence status.

The T16E 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.

T16E Ch.*x* Compare Data A Register (T16E_CA*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16E Ch.*x* Compare Data A Register (T16E_CA*x*)	**0x5300** (16 bits)	D15–0	**CA[15:0]**	Compare data A CA15 = MSB CA0 = LSB	0x0 to 0xffff	0x0	R/W	

D[15:0] CA[15:0]: Compare Data A

Sets compare data A. (Default: 0x0)

When CBUFEN/T16E_CTL*x* register is set to 0, this register can be used to directly read from or directly write to the compare data A register.

When CBUFEN is set to 1, data is read from and written to these registers via the compare data A buffer. The buffer contents are loaded into the compare data A register when the counter is reset.

The data set is compared against the counter data, and a cause of compare A interrupt is generated if the contents match. The timer output waveform changes at the same time (rises when INVOUT/T16E_CTL*x* register = 0 or falls when INVOUT = 1). These processes do not affect the counter data or count operations.

T16E Ch.*x* Compare Data B Register (T16E_CB*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16E Ch.*x* Compare Data B Register (T16E_CB*x*)	**0x5302** (16 bits)	D15–0	**CB[15:0]**	Compare data B CB15 = MSB CB0 = LSB	0x0 to 0xffff	0x0	R/W	

D[15:0] CB[15:0]: Compare Data B

Sets compare data B. (Default: 0x0)

When CBUFEN/T16E_CTL*x* register is set to 0, this register can be used to directly read from or directly write to the compare data B register.

When CBUFEN is set to 1, data is read from and written to these registers via the compare data B buffer. The buffer contents are loaded into the compare data B register when the counter is reset.

The data set is compared against the counter data, and a cause of compare B interrupt is generated if the contents match. The timer output waveform changes at the same time (falls when INVOUT/T16E_CTL*x* register = 0 or rises when INVOUT = 1) and the counter is reset.

T16E Ch.*x* Counter Data Register (T16E_TC*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16E Ch.*x* Counter Data Register (T16E_TC*x*)	**0x5304** (16 bits)	D15–0	**TC[15:0]**	Counter data TC15 = MSB TC0 = LSB	0x0 to 0xffff	0x0	R/W	

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

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

The counter value can also be set by writing data to this register.

T16E Ch.*x* Control Register (T16E_CTL*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16E Ch.*x* Control Register (T16E_CTL*x*)	0x5306 (16 bits)	D15–9	–	reserved	–	–	–	0 when being read.
		D8	INITOL	Initial output level	1 High 0 Low	0	R/W	
		D7	–	reserved	–	–	–	0 when being read.
		D6	SELFM	Fine mode select	1 Fine mode 0 Normal mode	0	R/W	
		D5	CBUFEN	Comparison buffer enable	1 Enable 0 Disable	0	R/W	
		D4	INVOUT	Inverse output	1 Invert 0 Normal	0	R/W	
		D3	CLKSEL	Input clock select	1 External 0 Internal	0	R/W	
		D2	OUTEN	Clock output enable	1 Enable 0 Disable	0	R/W	
		D1	T16ERST	Timer reset	1 Reset 0 Ignored	0	W	0 when being read.
		D0	T16ERUN	Timer run/stop control	1 Run 0 Stop	0	R/W	

D[15:9] Reserved

D8 INITOL: Initial Output Level Bit

Sets the initial timer output level.
1 (R/W): TOUT*x* = High, TOUTN*x* = Low
0 (R/W): TOUT*x* = Low, TOUTN*x* = High (default)

The timer output pin switches to the initial output level set here when the clock output is switched off by writing 0 to OUTEN. Note that this level will be inverted when INVOUT is 1.

D7 Reserved

D6 SELFM: Fine Mode Select Bit

Sets the clock output to fine mode.
1 (R/W): Fine mode
0 (R/W): Normal output (default)

When SELFM is set to 1, the clock output is set to fine mode, and the output clock duty becomes adjustable in count clock half-cycle steps. When SELFM is set to 0, normal clock output is performed.

D5 CBUFEN: Comparison Buffer Enable Bit

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

When CBUFEN is set to 1, compare data is read and written via the compare data buffer. The buffer contents are loaded into the compare data register when the counter is reset via software or by the compare B signal.
When CBUFEN is set to 0, compare data is read and written directly from/to the compare data register.

D4 INVOUT: Inverse Output Bit

Selects the timer output signal polarity.
1 (R/W): Inverted (TOUT*x* = active low, TOUTN*x* = active high)
0 (R/W): Normal (TOUT*x* = active high, TOUTN*x* = active low) (default)

Writing 1 to INVOUT generates an active low signal (off level = high) for the TOUT*x* output. When INVOUT is 0, an active high signal (off level = low) is generated.
Writing 1 to this bit also inverts the initial output level set by INITOL.
The signal level above is inverted for the TOUTN*x* output.

D3 CLKSEL: Input Clock Select Bit

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

Writing 0 to CLKSEL selects the internal clock (PCLK) for the timer input clock, while writing 1 selects an external clock (a clock input via the EXCL*x* pin) and it functions as an event counter.
To input an external clock/pulse to T16E, the I/O port shared with an EXCL*x* input must be set to input mode.

D2 OUTEN: Clock Output Enable Bit

Controls the TOUT*x* and TOUTN*x* signal (timer output clock) outputs.
1 (R/W): Enabled
0 (R/W): Disabled (default)

Writing 1 to OUTEN outputs the TOUT*x* and TOUTN*x* signals from the corresponding output pins. Writing 0 to OUTEN stops the output, and the output pins go to the off level according to the INVOUT and INITOL settings. The TOUT*x*/TOUTN*x* port function must be selected using the port function select register before outputting the TOUT*x* and TOUTN*x* signals.

D1 T16ERST: Timer Reset Bit

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

Writing 1 to T16ERST resets the counter.

D0 T16ERUN: Timer Run/Stop Control Bit

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

T16E starts counting when 1 is written to T16ERUN and stops when 0 is written. The counter data is retained when stopped until the subsequent reset or run. Counting can be resumed when switched from stop to run from the data retained.

T16E Ch.*x* Clock Division Ratio Select Register (T16E_DF*x*)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
T16E Ch.*x* Clock Division Ratio Select Register (T16E_DF*x*)	0x5308 (16 bits)	D15–4	–	reserved	–		–	–	0 when being read.
		D3–0	T16EDF[3:0]	Clock division ratio select	T16EDF[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] T16EDF[3:0]: Clock Division Ratio Select Bits

Selects a PCLK division ratio to generate the count clock.

Table 12.9.2 PCLK Division Ratio Selection

T16EDF[3:0]	Division ratio	T16EDF[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.

T16E Ch.*x* Interrupt Mask Register (T16E_IMSK*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16E Ch.*x* Interrupt Mask Register (T16E_IMSK*x*)	0x530a (16 bits)	D15–2	–	reserved	–	–	–	0 when being read.
		D1	CBIE	Compare B interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	CAIE	Compare A interrupt enable	1 Enable 0 Disable	0	R/W	

D[15:2] Reserved

D1 CBIE: Compare B Interrupt Enable Bit

Enables or disables compare B match 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 match 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.

T16E Ch.*x* Interrupt Flag Register (T16E_IFLG*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16E Ch.*x* Interrupt Flag Register (T16E_IFLG*x*)	0x530c (16 bits)	D15–2	–	reserved	–	–	–	0 when being read.
		D1	CBIF	Compare B interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D0	CAIF	Compare A interrupt flag		0	R/W	

D[15:2] Reserved

D1 CBIF: Compare B Interrupt Flag 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 T16E 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 T16E interrupt flag that is set to 1 when the counter reaches the value set in the compare A register. CAIF is reset by writing 1.

13 16-bit PWM Timers (T16A2)

Note: T16A2 is available only in the S1C17624/604.

13.1 T16A2 Module Overview

The S1C17624/604 includes a 16-bit PWM timer (T16A2) module that consists of counter blocks and comparator/capture blocks. This timer can be used as an interval timer, PWM waveform generator, external event counter and a count capture unit to measure external event periods.
The features of T16A2 are listed below.

- Two channels of 16-bit up counter blocks
- Two 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 13.1.1 shows the T16A2 configuration.

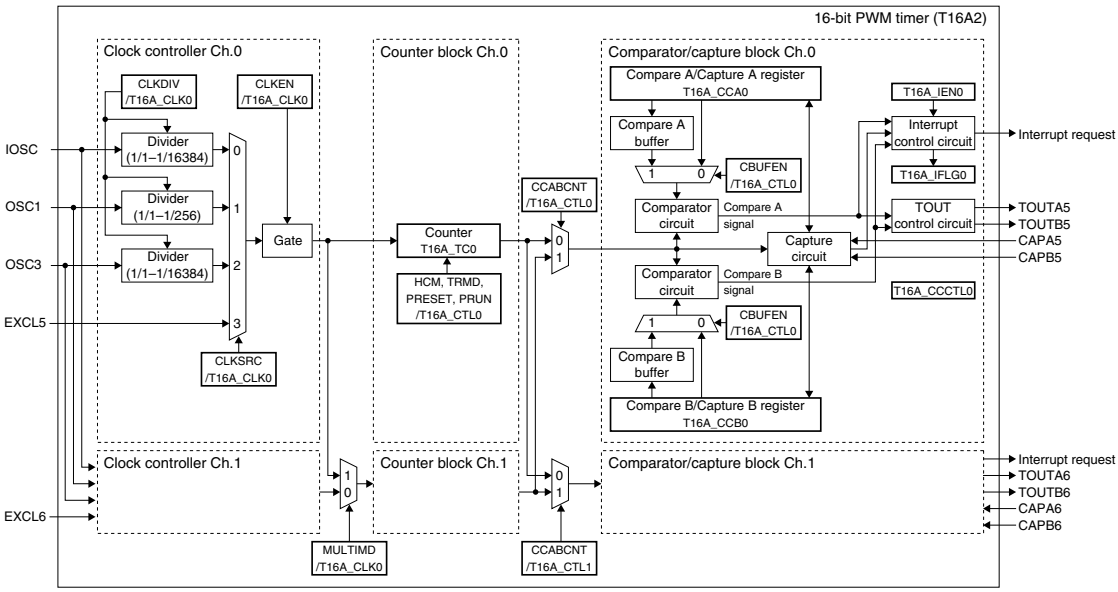


Figure 13.1.1 T16A2 Configuration

Clock controller

T16A2 includes two 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, OSC3, or OSC1 division clock, or the external count clock input from outside the IC. The T16A2 module allows software to run and stop the counter 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 provides a counter comparison function (comparator mode) and a count capture function (capture mode). When comparator mode is selected via software, the comparator/capture block can be used as a PWM waveform or clock generator. When capture mode is selected, this block can be used as a count capture unit for measuring external event periods/cycles. The comparator circuit generates the compare A and B signals that represent matching between compare A/B register values (set via software) and the counter value, and outputs the signals to the TOUT control circuit and the interrupt control circuit. The TOUT control circuit generates a PWM or other signal from the compare A and B signals and outputs it to the external TOUTA*x* and TOUTB*x* pins. The capture circuit loads the counter value to the capture A or B register using the CAPA*x* or CAPB*x* input signal that represents external events issued as a trigger. The interrupt control circuit outputs an interrupt signal to the interrupt controller (ITC) module according to the interrupt condition that has been set.
Comparator mode and capture mode cannot be used simultaneously in the same channel.

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 four systems (details are described later).

Notes: • The letter '*x*' in register names refers to a channel number (0 or 1).
Example: T16A_CTL*x* register
Ch.0: T16A_CTL0 register
Ch.1: T16A_CTL1 register

• The letter '*x*' in EXCL*x*, CAPA*x*, CAPB*x*, TOUTA*x*, and TOUTB*x* pins refers to a signal number (Ch.0 = 5, Ch.1 = 6).
Ch.0: EXCL5, CAPA5, CAPB5, TOUTA5, TOUTB5
Ch.1: EXCL6, CAPA6, CAPB6, TOUTA6, TOUTB6

13.2 T16A2 Input/Output Pins

Table 13.2.1 lists the input/output pins for the T16A2 module.

Table 13.2.1 List of T16A2 Pins

Pin name	I/O	Qty	Function
EXCL5 (for Ch.0) EXCL6 (for Ch.1)	I	2	External clock input pins Inputs an external clock for the event counter function.
CAPA5, CAPB5 (for Ch.0) CAPA6, CAPB6 (for Ch.1)	I	4	Counter-capture trigger signal input pins (effective in capture mode) The specified edge (falling edge, rising edge, or both) of the signal input to the CAPA*x* pin captures the counter data into the capture A register. The CAPB*x* pin input signal captures the counter data into the capture B register.
TOUTA5, TOUTB5 (for Ch.0) TOUTA6, TOUTB6 (for Ch.1)	O	4	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 T16A2 input/output pins (CAPA*x*, CAPB*x*, TOUTA*x*, and TOUTB*x*) are shared with I/O ports and are initially set as general purpose I/O port pins. The pin functions must be switched using the port function select bits to use the general purpose I/O port pins as T16A2 input/output pins. Also the external clock input pins (EXCL*x*) are shared with I/O ports. Setting the port to input mode enables it to be used as the T16A2 input pin with a general-purpose input function. For detailed information on pin function switching and port control, see the "I/O Ports (P)" chapter.

13.3 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.

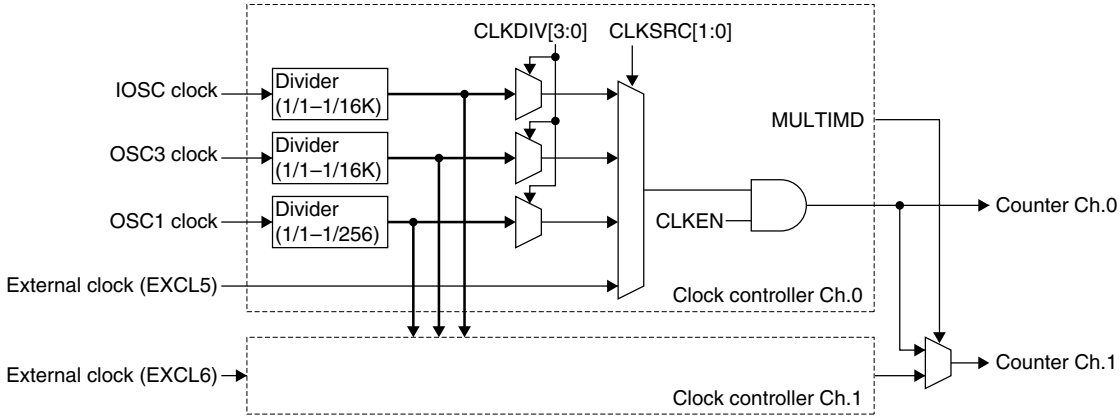


Figure 13.3.1 Clock Controller

Clock source selection

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

Table 13.3.1 Clock Source Selection

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

(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 13.3.2 External Clock Input Pins

Channel	External clock input pin
T16A2 Ch.0	EXCL5
T16A2 Ch.1	EXCL6

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 13.3.3 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 T16A2 is not required, disable the clock supply to reduce current consumption.

Multi-comparator/capture mode

The T16A2 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 T16A2 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 T16A2 count is stopped before setting the count clock.

13.4 T16A2 Operating Modes

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

13.4.1 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 13.6, "Timer Output Control," for more information.

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

Compare buffers

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

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 13.4.1.1 List of Counter Capture Trigger Signal Input Pins

Channel	Trigger input pins	
	Capture A	Capture B
T16A2 Ch.0	CAPA5	CAPB5
T16A2 Ch.1	CAPA6	CAPB6

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 13.4.1.2 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.

13.4.2 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 is reset to 0 or 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 counter is reset or it overflows. The counter should be set to this mode to set a specific wait time or for pulse width measurement.

The counter is not cleared to 0 after the count operation is completed in one-shot mode. The counter must be reset to clear its value to 0. For more information on resetting methods, see Section 13.5.1, "Counter Reset."

13.4.3 Normal Channel Mode and Multi-Comparator/Capture Mode

One channel of the T16A2 module basically consists of a counter block and a comparator/capture block. The T16A2 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/T16A_CLK0 register.

Normal channel mode (MULTIMD = 0, default)

Set the T16A2 module to this mode when using it as two 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 13.4.3.1 Counter Selection

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

(Default: 0x0)

When using the T16A2 module in normal channel mode, be sure to connect the counter block to the comparator/capture block in the same channel.

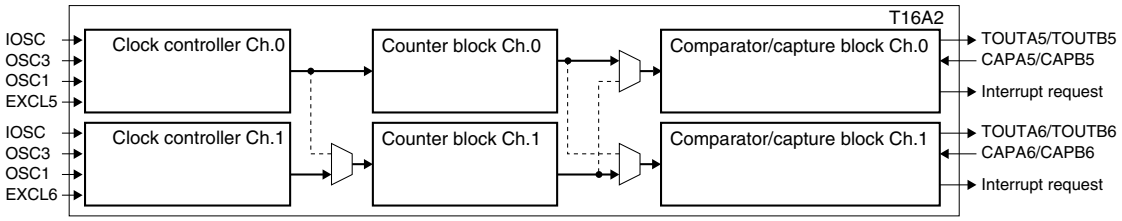


Figure 13.4.3.1 Timer Configuration in Normal Channel Mode

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 other channel are ineffective.

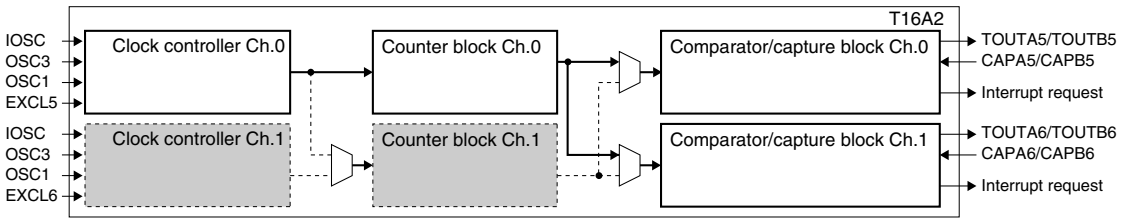


Figure 13.4.3.2 Timer Configuration in Multi-Comparator/Capture Mode

13.4.4 Normal Clock Mode and Half Clock Mode

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

Normal clock mode (HCM = 0, default)

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

Half clock mode (HCM = 1)

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

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

- The dual-edge counter value cannot be read.
- Do not use the compare A interrupt in half clock mode.

13.5 Counter Control

13.5.1 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.

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

13.5.2 Counter RUN/STOP Control

Make the following settings before starting the count operation.
(1) Switch the input/output pin functions to be used for T16A2. See the “I/O Port (P)” chapter.
(2) Select operating modes. See Section 13.4.
(3) Select the clock source. See Section 13.3.
(4) Configure the timer outputs (TOUT). See Section 13.6.
(5) If using interrupts, set the interrupt level and enable the T16A2 interrupts. See Section 13.7.
(6) Reset the counter to 0. See Section 13.5.1.
(7) Set comparison data (in comparator mode). See Section 13.4.1.

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.

Note: After the T16A_CCA*x* and T16A_CCB*x* registers have been set, wait for one or more T16A2 count clock cycles and then run the counter.

13.5.3 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.

13.5.4 Counter Operation and Interrupt Timing Charts

Comparator mode

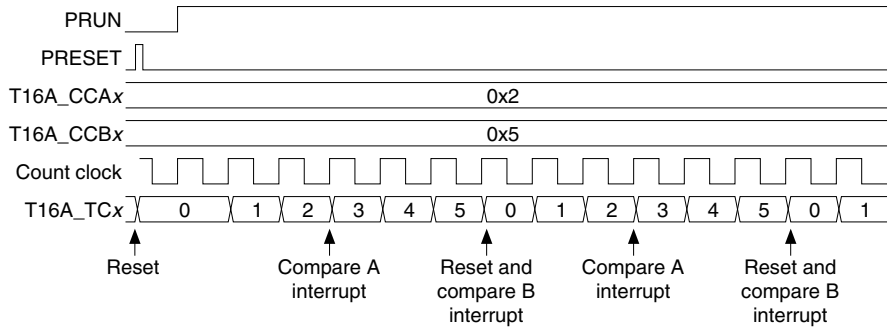


Figure 13.5.4.1 Operation Timing in Comparator Mode

Capture mode

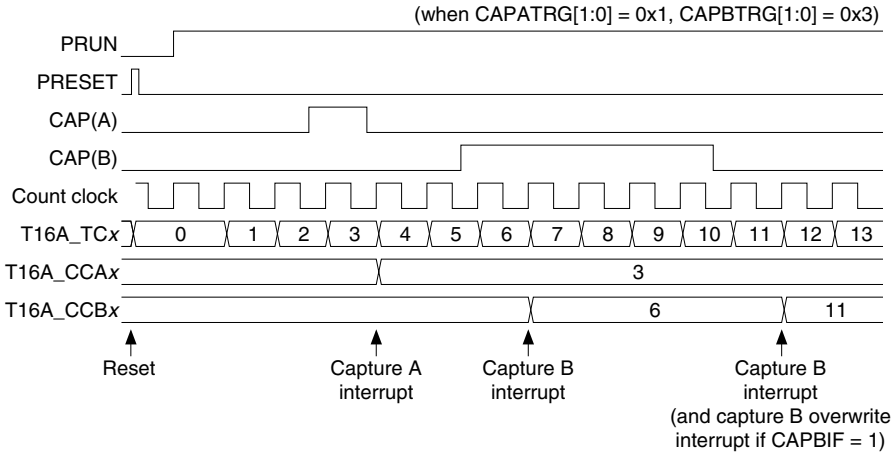


Figure 13.5.4.2 Operation Timing in Capture Mode

13.6 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 T16A2 module can output up to four TOUT signals. Figure 13.6.1 shows the TOUT output circuit (one timer channel).

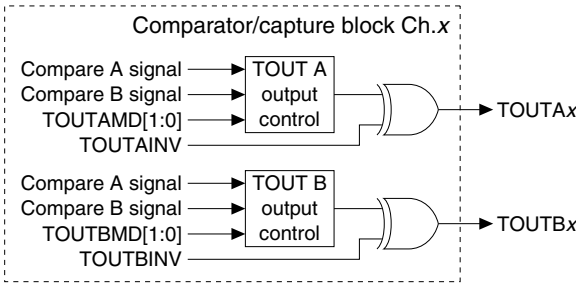


Figure 13.6.1 TOUT Output Circuit

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 13.6.1 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 13.6.1 List of TOUT Output Pins

Channel	TOUT output pin	
	System A	System B
T16A2 Ch.0	TOUTA5	TOUTB5
T16A2 Ch.1	TOUTA6	TOUTB6

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 13.6.2 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 13.6.2 illustrates the TOUT output waveform.

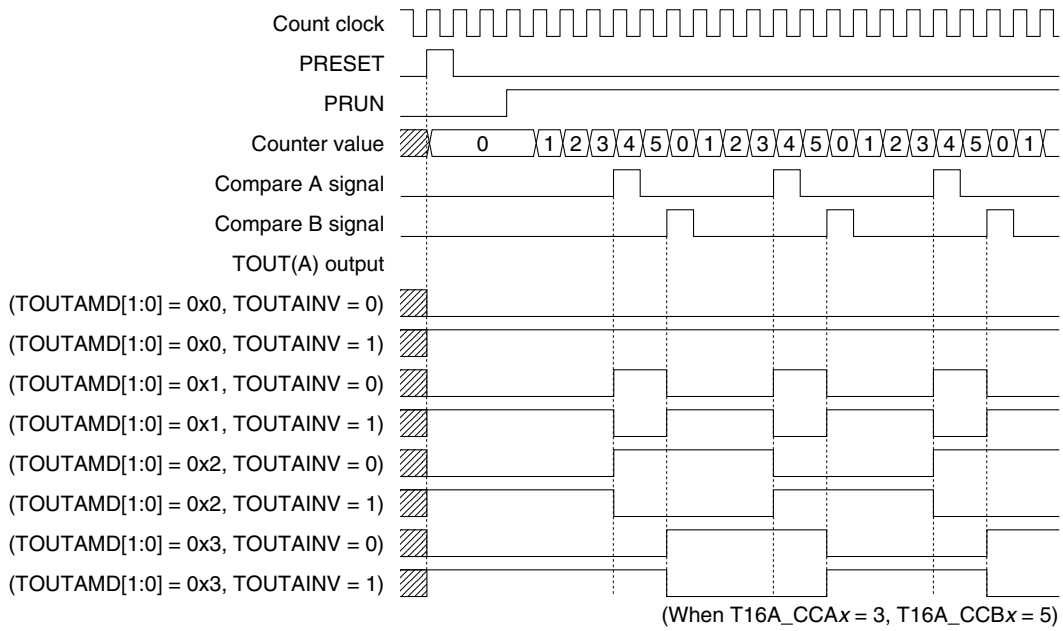
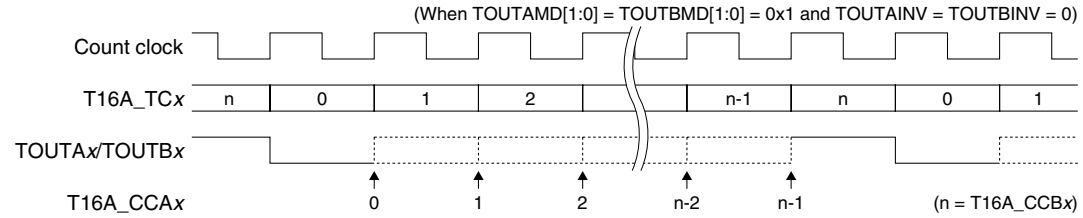


Figure 13.6.2 TOUT Output Waveform

PWM waveform output timings

Normal clock mode (HCM = 0)



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

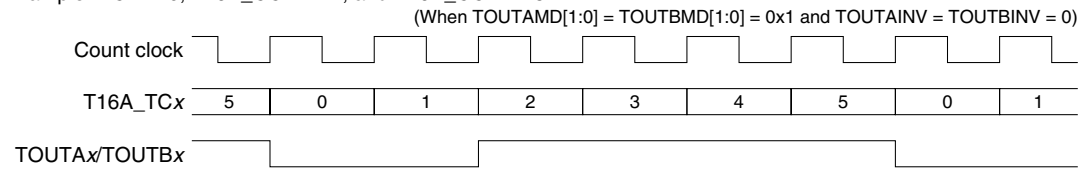
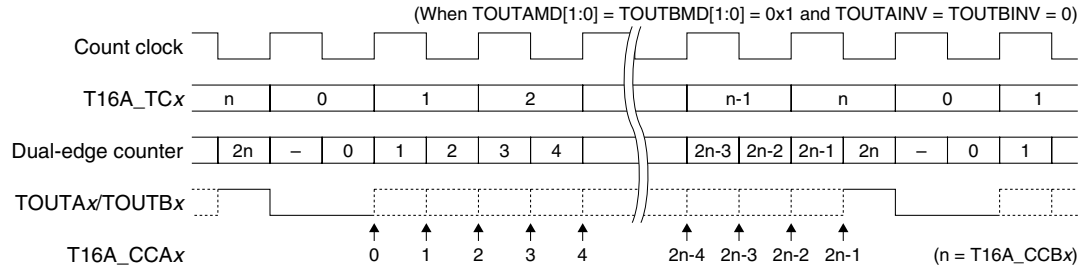


Figure 13.6.3 PWM Waveform Output Timings in Normal Clock Mode

Half clock mode (HCM = 1)



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

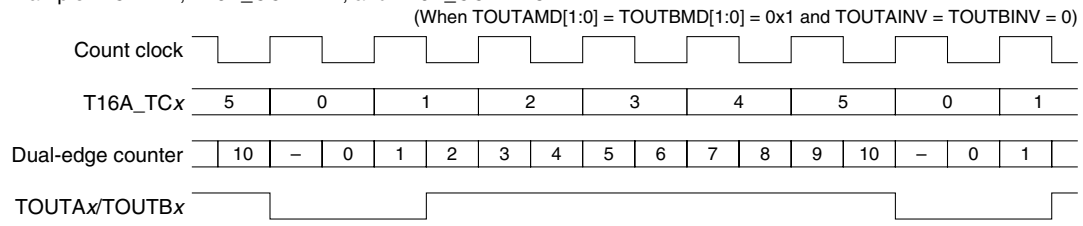


Figure 13.6.4 PWM Waveform Output Timings in Half Clock Mode

13.7 T16A2 Interrupts

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

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

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 T16A2 module to identify the interrupt cause that has been occurred.

Interrupts in comparator mode

Compare A interrupt

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

Compare B interrupt

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

Interrupts in capture mode

Capture A interrupt

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

Capture B interrupt

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

Capture A overwrite interrupt

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

Capture B overwrite interrupt

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

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

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

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

13.8 Control Register Details

Table 13.8.1 List of T16A2 Registers

Address	Register name		Function
0x5068	T16A_CLK0	T16A Clock Control Register Ch.0	Controls the T16A2 Ch.0 clock.
0x5069	T16A_CLK1	T16A Clock Control Register Ch.1	Controls the T16A2 Ch.1 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.

The T16A2 registers are described in detail below.

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

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

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16A Clock Control Register Ch.*x* (T16A_CLK*x*)	**0x5068** **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	
		D1	**MULTIMD**	Multi-comparator/capture mode select	1 Multi, 0 Normal	0	R/W	T16A_CLK0
		–	–	reserved	–	–	–	T16A_CLK1 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 13.8.2 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 13.8.3 Clock Source Selection

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

(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 T16A2 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 register)

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 (16 bits)	D15–7	–	reserved	–	–	–	0 when being read.
		D6	HCM	Half clock mode enable	1 Enable 0 Disable	0	R/W	
		D5–4	CCABCNT [1:0]	Counter select	CCABCNT[1:0] Counter Ch. 0x3, 0x2 reserved 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	

D[15:7] Reserved

D6 HCM: Half Clock Mode Enable Bit

Sets T16A2 to half clock mode.
1 (R/W): Enabled (half clock mode)
0 (R/W): Disabled (normal clock mode) (default)

Setting HCM to 1 places T16A2 into half clock mode. In half clock mode, T16A2 uses the dual-edge counter, which counts at the rising and falling edges of the count clock, to generate a compare A signal when the dual-edge counter value matches the T16A_CCA*x* register. This makes it possible to control the duty ratio with double accuracy as compared to normal clock mode.
Setting HCM to 0 places T16A2 into normal clock mode. In normal clock mode, T16A2 generates a compare A signal when the T16A_TC*x* register value matches the T16A_CCA*x* register.

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

- The dual-edge counter value cannot be read.
- Do not use the compare A interrupt in half clock mode.

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 13.8.4 Counter Selection

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

(Default: 0x0)

When using the T16A2 module in normal channel mode (T16A2MULTIMD = 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)

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

Setting CBUFEN to 0 disables the compare buffer. The compare A and B signals will be generated by comparing the counter values with the compare A and B register values.

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

D2 TRMD: Count Mode Select Bit

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

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

D1 PRESET: Counter Reset Bit

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

Writing 1 to this bit resets the counter to 0.

Note: Make sure the counter is halted (CLKEN/T16A_CLK*x* register = 0) before setting PRESET.

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

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

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

Table 13.8.5 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 (TOUTB*x* output) is changed by the compare A and compare B signals. These bits are also used to turn the TOUT B output On and Off.

Table 13.8.6 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 (TOUTB*x* output) polarity.
1 (R/W): Inverted (active Low)
0 (R/W): Normal (active High) (default)

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

D8 CCBMD: T16A_CCB Register Mode Select Bit

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

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

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

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

Table 13.8.7 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 (TOUTA*x* output) is changed by the compare A and compare B signals. These bits are also used to turn the TOUT A output On and Off.

Table 13.8.8 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 (TOUTA*x* output) polarity.
1 (R/W): Inverted (active Low)
0 (R/W): Normal (active High) (default)

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

D0 CCAMD: T16A_CCA Register Mode Select Bit

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

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

T16A Comparator/Capture Ch.*x* A Data 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** (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.
The counter value comparison timing varies according to the CBUFEN/T16A_CTL*x* register value. For more information, see "Comparator mode (CCAMD/CCBMD = 0, default)" in Section 13.4.1.

Note: After the T16A_CCA*x* register has been set, wait for one or more T16A2 count clock cycles and then run the counter.

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

T16A Comparator/Capture Ch.*x* B Data 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** (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.
The counter value comparison timing varies according to the CBUFEN/T16A_CTL*x* register value. For more information, see "Comparator mode (CCAMD/CCBMD = 0, default)" in Section 13.4.1.

Note: After the T16A_CCB*x* register has been set, wait for one or more T16A2 count clock cycles and then run the counter.

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

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

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

D[15:6] Reserved

D5 CAPBOWIE: Capture B Overwrite Interrupt Enable Bit

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

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

D4 CAPAOWIE: Capture A Overwrite Interrupt Enable Bit

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

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

D3 CAPBIE: Capture B Interrupt Enable Bit

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

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

D2 CAPAIE: Capture A Interrupt Enable Bit

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

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

D1 CBIE: Compare B Interrupt Enable Bit

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

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

D0 CAIE: Compare A Interrupt Enable Bit

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

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

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

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
T16A Comparator/ Capture Ch.*x* Interrupt Flag Register (T16A_IFLG*x*)	0x540c 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	

D[15:6] Reserved

D5 CAPBOWIF: Capture B Overwrite Interrupt Flag Bit

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

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

D4 CAPAOWIF: Capture A Overwrite Interrupt Flag Bit

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

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

D3 CAPBIF: Capture B Interrupt Flag Bit

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

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

D2 CAPAIF: Capture A Interrupt Flag Bit

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

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

D1 CBIF: Compare B Interrupt Flag Bit

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

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

D0 CAIF: Compare A Interrupt Flag Bit

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

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

14 8-bit OSC1 Timer (T8OSC1)

14.1 T8OSC1 Module Overview

The S1C17624/604/622/602/621 includes a single-channel 8-bit OSC1 timer that uses OSC1 as its clock source. The features of the T8OSC1 module are listed below.

- 8-bit up counter with a comparator
- The counter value can be compared with two specified comparison values (compare data and PWM duty data) by the comparator.
- Can generate interrupts when the counter value matches compare data.
- Can generate a PWM signal from matching between the counter and two specified comparison values (compare data and PWM duty data), and output it outside the IC.

Figure 14.1.1 shows the T8OSC1 module configuration.

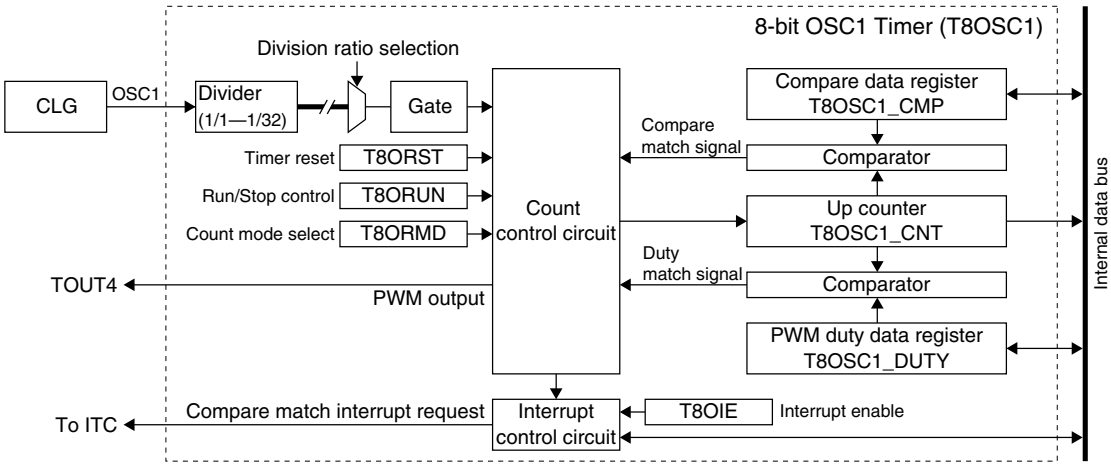


Figure 14.1.1 T8OSC1 Module Configuration

The 8-bit OSC1 timer includes an 8-bit up-counter (T8OSC1_CNT register), an 8-bit compare data register (T8OSC1_CMP register), and an 8-bit PWM duty data register (T8OSC1_DUTY register).

The up-counter can be reset to 0 via software and counts up using the OSC1 division clock. The count value can be read via software.

The compare data and PWM duty data registers store the data used for comparisons against up-counter contents. If the counter value matches the contents of each data register, the comparator outputs a signal to control the interrupts and the PWM output signal. The compare data register can be used to set the interrupt generating and PWM output clock cycles. The PWM duty data register can be used to set the PWM output clock duty ratio.

14.2 T8OSC1 Output Pin

Table 14.2.1 shows the T8OSC1 module output pin.

Table 14.2.1 T8OSC1 Pin

Pin name	I/O	Qty	Function
TOUT4	O	1	PWM signal (non-inverted) output pin Outputs the PWM signal generated by the timer.

The T8OSC1 output pin (TOUT4) is shared with an I/O port and is initially set as a general-purpose I/O port pin. The pin function must be switched using the port function select bit to use the general purpose I/O port pin as the T8OSC1 output pin. For detailed information on pin function switching, see the "I/O Ports (P)" chapter.

14.3 Count Clock

The T8OSC1 module includes a divider and a gate circuit for controlling the count clock.

Clock division ratio selection

The T8OSC1 module generates the count clock by dividing the OSC1 clock output from the CLG.
Use T8O1CK[2:0]/OSC_T8OSC1 register to select the division ratio.

Table 14.3.1 OSC1 Division Ratio Selection

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

(Default: 0x0)

Clock enable

The count clock supply is enabled with T8O1CE/OSC_T8OSC1 register. The T8O1CE default setting is 0, which stops the clock. Setting T8O1CE to 1 feeds the clock generated as above to the counter. If no T8OSC1 operation is required, stop the clock to reduce current consumption.

Note: Be sure to set T8O1CE to 0 before selecting a clock division ratio.

14.4 Count Mode

T8OSC1 features two count modes: Repeat mode and One-shot mode. These modes are selected using T8ORMD/T8OSC1_CTL register.

Repeat mode (T8ORMD = 0, default)

Setting T8ORMD to 0 sets T8OSC1 to repeat mode.
In this mode, once the count starts, the timer continues running until stopped by the application program. If the counter matches the compare data, the timer resets the counter and continues counting. The interrupt signal is output at the same time. T8OSC1 should be set to this mode to generate periodic interrupts at desired intervals or to perform PWM output.

One-shot mode (T8ORMD = 1)

Setting T8ORMD to 1 sets T8OSC1 to One-shot mode.
In this mode, the timer stops automatically as soon as the counter matches the compare data. This means only one interrupt can be generated after the timer starts. Note that the timer resets the counter, then stops after a complete match has occurred. T8OSC1 should be set to this mode to set a specific wait time.

Notes:
- Make sure the timer count is halted before changing count mode settings.
- If count operation is activated while the count mode is set to one-shot mode, and the CPU enters halt state, the counter does not stop even when a compare match occurs, disabling one-shot operation.

14.5 Counter Reset

The counter can be reset to 0 by writing 1 to T8ORST/T8OSC1_CTL 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 matches compare data after the count starts.

14.6 Compare Data Settings

Compare data is written to T8OCMP[7:0]/T8OSC1_CMP register.
After an initial reset, the compare data register is set to 0x0.

The timer compares the count data against the compare data register and generates a compare match signal as well as resets the counter if the values are equal. This compare match signal can generate an interrupt.
The compare match cycle can be calculated as follows:

$$\text{Compare match interval} = \frac{\text{CMP} + 1}{\text{ct\_clk}} \text{ [s]}$$

$$\text{Compare match cycle} = \frac{\text{ct\_clk}}{\text{CMP} + 1} \text{ [Hz]}$$

CMP: Compare data (T8OSC1_CMP register value)
ct_clk: Count clock frequency

When T8OSC1 is used to generate a PWM signal, the compare data determines the frequency of the output signal. (For a discussion of PWM output, refer to Section 14.8.)

14.7 Timer RUN/STOP Control

Make the following settings before starting T8OSC1.
(1) To output the PWM signal, switch the output pin function to be used for T8OSC1. Refer to the "I/O Ports (P)" chapter.
(2) Set the count mode (one-shot or repeat mode). See Section 14.4.
(3) Select the count clock. See Section 14.3.
(4) If using interrupts, set the interrupt level and enable T8OSC1 interrupts. See Section 14.9.
(5) Reset the counter. See Section 14.5.
(6) Set the compare data. See Section 14.6.
(7) To output the PWM signal, set the PWM duty data. See Section 14.8.

T8OSC1 includes T8ORUN/T8OSC1_CTL register to control Run/Stop.
The timer starts counting when T8ORUN is written as 1. Writing 0 to T8ORUN stops the count.
This control does not affect the counter data. The counter data is retained even when the count is halted, allowing resumption of the count from that data.
If T8ORUN and T8ORST are written as 1 simultaneously, the timer starts counting after the reset.

If the counter matches the compare data register setting during counting, a compare match signal is output and a cause of compare match interrupt occurs. The counter is reset to 0 at the same time. If interrupts are enabled, an interrupt request is sent to the interrupt controller (ITC).
If one-shot mode is set, the timer stops the count.
If repeat mode is set, the timer continues to count from 0.

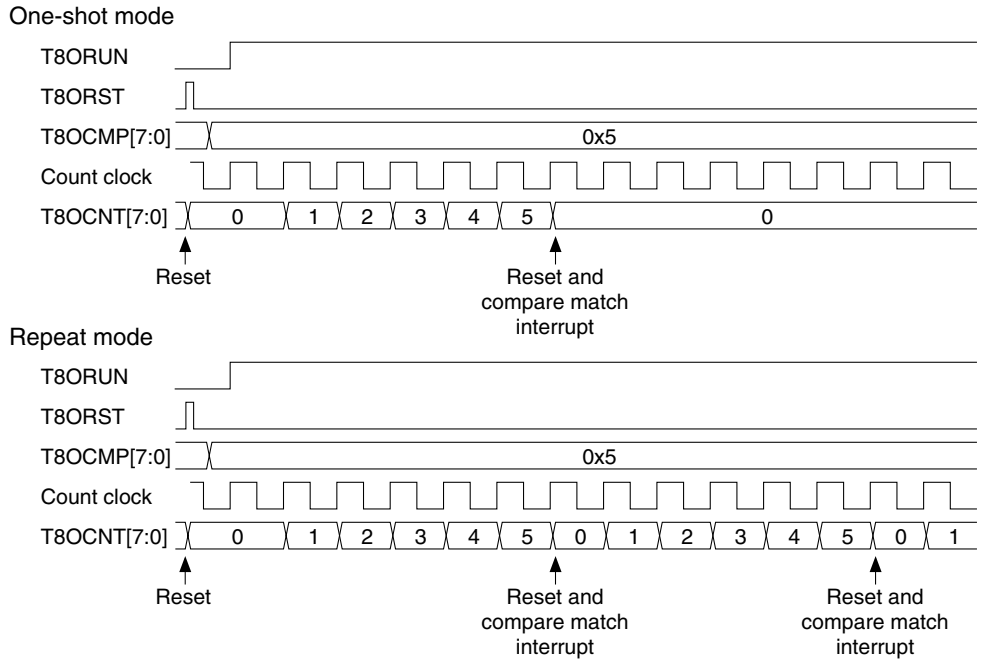


Figure 14.7.1 Basic Counter Operation Timing

14.8 PWM Output

The T8OSC1 module can generate a PWM signal in accordance with the compare data and PWM duty data settings and output it from the TOUT4 pin.

Output pin setting

To use the PWM output pin (TOUT4), the pin function of the corresponding port must be switched in advance. See the "I/O Ports (P)" chapter for switching the pin function.

PWM waveform control

The PWM waveform cycle can be set by the compare data register (see Section 14.6) and the duty ratio can be adjusted by the PWM duty data register. The timer outputs a low level signal until the counter value matches the value of the PWM duty data register. When the counter value exceeds the value of the PWM duty data, the output pin changes to high. Once the counter counts up to the compare data register value, the counter is reset and the output pin returns to low. Figure 14.8.1 shows the output waveform.

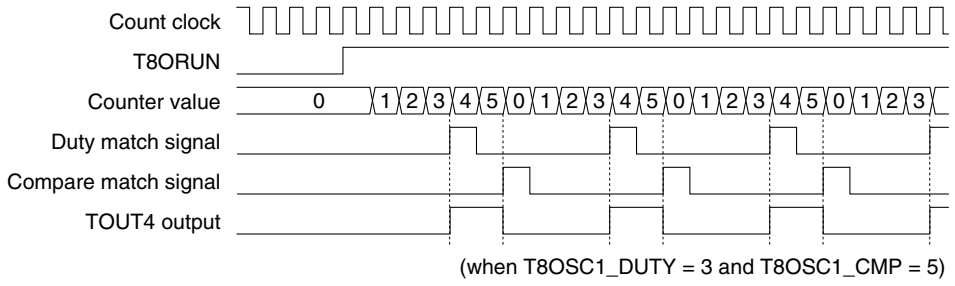


Figure 14.8.1 PWM Output Waveform

Precautions

(1) When using the timer output, set the following: PWM duty data ≥ 0, compare data ≥ 1. The minimum setting value is 0 for PWM duty data and 1 for compare data. The timer output cycle is 1/2 of the count clock.

(2) When the PWM duty data is set greater than the compare data, only the compare match signal will be generated. No duty match signal will be generated. In that case, the TOUT4 output is fixed at low.

14.9 T8OSC1 Interrupts

The T8OSC1 module outputs an interrupt request to the interrupt controller (ITC) by compare match.

Compare match interrupt

This interrupt request is generated when the counter matches the compare data register value during counting. It sets the interrupt flag T8OIF/T8OSC1_IFLG register in the T8OSC1 module to 1.
To use this interrupt, set T8OIE/T8OSC1_IMSK register to 1. If T8OIE is set to 0 (default), T8OIE is not set to 1, and an interrupt request for this cause is not sent to the ITC.

If T8OIF is set to 1, the T8OSC1 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: • To prevent interrupt recurrences, the T8OSC1 module interrupt flag T8OIF must be reset in the interrupt handler routine following a T8OSC1 interrupt.

• To prevent generating unnecessary interrupts, reset T8OIF before enabling T8OSC1 interrupts using T8OIE.

14.10 Control Register Details

Table 14.10.1 List of T8OSC1 Registers

Address	Register name		Function
0x5065	OSC_T8OSC1	T8OSC1 Clock Control Register	Controls the count clock.
0x50c0	T8OSC1_CTL	T8OSC1 Control Register	Sets the timer mode and starts/stops the timer.
0x50c1	T8OSC1_CNT	T8OSC1 Counter Data Register	Counter data
0x50c2	T8OSC1_CMP	T8OSC1 Compare Data Register	Sets compare data.
0x50c3	T8OSC1_IMSK	T8OSC1 Interrupt Mask Register	Sets the interrupt mask.
0x50c4	T8OSC1_IFLG	T8OSC1 Interrupt Flag Register	Indicates and reset interrupt occurrence status.
0x50c5	T8OSC1_DUTY	T8OSC1 PWM Duty Data Register	Sets data for PWM output.

The T8OSC1 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.

T8OSC1 Clock Control Register (OSC_T8OSC1)

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

D[7:4] Reserved

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

Selects the division ratio for generating the count clock.

Table 14.10.2 OSC1 Division Ratio Selection

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

(Default: 0x0)

Note: Make sure that T8O1CE = 0 before setting the clock division ratio.

D0 T8O1CE: 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 T8O1CE default setting is 0, which disables the clock supply. Setting T8O1CE to 1 sends the clock selected as above to the counter.

T8OSC1 Control Register (T8OSC1_CTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T8OSC1 Control Register (T8OSC1_CTL)	0x50c0 (8 bits)	D7–5	–	reserved	–	–	–	0 when being read.
		D4	T8ORST	Timer reset	1 Reset 0 Ignored	0	W	
		D3–2	–	reserved	–	–	–	
		D1	T8ORMD	Count mode select	1 One shot 0 Repeat	0	R/W	
		D0	T8ORUN	Timer run/stop control	1 Run 0 Stop	0	R/W	

D[7:5] Reserved

D4 T8ORST: Timer Reset Bit

Resets the timer.

1 (W): Reset

0 (W): Ignored

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

Writing 1 to this bit resets the counter to 0.

D[3:2] Reserved

D1 T8ORMD: Count Mode Select Bit

Selects the count mode.

1 (R/W): One-shot mode

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

Setting T8ORMD to 0 sets T8OSC1 to repeat mode. In this mode, once the count starts, the timer continues to run until stopped by the application. If the counter matches the compare data register value, the timer resets the counter and continues counting. This means the timer periodically outputs a compare match signal. Set T8OSC1 to this mode to generate periodic interrupts at the desired interval or to perform PWM output.

Setting T8ORMD to 1 sets T8OSC1 to one-shot mode. In this mode, the timer stops automatically when the counter matches the compare data register value. This means an interrupt can be generated only once after the timer has been started. Note that the timer resets the counter and then stops after a compare match has occurred. Set T8OSC1 to this mode to create a specific wait time.

Note: Set the count mode only while T8OSC1 count is stopped.

D0 T8ORUN: Timer Run/Stop Control Bit

Controls the timer Run/Stop.

1 (R/W): Run

0 (R/W): Stop (default)

The timer starts counting when T8ORUN is written as 1 and stops when written as 0. When the timer is stopped, the counter data is retained until reset or until the next Run state.

T8OSC1 Counter Data Register (T8OSC1_CNT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T8OSC1 Counter Data Register (T8OSC1_CNT)	**0x50c1** (8 bits)	D7–0	**T8OCNT[7:0]**	Timer counter data T8OCNT7 = MSB T8OCNT0 = LSB	0x0 to 0xff	0x0	R	

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

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

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

Note: The correct counter value may not be read out (reading is unstable) if the register is read while counting is underway.

Obtain the counter value by one of the following methods:

- Read the counter value while the counter is halted.
- Read the counter twice in succession. Treat the value as valid if the values read are identical.

T8OSC1 Compare Data Register (T8OSC1_CMP)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T8OSC1 Compare Data Register (T8OSC1_CMP)	**0x50c2** (8 bits)	D7–0	**T8OCMP[7:0]**	Compare data T8OCMP7 = MSB T8OCMP0 = LSB	0x0 to 0xff	0x0	R/W	

D[7:0] T8OCMP[7:0]: Compare Data

Sets compare data. (Default: 0x0)

The data set is compared against the counter data, and a compare match interrupt cause is generated if the contents match. And the counter is reset to 0.

T8OSC1 Interrupt Mask Register (T8OSC1_IMSK)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T8OSC1 Interrupt Mask Register (T8OSC1_IMSK)	**0x50c3** (8 bits)	D7–1	–	reserved	–	–	–	0 when being read.
		D0	**T8OIE**	T8OSC1 interrupt enable	1 Enable 0 Disable	0	R/W	

D[7:1] Reserved

D0 T8OIE: T8OSC1 Interrupt Enable Bit

Enables or disables compare match interrupts.

1 (R/W): Interrupt enabled

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

Setting T8OIE to 1 enables T8OSC1 interrupt requests to the ITC. Setting it to 0 disables interrupts.

T8OSC1 Interrupt Flag Register (T8OSC1_IFLG)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T8OSC1 Interrupt Flag Register (T8OSC1_IFLG)	**0x50c4** (8 bits)	D7–1	–	reserved	–	–	–	0 when being read.
		D0	**T8OIF**	T8OSC1 interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.

D[7:1] Reserved

D0 T8OIF: T8OSC1 Interrupt Flag Bit

Indicates whether the cause of compare match 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

T8OIF is the T8OSC1 module interrupt flag. Setting T8OIE/T8OSC1_IMSK register to 1 sets this to 1 when the counter matches the compare data register value during counting. At the same time the T8OSC1 interrupt request signal is output to the ITC and an interrupt occurs if the ITC and S1C17 Core interrupt conditions are met.
T8OIF is reset by writing 1.

Notes: • To prevent interrupt recurrences, the T8OSC1 module interrupt flag T8OIF must be reset in the interrupt handler routine following a T8OSC1 interrupt.

• To prevent generating unnecessary interrupts, reset T8OIF before enabling compare match interrupts using T8OIE.

T8OSC1 PWM Duty Data Register (T8OSC1_DUTY)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T8OSC1 PWM Duty Data Register (T8OSC1_DUTY)	**0x50c5** (8 bits)	D7–0	**T8ODTY[7:0]**	PWM output duty data T8ODTY7 = MSB T8ODTY0 = LSB	0x0 to 0xff	0x0	R/W	

D[7:0] T8ODTY[7:0]: PWM Output Duty Data Bits

Sets the data that determines the duty ratio of PWM waveform. (Default: 0x0)
The set data is compared against the counter data. If the contents match, the timer output waveform rises. If the counter data matches the compare data, the timer output waveform falls. These processes do not affect the counter data or count process.

15 Clock Timer (CT)

15.1 CT Module Overview

The S1C17624/604/622/602/621 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 15.1.1 shows the CT configuration.

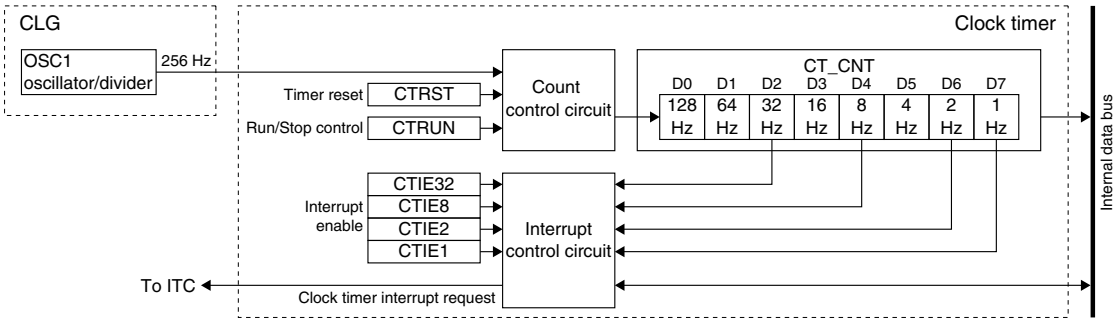


Figure 15.1.1 CT Configuration

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.

15.2 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.

15.3 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.

15.4 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 15.5.
(2) Reset the timer. See Section 15.3.

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).

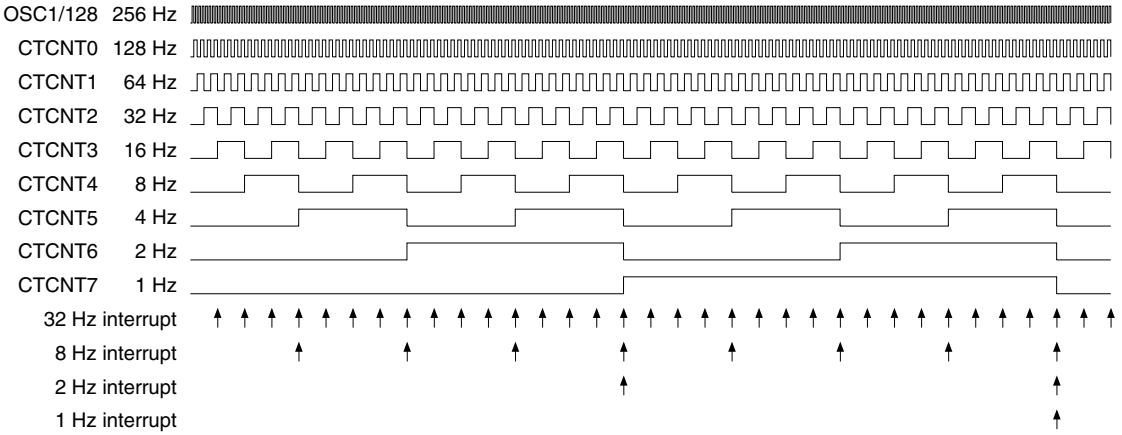


Figure 15.4.1 Clock Timer Timing Chart

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 15.4.2 shows the Run/Stop control timing chart.

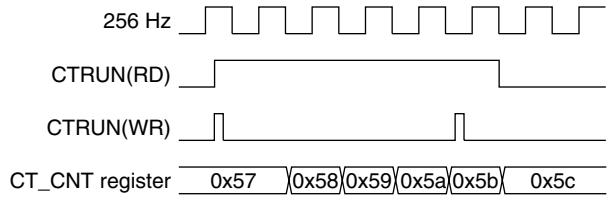


Figure 15.4.2 Run/Stop Control Timing Chart

• 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.

15.5 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 15.5.1 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.

15.6 Control Register Details

Table 15.6.1 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

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

16 Stopwatch Timer (SWT)

16.1 SWT Module Overview

The S1C17624/604/622/602/621 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 16.1.1 shows the SWT configuration.

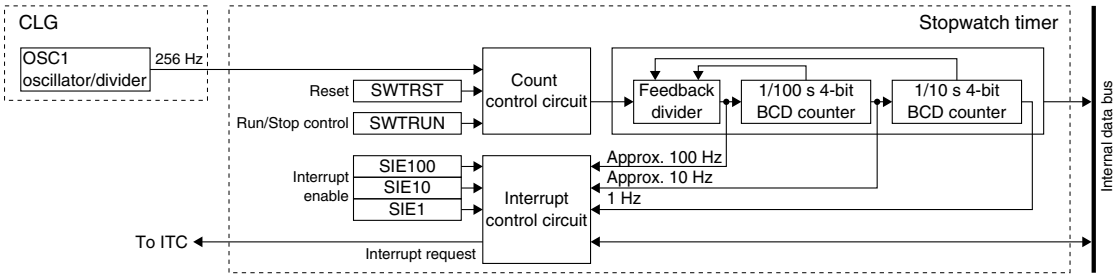


Figure 16.1.1 SWT Configuration

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.

16.2 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.

16.3 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 16.3.1.

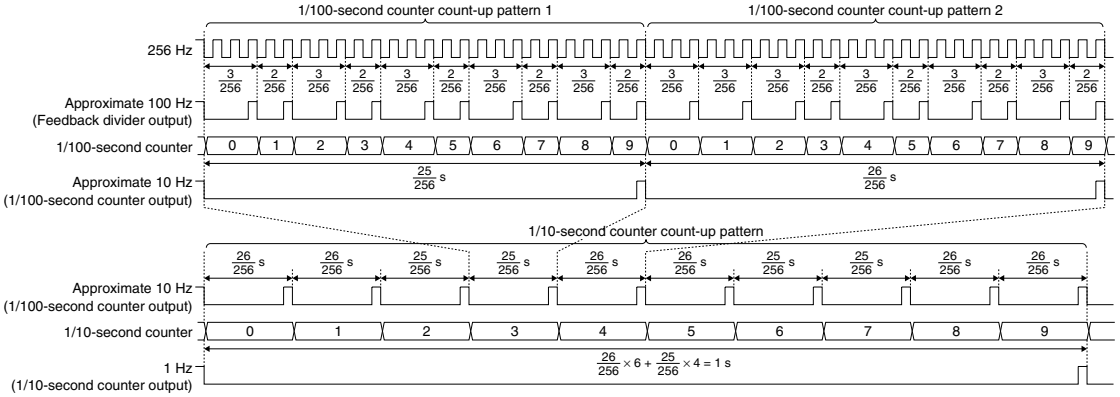


Figure 16.3.1 SWT Count-Up Patterns

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.

16.4 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.

16.5 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 16.6.
(2) Reset the timer. See Section 16.4.

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).

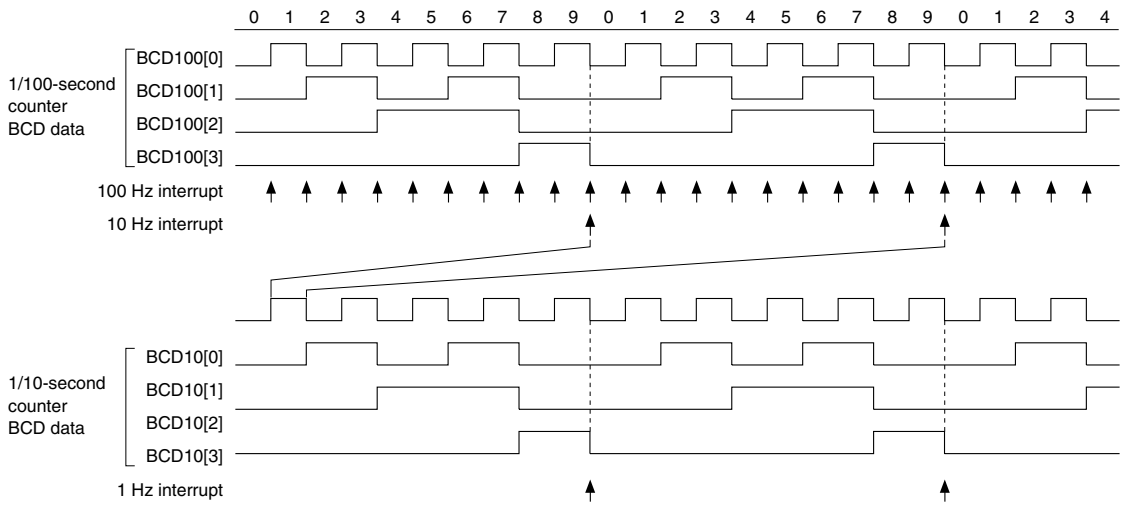


Figure 16.5.1 SWT Timing Chart

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 16.5.2 shows the Run/Stop control timing chart.

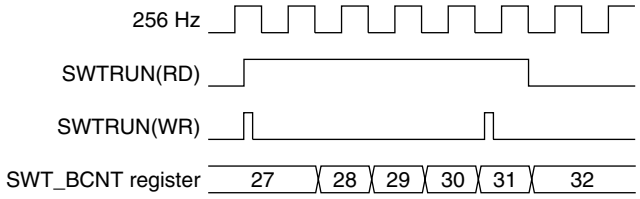


Figure 16.5.2 Run/Stop Control Timing Chart

• 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.

16.6 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 16.6.1 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.

16.7 Control Register Details

Table 16.7.1 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

17 Watchdog Timer (WDT)

17.1 WDT Module Overview

The S1C17624/604/622/602/621 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 17.1.1 shows the WDT configuration.

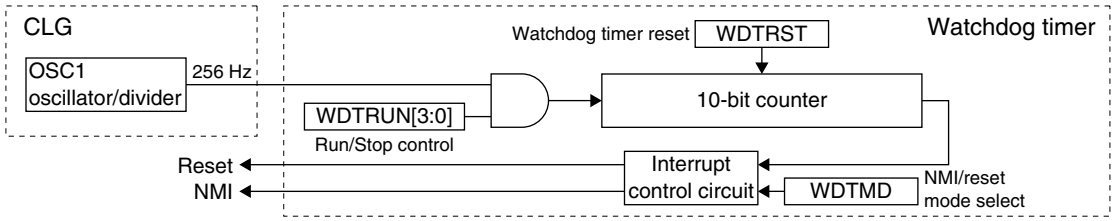


Figure 17.1.1 WDT Configuration

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.

17.2 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.

17.3 WDT Control

17.3.1 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.

17.3.2 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.

17.3.3 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.

17.3.4 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].

17.4 Control Register Details

Table 17.4.1 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.

18 UART

18.1 UART Module Overview

The S1C17624/604/622/602/621 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 460,800 bps (150 to 115,200 bps in IrDA mode)
- Transfer clock: Internal clock (T8F 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 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, and receive error interrupts.

Figure 18.1.1 shows the UART configuration.

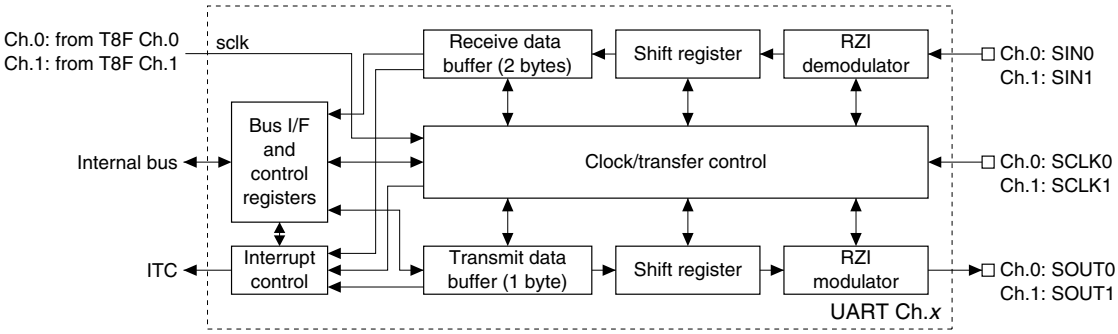


Figure 18.1.1 UART Configuration

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 and pin names indicate a channel number (0 or 1).
Example: UART_CTL*x* register
Ch.0: UART_CTL0 register
Ch.1: UART_CTL1 register

18.2 UART Input/Output Pins

Table 18.2.1 lists the UART input/output pins.

Table 18.2.1 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.

18.3 Transfer Clock

Either the internal clock or an external clock can be selected as the UART transfer clock using SSCK/UART_MOD*x* register.

Note: Make sure that the UART is halted (RXEN/UART_CTL*x* register = 0) before altering SSCK.

Internal clock

Setting SSCK to 0 (default) selects the internal clock. UART Ch.0 uses the T8F Ch.0 output clock as the transfer clock, while UART Ch.1 uses the T8F Ch.1 output clock. Thus, T8F must be programmed to output a clock suited to the transfer rate.
For more information on T8F control, see the "Fine Mode 8-bit Timers (T8F)" chapter.

External clock

Setting SSCK to 1 selects an external clock. In this case, input an external clock from the SCLK*x* pin.

Notes: • The UART generates a sampling clock by dividing the T8F output or the external clock by 16. Take this into consideration when setting the transfer rate.

• When supplying an external clock via the SCLK*x* pin, the clock frequency must be less than half of the PCLK with a duty ratio of 50%.

18.4 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.

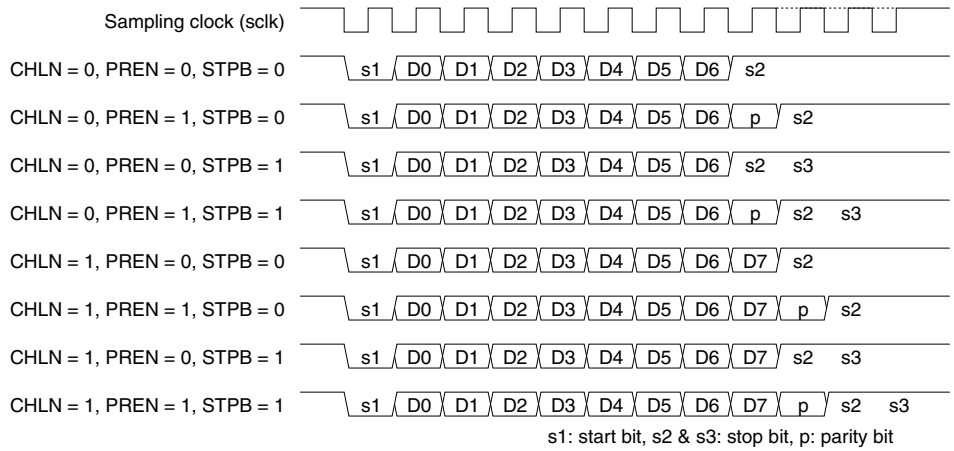


Figure 18.4.1 Transfer Data Format

18.5 Data Transfer Control

Make the following settings before starting data transfers.

(1) Select the input clock. (See Section 18.3.)
Program T8F to output the transfer clock.
(2) Set the transfer data format. (See Section 18.4.)
(3) To use the IrDA interface, set IrDA mode. (See Section 18.8.)
(4) Set interrupt conditions to use UART interrupts. (See Section 18.7.)

Note: Make sure the UART is halted (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.

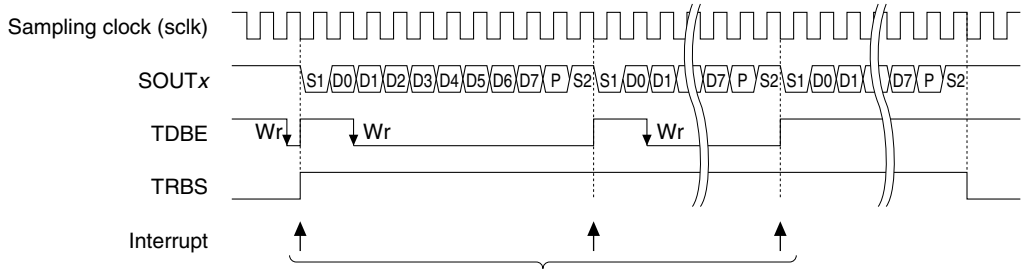
Note: Do not set RXEN to 0 while the UART is sending or receiving data.

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 two status flags: TDBE/UART_ST*x* register and TRBS/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 18.7). 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.



S1: Start bit, S2: Stop bit, P: Parity bit, Wr: Data write to transmit data buffer

Figure 18.5.1 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.

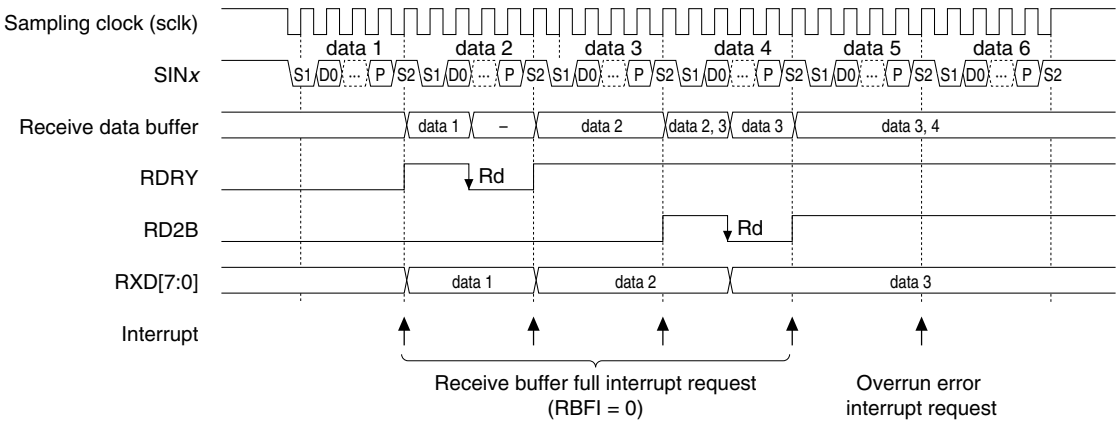
(2) RDRY = 1, RD2B = 0
One 8-bit data has been received. Read the receive data buffer contents once. This resets the RDRY flag. The buffer reverts to state (1) above.
If the receive data buffer contents are read twice, the second data read will be invalid.

(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 18.6.

The volume of data received can be checked by reading these flags.
The UART allows receive buffer full interrupts to be generated once data has been received in the receive data buffer. These interrupts can be used to read the receive data buffer. 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 18.6 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 18.5.2 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.
The data being transferred cannot be guaranteed if RXEN is set to 0 while data is being sent or received. Before setting RXEN to 0, check the data transfer status with software in consideration of the communication procedure. The data transmit status can be checked using the TRBS flag.

Note: Setting RXEN to 0 empties the transmit data buffer, clearing any remaining data. The data being transferred cannot be guaranteed if RXEN is set to 0 while data is being sent or received.
Make sure that the TDBE flag is 1 and the TRBS and RDRY flags are both 0 before disabling data transfer.

18.6 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 18.7.

Parity error

If PREN/UART_MOD*x* register has been set to 1 (parity enabled), data received is checked for parity.
Data received in the shift register is checked for parity when sent to the receive data buffer. The matching is checked against the PMD/UART_MOD*x* register setting (odd or even parity). If the result is a non-match, a parity error is issued, and the parity error flag PER/UART_ST*x* register is set to 1. Even if this error occurs, the data received is sent to the receive data buffer, and the receiving operation continues. However, the received data cannot be guaranteed if a parity error occurs. The PER flag is reset to 0 by writing 1.

Framing error

A framing error occurs if the stop bit is received as 0 and the UART determines loss of sync. If the stop bit is set to two bits, only the first bit is checked.

The framing error flag FER/UART_ST*x* register is set to 1 if this error occurs. The received data is still transferred to the receive data buffer if this error occurs and the receiving operation continues, but the data cannot be guaranteed, even if no framing error occurs for subsequent data receiving. The FER flag is reset to 0 by writing 1.

Overrun error

Even if the receive data buffer is full (two 8-bit data already received), the third data can be received in the shift register. However, if the receive data buffer is not emptied (by reading out data received) by the time this data has been received, the third data received in the shift register will not be sent to the buffer and generate an overrun error. If an overrun error occurs, the overrun error flag OER/UART_ST*x* register is set to 1. The receiving operation continues even if this error occurs. The OER flag is reset to 0 by writing 1.

18.7 UART Interrupts

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

- Transmit buffer empty interrupt
- Receive buffer full interrupt
- Receive error interrupt

Each UART channel outputs one interrupt signal shared by the three 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.

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.

18.8 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.

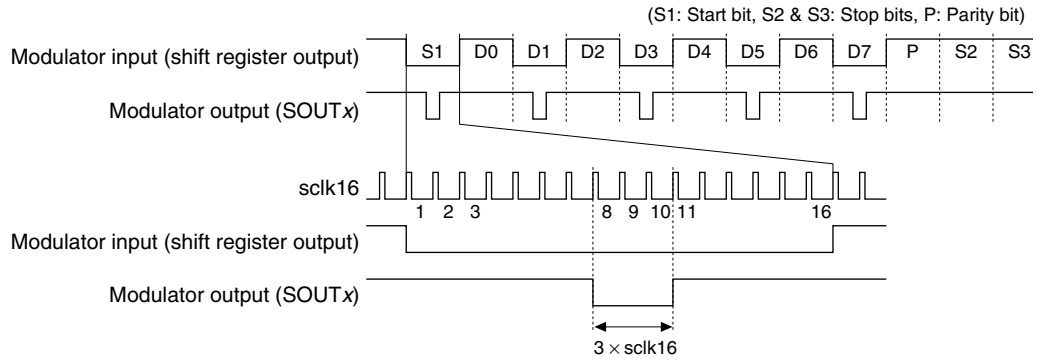


Figure 18.8.1 Transmission Signal Waveform

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).

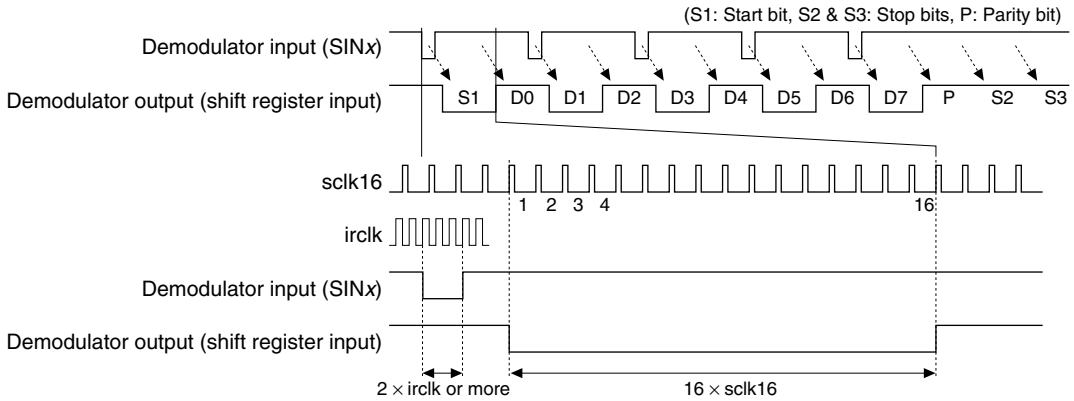


Figure 18.8.2 Receive Signal Waveform

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.

IrDA receive detection clock selection

The input pulse detection clock is generated by dividing PCLK. The division ratio can be selected using IRCLK[2:0]/UART_EXP*x* register.

Table 18.8.1 IrDA Receive Detection Clock (PCLK Division Ratio) Selection

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

(Default: 0x0)

Note: This clock must be selected as a clock faster than sclk16.
The demodulator circuit treats Low pulses with a width of at least two IrDA receive detection clock cycles as valid and converts them to 16 × sclk16 cycle width Low pulses. Select a clock to enable detection of input pulses with a minimum width of 1.41 µs.

Serial data transfer control

Data transfer control in IrDA mode is identical to that for normal interfaces. For detailed information on data format settings and data transfer and interrupt control methods, refer to the preceding sections.

18.9 Control Register Details

Table 18.9.1 List of UART Registers

Address	Register name		Function
0x4100	UART_ST0	UART Ch.0 Status Register	Indicates transfer, buffer and error statuses.
0x4101	UART_TXD0	UART Ch.0 Transmit Data Register	Transmit data
0x4102	UART_RXD0	UART Ch.0 Receive Data Register	Receive data
0x4103	UART_MOD0	UART Ch.0 Mode Register	Sets transfer data format.
0x4104	UART_CTL0	UART Ch.0 Control Register	Controls data transfer.
0x4105	UART_EXP0	UART Ch.0 Expansion Register	Sets IrDA mode.
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.

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, IRCLK[2:0])

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	**–**	reserved	–				–	–	0 when being read.
		D6	**FER**	Framing error flag	1	Error	0	Normal	0	R/W	Reset by writing 1.
		D5	**PER**	Parity error flag	1	Error	0	Normal	0	R/W	
		D4	**OER**	Overrun error flag	1	Error	0	Normal	0	R/W	
		D3	**RD2B**	Second byte receive flag	1	Ready	0	Empty	0	R	
		D2	**TRBS**	Transmit busy flag	1	Busy	0	Idle	0	R	Shift register status
		D1	**RDRY**	Receive data ready flag	1	Ready	0	Empty	0	R	
		D0	**TDBE**	Transmit data buffer empty flag	1	Empty	0	Not empty	1	R	

D7 Reserved

D6 FER: Framing Error Flag Bit

Indicates whether a framing error has occurred or not.

1 (R): Error occurred
0 (R): No error (default)
1 (W): Reset to 0
0 (W): Ignored

FER is set to 1 when a framing error occurs. Framing errors occur when data is received with the stop bit set to 0. FER is reset by writing 1.

D5 PER: Parity Error Flag Bit

Indicates whether a parity error has occurred or not.

1 (R): Error occurred
0 (R): No error (default)
1 (W): Reset to 0
0 (W): Ignored

PER is set to 1 when a parity error occurs. Parity checking is enabled only when PREN/ UART_MOD*x* register is set to 1 and is performed when received data is transferred from the shift register to the receive data buffer. PER is reset by writing 1.

D4 OER: Overrun Error Flag Bit

Indicates whether an overrun error has occurred or not.

1 (R): Error occurred
0 (R): No error (default)
1 (W): Reset to 0
0 (W): Ignored

OER is set to 1 when an overrun error occurs. Overrun errors occur if the receive data buffer is full when data is received in the shift register. The receive data buffer is not overwritten even if this error occurs. The shift register is overwritten as soon as the error occurs.
OER is reset by writing 1.

D3 RD2B: Second Byte Receive Flag Bit

Indicates that the receive data buffer contains two received data.

1 (R): Second byte can be read
0 (R): Second byte not received (default)

RD2B is set to 1 when the second byte of data is loaded into the receive data buffer and is reset to 0 when the first data is read from the receive data buffer.

D2 TRBS: Transmit Busy Flag Bit

Indicates the transmit shift register status.

1 (R): Operating
0 (R): Standby (default)

TRBS is set to 1 when transmit data is loaded from the transmit data buffer into the shift register and is reset to 0 when the data transfer is completed. Inspect TRBS to determine whether the transmit circuit is operating or at standby.

D1 RDRY: Receive Data Ready Flag Bit

Indicates that the receive data buffer contains valid received data.

1 (R): Data can be read
0 (R): Buffer empty (default)

RDRY is set to 1 when received data is loaded into the receive data buffer and is reset to 0 when all data has been read from the receive data buffer.

D0 TDBE: Transmit Data Buffer Empty Flag Bit

Indicates the transmit data buffer status.

1 (R): Buffer empty (default)

0 (R): Data exists

TDBE is reset to 0 when transmit data is written to the transmit data buffer and is set to 1 when the data is transferred to the shift register.

UART Ch.*x* Transmit Data 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	**SSCK**	Input clock select	1	External	0	Internal	0	R/W	

D[7:5] Reserved

D4 **CHLN: Character Length Select Bit**

Selects the serial transfer data length.
1 (R/W): 8 bits
0 (R/W): 7 bits (default)

D3 **PREN: Parity Enable Bit**

Enables the parity function.
1 (R/W): With parity
0 (R/W): No parity (default)

PREN is used to select 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 **SSCK: Input Clock Select Bit**

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

Select whether the internal clock (T8F output clock) or an external clock (input via SCLK*x* pin) is used. Writing 1 to SSCK selects an external clock; Writing 0 to it selects the internal clock.

UART Ch.*x* Control Registers (UART_CTL*x*)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
UART Ch.*x* Control Register (UART_CTL*x*)	0x4104 0x4124 (8 bits)	D7	–	reserved	–				–	–	0 when being read.
		D6	REIEN	Receive error int. enable	1	Enable	0	Disable	0	R/W	
		D5	RIEN	Receive buffer full int. enable	1	Enable	0	Disable	0	R/W	
		D4	TIEN	Transmit buffer empty int. enable	1	Enable	0	Disable	0	R/W	
		D3–2	–	reserved	–				–	–	0 when being read.
		D1	RBFI	Receive buffer full int. condition setup	1	2 bytes	0	1 byte	0	R/W	
		D0	RXEN	UART enable	1	Enable	0	Disable	0	R/W	

D7 **Reserved**

D6 **REIEN: Receive Error Interrupt Enable Bit**

Enables 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.
Disabling transfers by writing 0 to RXEN also clears transmit data buffer.

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

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
UART Ch.*x* Expansion Register (UART_EXP*x*)	**0x4105** **0x4125** (8 bits)	D7	**–**	reserved	–		–	–	0 when being read.
		D6–4	**IRCLK[2:0]**	IrDA receive detection clock division ratio select	IRCLK[2:0]	Division ratio	0x0	R/W	Source clock = PCLK
					0x7	1/128			
					0x6	1/64			
					0x5	1/32			
					0x4	1/16			
					0x3	1/8			
					0x2	1/4			
					0x1	1/2			
					0x0	1/1			
		D3–1	**–**	reserved	–		–	–	0 when being read.
		D0	**IRMD**	IrDA mode select	1 On	0 Off	0	R/W	

D7 Reserved

D[6:4] IRCLK[2:0]: IrDA Receive Detection Clock Division Ratio Select Bits

Selects a PCLK division ratio to generate the IrDA input pulse detection clock.

Table 18.9.2 IrDA Receive Detection Clock (PCLK Division Ratio) Selection

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

(Default: 0x0)

This clock must be selected as a clock faster than sclk16.
The demodulator circuit treats Low pulses with a width of at least 2 IrDA receive detection clock cycles as valid. Select an appropriate clock to enable detection of input pulses with a minimum width of 1.41 µs.

D[3:1] Reserved

D0 IRMD: IrDA Mode Select Bit

Switches the IrDA interface function on and off.
1 (R/W): On
0 (R/W): Off (default)

Set 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.

19 SPI

19.1 SPI Module Overview

The S1C17624/604/622/602/621 includes a synchronized serial interface module (SPI).
The following shows the main features of the SPI:

- Number of channels: 1 channel
- Supports both master and slave modes.
- Data length: 8 bits fixed
- Supports both MSB first and LSB first modes.
- Contains one-byte receive data buffer and one-byte transmit data buffer.
- Supports full-duplex communications.
- Data transfer timing (clock phase and polarity variations) is selectable from among 4 types.
- Can generate receive buffer full and transmit buffer empty interrupts.

Figure 19.1.1 shows the SPI module configuration.

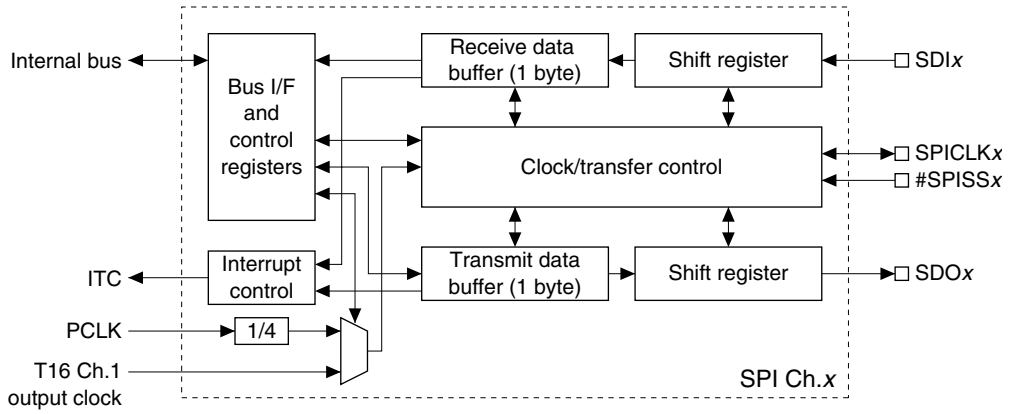


Figure 19.1.1 SPI Module Configuration

Notes: • In the S1C17602/621, the transmit buffer empty interrupt can only be used in master mode. In the S1C17624/604/622, the transmit buffer empty interrupt can be used in both master and slave modes.

• The letter '*x*' in register and pin names refers to a channel number (0).
Example: SPI_CTL*x* register
Ch.0: SPI_CTL0 register

19.2 SPI Input/Output Pins

Table 19.2.1 lists the SPI pins.

Table 19.2.1 List of SPI Pins

Pin name	I/O	Qty	Function
SDI0 (Ch.0)	I	1	SPI data input pin Inputs serial data from SPI bus.
SDO0 (Ch.0)	O	1	SPI data output pin Outputs serial data to SPI bus.

Pin name	I/O	Qty	Function
SPICLK0 (Ch.0)	I/O	1	SPI external clock input/output pin Outputs SPI clock when SPI is in master mode. Inputs external clock when SPI is used in slave mode.
#SPISS0 (Ch.0)	I	1	SPI slave select signal (active Low) input pin SPI (Slave mode) is selected as a slave device by Low input to this pin.

Note: Use an I/O (P) port to output the slave select signal when the SPI module is configured to master mode.

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

19.3 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.

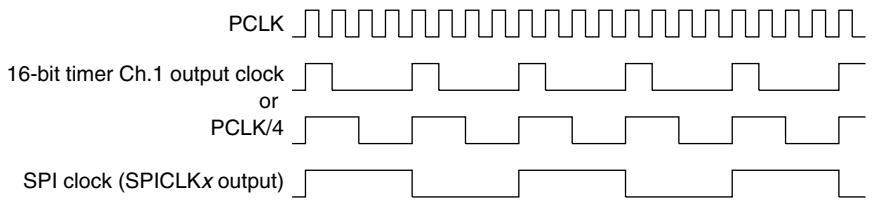


Figure 19.3.1 Master Mode SPI Clock

In slave mode, the SPI clock is input via the SPICLK*x* pin.

19.4 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.

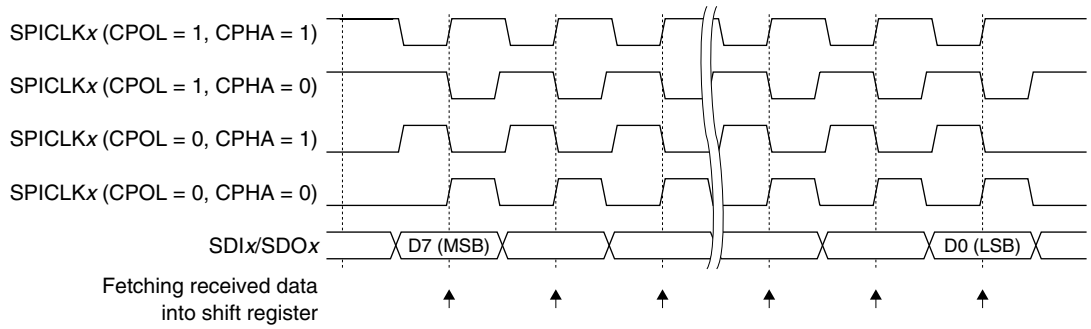


Figure 19.4.1 Clock and Data Transfer Timing

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.

19.5 Data Transfer Control

Make the following settings before starting data transfers.
(1) Select the SPI clock source. (See Section 19.3.)
(2) Select master mode or slave mode. (See Section 19.4.)
(3) Set clock conditions. (See Section 19.4.)
(4) Set the interrupt conditions to use SPI interrupts. (See Section 19.6.)

Note: Make sure the SPI is halted (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 19.4.1) 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 19.6). Subsequent data is sent after confirming that the transmit data buffer is empty either by using this interrupt or by inspecting the SPTBE flag. The 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.

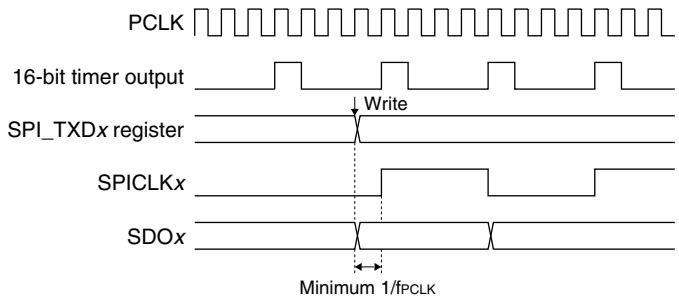


Figure 19.5.1 SDO*x* and SPICLK*x* Change Timings when CPHA = 0

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 19.4.1.) The received data is loaded into the receive data buffer once the 8 bits of data are received in the shift register.
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 19.6). The received data should be read out either by using this interrupt or by inspecting the SPRBF flag to confirm that the receive data buffer contains valid 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.

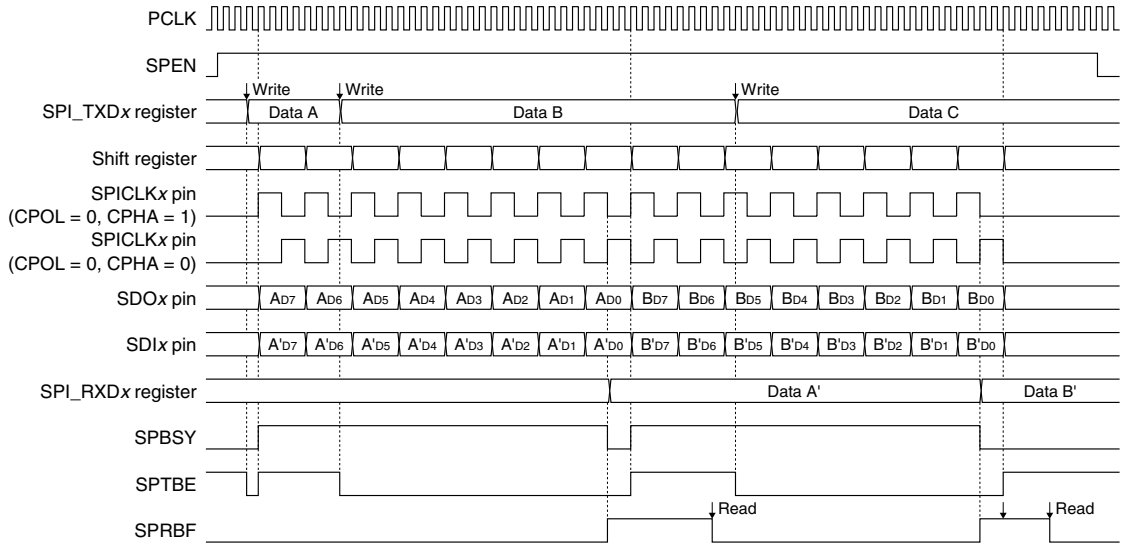


Figure 19.5.2 Data Transmission/Receiving Timing Chart (MSB first)

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.

19.6 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 0, the next transmit data can be written to the transmit data buffer by the interrupt handler routine.

Note: In the S1C17602/621, the transmit buffer empty interrupt can only be used in master mode. In the S1C17624/604/622, the transmit buffer empty interrupt can be used in both master and slave modes.

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.

19.7 Control Register Details

Table 19.7.1 List of SPI Registers

Address	Register name		Function
0x4320	SPI_ST0	SPI Ch.0 Status Register	Indicates transfer and buffer statuses.
0x4322	SPI_TXD0	SPI Ch.0 Transmit Data Register	Transmit data
0x4324	SPI_RXD0	SPI Ch.0 Receive Data Register	Receive data
0x4326	SPI_CTL0	SPI Ch.0 Control Register	Sets the SPI mode and enables data transfer.

The SPI registers are described in detail below. 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 Register (SPI_ST*x*)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
SPI Ch.*x* Status Register (SPI_ST*x*)	**0x4320** (16 bits)	D15–3	–	reserved	–				–	–	0 when being read.
		D2	**SPBSY**	Transfer busy flag (master)	1	Busy	0	Idle	0	R	
				ss signal low flag (slave)	1	ss = L	0	ss = H			
		D1	**SPRBF**	Receive data buffer full flag	1	Full	0	Not full	0	R	
		D0	**SPTBE**	Transmit data buffer empty flag	1	Empty	0	Not empty	1	R	

D[15:3] Reserved

D2 SPBSY: Transfer Busy Flag Bit (Master Mode)/ss Signal Low Flag Bit (Slave Mode)

Master mode

Indicates the SPI transfer status.

1 (R): Operating

0 (R): Standby (default)

SPBSY is set to 1 when the SPI starts data transfer in master mode and is maintained at 1 while transfer is underway. It is cleared to 0 once the transfer is complete.

Slave mode

Indicates the slave selection (#SPISS*x*) signal status.

1 (R): Low level (this SPI is selected)

0 (R): High level (this SPI is not selected) (default)

SPBSY is set to 1 when the master device asserts the #SPISS*x* signal to select this SPI module (slave device). It is returned to 0 when the master device clears the SPI module selection by negating the #SPISS*x* signal.

D1 SPRBF: Receive Data Buffer Full Flag Bit

Indicates the receive data buffer status.

1 (R): Data full

0 (R): No data (default)

SPRBF is set to 1 when data received in the shift register is sent to the receive data buffer (when receiving is completed), indicating that the data can be read. It reverts to 0 once the buffer data is read from the SPI_RXD*x* register.

D0 SPTBE: Transmit Data Buffer Empty Flag Bit

Indicates the transmit data buffer status.

1 (R): Empty (default)

0 (R): Data exists

SPTBE is set to 0 when transmit data is written to the SPI_TXD*x* register (transmit data buffer), and is set to 1 when the data is transferred to the shift register (when transmission starts).
Transmission data must be written to the SPI_TXD*x* register when this bit is 1.

SPI Ch.*x* Transmit Data Register (SPI_TXD*x*)

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

D[15:8] Reserved

D[7:0] SPTDB[7:0]: SPI Transmit Data Buffer Bits

Sets transmit data to be written to the transmit data buffer. (Default: 0x0)
In master mode, transmission is started by writing data to this register. In slave mode, the contents of this register are sent to the shift register and transmission begins when the clock is input from the master.
SPTBE/SPI_ST*x* register is set to 1 (empty) as soon as data written to this register has been transferred to the shift register. A transmit buffer empty interrupt is generated at the same time. The subsequent transmit data can then be written, even while data is being transmitted.
Serial converted data is output from the SDO*x* pin, with the bit set to 1 as High level and the bit set to 0 as Low level.

Note: Make sure that SPEN is set to 1 before writing data to the SPI_TXD*x* register to start data transmission/reception.

SPI Ch.*x* Receive Data Register (SPI_RXD*x*)

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

D[15:8] Reserved

D[7:0] SPRDB[7:0]: SPI Receive Data Buffer Bits

Contains the received data. (Default: 0x0)
SPRBF/SPI_ST*x* register is set to 1 (data full) as soon as data is received and the shift register data has been transferred to the receive data buffer. A receive buffer full interrupt is generated at the same time. Data can then be read until subsequent data is received. If receiving the subsequent data is completed before the register has been read out, the new received data overwrites the contents.
Serial data input from the SDI*x* pin is converted to parallel, with the High level bit set to 1 and the Low level bit set to 0. The data is the loaded into this register.
This register is read-only.

SPI Ch.*x* Control Register (SPI_CTL*x*)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
SPI Ch.*x* Control Register (SPI_CTL*x*)	**0x4326** (16 bits)	D15–10	**–**	reserved	–				–	–	0 when being read.
		D9	**MCLK**	SPI clock source select	1	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: In the S1C17602/621, 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.

Note: In the S1C17602/621, the transmit buffer empty interrupt can only be used in master mode. In the S1C17624/604/622, the transmit buffer empty interrupt can be used in both master and slave modes.

D3 CPHA: Clock Phase Select Bit

Selects the SPI clock phase. (Default: 0)
Set the data transfer timing together with CPOL. (See Figure 19.7.1.)

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 19.7.1.)

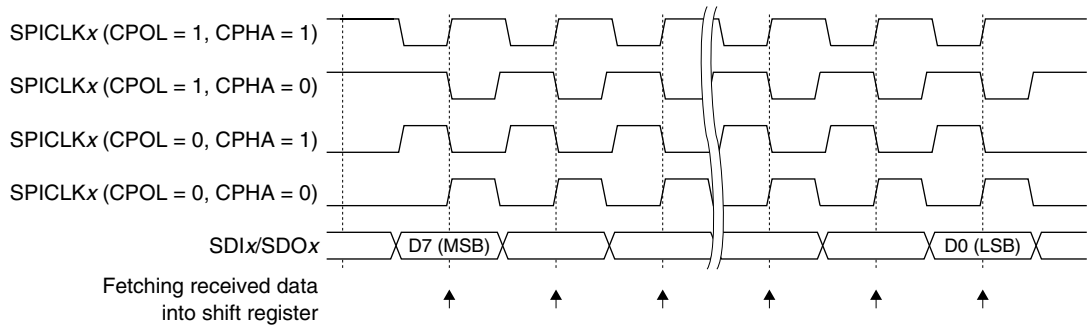


Figure 19.7.1 Clock and Data Transfer Timing

D1 MSSL: Master/Slave Mode Select Bit

Sets the SPI module to master or slave mode.
1 (R/W): Master mode
0 (R/W): Slave mode (default)

Setting MSSL to 1 selects master mode; setting it to 0 selects slave mode. Master mode performs data transfer with the 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.

20 I²C Master (I2CM)

20.1 I2CM Module Overview

The S1C17624/604/622/602/621 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 20.1.1 shows the I2CM configuration.

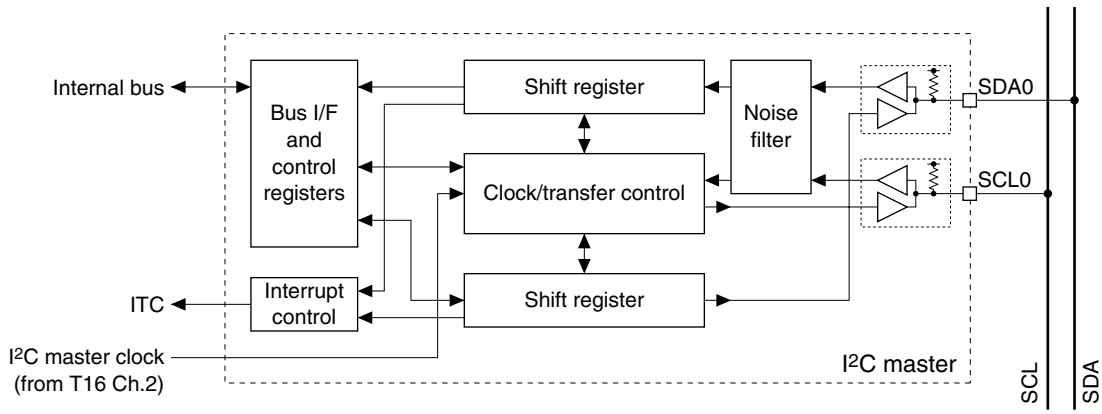


Figure 20.1.1 I2CM Module Configuration

20.2 I2CM Input/Output Pins

Table 20.2.1 lists the I2CM pins.

Table 20.2.1 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 VDD with an external pull-up resistor. Be sure to avoid pulling these pins up to a voltage that exceeds the VDD level.

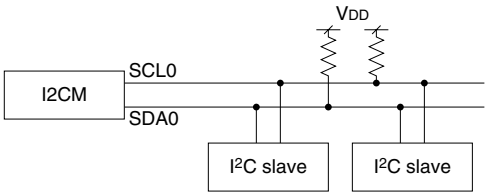


Figure 20.2.1 I²C Connection Example

20.3 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.

20.4 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.

20.5 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 20.4.)
(3) Set the interrupt conditions to use I2CM interrupts. (See Section 20.6.)

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.

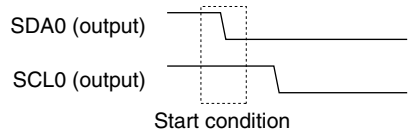


Figure 20.5.1 Start Condition

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 20.5.2 shows the configuration of the address data.

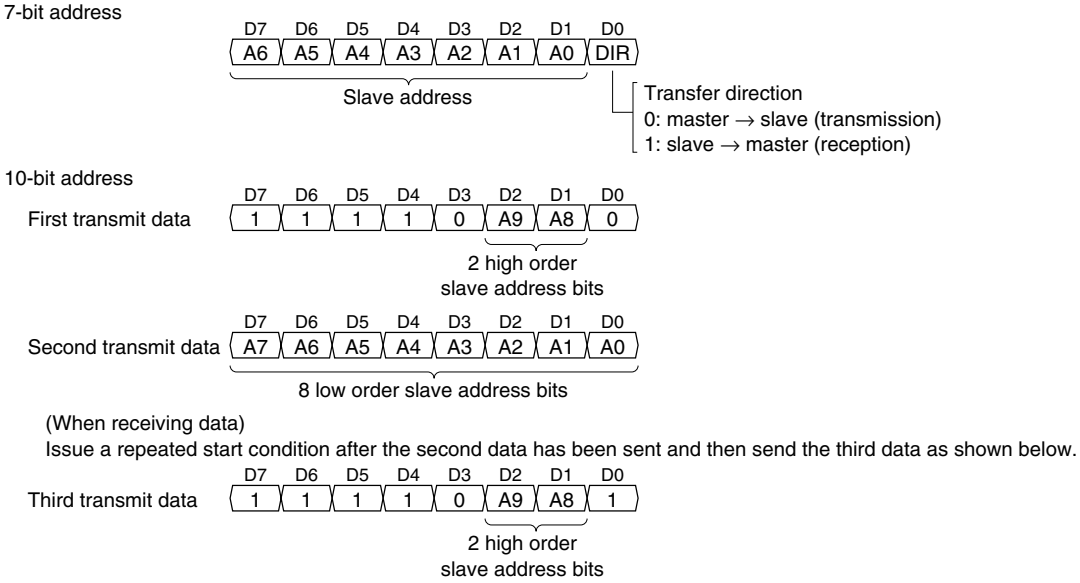


Figure 20.5.2 Transmit Data Specifying Slave Address and Transfer Direction

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).

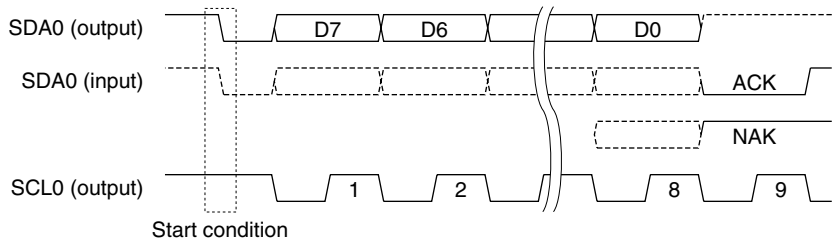


Figure 20.5.3 ACK and NAK

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.

Note: To wait for reception using polling in the S1C17602/621, follow the procedures given below using the RBUSY flag. Interrupts to the CPU are disabled because polling accurately determines the two state transitions 3 and 4.
1. Disable interrupts to the CPU using the `di` instruction.
2. Write 1 to RXE to prepare for receiving.
3. Wait for RBUSY to become 1 (reception start).
4. Wait for RBUSY to become 0 (reception end).
5. Read out RTDT (received data).
6. Enables interrupts to the CPU using the `ei` instruction.

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.

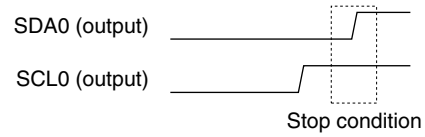


Figure 20.5.4 Stop Condition

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.

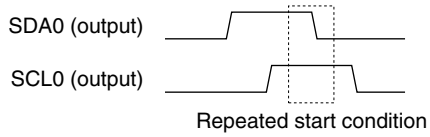


Figure 20.5.5 Repeated Start Condition

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

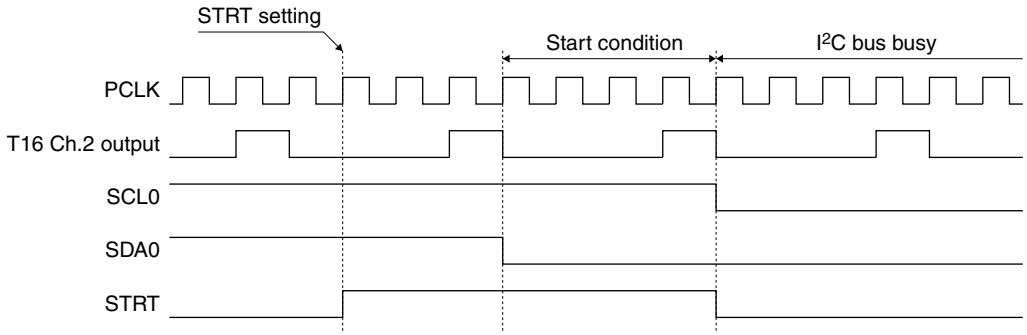


Figure 20.5.6 Start Condition Generation

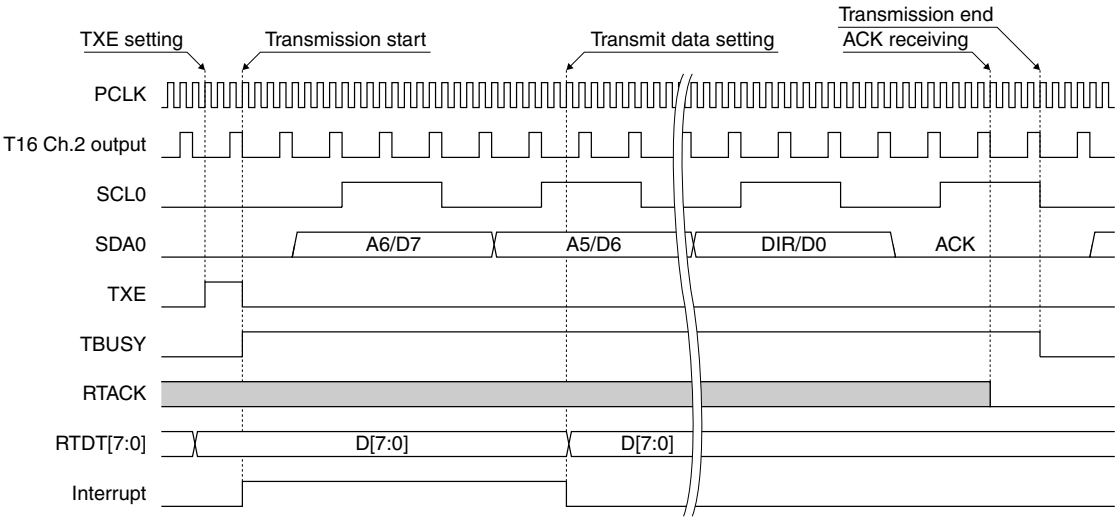


Figure 20.5.7 Slave Address Transmission/Data Transmission

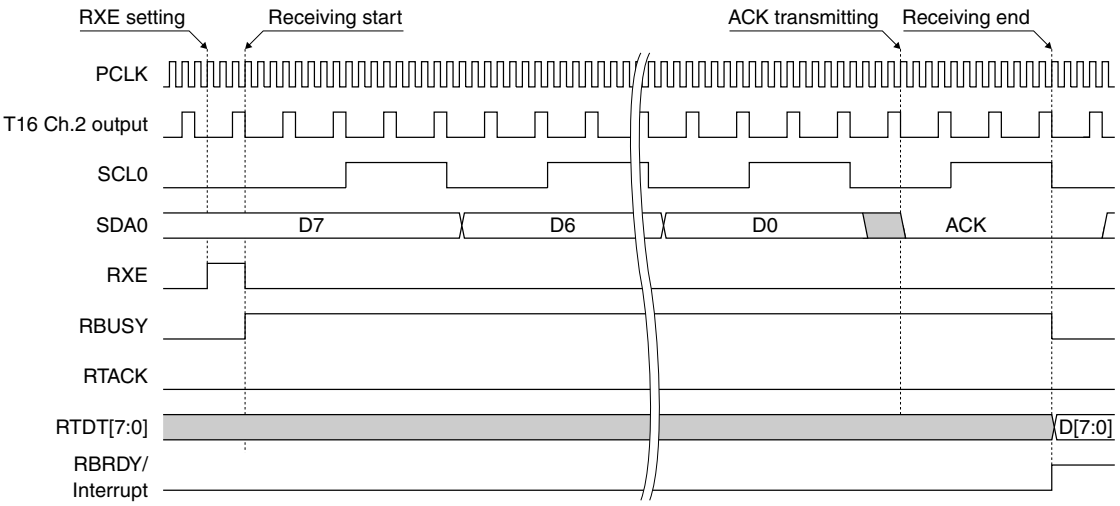


Figure 20.5.8 Data Receiving

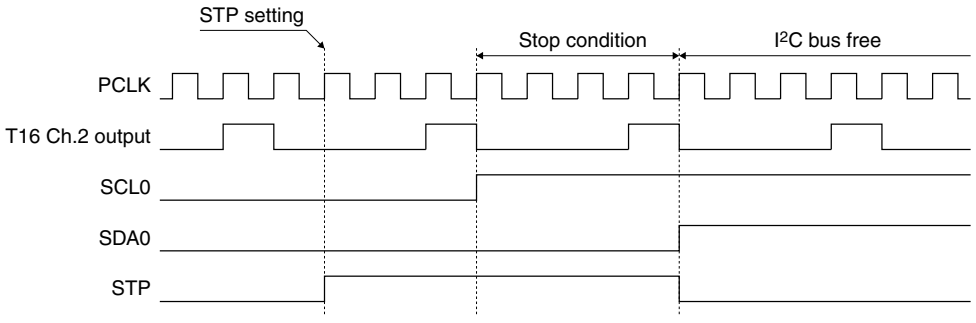


Figure 20.5.9 Stop Condition Generation

20.6 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.
The transmit buffer empty interrupt will only occur during data transmission.

To clear the cause of transmit buffer empty interrupt

The cause of transmit buffer empty interrupt can be cleared by writing data to RTDT[7:0]/I2CM_DAT register. If TXE/I2CM_DAT register is set to 0 at the same time, I2CM only clear the cause of interrupt without sending the data written.

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].
The receive buffer full interrupt will only occur during data reception.

To clear the cause of receive buffer full interrupt

The cause of receive buffer full interrupt can be cleared by reading data from RTDT[7:0]/I2CM_DAT register.

Note: After an I2CM interrupt occurs, determine whether a transmit buffer empty interrupt or a receive buffer full interrupt has occurred according to the I²C master transmit/receive processing being executed at that time. Note that it cannot be checked using a register.

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

20.7 Control Register Details

Table 20.7.1 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.

Note: Use the RBUSY flag when awaiting reception using polling in the S1C17602/621. The RBRDY flag cannot be used to await reception with polling. For more information on awaiting reception control procedures using polling, refer to "Data reception control" in Section 20.5.
In the S1C17624/604/622, the RBRDY flag can be used to await reception with polling. Access the I2CM_DAT register in 16-bit size to read both RBRDY and RTDT[7:0] at a time and use the RTDT[7:0] value as valid receive data when RBRDY = 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.

21 I2C Slave (I2CS)

21.1 I2CS Module Overview

The S1C17624/604/622/602/621 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 21.1.1 shows the I2CS configuration.

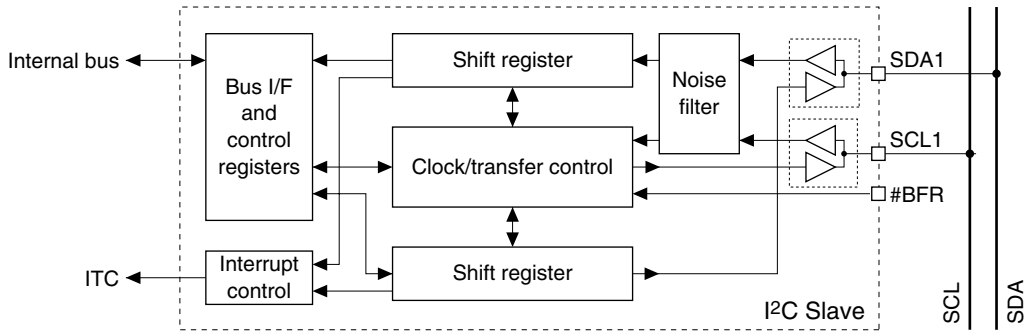


Figure 21.1.1 I2CS Module Configuration

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

21.2 I2CS Input/Output Pins

Table 21.2.1 lists the I2CS pins.

Table 21.2.1 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 VDD with an external pull-up resistor. Be sure to avoid pulling these pins up to a voltage that exceeds the VDD level.

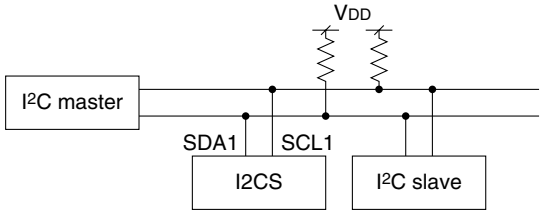


Figure 21.2.1 I2C Connection Example

21.3 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 21.4.3.

21.4 Initializing I2CS

21.4.1 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) cycles 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.

21.4.2 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.

21.4.3 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.

21.5 Data Transfer Control

Make the following settings before starting data transfers.
(1) Initialize the I2CS module. See Section 21.4.
(2) Set the interrupt conditions to use I2CS interrupt. See Section 21.6.

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).

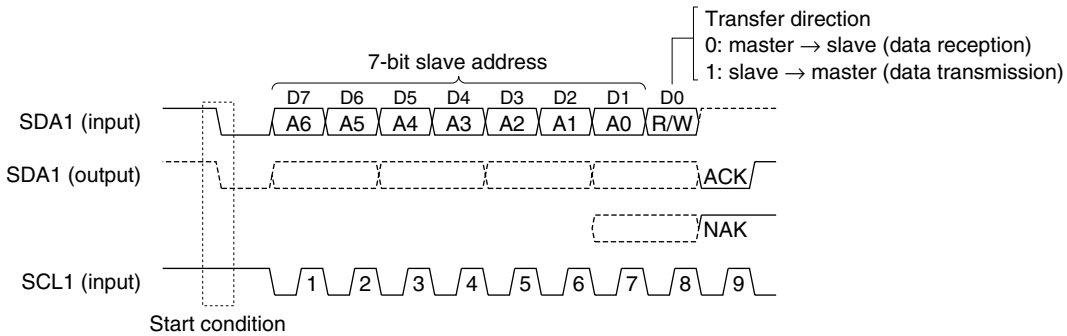


Figure 21.5.1 Receiving Slave Address and Data Direction Bit

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.

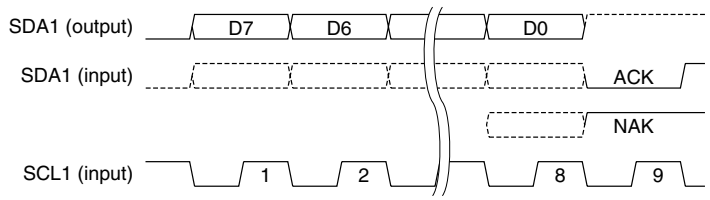


Figure 21.5.2 ACK and NAK

The ACK bit indicates that the master could receive data. It is also a transmit request bit, therefore, the next transmit data must be written in advance. Receiving 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.

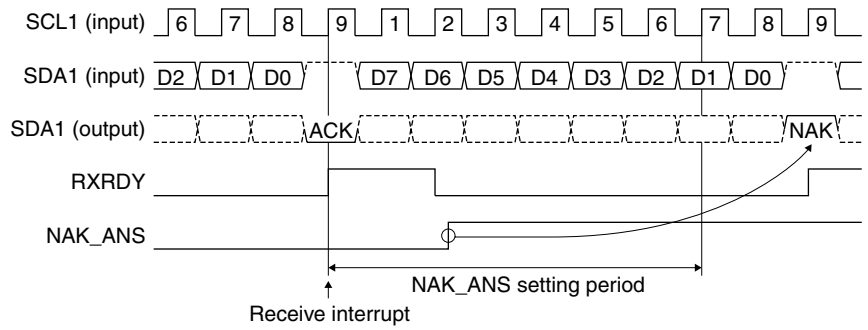


Figure 21.5.3 NAK_ANS Setting and NAK Response Timing

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.

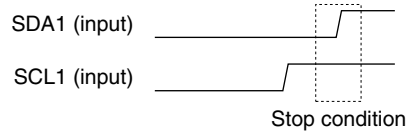


Figure 21.5.4 Stop Condition

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

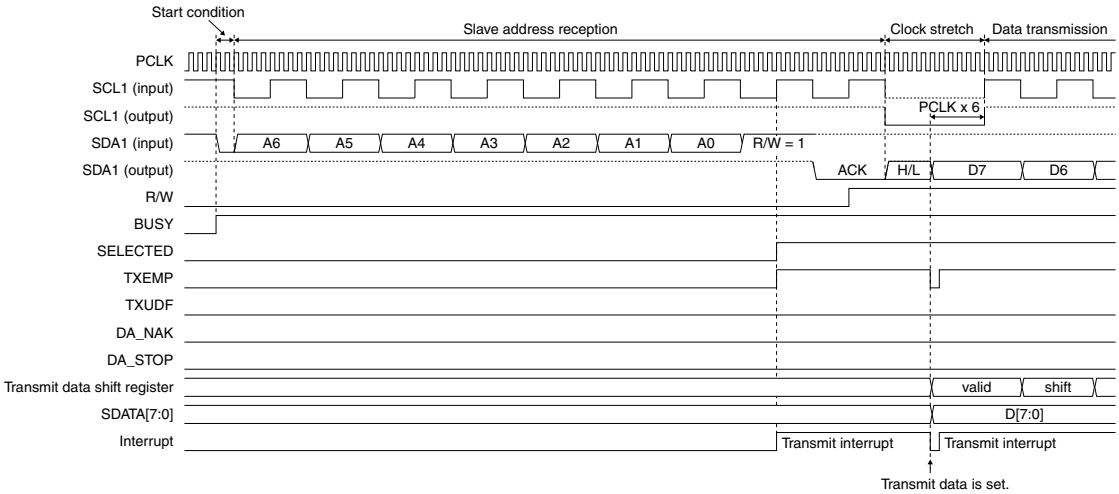


Figure 21.5.5 I2CS Timing Chart 1 (start condition → data transmission)

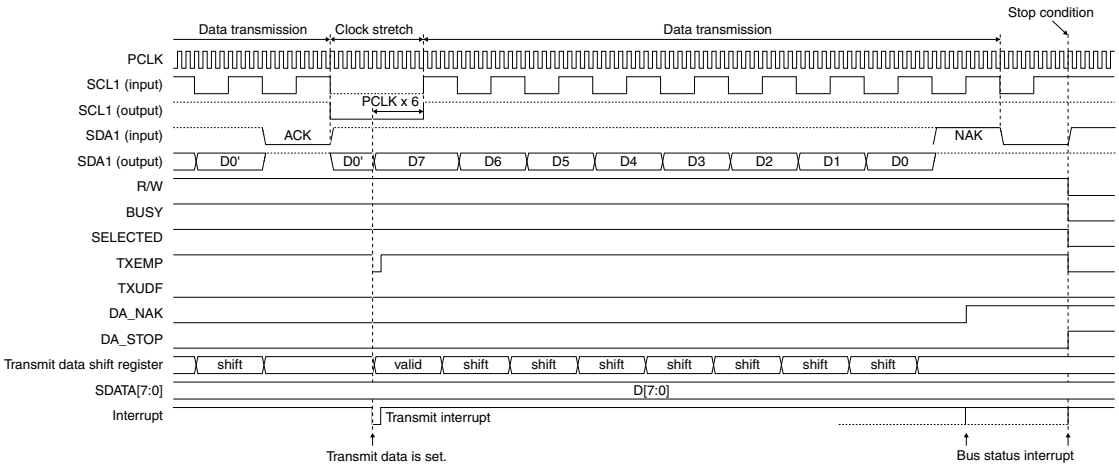


Figure 21.5.6 I2CS Timing Chart 2 (data transmission → stop condition)

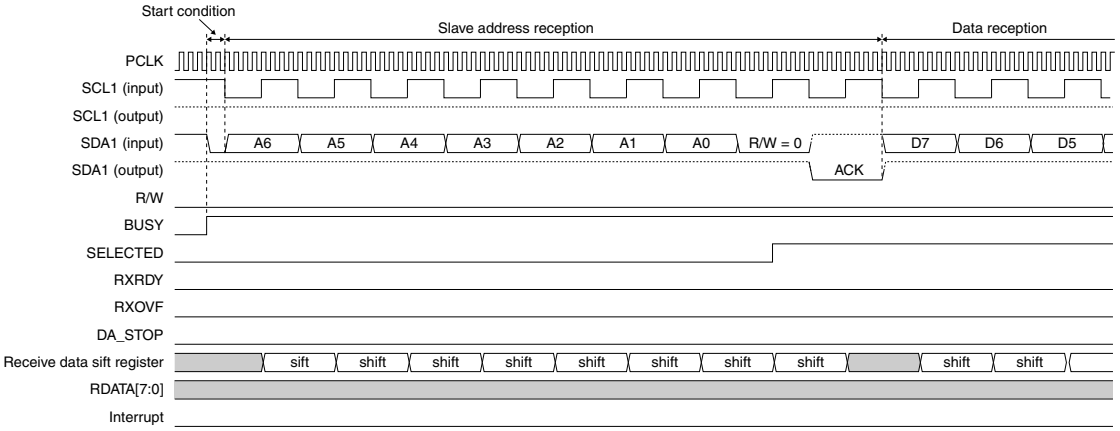


Figure 21.5.7 I2CS Timing Chart 3 (start condition → data reception)

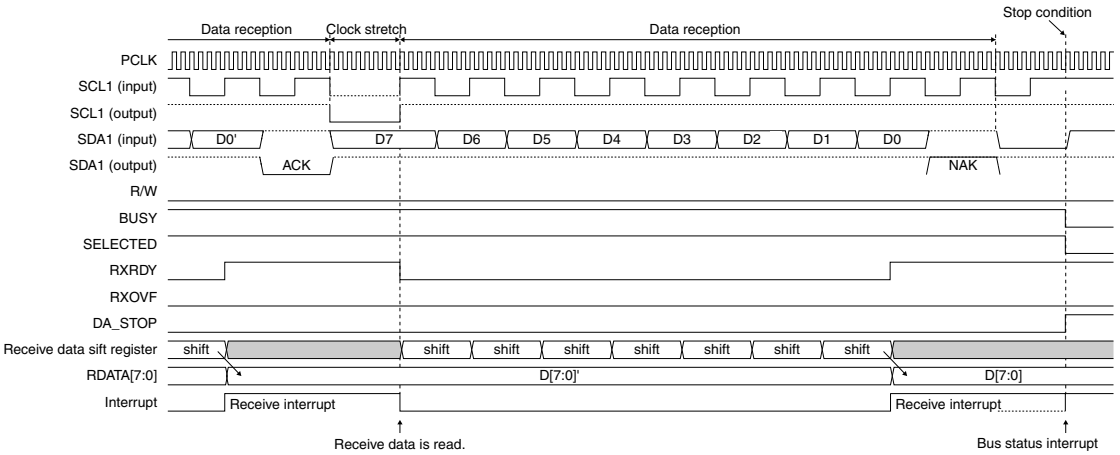


Figure 21.5.8 I2CS Timing Chart 4 (data reception → stop condition)

21.6 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 21.5 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.

21.7 Control Register Details

Table 21.7.1 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.

22 IR Remote Controller (REMC)

22.1 REMC Module Overview

The S1C17624/604/622/602/621 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 22.1.1 shows the configuration of the REMC module.

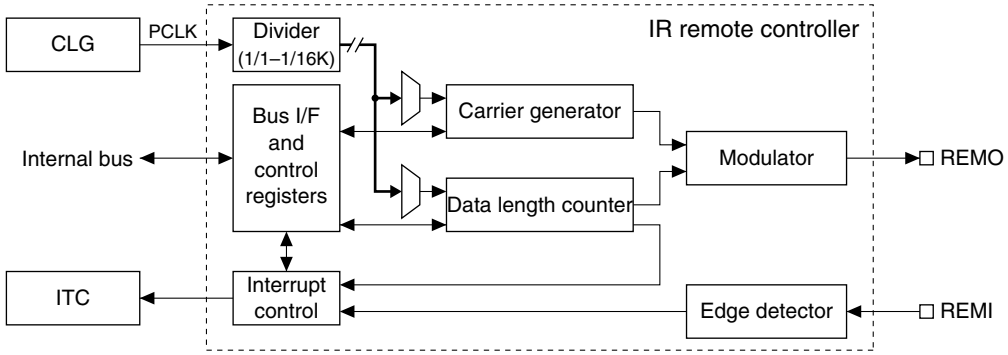


Figure 22.1.1 REMC Module Configuration

22.2 REMC Input/Output Pins

Table 22.2.1 lists the REMC input/output pins.

Table 22.2.1 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.

22.3 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 22.3.1 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 22.3.1.

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

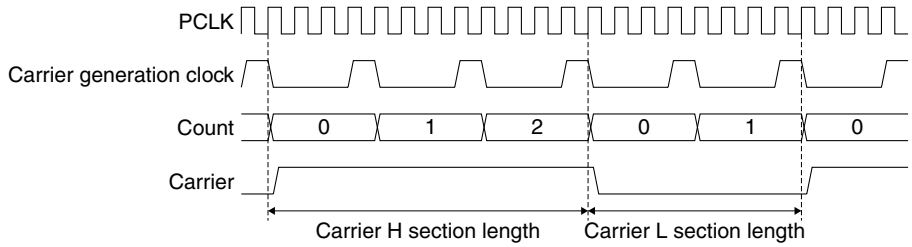


Figure 22.3.1 Carrier Signal Generation

22.4 Data Length Counter Clock Settings

The data length counter is an 8-bit counter for setting data lengths when transmitting data.
When a value corresponding to the data pulse width is written during data transmission, the data length counter begins counting down from that value 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 22.4.1 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.

22.5 Data Transfer Control

Make the following settings before starting data transfers.
(1) Configure the carrier signal. (See Section 22.3.)
(2) Select the data length counter clock. (See Section 22.4.)
(3) Set the interrupt conditions. (See Section 22.6.)

Note: Make sure the REMC module is halted (REMEN/REMC_CFG register = 0) before changing the above settings.

Data transmission control

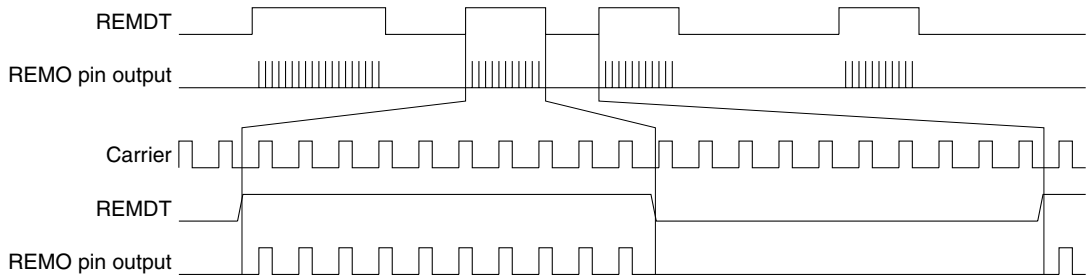


Figure 22.5.1 Data Transmission

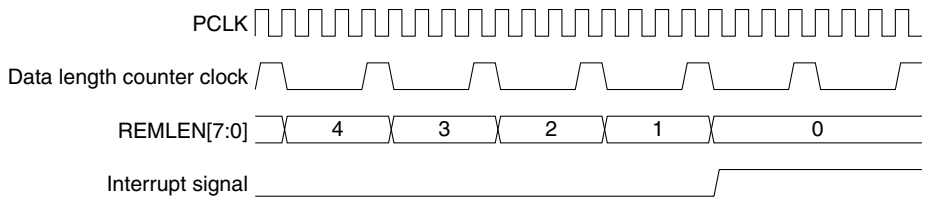


Figure 22.5.2 Underflow Interrupt Generation Timing

(1) Data transmit mode setting
Set REMC to transmit mode by writing 0 to REMMD/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

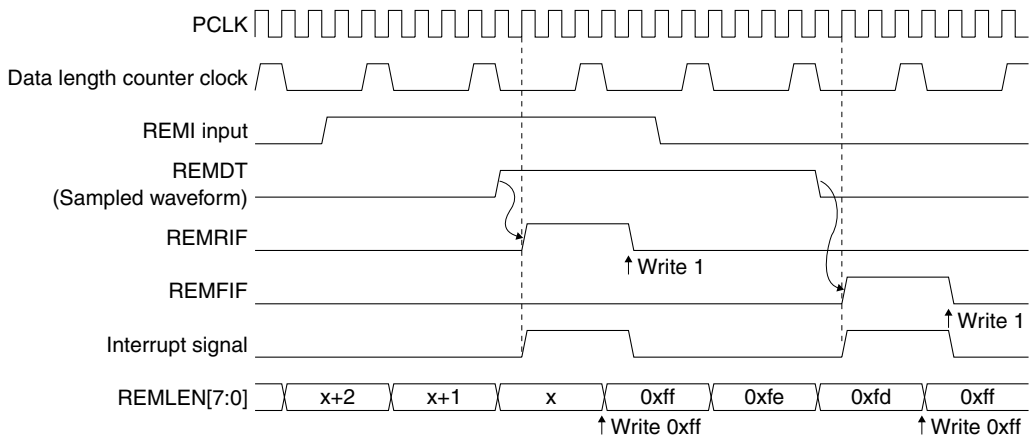


Figure 22.5.3 Data Reception

(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.

22.6 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.

22.7 Control Register Details

Table 22.7.1 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 22.7.2 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 22.7.3 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 22.7.1.

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 22.7.1.

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

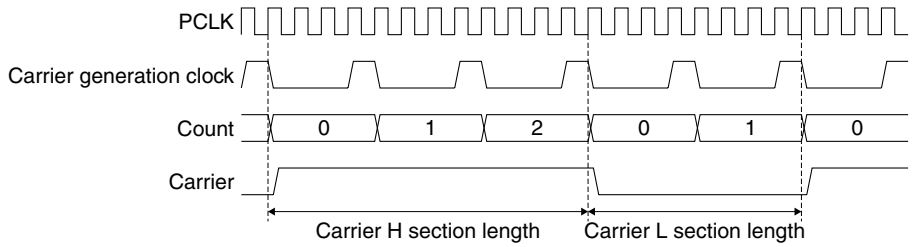


Figure 22.7.1 Carrier Signal Generation

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)

23 LCD Driver (LCD)

23.1 LCD Module Overview

The S1C17624/622 and S1C17604/602/621 include a dot-matrix LCD driver capable of driving an LCD panel with up to 416 segments (52 segments × 8 commons) and 288 segments (36 segments × 8 commons), respectively.
The main features of the LCD driver are listed below.

• Number of SEG and COM outputs	S1C17624/622: 52 SEG × 8 COM or 56 SEG × 4/3/2/1 COM S1C17604/602/621: 36 SEG × 8 COM or 40 SEG × 4/3/2/1 COM
• Drive bias	1/3 bias (fixed)
• Display data RAM	S1C17624/622: 56 bytes S1C17604/602/621: 40 bytes
• Frame frequency configuration	Adjustable with a four-bit counter
• LCD display mode	Normal display mode All on mode All off mode Inverted display mode
• COM and SEG pins	Supports inverting memory bit assignment to the COM and SEG pins.
• LCD contrast adjustment	Selectable from among 16 values
• Other functions	LFRO signal output, frame interrupt

Figure 23.1.1 shows the LCD driver and drive power supply configuration.

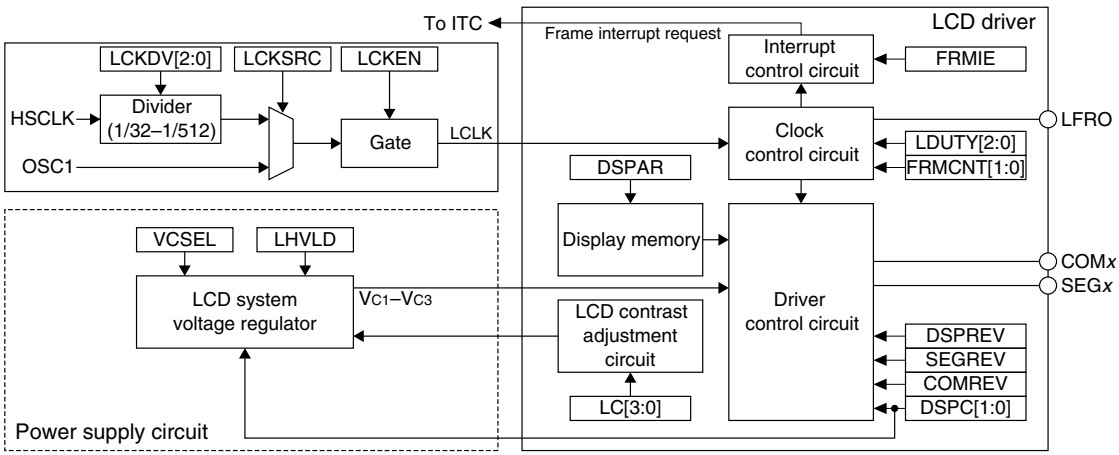


Figure 23.1.1 LCD Driver and Driver Power Supply Configuration

23.2 LCD Power Supply

The LCD drive voltages V_{C1} to V_{C3} are generated by the on-chip LCD voltage regulator. No external power supply is needed. For more information on the LCD power supply, see the "Power Supply" chapter.

23.3 LCD Clock

Figure 23.3.1 shows the LCD clock supply system.

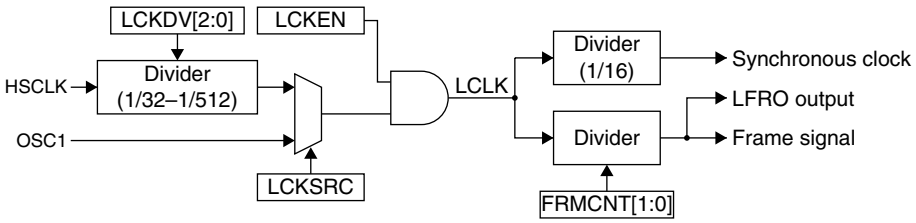


Figure 23.3.1 LCD Clock System

23.3.1 LCD Operating Clock (LCLK)

Clock source selection

Use LCKSRC/OSC_LCLK register to select the clock source from HSCLK (IOSC or OSC3) and OSC1. Setting LCKSRC to 1 (default) selects OSC1 and setting it to 0 selects HSCLK.

Clock division ratio selection

When the clock source is OSC1

No division ratio needs to be selected when OSC1 is selected for the clock source. The OSC1 clock (typ. 32.768 kHz) is directly used as LCLK.

When the clock source is HSCLK

When HSCLK is selected for the clock source, use LCKDV[2:0]/OSC_LCLK register to select the division ratio.

Table 23.3.1.1 HSCLK Division Ratio Selection

LCKDV[2:0]	Division ratio
0x7–0x5	Reserved
0x4	1/512
0x3	1/256
0x2	1/128
0x1	1/64
0x0	1/32

(Default: 0x0)

Clock enable

The LCLK supply is enabled with LCKEN/OSC_LCLK register. The LCKEN default setting is 0, which stops the clock. Setting LCKEN to 1 feeds the clock generated as above to the LCD driver. If no LCD display is required, stop the clock to reduce current consumption.
If LCLK is not supplied, the LCD cannot display. However, the LCD driver control registers and display memory can be accessed even if LCLK is stopped.

Note: Be sure to set LCKEN to 0 before selecting a clock division ratio.

23.3.2 Frame Signal

The LCD driver generates the frame signal by dividing LCLK. The clock division ratio can be set using FRMCNT[1:0]/LCD_CCTL register. Figures 23.4.2.1 to 23.4.2.5 show one cycle of the frame frequency as “1 frame.” Tables 23.3.2.1 and 23.3.2.2 list the frame frequencies that can be programmed.

When the clock source is OSC1

Table 23.3.2.1 Frame Frequency Settings (when OSC1 = 32.768 kHz)

Drive duty (LDUTY[2:0] setting)	FRMCNT[1:0] setting (LCLK division ratio)			
	0x0	0x1	0x2	0x3
1/8 duty (0x4)	128 Hz (1/256)	64 Hz (1/512) *	48.19 Hz (1/680)	32 Hz (1/1024)
1/4 duty (0x3)	128 Hz (1/256)	64 Hz (1/512)	48.19 Hz (1/680)	32 Hz (1/1024)
1/3 duty (0x2)	130.04 Hz (1/252)	65.02 Hz (1/504)	48.12 Hz (1/681)	32.5 Hz (1/1008)
1/2 duty (0x1)	128 Hz (1/256)	64 Hz (1/512)	48.19 Hz (1/680)	32 Hz (1/1024)
Static (0x0)	128 Hz (1/256)	64 Hz (1/512)	48.19 Hz (1/680)	32 Hz (1/1024)

* Default setting

When the clock source is HSCLK

Table 23.3.2.2 Frame Frequency Settings

Drive duty (LDUTY[2:0] setting)	FRMCNT[1:0] setting			
	0x0	0x1	0x2	0x3
1/8 duty (0x4)	$\frac{f_{HSCLK} \times LCKDV}{256}$	$\frac{f_{HSCLK} \times LCKDV}{512}$ *	$\frac{f_{HSCLK} \times LCKDV}{680}$	$\frac{f_{HSCLK} \times LCKDV}{1024}$
1/4 duty (0x3)	$\frac{f_{HSCLK} \times LCKDV}{256}$	$\frac{f_{HSCLK} \times LCKDV}{512}$	$\frac{f_{HSCLK} \times LCKDV}{680}$	$\frac{f_{HSCLK} \times LCKDV}{1024}$
1/3 duty (0x2)	$\frac{f_{HSCLK} \times LCKDV}{252}$	$\frac{f_{HSCLK} \times LCKDV}{504}$	$\frac{f_{HSCLK} \times LCKDV}{681}$	$\frac{f_{HSCLK} \times LCKDV}{1008}$
1/2 duty (0x1)	$\frac{f_{HSCLK} \times LCKDV}{256}$	$\frac{f_{HSCLK} \times LCKDV}{512}$	$\frac{f_{HSCLK} \times LCKDV}{680}$	$\frac{f_{HSCLK} \times LCKDV}{1024}$
Static (0x0)	$\frac{f_{HSCLK} \times LCKDV}{256}$	$\frac{f_{HSCLK} \times LCKDV}{512}$	$\frac{f_{HSCLK} \times LCKDV}{680}$	$\frac{f_{HSCLK} \times LCKDV}{1024}$

* Default setting, f_{HSCLK}: HSCLK (IOSC or OSC3) clock frequency, LCKDV: HSCLK division ratio (1/32 to 1/512)

The frame signal generated can be output to an external device via the LFRO pin. To output the frame signal, set LFROUT/LCD_CCTL register to 1. However, the output pin must be switched for LFRO output using the port function select bit, as the pin is configured for an I/O port by default. For detailed information on pin function switching, see the "I/O Ports (P)" chapter.

23.4 Drive Duty Control

23.4.1 Drive Duty Switching

Drive duty can be set to 1/8, 1/4, 1/3, 1/2 or static drive using LDUTY[2:0]/LCD_CCTL register. Tables 23.4.1.1 and 23.4.1.2 show the correspondence between LDUTY[2:0] settings, drive duty, and maximum number of display segments.

S1C17624/622

Table 23.4.1.1 Drive Duty Settings (S1C17624/622)

LDUTY[2:0]	Duty	Valid COM pins	Valid SEG pins	Max. number of display segments
0x7–0x5	Reserved	–	–	–
0x4	1/8	COM0 to COM7	SEG0 to SEG51	416 segments
0x3	1/4	COM0 to COM3	SEG0 to SEG55	224 segments
0x2	1/3	COM0 to COM2	SEG0 to SEG55	168 segments
0x1	1/2	COM0 to COM1	SEG0 to SEG55	112 segments
0x0	Static	COM0	SEG0 to SEG55	56 segments

(Default: 0x4)

The COM4/SEG55 to COM7/SEG52 pins are configured to COM pins when 1/8 duty is selected or SEG pins when other duty is selected.

S1C17604/602/621

Table 23.4.1.2 Drive Duty Settings (S1C17604/602/621)

LDUTY[2:0]	Duty	Valid COM pins	Valid SEG pins	Max. number of display segments
0x7–0x5	Reserved	–	–	–
0x4	1/8	COM0 to COM7	SEG0 to SEG35	288 segments
0x3	1/4	COM0 to COM3	SEG0 to SEG39	160 segments
0x2	1/3	COM0 to COM2	SEG0 to SEG39	120 segments
0x1	1/2	COM0 to COM1	SEG0 to SEG39	80 segments
0x0	Static	COM0	SEG0 to SEG39	40 segments

(Default: 0x4)

The COM4/SEG39 to COM7/SEG36 pins are configured to COM pins when 1/8 duty is selected or SEG pins when other duty is selected.

The drive bias is fixed at 1/3 (three potentials V_{C1}, V_{C2}, V_{C3}) for all duty settings.

23.4.2 Drive Waveform

Figures 23.4.2.1 to 23.4.2.5 shows the drive waveforms according to the duty selections.

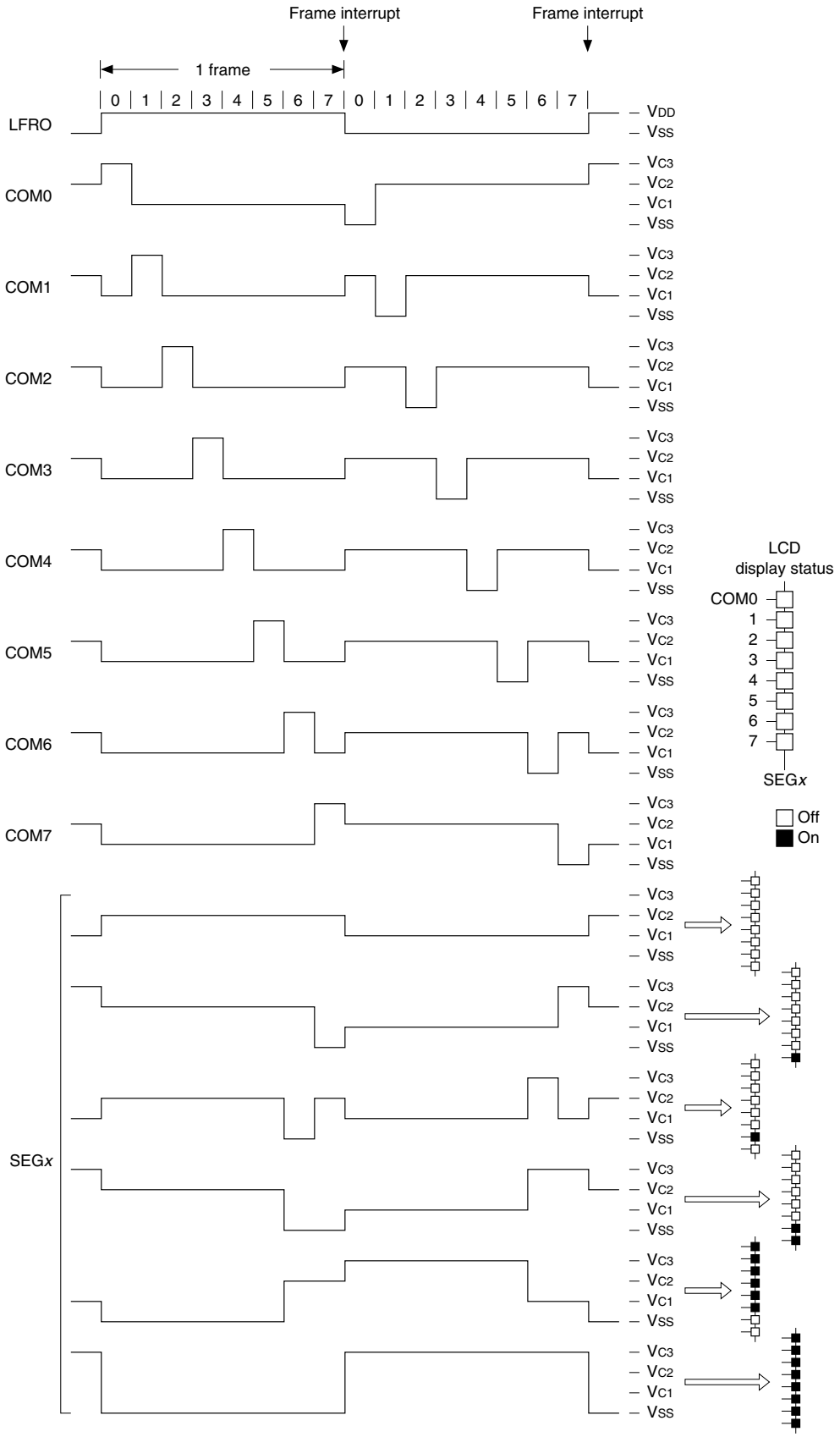


Figure 23.4.2.1 1/8 Duty Drive Waveform

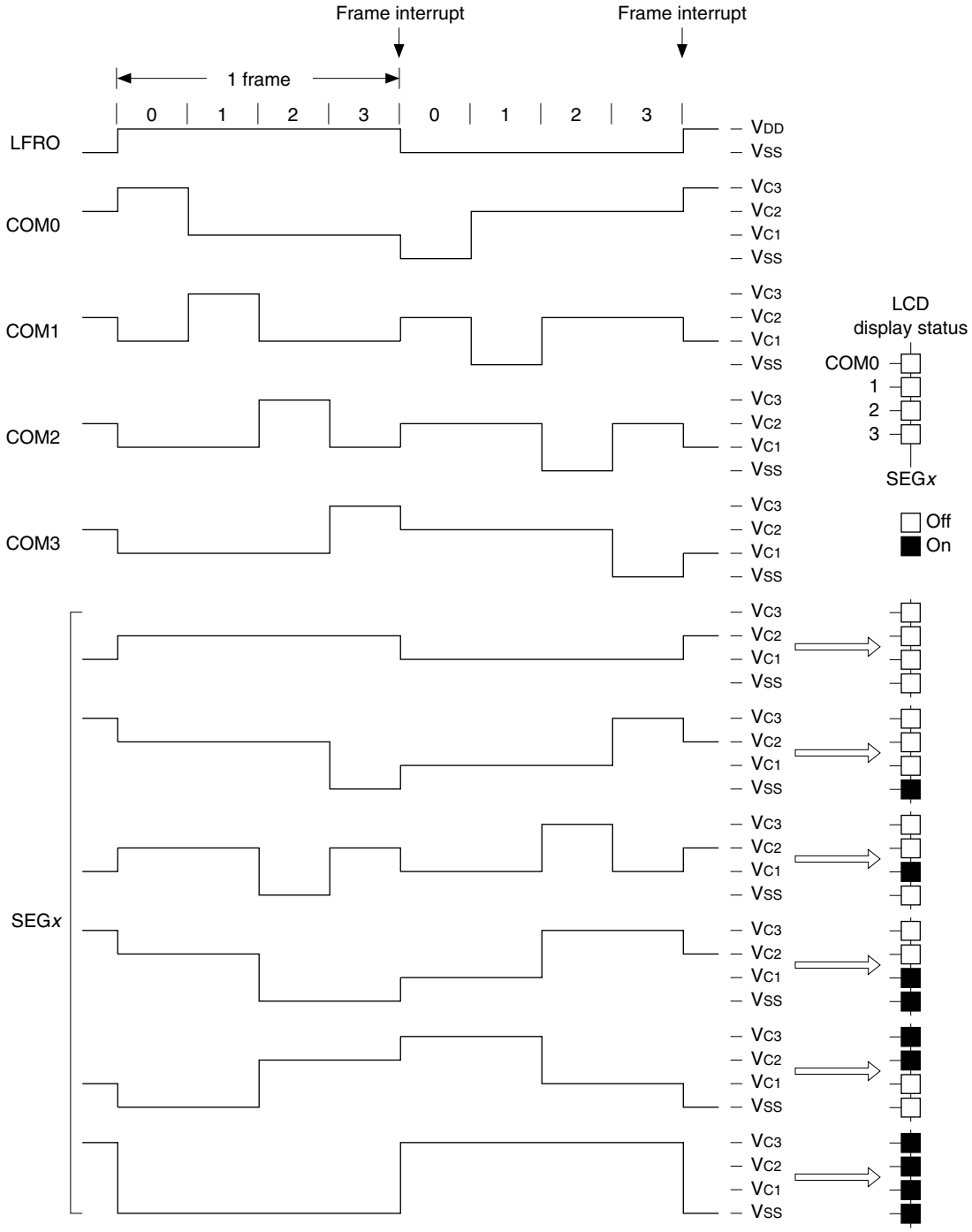


Figure 23.4.2.2 1/4 Duty Drive Waveform

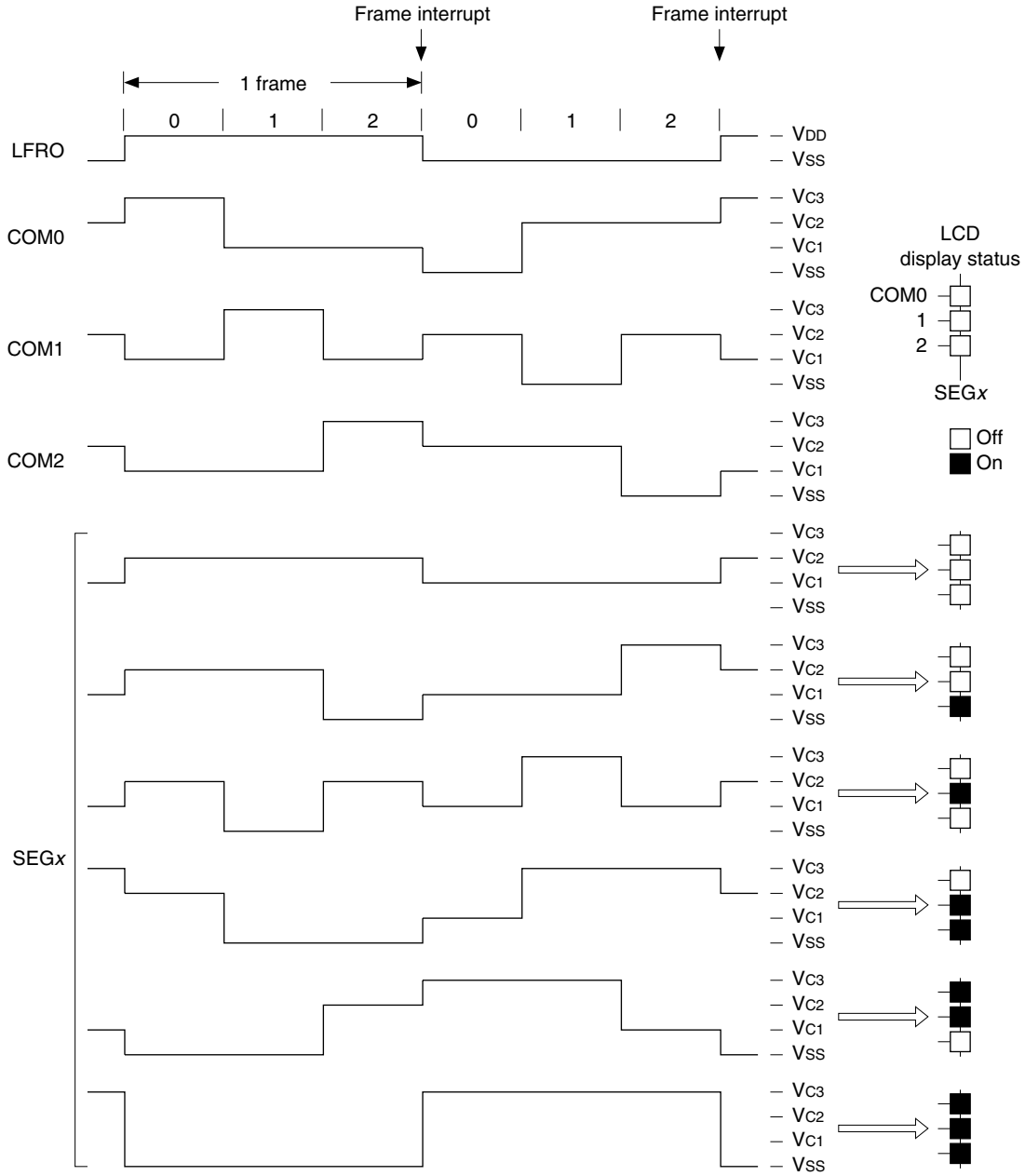


Figure 23.4.2.3 1/3 Duty Drive Waveform

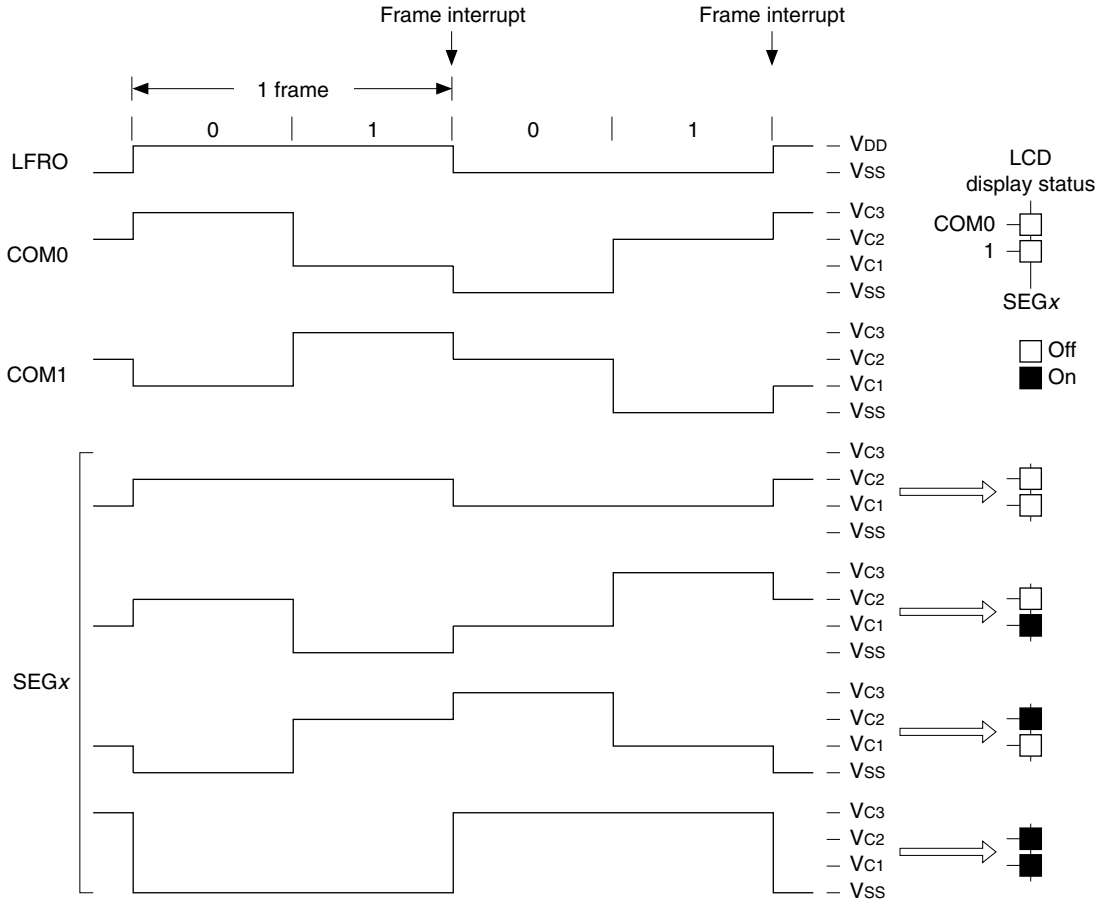


Figure 23.4.2.4 1/2 Duty Drive Waveform

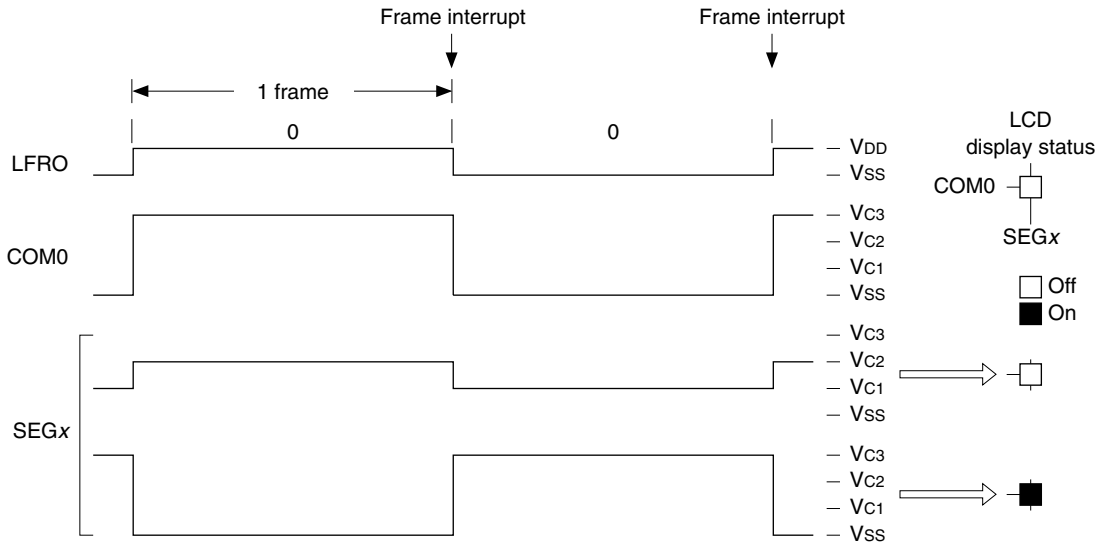


Figure 23.4.2.5 Static Drive Waveform

23.5 Display Memory

The S1C17624/622 includes a 56-byte display memory (address 0x53c0 to address 0x53f7). The S1C17604/602/621 includes a 40-byte display memory (address 0x53c0 to address 0x53e7). The correspondence between memory bits and COM/SEG pins varies depending on the conditions selected, as follows.

(1) Drive duty (1/8, 1/4, 1/3, 1/2 duty or static drive)
(2) SEG pin assignment (normal or inverted)
(3) COM pin assignment (normal or inverted)

Figures 23.5.1 to 23.5.10 show the correspondence between display memory and COM/SEG pins for each drive duty.

Writing 1 to a display memory bit corresponding to pixels on the LCD panel turns that pixel on, while writing 0 turns the pixel off. Since the display memory is a RAM allowing reading and writing, bits can be controlled individually using logic operation instructions (read-modify-write instructions).

Bits not assigned to the display area within the display memory can be used as general-purpose RAM that can be read and written to.

Display area selection (when 1/4, 1/3, 1/2 duty or static drive is selected)

When 1/4, 1/3, 1/2 duty or static drive is selected, two screen areas can be reserved within the display memory, and DSPAR/LCD_DCTL register can be used to switch between the screens. Setting DSPAR to 0 selects display area 0; setting to 1 selects display area 1.

SEG pin assignment

The display memory address assignment for the SEG pins can be inverted using SEGREV/LCD_DCTL register. When SEGREV is set to 1 (default), memory addresses are assigned to SEG pins in ascending order. When SEGREV is set to 0, memory addresses are assigned to SEG pins in descending order. (See Figures 23.5.1 to 23.5.10.)

COM pin assignment

The display memory bit assignment for the COM pins can be inverted using COMREV/LCD_DCTL register. When COMREV is set to 1 (default), memory bits are assigned to COM pins in ascending order. When COMREV is set to 0, memory bits are assigned to COM pins in descending order. (See Figures 23.5.1 to 23.5.10.)

<table>
<tr><th rowspan="2">Bit</th><th colspan="11">Address</th><th rowspan="2">COMREV = 1</th><th rowspan="2">COMREV = 0</th></tr>
<tr><th>0x53c0</th><th>...</th><th>0x53f3</th><th>0x53f4</th><th>...</th><th>0x53f7</th><th>0x53f8</th><th>...</th><th>0x53ff</th></tr>
<tr><td>D0</td><td colspan="3" rowspan="8">Display area</td><td colspan="3" rowspan="8">Unused area (general-purpose memory)</td><td colspan="3" rowspan="8">Unavailable area</td><td>COM0</td><td>COM7</td></tr>
<tr><td>D1</td><td>COM1</td><td>COM6</td></tr>
<tr><td>D2</td><td>COM2</td><td>COM5</td></tr>
<tr><td>D3</td><td>COM3</td><td>COM4</td></tr>
<tr><td>D4</td><td>COM4</td><td>COM3</td></tr>
<tr><td>D5</td><td>COM5</td><td>COM2</td></tr>
<tr><td>D6</td><td>COM6</td><td>COM1</td></tr>
<tr><td>D7</td><td>COM7</td><td>COM0</td></tr>
<tr><td>SEGREV = 1</td><td>SEG0</td><td>...</td><td>SEG51</td><td colspan="9"></td><td colspan="2"></td></tr>
<tr><td>SEGREV = 0</td><td>SEG51</td><td>...</td><td>SEG0</td><td colspan="9"></td><td colspan="2"></td></tr>
</table>

Figure 23.5.1 S1C17624/622 Display Memory Map (1/8 duty)

Bit	Address								COMREV = 1	COMREV = 0
	0x53c0	...			0x53f7	0x53f8	...	0x53ff		
D0	Display area 0 (DSPAR = 0)					Unavailable area			COM0	COM3
D1									COM1	COM2
D2									COM2	COM1
D3									COM3	COM0
D4	Display area 1 (DSPAR = 1)								COM0	COM3
D5									COM1	COM2
D6									COM2	COM1
D7									COM3	COM0
SEGREV = 1	SEG0	...			SEG55					
SEGREV = 0	SEG55	...			SEG0					

Figure 23.5.2 S1C17624/622 Display Memory Map (1/4 duty)

Bit	Address								COMREV = 1	COMREV = 0
	0x53c0	...			0x53f7	0x53f8	...	0x53ff		
D0	Display area 0 (DSPAR = 0)					Unavailable area			COM0	COM2
D1									COM1	COM1
D2									COM2	COM0
D3	Unused area (general-purpose memory)								–	–
D4	Display area 1 (DSPAR = 1)								COM0	COM2
D5									COM1	COM1
D6									COM2	COM0
D7	Unused area (general-purpose memory)								–	–
SEGREV = 1	SEG0	...			SEG55					
SEGREV = 0	SEG55	...			SEG0					

Figure 23.5.3 S1C17624/622 Display Memory Map (1/3 duty)

Bit	Address								COMREV = 1	COMREV = 0
	0x53c0	...			0x53f7	0x53f8	...	0x53ff		
D0	Display area 0 (DSPAR = 0)					Unavailable area			COM0	COM1
D1									COM1	COM0
D2	Unused area (general-purpose memory)								–	–
D3									–	–
D4	Display area 1 (DSPAR = 1)								COM0	COM1
D5									COM1	COM0
D6	Unused area (general-purpose memory)								–	–
D7									–	–
SEGREV = 1	SEG0	...			SEG55					
SEGREV = 0	SEG55	...			SEG0					

Figure 23.5.4 S1C17624/622 Display Memory Map (1/2 duty)

Bit	Address								COMREV = 1	COMREV = 0
	0x53c0	...			0x53f7	0x53f8	...	0x53ff		
D0	Display area 0 (DSPAR = 0)					Unavailable area			COM0	COM0
D1	Unused area (general-purpose memory)								–	–
D2									–	–
D3									–	–
D4	Display area 1 (DSPAR = 1)								COM0	COM0
D5	Unused area (general-purpose memory)								–	–
D6									–	–
D7									–	–
SEGREV = 1	SEG0	...			SEG55					
SEGREV = 0	SEG55	...			SEG0					

Figure 23.5.5 S1C17624/622 Display Memory Map (Static Drive)

Bit	Address									COMREV = 1	COMREV = 0
	0x53c0	...			0x53e3	0x53e4	...	0x53e7	0x53e8	...	0x53ff		
D0	Display area					Unused area (general-purpose memory)			Unavailable area			COM0	COM7
D1												COM1	COM6
D2												COM2	COM5
D3												COM3	COM4
D4												COM4	COM3
D5												COM5	COM2
D6												COM6	COM1
D7												COM7	COM0
SEGREV = 1	SEG0	...			SEG35								
SEGREV = 0	SEG35	...			SEG0								

Figure 23.5.6 S1C17604/602/621 Display Memory Map (1/8 duty)

Bit	Address							COMREV = 1	COMREV = 0
	0x53c0	...			0x53e7	0x53e8	...	0x53ff		
D0	Display area 0 (DSPAR = 0)					Unavailable area			COM0	COM3
D1									COM1	COM2
D2									COM2	COM1
D3									COM3	COM0
D4	Display area 1 (DSPAR = 1)								COM0	COM3
D5									COM1	COM2
D6									COM2	COM1
D7									COM3	COM0
SEGREV = 1	SEG0	...			SEG39					
SEGREV = 0	SEG39	...			SEG0					

Figure 23.5.7 S1C17604/602/621 Display Memory Map (1/4 duty)

Bit	Address								COMREV = 1	COMREV = 0
	0x53c0	...				0x53e7	0x53e8	...	0x53ff	
D0	Display area 0 (DSPAR = 0)						Unavailable area			COM0	COM2
D1										COM1	COM1
D2										COM2	COM0
D3	Unused area (general-purpose memory)									–	–
D4	Display area 1 (DSPAR = 1)									COM0	COM2
D5										COM1	COM1
D6										COM2	COM0
D7	Unused area (general-purpose memory)									–	–
SEGREV = 1	SEG0	...				SEG39					
SEGREV = 0	SEG39	...				SEG0					

Figure 23.5.8 S1C17604/602/621 Display Memory Map (1/3 duty)

Bit	Address								COMREV = 1	COMREV = 0
	0x53c0	...				0x53e7	0x53e8	...	0x53ff	
D0	Display area 0 (DSPAR = 0)						Unavailable area			COM0	COM1
D1										COM1	COM0
D2	Unused area (general-purpose memory)									–	–
D3										–	–
D4	Display area 1 (DSPAR = 1)									COM0	COM1
D5										COM1	COM0
D6	Unused area (general-purpose memory)									–	–
D7										–	–
SEGREV = 1	SEG0	...				SEG39					
SEGREV = 0	SEG39	...				SEG0					

Figure 23.5.9 S1C17604/602/621 Display Memory Map (1/2 duty)

Bit	Address								COMREV = 1	COMREV = 0
	0x53c0	...				0x53e7	0x53e8	...	0x53ff	
D0	Display area 0 (DSPAR = 0)						Unavailable area			COM0	COM0
D1	Unused area (general-purpose memory)									–	–
D2										–	–
D3										–	–
D4	Display area 1 (DSPAR = 1)									COM0	COM0
D5	Unused area (general-purpose memory)									–	–
D6										–	–
D7										–	–
SEGREV = 1	SEG0	...				SEG39					
SEGREV = 0	SEG39	...				SEG0					

Figure 23.5.10 S1C17604/602/621 Display Memory Map (Static Drive)

23.6 Display Control

23.6.1 Display On/Off

The LCD display state is controlled using DSPC[1:0]/LCD_DCTL register.

Table 23.6.1.1 LCD Display Control

DSPC[1:0]	LCD display
0x3	All off (static)
0x2	All on (dynamic)
0x1	Normal display
0x0	Display off

(Default: 0x0)

For normal display, set DSPC[1:0] to 0x1. Note that the clock must be supplied. (See Section 23.3.)

If "Display off" is selected, the drive voltage supplied from the LCD system voltage regulator stops, and the V_{C1} to V_{C3} pins are all set to V_{SS} level.

Since "All on" and "All off" directly control the driving waveform output by the LCD driver, display memory data is not altered. COM pins are set to dynamic drive for "All on" and to static drive for "All off." This function can be used to make the display flash on and off without altering the display memory.

DSPC[1:0] is reset to 0x0 (Display off) after an initial reset.
DSPC[1:0] is also reset to 0x0 when the `slp` instruction is executed and it reverts to the previous setting after SLEEP mode is canceled.

23.6.2 LCD Contrast Adjustment

The LCD contrast can be adjusted to one of 16 levels using LC[3:0]/LCD_CADJ register. Contrast is adjusted by controlling the voltages V_{C1} to V_{C3} output by the internal LCD system voltage regulator.

Table 23.6.2.1 LCD Contrast Adjustment

LC[3:0]	Contrast
0xf	High (dark)
0xe	↑
:	:
0x1	↓
0x0	Low (light)

(Default: 0x7)

LC[3:0] is set to 0x7 after an initial reset. Initialization via software is required to achieve the required contrast.

23.6.3 Inverted Display

The LCD display can be inverted (black/white inversion) using merely control bit manipulation, without changing the display memory. Setting DSPREV/LCD_DCTL register to 0 inverts the display; setting to 1 returns the display to normal status.
Note that the display will not be inverted if "All off" is selected using DSPC[1:0]. The display will be inverted when "All on" is selected.

23.7 LCD Interrupt

The LCD module includes a function for generating interrupts using the frame signal.

Frame interrupt

This cause of interrupt occurs every frame and sets the interrupt flag FRMIF/LCD_IFLG register in the LCD module to 1. See Figures 23.4.2.1 to 23.4.2.5 for interrupt timings.
To use this interrupt, set FRMIE/LCD_IMSK register to 1. When FRMIE is set to 0 (default), interrupt requests for this interrupt cause are not sent to the interrupt controller (ITC).
If FRMIF is set to 1 while FRMIE is set to 1 (interrupt enabled), the LCD module outputs an interrupt request to the ITC. An interrupt is generated if the ITC and S1C17 Core interrupt conditions are satisfied.
For more information on interrupt processing, see the "Interrupt Controller (ITC)" chapter.

Notes: • To prevent interrupt recurrences, the LCD module interrupt flag FRMIF must be reset in the interrupt handler routine after an LCD interrupt has occurred.

• To prevent unwanted interrupts, FRMIF should be reset before enabling LCD interrupts with FRMIE.

23.8 Control Register Details

Table 23.8.1 List of LCD Registers

Address	Register name		Function
0x5063	OSC_LCLK	LCD Clock Select Register	Selects the LCD clock.
0x50a0	LCD_DCTL	LCD Display Control Register	Controls the LCD display.
0x50a1	LCD_CADJ	LCD Contrast Adjustment Register	Controls the contrast.
0x50a2	LCD_CCTL	LCD Clock Control Register	Controls the LCD drive duty.
0x50a3	LCD_VREG	LCD Voltage Regulator Control Register	Controls the LCD drive voltage regulator.
0x50a5	LCD_IMSK	LCD Interrupt Mask Register	Enables/disables interrupts.
0x50a6	LCD_IFLG	LCD Interrupt Flag Register	Indicates/resets interrupt occurrence status.

The LCD module registers are described in detail below. 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.

LCD Clock Select Register (OSC_LCLK)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
LCD Clock Select Register (OSC_LCLK)	**0x5063** (8 bits)	D7–5	–	reserved	–				–	–	0 when being read.
		D4–2	**LCKDV[2:0]**	LCD clock division ratio select	LCKDV[2:0]	Division ratio			0x0	R/W	When the clock source is HSCLK
					0x7–0x5	reserved					
					0x4	1/512					
					0x3	1/256					
					0x2	1/128					
					0x1	1/64					
					0x0	1/32					
		D1	**LCKSRC**	LCD clock source select	1	OSC1	0	HSCLK	1	R/W	
		D0	**LCKEN**	LCD clock enable	1	Enable	0	Disable	0	R/W	

D[7:5] Reserved

D[4:2] LCKDV[2:0]: LCD Clock Division Ratio Select Bits
Selects the division ratio when HSCLK (IOSC or OSC3) is selected as the LCD clock source.

Table 23.8.2 HSCLK Division Ratio Selection

LCKDV[2:0]	Division ratio
0x7–0x5	Reserved
0x4	1/512
0x3	1/256
0x2	1/128
0x1	1/64
0x0	1/32

(Default: 0x0)

No division ratio needs to be selected if OSC1 is selected as the LCD clock source.

D1 LCKSRC: LCD Clock Source Select Bit

Selects the LCD clock source.
1 (R/W): OSC1 (default)
0 (R/W): HSCLK (IOSC or OSC3)

D0 LCKEN: LCD Clock Enable Bit

Enables or disables the LCD clock supply to the LCD driver.
1 (R/W): Enabled (on)
0 (R/W): Disabled (off) (default)

The LCKEN default setting is 0, which stops the clock. Setting LCKEN to 1 feeds the clock selected as above to the LCD driver. If no LCD display is required, stop the clock to reduce current consumption.

LCD Display Control Register (LCD_DCTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
LCD Display Control Register (LCD_DCTL)	0x50a0 (8 bits)	D7	SEGREV	Segment output assignment control	1 Normal / 0 Reverse	1	R/W	
		D6	COMREV	Common output assignment control	1 Normal / 0 Reverse	1	R/W	
		D5	DSPAR	Display memory area control	1 Area 1 / 0 Area 0	0	R/W	
		D4	DSPREV	Reverse display control	1 Normal / 0 Reverse	1	R/W	
		D3–2	–	reserved	–	–	–	0 when being read.
		D1–0	DSPC[1:0]	LCD display control	DSPC[1:0]: 0x3 All off; 0x2 All on; 0x1 Normal display; 0x0 Display off	0x0	R/W	

D7 SEGREV: Segment Output Assignment Control Bit

Inverts memory assignments for SEG pins.
1 (R/W): Normal (default)
0 (R/W): Inverted

When SEGREV is set to 1 (default), memory addresses are assigned to SEG pins in ascending order. When SEGREV is set to 0, memory addresses are assigned to SEG pins in descending order. (See Figures 23.5.1 to 23.5.10.)

D6 COMREV: Common Output Assignment Control Bit

Inverts memory assignments for COM pins.
1 (R/W): Normal (default)
0 (R/W): Inverted

When COMREV is set to 1 (default), memory bits are assigned to COM pins in ascending order. When COMREV is set to 0, memory bits are assigned to COM pins in descending order. (See Figures 23.5.1 to 23.5.10.)

D5 DSPAR: Display Memory Area Control Bit

Selects the display area.
1 (R/W): Display area 1
0 (R/W): Display area 0 (default)

Selects which of the two display areas reserved in the display memory is displayed when driving the LCD in 1/4, 1/3, 1/2 duty or static drive. Setting DSPAR to 0 selects display area 0; setting to 1 selects display area 1. See Figures 23.5.1 to 23.5.10 for the display areas.

D4 DSPREV: Reverse Display Control Bit

Inverts (negative display) the LCD display.
1 (R/W): Normal display (default)
0 (R/W): Inverted display

Setting DSPREV to 0 inverts the LCD panel display; setting to 1 returns the display to normal status. This operation does not affect the contents of the display memory.

D[3:2] Reserved

D[1:0] DSPC[1:0]: LCD Display Control Bits

Controls the LCD display.

Table 23.8.3 LCD Display Control

DSPC[1:0]	LCD display
0x3	All off (static)
0x2	All on (dynamic)
0x1	Normal display
0x0	Display off

(Default: 0x0)

For normal display, set DSPC[1:0] to 0x1. Note that the clock must be supplied. (See Section 23.3.)

If "Display off" is selected, the drive voltage supplied from the LCD system voltage regulator stops, and the V_{C1} to V_{C3} pins are all set to V_{SS} level.

Since "All on" and "All off" directly control the driving waveform output by the LCD driver, display memory data is not altered. COM pins are set to dynamic drive for "All on" and to static drive for "All off." This function can be used to make the display flash on and off without altering the display memory.

DSPC[1:0] is reset to 0x0 (Display off) after an initial reset. DSPC[1:0] is also reset to 0x0 when the `slp` instruction is executed and it reverts to the previous setting after SLEEP mode is canceled.

LCD Contrast Adjustment Register (LCD_CADJ)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
LCD Contrast Adjustment Register (LCD_CADJ)	0x50a1 (8 bits)	D7–4	–	reserved	–		–	–	0 when being read.
		D3–0	LC[3:0]	LCD contrast adjustment	LC[3:0]	Display	0x7	R/W	
					0xf : 0x0	Dark : Light			

D[7:4] Reserved

D[3:0] LC[3:0]: LCD Contrast Adjustment Bits

Adjusts the LCD contrast by controlling voltages V_{C1} to V_{C3} output by the internal LCD system voltage regulator.

Table 23.8.4 LCD Contrast Adjustment

LC[3:0]	Contrast
0xf	High (dark)
0xe	↑
:	:
0x1	↓
0x0	Low (light)

(Default: 0x7)

LC[3:0] is set to 0x7 after an initial reset. Initialization via software is required to achieve the required contrast.

LCD Clock Control Register (LCD_CCTL)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
LCD Clock Control Register (LCD_CCTL)	0x50a2 (8 bits)	D7–6	FRMCNT[1:0]	Frame frequency control	FRMCNT[1:0]	Division ratio	0x1	R/W	Source clock: LCLK
					0x3	1/1024			
					0x2	1/680			
					0x1	1/512			
					0x0	1/256			
		D5	LFROUT	LFRO output control	1 On	0 Off	0	R/W	
		D4–3	–	reserved	–		–	–	0 when being read.
		D2–0	LDUTY[2:0]	LCD duty select	LDUTY[2:0]	Duty	0x4	R/W	
					0x7–0x5	reserved			
					0x4	1/8			
					0x3	1/4			
					0x2	1/3			
					0x1	1/2			
					0x0	Static			

D[7:6] FRMCNT[1:0]: Frame Frequency Control Bits

Sets the Frame frequency.

When the clock source is OSC1

Table 23.8.5 Frame Frequency Settings (when OSC1 = 32.768 kHz)

Drive duty (LDUTY[2:0] setting)	FRMCNT[1:0] setting (LCLK division ratio)			
	0x0	0x1	0x2	0x3
1/8 duty (0x4)	128 Hz (1/256)	64 Hz (1/512) *	48.19 Hz (1/680)	32 Hz (1/1024)
1/4 duty (0x3)	128 Hz (1/256)	64 Hz (1/512)	48.19 Hz (1/680)	32 Hz (1/1024)
1/3 duty (0x2)	130.04 Hz (1/252)	65.02 Hz (1/504)	48.12 Hz (1/681)	32.5 Hz (1/1008)
1/2 duty (0x1)	128 Hz (1/256)	64 Hz (1/512)	48.19 Hz (1/680)	32 Hz (1/1024)
Static (0x0)	128 Hz (1/256)	64 Hz (1/512)	48.19 Hz (1/680)	32 Hz (1/1024)

* Default setting

When the clock source is HSCLK

Table 23.8.6 Frame Frequency Settings

Drive duty (LDUTY[2:0] setting)	FRMCNT[1:0] setting			
	0x0	0x1	0x2	0x3
1/8 duty (0x4)	$\frac{f_{HSCLK} \times LCKDV}{256}$	$\frac{f_{HSCLK} \times LCKDV}{512}$ *	$\frac{f_{HSCLK} \times LCKDV}{680}$	$\frac{f_{HSCLK} \times LCKDV}{1024}$
1/4 duty (0x3)	$\frac{f_{HSCLK} \times LCKDV}{256}$	$\frac{f_{HSCLK} \times LCKDV}{512}$	$\frac{f_{HSCLK} \times LCKDV}{680}$	$\frac{f_{HSCLK} \times LCKDV}{1024}$
1/3 duty (0x2)	$\frac{f_{HSCLK} \times LCKDV}{252}$	$\frac{f_{HSCLK} \times LCKDV}{504}$	$\frac{f_{HSCLK} \times LCKDV}{681}$	$\frac{f_{HSCLK} \times LCKDV}{1008}$
1/2 duty (0x1)	$\frac{f_{HSCLK} \times LCKDV}{256}$	$\frac{f_{HSCLK} \times LCKDV}{512}$	$\frac{f_{HSCLK} \times LCKDV}{680}$	$\frac{f_{HSCLK} \times LCKDV}{1024}$
Static (0x0)	$\frac{f_{HSCLK} \times LCKDV}{256}$	$\frac{f_{HSCLK} \times LCKDV}{512}$	$\frac{f_{HSCLK} \times LCKDV}{680}$	$\frac{f_{HSCLK} \times LCKDV}{1024}$

* Default setting

f_{HSCLK}: HSCLK (IOSC or OSC3) clock frequency, LCKDV: HSCLK division ratio (1/32 to 1/512)

D5 LFROUT: LFRO Output Control Bit

Controls the frame signal (LFRO) output.

1 (R/W): Output enabled (On)

0 (R/W): Output disabled (Off) (default)

Setting LFROUT 1 outputs the frame signal generated by the LCD module from the LFRO pin. Setting it to 0 stops output and the LFRO pin goes a low level.

D[4:3] Reserved

D[2:0] LDUTY[2:0]: LCD Duty Select Bits

Selects the drive duty.

Table 23.8.7 Drive Duty Settings (S1C17624/622)

LDUTY[2:0]	Duty	Valid COM pins	Valid SEG pins	Max. number of display segments
0x7–0x5	Reserved	–	–	–
0x4	1/8	COM0 to COM7	SEG0 to SEG51	416 segments
0x3	1/4	COM0 to COM3	SEG0 to SEG55	224 segments
0x2	1/3	COM0 to COM2	SEG0 to SEG55	168 segments
0x1	1/2	COM0 to COM1	SEG0 to SEG55	112 segments
0x0	Static	COM0	SEG0 to SEG55	56 segments

(Default: 0x4)

Table 23.8.8 Drive Duty Settings (S1C17604/602/621)

LDUTY[2:0]	Duty	Valid COM pins	Valid SEG pins	Max. number of display segments
0x7–0x5	Reserved	–	–	–
0x4	1/8	COM0 to COM7	SEG0 to SEG35	288 segments
0x3	1/4	COM0 to COM3	SEG0 to SEG39	160 segments
0x2	1/3	COM0 to COM2	SEG0 to SEG39	120 segments
0x1	1/2	COM0 to COM1	SEG0 to SEG39	80 segments
0x0	Static	COM0	SEG0 to SEG39	40 segments

(Default: 0x4)

LCD Voltage Regulator Control Register (LCD_VREG)

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

For more information on the control bit, see "LCD Voltage Regulator Control Register (LCD_VREG)" in the "Power Supply" chapter.

LCD Interrupt Mask Register (LCD_IMSK)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
LCD Interrupt Mask Register (LCD_IMSK)	0x50a5 (8 bits)	D7–1	–	reserved	–	–	–	0 when being read.
		D0	FRMIE	Frame signal interrupt enable	1 Enable 0 Disable	0	R/W	

D[7:1] Reserved

D0 FRMIE: Frame Signal Interrupt Enable Bit

Enables or disables frame interrupts.

1 (R/W): Interrupt enabled

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

Setting FRMIE to 1 enables LCD interrupt requests to the ITC. Setting to 0 disables interrupts.

LCD Interrupt Flag Register (LCD_IFLG)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
LCD Interrupt Flag Register (LCD_IFLG)	0x50a6 (8 bits)	D7–1	–	reserved	–	–	–	0 when being read.
		D0	FRMIF	Frame signal interrupt flag	1 Occurred 0 Not occurred	0	R/W	Reset by writing 1.

D[7:1] Reserved

D0 FRMIF: Frame Signal Interrupt Flag Bit

Indicates the frame interrupt cause occurrence status.

1 (R): Cause of interrupt has occurred

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

1 (W): Flag is reset

0 (W): Ignored

FRMIF is set to 1 at the frame signal rising edge. FRMIF is reset to 0 by writing 1.

24 A/D Converter (ADC10)

24.1 ADC10 Module Overview

The S1C17624/604/622/602/621 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. 8 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} (= V_{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 24.1.1 shows the ADC10 configuration.

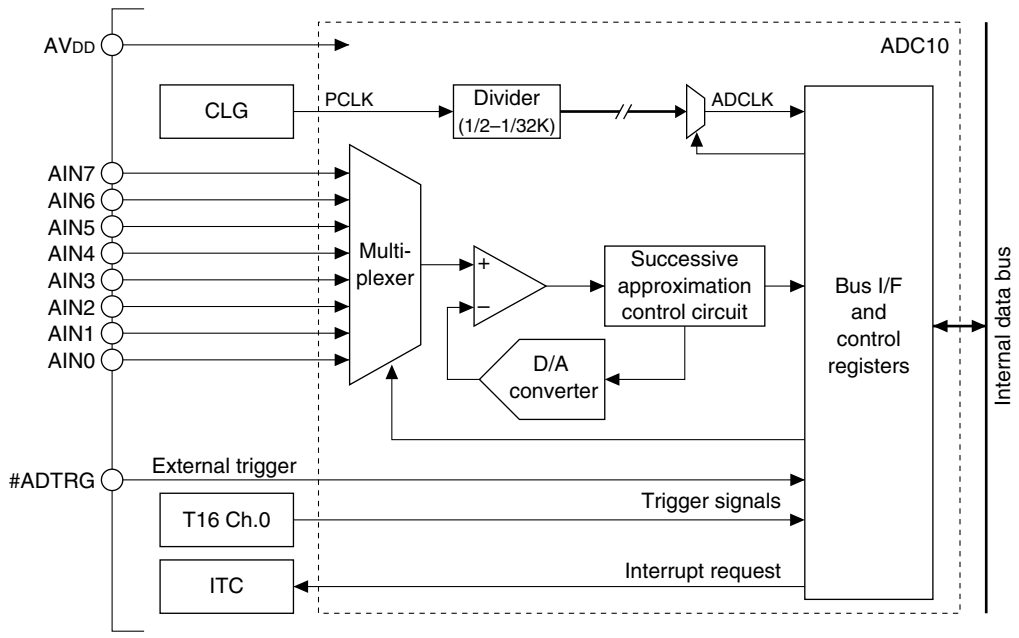


Figure 24.1.1 A/D Converter Configuration

24.2 ADC10 Input Pins

Table 24.2.1 lists the ADC10 input pins.

Table 24.2.1 List of ADC10 Input Pins

Pin name	I/O	Qty	Function
AIN[7:0]	I	8	Analog signal input pins AIN0 (Ch.0) to AIN7 (Ch.7) (see Note below) Input the analog signals to be A/D converted. The analog input voltage AV_{IN} must be within the range of $V_{SS} \leq AV_{IN} \leq AV_{DD}$ (= V_{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 ($AV_{DD} = V_{DD}$) Always supply the V_{DD} voltage even if the A/D converter is not used.

Note: The pins go to high impedance status when the port function is switched.

The A/D converter input pins (AIN[7:0], #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.

24.3 A/D Converter Settings

Make the following settings before starting A/D conversion.
(1) Set the analog input pins. See Section 24.2.
(2) Set the A/D conversion clock.
(3) Select the A/D conversion start and end channels.
(4) Select the A/D conversion mode.
(5) Select the A/D conversion trigger source.
(6) Set the sampling time.
(7) Select the conversion result storing mode.
(8) When using A/D converter interrupts, set interrupt conditions. See Section 24.5.

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.

24.3.1 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 24.3.1.1 using ADDF[3:0]/ADC_DIV 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 24.3.1.1 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)

24.3.2 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 24.3.2.1 Relationship between ADCS/ADCE and Input Channels

ADCS[2:0]/ADCE[2:0]	Channel selected
0x7	AIN7
0x6	AIN6
0x5	AIN5
0x4	AIN4
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

24.3.3 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.

24.3.4 Trigger Selection

Select a trigger source to start A/D conversion from among the three types listed in Table 24.3.4.1 using ADTS[1:0]/ADC10_TRG register.

Table 24.3.4.1 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.

24.3.5 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 24.3.5.1 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 24.3.5.1 shows an equivalent circuit of the analog input portion.

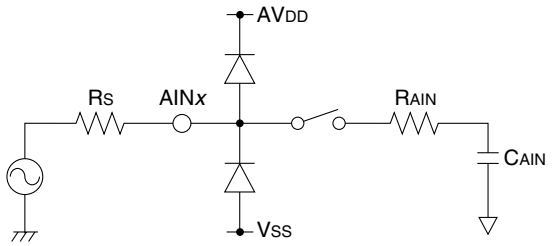


Figure 24.3.5.1 Equivalent Circuit of Analog Input Portion

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.

$$\text{Sampling rate [sps]} = \frac{f_{ADCLK}}{\text{Number of clock cycles set by ADST[2:0]} + 11}$$

24.3.6 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 24.3.6.1 Conversion Data Alignment

24.4 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.

24.4.1 Activating A/D Converter

After the settings described in Section 24.3 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.

24.4.2 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.

24.4.3 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.

24.4.4 Terminating A/D Conversion

One-time conversion mode (ADMS = 0)

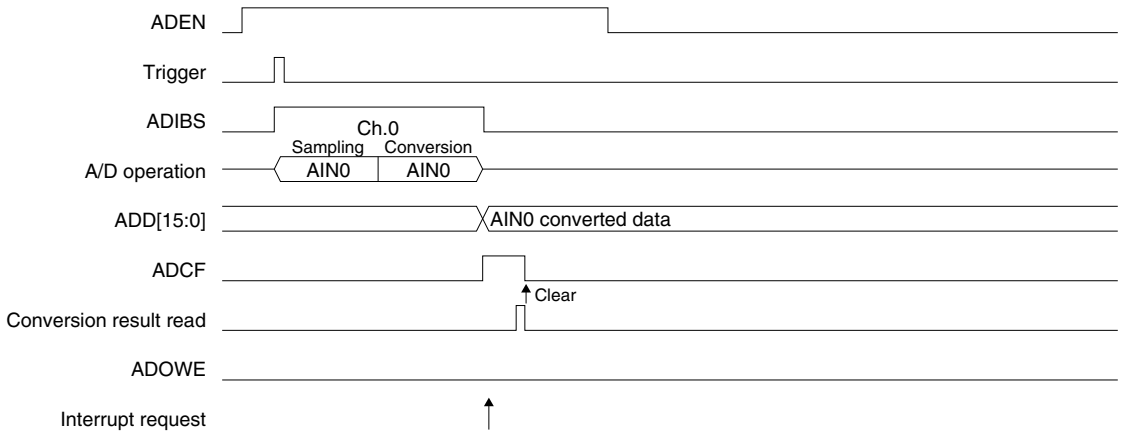
In one-time mode, the A/D converter performs A/D conversion within the channel range successively beginning with the conversion start channel specified by ADCS[2:0]/ADC10_TRG register and terminates once the conversion end channel specified by ADCE[2:0]/ADC10_TRG register has been completed. ADCTL/ADC10_CTL register is reset to 0 upon completion of the conversion sequence.

Continuous conversion mode (ADMS = 1)

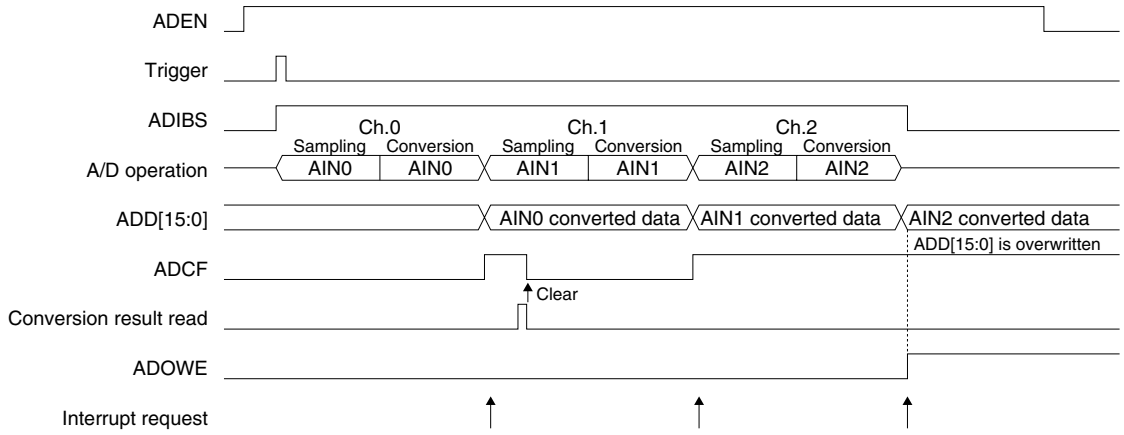
In continuous conversion mode, the A/D converter repeatedly performs A/D conversion from the conversion start channel to the conversion end channel. The hardware does not stop the conversion sequence. To stop A/D conversion, write 0 to ADCTL. Since the conversion sequence is forcibly terminated, the results of the conversion then underway cannot be obtained. ADEN/ADC10_CTL register must be set to 0 after a forced termination.

24.4.5 Timing Charts

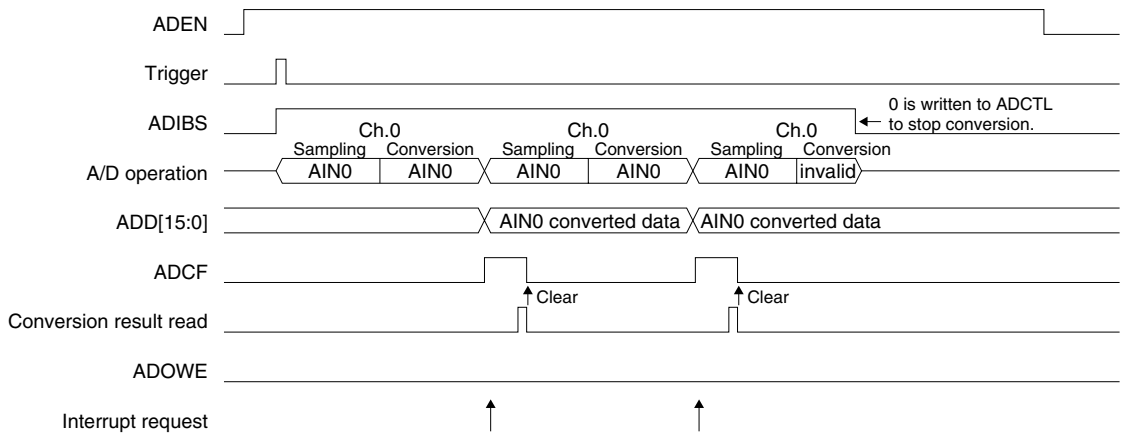
Figure 24.4.5.1 shows the operations of the A/D converter.



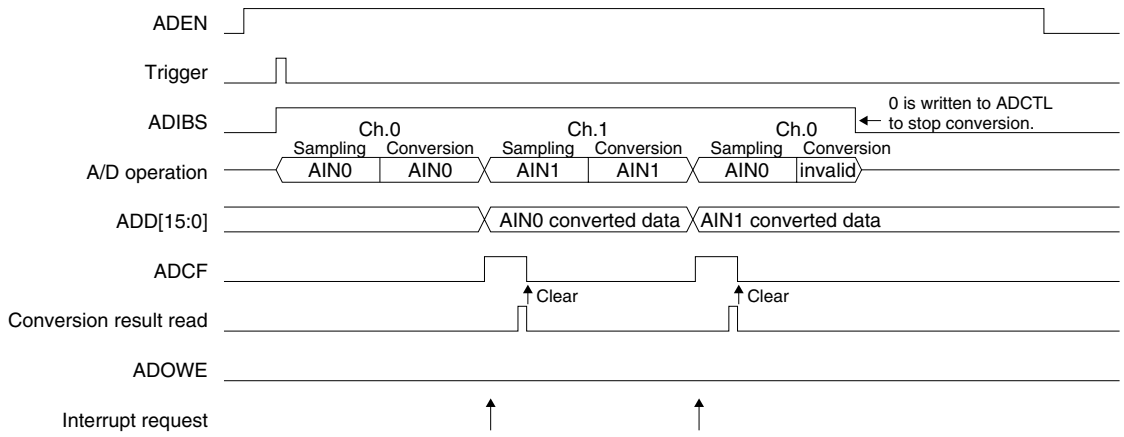
(1) Single channel (AIN0) one-time conversion mode (ADCS = 0, ADCE = 0, ADMS = 0)



(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 24.4.5.1 A/D Converter Operations

24.5 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.

24.6 Control Register Details

Table 24.6.1 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	ADC_DIV	A/D Clock Control Register	Controls A/D converter clock.

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 24.6.1 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 0x7	0x0	R/W	
		D10–8	**ADCS[2:0]**	Start channel select	0x0 to 0x7	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 7. (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 24.6.2 Relationship between ADCS/ADCE and Input Channels

ADCS[2:0]/ADCE[2:0]	Channel selected
0x7	AIN7
0x6	AIN6
0x5	AIN5
0x4	AIN4
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 7. (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 24.6.3 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 24.6.4 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 0x7	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 7) 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. However, setting ADEN to 0 does not stop A/D conversion being currently executed. To stop A/D conversion, write 0 to ADCTL.
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 (ADC_DIV)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
A/D Clock Control Register (ADC_DIV)	**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 24.6.5 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).

25 R/F Converter (RFC)

25.1 RFC Module Overview

The S1C17624/604/622/602/621 includes an R/F converter (RFC) module with two conversion channels. It is capable of being used as a CR oscillation type A/D converter. A thermo-hygrometer can easily be implemented by connecting only resistive or capacitive sensors (e.g., thermistor and humidity sensor) and a few passive elements (resistors and capacitors) to the R/F converter.
The following shows the features of the RFC module:

- Conversion method: Resistance to frequency conversion type
- Number of conversion channels: Max. 2 channels
- Oscillation mode: DC oscillation mode (for resistive sensors)
  AC oscillation mode (for resistive sensors)
  DC oscillation mode (for capacitive sensors)
- Counter length: 24 bits
- Five types of interrupts can be generated: Reference oscillation completion interrupt
  Sensor A oscillation completion interrupt
  Sensor B oscillation completion interrupt
  Measurement counter overflow error interrupt
  Time base counter overflow error interrupt

Figure 25.1.1 shows the RFC configuration.

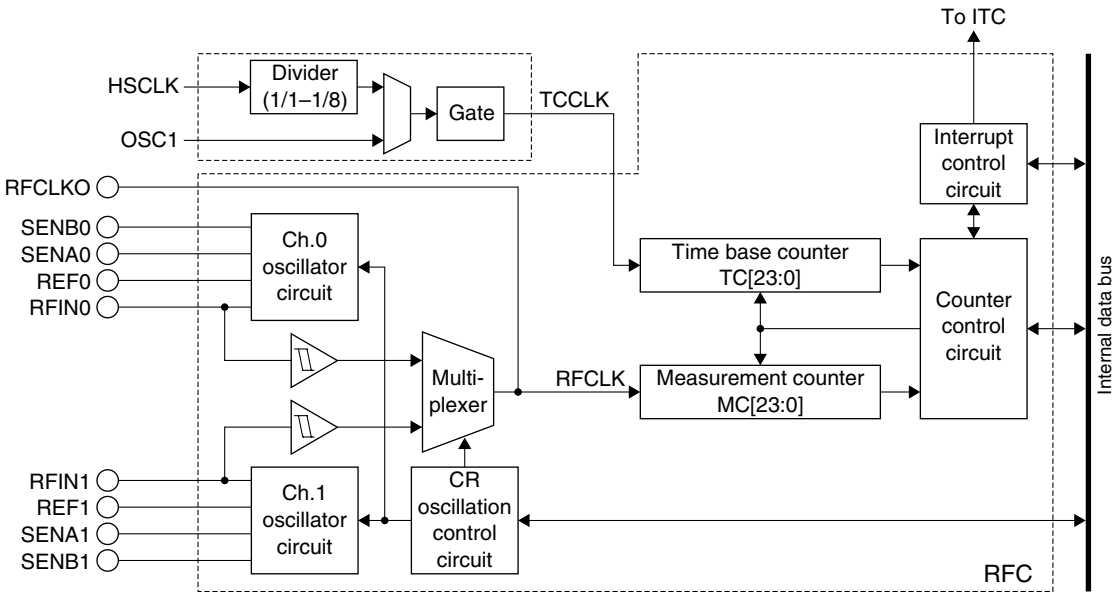


Figure 25.1.1 R/F Converter Configuration

The R/F converter converts the resistance or capacitance of the sensor connected into frequency (RFCLK) using the embedded CR oscillator circuit, and counts this frequency using the measurement counter for a set period of time to provide the digital value equivalent to the sensor value. The time base counter is also included for generating the measurement time by counting an internal clock (TCCLK). In addition to CR oscillation using a sensor (sensor oscillation), the R/F converter performs CR oscillation using a reference element with less variation in the characteristics due to external factors (reference oscillation). This removes error factors such as voltage fluctuations and unevenness in quality to realize precise measurements. The CR oscillator circuit supports AC driving and external clock input as well as general DC driving, allowing use of various sensors.

25.2 RFC Input/Output Pins

Table 25.2.1 lists the RFC input/output pins.

Table 25.2.1 List of R/F Converter Input/Output Pins

Pin name	I/O	Qty	Function
SENB0/SENB1	I/O	2	Sensor B oscillation control pin (see Note 1 below)
SENA0/SENA1	I/O	2	Sensor A oscillation control pin (see Note 1 below)
REF0/REF1	I/O	2	Reference oscillation control pin (see Note 1 below)
RFIN0/RFIN1	I/O	2	RFCLK input and oscillation control pin (see Note 2 below)
RFCLKO	O	1	RFCLK monitoring output pin Outputs RFCLK to monitor the oscillation frequency.

Notes: 1. The pins go to high impedance status when the port function is switched for the R/F converter.
2. The RFIN*x* pin goes to Vss level when the port function is switched for the R/F converter. A large current may flow through the RFIN*x* pin if the pin is externally biased.

The R/F converter input/output pins 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 R/F converter input/output pins.
For detailed information on pin function switching, see the "I/O Ports (P)" chapter.

25.3 Operation Clock

The RFC module includes a clock source selector, dividers, and a gate circuit for controlling the operation clock.

Note: The operation clock (TCCLK) must be enabled before setting the R/F converter. Otherwise, the R/F converter cannot operate normally.

Clock source selection

Use RFTCKSRC/OSC_RFC register to select the clock source from HSCLK (IOSC or OSC3) and OSC1. Setting RFTCKSRC to 1 (default) selects OSC1 and setting it to 0 selects HSCLK.

Clock division ratio selection

When the clock source is OSC1

No division ratio needs to be selected when OSC1 is selected for the clock source. The OSC1 clock (typ. 32.768 kHz) is directly used as TCCLK.

When the clock source is HSCLK

When HSCLK is selected for the clock source, use RFTCKDV[1:0]/OSC_RFC register to select the division ratio.

Table 25.3.1 HSCLK Division Ratio Selection

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

(Default: 0x0)

The time base counter uses the clock selected here for counting. Selecting a high-speed clock improves the conversion accuracy. However the clock must be selected so that the time base counter will not overflow in the reference oscillation phase.

Clock enable

The clock supply is enabled with RFTCKEN/OSC_RFC register. The RFTCKEN default setting is 0, which stops the clock. Setting RFTCKEN to 1 feeds the clock generated as above to the RFC circuit. If no RFC operation is required, stop the clock to reduce current consumption.

Note: Be sure to set RFTCKEN to 0 before selecting a clock division ratio.

25.4 Operating Modes

The R/F converter features three oscillation modes that use the RFC internal oscillator circuit and a mode for measuring an external input clock. Also it includes a CR oscillation clock (RFCLK) monitoring function and continuous oscillation function for measuring the oscillation clock frequency. Each channel can be set to a different mode.

25.4.1 Oscillation Mode

In measurements using the RFC internal oscillator circuit, operate the oscillator with the reference element and then the sensor for the same duration in time to count each oscillation frequency. The sensor value can be determined from the difference between the two count values by software. The R/F converter supports DC bias resistive or capacitive sensors and AC bias resistive sensors. The RFC internal oscillator circuit can operate in three oscillation modes corresponding to the sensor to be used that is specified by SMODE[1:0]/RFC_CTL register.

Table 25.4.1.1 Oscillation Mode Selection

SMODE[1:0]	Oscillation mode
0x3	Reserved
0x2	DC oscillation mode for measuring capacitive sensors
0x1	AC oscillation mode for measuring resistive sensors
0x0	DC oscillation mode for measuring resistive sensors

(Default: 0x0)

DC oscillation mode for measuring resistive sensors (SMODE[1:0] = 0x0, default)

This mode drives the oscillator with the reference resistor and resistive sensor by applying DC bias voltage. Select this mode when a DC bias resistive sensor is connected. This mode enables two resistive sensors to be connected to a channel. One reference resistor and one reference capacitor is also required.

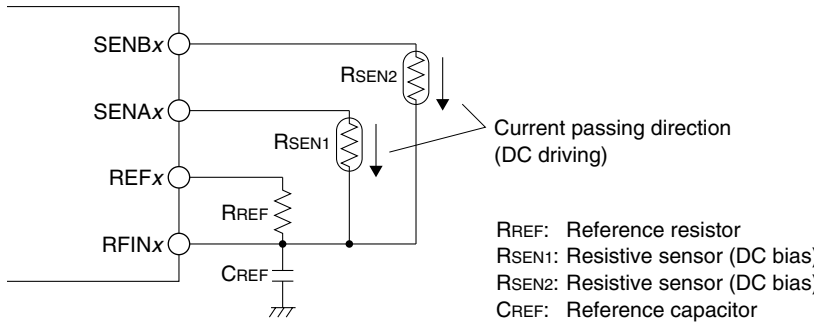


Figure 25.4.1.1 Connection Example in DC Oscillation Mode for Measuring Resistive Sensors

When one resistive sensor only is used, leave the unused pin open.

AC oscillation mode for measuring resistive sensors (SMODE[1:0] = 0x1)

This mode drives the oscillator with the reference resistor and resistive sensor by applying AC bias voltage. Select this mode when an AC bias resistive sensor is connected. This mode enables only one resistive sensor to be connected to a channel. One reference resistor and one reference capacitor is also required.

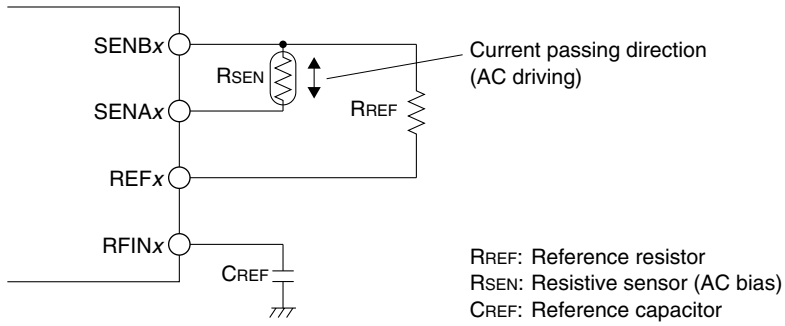


Figure 25.4.1.2 Connection Example in AC Oscillation Mode for Measuring Resistive Sensors

DC oscillation mode for measuring capacitive sensors (SMODE[1:0] = 0x2)

This mode drives the oscillator with the reference capacitor and capacitive sensor by applying DC bias voltage. Select this mode when a DC bias capacitive sensor is connected. This mode enables only one capacitive sensor to be connected to a channel. One reference resistor and one reference capacitor is also required.

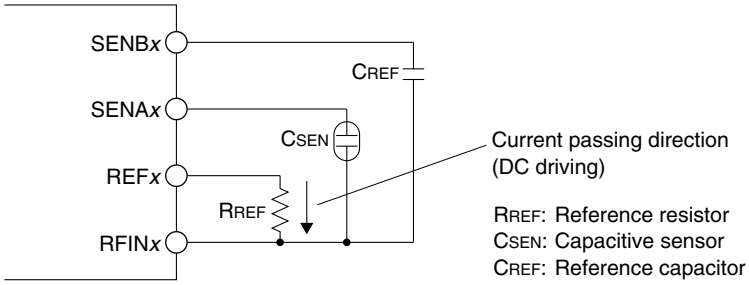


Figure 25.4.1.3 Connection Example in DC Oscillation Mode for Measuring Capacitive Sensors

25.4.2 External Clock Input Mode (Event Counter Mode)

This mode enables to input clocks/pulses from an external circuit such as an oscillator and count them same as those of internal oscillation clocks. It supports rectangular waves, triangular waves, and sign waves to be input. (For the threshold voltage of the Schmitt input buffer, see "Electrical Characteristics.")
Setting EVTEN/RFC_CTL register to 1 enables this function. The measurement control procedure is the same as that when the internal oscillator circuit is used.

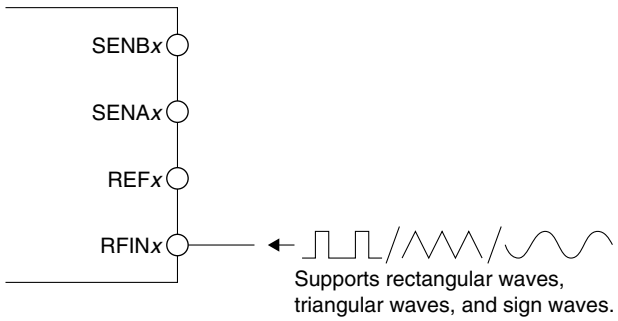


Figure 25.4.2.1 External Clock Input

The unused pins should be left open.

25.4.3 Functions for Measuring CR Oscillation Clock Frequency

CR Oscillation Clock (RFCLK) Monitoring Function

The CR oscillation clock (RFCLK) during converting can be output from the RFCLKO pin for monitoring. Use this output to measure the oscillation frequency.

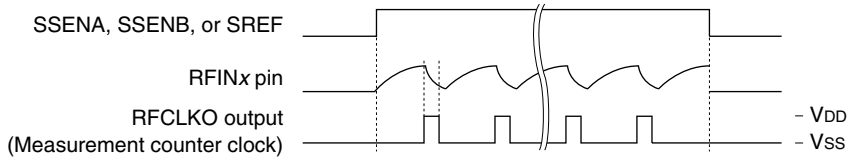


Figure 25.4.3.1 CR Oscillation Clock (RFCLK) Waveform

Continuous oscillation function

The CR oscillations by the sensor and reference element will automatically stop due to stop conditions. Setting 1 to CONEN/RFC_CTL register enables the continuous oscillation function and CR oscillation will continue until stopped by software. Using this function with the CR oscillation monitoring function helps easily measure the CR oscillation clock frequency.

25.5 RFC Counters

The R/F converter includes two kinds of counters: measurement counter (MC) counting the reference element and sensor oscillation clocks, and time base counter (TC) counting the TCCLK clock.

Measurement counter (MC)

The measurement counter is a 24-bit presettable up counter. Counting the reference oscillation clock and the sensor oscillation clock for the same duration in time using this counter minimizes errors caused by voltage, and unevenness of IC quality, as well as external parts and on-board parasitic elements. The counter values should be corrected via software after the reference and sensor oscillation are completed according to the sensor characteristics to determine the value being currently detected by the sensor.

Time base counter (TC)

The time base counter is a 24-bit presettable up/down counter. The time base counter counts up by TCCLK during reference oscillation to measure the reference oscillation time. During sensor oscillation, it counts down from the reference oscillation time and stops the sensor oscillation when it reaches 0x0. This means that the sensor oscillation time becomes equal to the reference oscillation time. The value counted during reference oscillation should be saved in the memory. It can be reused at the subsequent sensor oscillations omitting reference oscillations.

Counter initial value

To obtain the difference between the reference oscillation and sensor oscillation clock count values from the measurement counter simply, appropriate initial values must be set to the counters before starting reference oscillation and sensor oscillation. Connecting the reference element and sensor with the same resistance/capacitance will result <Initial value> = <Counter value at the end of sensor oscillation> (if no error introduced). Setting a small initial value to the measurement counter improves measurement accuracy. However, the measurement counter may overflow during sensor oscillation when the sensor value decreases below the reference element value (the measurement will be canceled). The initial value for the measurement counter should be determined taking the range of sensor value into consideration.

The time base counter should be cleared to 0x0 before starting reference oscillation.

25.6 Conversion Operations

The conversion operations by the R/F converter should be controlled in the following procedure regardless of the operating mode: initial settings, reference oscillation control, and sensor oscillation control. The R/F converter channels are controlled individually and both channels cannot operate simultaneously. This section describes these control procedure. Although the following explanations assume that the internal oscillator circuit is used, the control procedures are the same even in external clock input mode. When the R/F converter is used in external clock input mode, select which oscillation is performed either reference or sensor and determine the counter initial values depending on the purpose for using.

25.6.1 Initial Settings

Clock and pin configurations

(1) Select the R/F converter operating clock (TCCLK) and enable the clock supply. (See Section 25.3.)

(2) Configure the pins to be used for the R/F converter by switching from general-purpose input/output ports. See the “I/O Ports (P)” chapter.

R/F converter channel and mode settings

(1) Set RFCEN/RFC_CTL register to 1 to enable the R/F converter.

(2) Select the channel to perform conversion using CHSEL/RFC_CTL register. Setting CHSEL to 0 (default) selects Ch. 0 and setting 1 selects Ch.1.

(3) Set the oscillation mode using SMODE[1:0]/RFC_CTL register. (See Section 25.4.1.)

25.6.2 Reference Oscillation Control

First, perform oscillation with the reference resistor/capacitor and obtain the time base counter value to perform sensor oscillation for the same period of time.

(1) Set the initial value (0x0 - n) to MC[23:0]/RFC_MC(H/L) registers (measurement counter). (See Section 25.5.)

(2) Set 0x0 to TC[23:0]/RFC_TC(H/L) registers (time base counter).

(3) Reset the cause-of-interrupt flags OVTCIF and EREFIF in the RFC_IFLG register by writing 1.

(4) Set SREF/RFC_TRG register to 1 to start reference oscillation.
The CR oscillator circuit starts oscillating with the reference resistor/capacitor and outputs the clock to the measurement counter. The measurement counter starts counting up using the CR oscillation clock from the initial value that has been set. The time base counter starts counting up using TCCLK from 0x0.

Notes: • In the S1C17602/621, after a value is set to TC[23:0], wait a time interval at least three TCCLK cycles before starting oscillation.

• For restrictions and precautions on control bit settings, see the descriptions in the "Control Register Details" section. The control bits may not be set to the desired values depending on conditions.

(5) When the measurement counter or the time base counter overflows (0xffffff → 0x0), SREF is reset to 0 and the reference oscillation stops automatically.

(6-1) The measurement counter overflow sets EREFIF to 1 indicating that the reference oscillation has been terminated normally. An interrupt can be generated at this point. Read the time base counter value (TC[23:0] = X) and store it to the memory by the interrupt handler routine. When this interrupt is not used, perform the same processing after checking if EREFIF has been set.

(6-2) The time base counter overflow sets OVTCIF to 1 indicating that the reference oscillation has been terminated abnormally. An interrupt can be generated at this point. Handle this error in the interrupt handler routine. When this interrupt is not used, perform the same processing after checking if OVTCIF has been set.

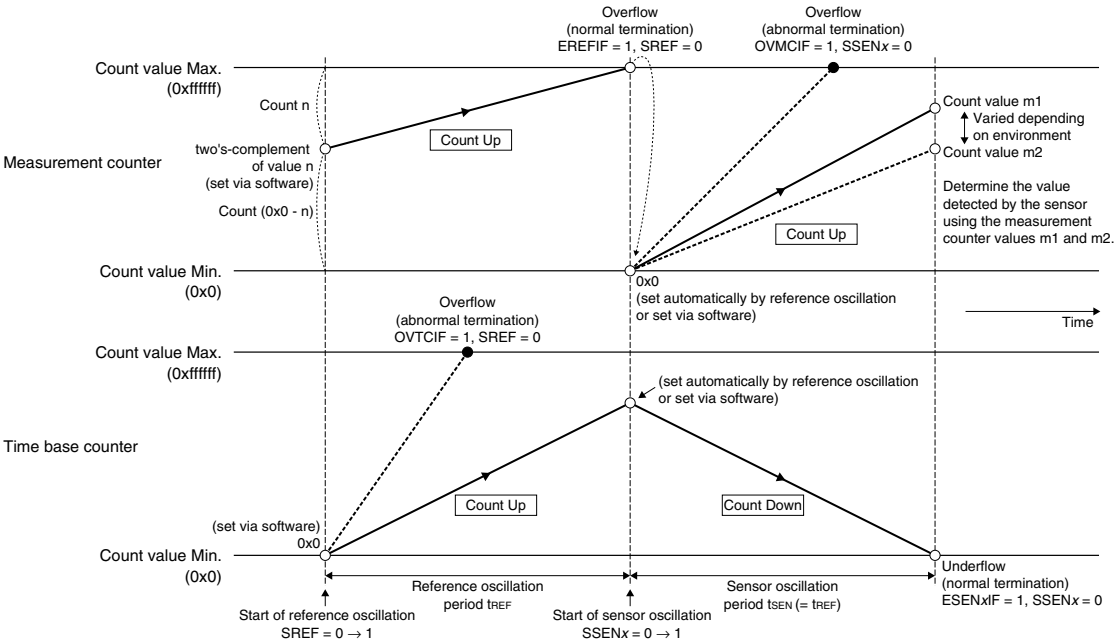


Figure 25.6.2.1 Counter Operations During Reference/Sensor Oscillation

25.6.3 Sensor Oscillation Control

Perform oscillation with the sensor for the period of time obtained by the time base counter in reference oscillation and count the oscillation clock by the measurement counter.

(1) Initialize MC[23:0] (measurement counter) by writing 0x0. This can be omitted after a reference oscillation has completed.

(2) Initialize TC[23:0] (time base counter) by writing the value (X) that has been counted in the time base counter during reference oscillation. This can be omitted after a reference oscillation has completed.

(3) Reset the cause-of-interrupt flags OVMCIF, ESENBIF, and ESENAIF in the RFC_IFLG register by writing 1.

(4) Set SSENA/RFC_TRG register (for sensor A) or SSENB/RFC_TRG register (for sensor B) to 1 to start sensor oscillation.
The CR oscillator circuit starts oscillating with the sensor and outputs the clock to the measurement counter. The measurement counter starts counting up using the CR oscillation clock from 0x0. The time base counter starts counting down using TCCLK from the initial value (X) that has been set.

Note: For restrictions and precautions on control bit settings, see the descriptions in the "Control Register Details" section. The control bits may not be set to the desired values depending on conditions.

(5) When the time base counter reaches 0x0 or the measurement counter overflows (0xffffff → 0x0), SSENA or SSENB is reset to 0 and the sensor oscillation stops automatically.

(6-1) The time base counter reached 0x0 sets ESENAIF (for sensor A) or ESENBIF (for sensor B) to 1 indicating that the sensor oscillation has been terminated normally. An interrupt can be generated at this point. Read the measurement counter value (MC[23:0] = m) and process the detection results by the interrupt handler routine. When this interrupt is not used, perform the same processing after checking if ESENAIF or ESENBIF has been set.

(6-2) The measurement counter overflow sets OVMCIF to 1 indicating that the sensor oscillation has been terminated abnormally. An interrupt can be generated at this point. Handle this error in the interrupt handler routine. When this interrupt is not used, perform the same processing after checking if OVMCIF has been set.

25.6.4 Forced Termination

To abort reference oscillation or sensor oscillation, write 0 to SREF (reference oscillation), SSENA (sensor A oscillation), or SSENB (sensor B oscillation) in the RFC_TRG register used to start the oscillation. The counters maintain the value at they stopped, note, however, that the conversion results cannot be guaranteed if the oscillation is resumed. When resuming oscillation, initialize the counters.

25.6.5 Conversion Error

Performing reference oscillation and sensor oscillation with the same resistor and capacitor results n ≈ m. The difference between n and m is a conversion error. The conversion error may be introduced caused by temperature, voltage, and unevenness of IC quality, as well as external parts and on-board parasitic elements. For sample errors, see "Electrical Characteristics."

25.7 RFC Interrupts

The RFC module includes a function for generating the following five different types of interrupts.
- Reference oscillation completion interrupt
- Sensor A oscillation completion interrupt
- Sensor B oscillation completion interrupt
- Measurement counter overflow error interrupt
- Time base counter overflow error interrupt

The RFC module outputs one interrupt signal shared by the five above interrupt causes to the interrupt controller (ITC). Inspect the interrupt flag to determine the interrupt cause occurred.

Reference oscillation completion interrupt

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

When the measurement counter overflows and a reference oscillation is completed normally, the R/F converter sets EREFIF/RFC_IFLG register to 1. If reference oscillation completion interrupts are enabled (EREFIE = 1), an interrupt request is sent simultaneously to the ITC.

Sensor A oscillation completion interrupt

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

When the time base counter reaches 0x0 and a sensor A oscillation is completed normally, the R/F converter sets ESENAIF/RFC_IFLG register to 1. If sensor A oscillation completion interrupts are enabled (ESENAIE = 1), an interrupt request is sent simultaneously to the ITC.

Sensor B oscillation completion interrupt

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

When the time base counter reaches 0x0 and a sensor B oscillation is completed normally, the R/F converter sets ESENBIF/RFC_IFLG register to 1. If sensor B oscillation completion interrupts are enabled (ESENBIE = 1), an interrupt request is sent simultaneously to the ITC.

Measurement counter overflow error interrupt

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

When the measurement counter overflows and a sensor oscillation is terminated abnormally, the R/F converter sets OVMCIF/RFC_IFLG register to 1. If measurement counter overflow error interrupts are enabled (OVMCIE = 1), an interrupt request is sent simultaneously to the ITC.

Time base counter overflow error interrupt

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

When the time base counter overflows and a reference oscillation is terminated abnormally, the R/F converter sets OVTCIF/RFC_IFLG register to 1. If time base counter overflow error interrupts are enabled (OVTCIE = 1), an interrupt request is sent simultaneously to the ITC.

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

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

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

25.8 Control Register Details

Table 25.8.1 List of RFC Registers

Address	Register name		Function
0x5067	OSC_RFC	RFC Clock Control Register	Selects the operating clock.
0x53a0	RFC_CTL	RFC Control Register	Controls R/F converter.
0x53a2	RFC_TRG	RFC Oscillation Trigger Register	Controls oscillations.
0x53a4	RFC_MCL	RFC Measurement Counter Low Register	Measurement counter data
0x53a6	RFC_MCH	RFC Measurement Counter High Register	
0x53a8	RFC_TCL	RFC Time Base Counter Low Register	Time base counter data
0x53aa	RFC_TCH	RFC Time Base Counter High Register	
0x53ac	RFC_IMSK	RFC Interrupt Mask Register	Enables/disables interrupts.
0x53ae	RFC_IFLG	RFC Interrupt Flag Register	Indicates/resets interrupt occurrence status.

The R/F converter 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.

RFC Clock Control Registers (OSC_RFC)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RFC Clock Control Register (OSC_RFC)	0x5067 (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.
		D3–2	RFTCKDV [1:0]	RFC clock division ratio select	RFTCKDV[1:0] Division ratio 0x3 1/8 0x2 1/4 0x1 1/2 0x0 1/1	0x0	R/W	When the clock source is HSCLK
		D1	RFTCKSRC	RFC clock source select	1 OSC1 0 HSCLK	1	R/W	
		D0	RFTCKEN	RFC clock enable	1 Enable 0 Disable	0	R/W	

D[7:4] Reserved

D[3:2] RFTCKDV[1:0]: RFC Clock Division Ratio Select Bits

Selects the division ratio for generating the TCCLK clock when HSCLK (IOSC or OSC3) is used as the clock source.

Table 25.8.2 HSCLK Division Ratio Selection

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

(Default: 0x0)

D1 RFTCKSRC: RFC Clock Source Select Bit

Selects the count clock source.

1 (R/W): OSC1 (default)

0 (R/W): HSCLK (IOSC or OSC3)

D0 RFTCKEN: RFC Clock Enable Bit

Enables or disables the TCCLK clock supply.

1 (R/W): Enabled (on)

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

The RFTCKEN default setting is 0, which disables the clock supply. Setting RFTCKEN to 1 sends the clock selected to the R/F converter.

RFC Control Register (RFC_CTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RFC Control Register (RFC_CTL)	0x53a0 (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7	CONEN	Continuous oscillation enable	1 Enable 0 Disable	0	R/W	
		D6	EVTEN	Event counter mode enable	1 Enable 0 Disable	0	R/W	
		D5–4	SMODE[1:0]	Sensor oscillation mode select	SMODE[1:0] Sensor 0x3 reserved 0x2 DC capacitive 0x1 AC resistive 0x0 DC resistive	0x0	R/W	
		D3–2	–	reserved	–	–	–	0 when being read.
		D1	CHSEL	Conversion channel select	1 Ch.1 0 Ch.0	0	R/W	
		D0	RFCEN	RFC enable	1 Enable 0 Disable	0	R/W	

D[15:8] Reserved

D7 CONEN: Continuous Oscillation Enable Bit

Enables continuous oscillation by disabling the automatic CR oscillation stop function.

1 (R/W): Continuous oscillation enabled

0 (R/W): Continuous oscillation disabled (default)

Setting 1 to CONEN disables the reference oscillation/sensor oscillation stop conditions so that the CR oscillator will continue oscillating. Set SREF (reference oscillation), SSENA (sensor A oscillation), or SSENB (sensor B oscillation) in the RFC_TRG register to 1 to start oscillation even in this mode, and set to 0 to stop oscillation.
Using this function with the CR oscillation monitoring function helps easily measure the CR oscillation clock frequency.

D6 EVTEN: Event Counter Mode Enable Bit

Enables external clock input mode (event counter mode).
1 (R/W): External clock input mode
0 (R/W): Normal mode (default)

Setting EVTEN to 1 enables the external clock input to the RFIN*x* pin. SREF (reference oscillation), SSENA (sensor A oscillation), or SSENB (sensor B oscillation) should be used to control starting oscillation (starting conversion) to perform converting operation even in this mode.

Note: Do not input an external clock before setting EVTEN to 1. The RFIN*x* pin is pulled down to Vss when the pin function is switched for the R/F converter.

D[5:4] SMODE[1:0]: Sensor Oscillation Mode Select Bits

Selects an oscillation mode.

Table 25.8.3 Oscillation Mode Selection

SMODE[1:0]	Oscillation mode
0x3	Reserved
0x2	DC oscillation mode for measuring capacitive sensors
0x1	AC oscillation mode for measuring resistive sensors
0x0	DC oscillation mode for measuring resistive sensors

(Default: 0x0)

For more information on the oscillation mode, see Section 25.4.1.

D[3:2] Reserved

D1 CHSEL: Conversion Channel Select Bit

Selects the channel to perform conversion.
1 (R/W): Ch.1
0 (R/W): Ch.0 (default)

The D[7:4] settings in this register and oscillation control using the RFC_TRG register are effective only for the channel specified by CHSEL.

D0 RFCEN: RFC Enable Bit

Enables or disables the R/F converter.
1 (R/W): Enabled
0 (R/W): Disabled (default)

Setting RFCEN to 1 enables the R/F converter to start converting operations. When RFCEN is 0, manipulations of the RFC_TRG register for oscillation control are ineffective.

RFC Oscillation Trigger Register (RFC_TRG)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RFC Oscillation Trigger Register (RFC_TRG)	0x53a2 (16 bits)	D15–3	–	reserved	–	–	–	0 when being read.
		D2	SSENB	Sensor B oscillation control/status	1 Start/Run 0 Stop	0	R/W	
		D1	SSENA	Sensor A oscillation control/status	1 Start/Run 0 Stop	0	R/W	
		D0	SREF	Reference oscillation control/status	1 Start/Run 0 Stop	0	R/W	

D[15:3] Reserved

D2 SSENB: Sensor B Oscillation Control/Status Bit

Controls CR oscillation for sensor B. This bit also indicates the CR oscillation status.

1 (W): Start oscillation
0 (W): Stop oscillation
1 (R): Being oscillated
0 (R): Stopped (default)

Sensor B cannot be used in AC oscillation mode for resistive sensors and DC oscillation mode for capacitive sensors.

D1 SSENA: Sensor A Oscillation Control/Status Bit

Controls CR oscillation for sensor A. This bit also indicates the CR oscillation status.

1 (W): Start oscillation
0 (W): Stop oscillation
1 (R): Being oscillated
0 (R): Stopped (default)

D0 SREF: Reference Oscillation Control/Status Bit

Controls CR oscillation for the reference element. This bit also indicates the CR oscillation status.

1 (W): Start oscillation
0 (W): Stop oscillation
1 (R): Being oscillated
0 (R): Stopped (default)

Notes:
- SREF, SSENA, and SSENB are all ineffective when RFCEN/RFC_CTL register is 0 (converting operation disabled).
- Writing 1 to SSENB does not start oscillation when SMODE[1:0]/RFC_CTL register is 0x1 (AC oscillation mode for resistive sensors) or 0x2 (DC oscillation mode for capacitive sensors).
- When writing 1 to SREF, SSENA, or SSENB to start oscillation, be sure to avoid that more than one bit are set to 1.
- Be sure to reset the interrupt flags in the RFC_IFLG register (EREFIF, ESENAIF, ESENBIF, OVMCIF, and OVTCIF) before starting oscillation using SREF, SSENA, and SSENB.

RFC Measurement Counter Low and High Registers (RFC_MCL, RFC_MCH)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RFC Measurement Counter Low Register (RFC_MCL)	**0x53a4** (16 bits)	D15–0	**MC[15:0]**	Measurement counter low-order 16-bit data	0x0–0xffff	0x0	R/W	
RFC Measurement Counter High Register (RFC_MCH)	**0x53a6** (16 bits)	D15–8	**–**	reserved	–	–	–	0 when being read.
		D7–0	**MC[23:16]**	Measurement counter high-order 8-bit data	0x0–0xff	0x0	R/W	

D[7:0]/RFC_MCH, D[15:0]/RFC_MCL
MC[23:0]: Measurement Counter Bits

Measurement counter data can be read and written to. (Default: 0x0)

Note: The measurement counter must be set from the low-order value (MC[15:0]/RFC_MCL register) first. The counter may not be set to the correct value if the high-order value (MC[23:16]/RFC_MCH register) is written first.

RFC Time Base Counter Low and High Registers (RFC_TCL, RFC_TCH)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RFC Time Base Counter Low Register (RFC_TCL)	0x53a8 (16 bits)	D15–0	TC[15:0]	Time base counter low-order 16-bit data	0x0–0xffff	0x0	R/W	
RFC Time Base Counter High Register (RFC_TCH)	0x53aa (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	TC[23:16]	Time base counter high-order 8-bit data	0x0–0xff	0x0	R/W	

D[7:0]/RFC_TCH, D[15:0]/RFC_TCL

TC[23:0]: Time Base Counter Bits

Time base counter data can be read and written to. (Default: 0x0)

Note: The time base counter must be set from the low-order value (TC[15:0]/RFC_TCL register) first. The counter may not be set to the correct value if the high-order value (TC[23:16]/RFC_TCH register) is written first.

RFC Interrupt Mask Register (RFC_IMSK)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RFC Interrupt Mask Register (RFC_IMSK)	0x53ac (16 bits)	D15–5	–	reserved	–	–	–	0 when being read.
		D4	OVTCIE	TC overflow error interrupt enable	1 Enable 0 Disable	0	R/W	
		D3	OVMCIE	MC overflow error interrupt enable	1 Enable 0 Disable	0	R/W	
		D2	ESENBIE	Sensor B oscillation completion interrupt enable	1 Enable 0 Disable	0	R/W	
		D1	ESENAIE	Sensor A oscillation completion interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	EREFIE	Reference oscillation completion interrupt enable	1 Enable 0 Disable	0	R/W	

D[15:5] Reserved

D4 OVTCIE: TC Overflow Error Interrupt Enable Bit

Enables or disables time base counter overflow error interrupts.

1 (R/W): Interrupt enabled

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

D3 OVMCIE: MC Overflow Error Interrupt Enable Bit

Enables or disables measurement counter overflow error interrupts.

1 (R/W): Interrupt enabled

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

D2 ESENBIE: Sensor B Oscillation Completion Interrupt Enable Bit

Enables or disables sensor B oscillation completion interrupts.

1 (R/W): Interrupt enabled

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

D1 ESENAIE: Sensor A Oscillation Completion Interrupt Enable Bit

Enables or disables sensor A oscillation completion interrupts.

1 (R/W): Interrupt enabled

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

D0 EREFIE: Reference Oscillation Completion Interrupt Enable Bit

Enables or disables reference oscillation completion interrupts.

1 (R/W): Interrupt enabled

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

RFC Interrupt Flag Register (RFC_IFLG)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
RFC Interrupt Flag Register (RFC_IFLG)	0x53ae (16 bits)	D15–5	–	reserved	–				–	–	0 when being read.
		D4	OVTCIF	TC overflow error interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D3	OVMCIF	MC overflow error interrupt flag					0	R/W	
		D2	ESENBIF	Sensor B oscillation completion interrupt flag					0	R/W	
		D1	ESENAIF	Sensor A oscillation completion interrupt flag					0	R/W	
		D0	EREFIF	Reference oscillation completion interrupt flag					0	R/W	

D[15:5] Reserved

D4 OVTCIF: TC Overflow Error Interrupt Flag Bit

Indicates the time base counter overflow error 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

OVTCIF is set to 1 when a reference oscillation is terminated abnormally due to time base counter overflow. OVTCIF is reset to 0 by writing 1.

D3 OVMCIF: MC Overflow Error Interrupt Flag Bit

Indicates the measurement counter overflow error 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

OVMCIF is set to 1 when a sensor oscillation is terminated abnormally due to measurement counter overflow. OVMCIF is reset to 0 by writing 1.

D2 ESENBIF: Sensor B Oscillation Completion Interrupt Flag Bit

Indicates the sensor B oscillation completion 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

ESENBIF is set to 1 when the time base counter reaches 0x0 and a sensor B oscillation is completed normally. ESENBIF is reset to 0 by writing 1.

D1 ESENAIF: Sensor A Oscillation Completion Interrupt Flag Bit

Indicates the sensor A oscillation completion 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

ESENAIF is set to 1 when the time base counter reaches 0x0 and a sensor A oscillation is completed normally. ESENAIF is reset to 0 by writing 1.

D0 EREFIF: Reference Oscillation Completion Interrupt Flag Bit

Indicates the reference oscillation completion 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

EREFIF is set to 1 when the measurement counter overflows and a reference oscillation is completed normally. EREFIF is reset to 0 by writing 1.

26 Supply Voltage Detector (SVD)

26.1 SVD Module Overview

The S1C17624/604/622/602/621 includes an SVD (supply voltage detector) circuit to monitor the power voltage supplied to the VDD pin. It generates an interrupt when the power supply voltage drops below the detection level set with software. The detection results can also be read via software.
The following shows the features of the SVD module:

- Power supply voltage to be detected: VDD
- Detection voltage levels: 15 levels (1.8 V to 3.2 V)
- Interrupt supported: 1 system (power supply voltage drop detection interrupt)

Figure 26.1.1 shows the SVD configuration.

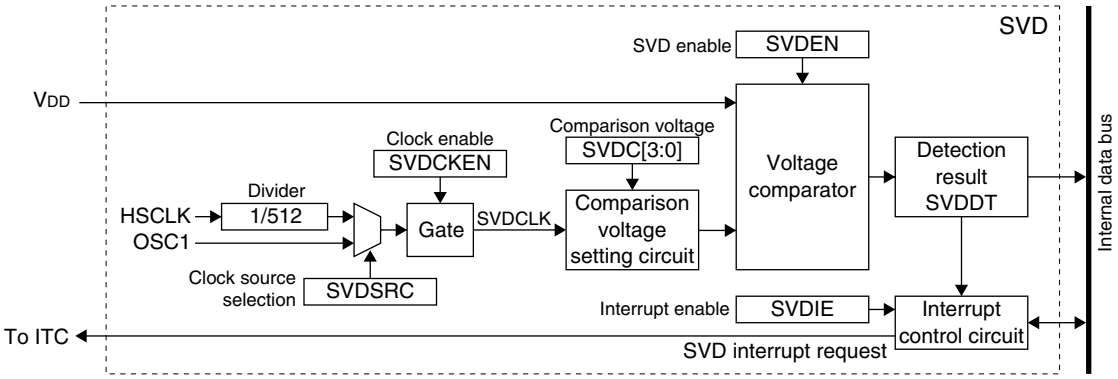


Figure 26.1.1 SVD Configuration

26.2 Operating Clock

The SVD module includes a clock source selector, dividers, and a gate circuit for controlling the operation clock.

Clock selection

Use SVDSRC[1:0]/OSC_SVD register to select the clock source from HSCLK (IOSC or OSC3) and OSC1. Setting SVDSRC to 1 (default) selects OSC1 and setting it to 0 selects HSCLK.

When OSC1 is selected as the clock source, the OSC1 clock (typ. 32.768 kHz) is directly used as SVDCLK.
When HSCLK is selected as the clock source, SVDCLK is generated by dividing HSCLK (IOSC or OSC3 clock) by 512.

Clock enable

The clock supply is enabled with SVDCKEN/OSC_SVD register. The SVDCKEN default setting is 0, which stops the clock. Setting SVDCKEN to 1 feeds the clock selected to the SVD circuit. If no SVD operation is required, stop the clock to reduce current consumption.
If SVDCLK is not supplied, the SVD circuit cannot detect voltage levels. However, the SVD control registers can be accessed even if SVDCLK is stopped.

Note: Be sure to set SVDCKEN to 0 before selecting the clock source.

26.3 Comparison Voltage Setting

The SVD circuit compares the power supply voltage (V_{DD}) against the comparison voltage set by software and outputs results indicating whether the power supply voltage exceeds this comparison voltage. The comparison voltage can be selected from among the 15 levels listed in Table 26.3.1 with the SVDC[3:0]/SVD_CMP register.

Table 26.3.1 Comparison Voltage Settings

SVDC[3:0]	Comparison voltage
0xf	3.2 V
0xe	3.1 V
0xd	3.0 V
0xc	2.9 V
0xb	2.8 V
0xa	2.7 V
0x9	2.6 V
0x8	2.5 V
0x7	2.4 V
0x6	2.3 V
0x5	2.2 V
0x4	2.1 V
0x3	2.0 V
0x2	1.9 V
0x1	1.8 V
0x0	Reserved

(Default: 0x0)

26.4 SVD Control

Power supply voltage detection using the SVD circuit is initiated by writing 1 to SVDEN/SVD_EN register and is stopped by writing 0.
The results can be read out from the SVDDT/SVD_RSLT register.
The detection results and SVDDT readings are as follows.

- When power supply voltage (V_{DD}) ≥ comparison voltage: SVDDT = 0
- When power supply voltage (V_{DD}) < comparison voltage: SVDDT = 1

When SVD interrupts are enabled and SVDEN is set to 1, an interrupt occurs as soon as the power supply voltage drops below the comparison voltage, and the detection result becomes 1. This interrupt can be used to indicate battery depletion and to initiate the heavy load protection function. See the following section for more information on interrupt control.
Note that if a temporary voltage drop causes an interrupt, the interrupt will not be cleared even when the voltage subsequently returns to a value exceeding the comparison voltage. The SVDDT should be checked in the interrupt handler routine.

Notes:
- An SVD circuit-enable response time is required to obtain stable detection results after SVDEN is altered from 0 to 1. Also when SVDC[3:0] is altered, an SVD circuit response time is required to obtain stable detection results. For these response times, see "Electrical Characteristics."
- Operating the SVD circuit increases current consumption. If power supply voltage detection is not required, stop SVD operations by setting SVDEN to 0.

26.5 SVD Interrupt

The SVD module includes a function for generating interrupts when power supply voltage drops are detected.

Power supply voltage drop detection interrupt

This cause of interrupt is generated when the power supply voltage (V_{DD}) detected value drops below the comparison voltage while SVD is operating (SVDEN = 1). It sets the interrupt flag SVDIF/SVD_IFLG register in the SVD module to 1. Once set, SVDIF is not reset even if the power supply voltage subsequently returns to a value exceeding the comparison voltage. SVDIF is reset to 0 by writing 1.

To use this interrupt, set SVDIE/SVD_IMSK register to 1. When SVDIE is set to 0 (default), interrupt requests for this cause will not be sent to the interrupt controller (ITC).
If SVDIF is set to 1 while SVDIE is set to 1 (interrupt enabled), the SVD module outputs an interrupt request to the ITC. An interrupt is generated if the ITC and S1C17 Core interrupt conditions are satisfied.

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

Notes: • To prevent interrupt recurrences, the SVD module interrupt flag SVDIF must be reset in the interrupt handler routine after an SVD interrupt has occurred.

• To prevent unwanted interrupts, SVDIF should be reset before enabling SVD interrupts with SVDIE.

26.6 Control Register Details

Table 26.6.1 List of SVD Registers

Address	Register name		Function
0x5066	OSC_SVD	SVD Clock Control Register	Selects the operating clock.
0x5100	SVD_EN	SVD Enable Register	Enables/disables the SVD operation.
0x5101	SVD_CMP	SVD Comparison Voltage Register	Sets the comparison voltage.
0x5102	SVD_RSLT	SVD Detection Result Register	Voltage detection results
0x5103	SVD_IMSK	SVD Interrupt Mask Register	Enables/disables interrupts.
0x5104	SVD_IFLG	SVD Interrupt Flag Register	Indicates/resets interrupt occurrence status.

The SVD 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.

SVD Clock Control Register (OSC_SVD)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
SVD Clock Control Register (OSC_SVD)	0x5066 (8 bits)	D7–2	–	reserved	–				–	–	0 when being read.
		D1	SVDSRC	SVD clock source select	1	OSC1	0	HSCLK/512	1	R/W	
		D0	SVDCKEN	SVD clock enable	1	Enable	0	Disable	0	R/W	

D[7:2] Reserved

D1 SVDSRC: SVD Clock Source Select Bit

Selects the clock source for the SVD circuit.
1 (R/W): OSC1 (default)
0 (R/W): HSCLK/512

When OSC1 is selected as the clock source, the OSC1 clock (typ. 32.768 kHz) is directly used as SVDCLK. When HSCLK is selected as the clock source, SVDCLK is generated by dividing HSCLK (IOSC or OSC3 clock) by 512.

D0 SVDCKEN: SVD Clock Enable Bit

Enables or disables the operation clock supply to the SVD circuit.
1 (R/W): Enabled (on)
0 (R/W): Disabled (off) (default)

The SVDCKEN default setting is 0, which disables the clock supply. Setting SVDCKEN to 1 feeds the clock selected to the SVD circuit.

SVD Enable Register (SVD_EN)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
SVD Enable Register (SVD_EN)	0x5100 (8 bits)	D7–1	–	reserved	–				–	–	0 when being read.
		D0	SVDEN	SVD enable	1	Enable	0	Disable	0	R/W	

D[7:1] Reserved

D0 SVDEN: SVD Enable Bit

Enables or disables SVD operations.
1 (R/W): Enabled
0 (R/W): Disabled(default)

Setting SVDEN to 1 initiates power supply voltage detection; setting to 0 stops detection.

Notes: • An SVD circuit-enable response time is required to obtain stable detection results after SVDEN is altered from 0 to 1. Also when SVDC[3:0] is altered, an SVD circuit response time is required to obtain stable detection results. For these response times, see "Electrical Characteristics."

• Operating the SVD circuit increases current consumption. If power supply voltage detection is not required, stop SVD operations by setting SVDEN to 0.

SVD Comparison Voltage Register (SVD_CMP)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
SVD Comparison Voltage Register (SVD_CMP)	**0x5101** (8 bits)	D7–4	**–**	reserved	–		–	–	0 when being read.
		D3–0	**SVDC[3:0]**	SVD comparison voltage select	SVDC[3:0]	Voltage	0x0	R/W	
					0xf	3.2 V			
					0xe	3.1 V			
					0xd	3.0 V			
					0xc	2.9 V			
					0xb	2.8 V			
					0xa	2.7 V			
					0x9	2.6 V			
					0x8	2.5 V			
					0x7	2.4 V			
					0x6	2.3 V			
					0x5	2.2 V			
					0x4	2.1 V			
					0x3	2.0 V			
					0x2	1.9 V			
					0x1	1.8 V			
					0x0	reserved			

D[7:4] Reserved

D[3:0] SVDC[3:0]: SVD Comparison Voltage Select Bits

Selects one of 15 comparison voltages for detecting voltage drops.

Table 26.6.2 Comparison Voltage Settings

SVDC[3:0]	Comparison voltage
0xf	3.2 V
0xe	3.1 V
0xd	3.0 V
0xc	2.9 V
0xb	2.8 V
0xa	2.7 V
0x9	2.6 V
0x8	2.5 V
0x7	2.4 V
0x6	2.3 V
0x5	2.2 V
0x4	2.1 V
0x3	2.0 V
0x2	1.9 V
0x1	1.8 V
0x0	Reserved

(Default: 0x0)

The SVD circuit compares the power supply voltage (V_{DD}) against the comparison voltage set by SVDC[3:0], and outputs results indicating whether the power supply voltage exceeds this comparison voltage.

SVD Detection Result Register (SVD_RSLT)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
SVD Detection Result Register (SVD_RSLT)	0x5102 (8 bits)	D7–1	–	reserved	–				–	–	0 when being read.
		D0	SVDDT	SVD detection result	1	Low	0	Normal	×	R	

D[7:1] Reserved

D0 SVDDT: SVD Detection Result Bit

Indicates the power supply voltage detection results.

1 (R): Power supply voltage (V_{DD}) < comparison voltage

0 (R): Power supply voltage (V_{DD}) ≥ comparison voltage

The SVD circuit compares the power supply voltage (V_{DD}) against the voltage set in SVDC[3:0]/SVD_CMP register while SVDEN/SVD_EN register = 1. The current power supply voltage status can be checked by reading SVDDT.

SVD Interrupt Mask Register (SVD_IMSK)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
SVD Interrupt Mask Register (SVD_IMSK)	0x5103 (8 bits)	D7–1	–	reserved	–				–	–	0 when being read.
		D0	SVDIE	SVD interrupt enable	1	Enable	0	Disable	0	R/W	

D[7:1] Reserved

D0 SVDIE: SVD Interrupt Enable Bit

Enables or disables interrupts when a power supply voltage drop is detected.

1 (R/W): Interrupt enabled

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

Setting SVDIE to 1 enables SVD interrupt requests to the ITC; setting to 0 disables interrupts.

SVD Interrupt Flag Register (SVD_IFLG)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
SVD Interrupt Flag Register (SVD_IFLG)	0x5104 (8 bits)	D7–1	–	reserved	–				–	–	0 when being read.
		D0	SVDIF	SVD interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.

D[7:1] Reserved

D0 SVDIF: SVD Interrupt Flag Bit

Indicates the power supply voltage drop detection 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

SVDIF is set to 1 when a power supply voltage drop is detected. SVDIF is reset to 0 by writing 1.

27 On-chip Debugger (DBG)

27.1 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" chapter.
The start address for this debugging work area can be read from the DBRAM register (0xffff90).

Debugging tools

Debugging involves connecting ICDmini (S5U1C17001H) to the S1C17624/604/622/602/621 debug pins and inputting the debug instruction from the debugger on the personal computer.
The following tools are required:
- S1C17 Family In-Circuit Debugger ICDmini (S5U1C17001H)
- S1C17 Family C compiler package (e.g., S5U1C17001C)

Debug pins

The following debug pins are used to connect ICDmini (S5U1C17001H).

Table 27.1.1 List of Debug Pins

Pin name	I/O	Qty	Function
DCLK	O	1	On-chip debugger clock output pin Outputs a clock to the ICDmini (S5U1C17001H).
DSIO	I/O	1	On-chip debugger data input/output pin Used to input/output debugging data and input the break signal.
DST2	O	1	On-chip debugger status signal output pin Outputs the processor status during debugging.

The on-chip debugger input/output pins (DCLK, DST2, DSIO) are shared with I/O ports and are initially set as the debug pins. If the debugging function is not used, these pins can be switched using the port function select bits to enable use as general-purpose I/O port pins.
For detailed information on pin function switching, see the "I/O Ports (P)" chapter.

27.2 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 PRUND/PSC_CTL register to 1. Setting PRUND 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 O1DBG/MISC_OSC1 register to 1. Setting O1DBG to 0 (default) will stop these peripheral circuits in debug mode.
Some peripheral circuits, such as SPI, I2CS, and T16A2, that run with an external input clock will not stop operating even if the S1C17 Core enters debug mode.
The LCD driver continues the operating status at occurrence of the debug interrupt.

27.3 Additional Debugging Function

The S1C17624/604/622/602/621 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 S1C17624/604/622/602/621 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 S1C17624/604/622/602/621 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.

27.4 Control Register Details

Table 27.4.1 List of Debug Registers

Address	Register name		Function
0x4020	PSC_CTL	Prescaler Control Register	Controls prescalers.
0x5322	MISC_OSC1	OSC1 Peripheral Control Register	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.

Prescaler Control Register (PSC_CTL)

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

D[7:2] Reserved

D1 PRUND: 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 PRUND 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 PRUND to 1 to maintain running status for these peripheral circuits in debug mode.

D0 PRUN: Prescaler Run/Stop Control Bit

Runs/stops the prescaler.
1 (R/W): Run
0 (R/W): Stop (default)

Write 1 to PRUN to run the prescaler for the peripheral circuit shown below and write 0 to stop the prescaler.
- 16-bit timer (T16)
- Fine mode 8-bit timer (T8F)
- 16-bit PWM timer (T16E)
- IR remote controller (REMC)
- I/O port (P)
- UART
- SPI
- I²C master (I2CM)
- A/D converter (ADC10)

OSC1 Peripheral Control Register (MISC_OSC1)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
OSC1 Peripheral Control Register (MISC_OSC1)	0x5322 (16 bits)	D15–1	–	reserved	–	–	–	0 when being read.
		D0	O1DBG	Run/stop select in debug mode (except PCLK peripheral circuits)	1 Run 0 Stop	0	R/W	

D[15:1] Reserved

D0 O1DBG: 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 O1DBG 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 O1DBG to 1 to maintain running status for these peripheral circuits in debug mode.
Some peripheral circuits, such as SPI, I2CS, and T16A2, that run with an external input clock will not stop operating even if the S1C17 Core enters debug mode.
The LCD driver continues the operating status at occurrence of the debug interrupt.

IRAM Size Select Register (MISC_IRAMSZ)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
IRAM Size Select Register (MISC_IRAMSZ) (S1C17624/604)	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	–	reserved	–	–	–	0x1 when being read.
		D3	–	reserved	–	–	–	0 when being read.
		D2–0	IRAMSZ[2:0]	IRAM size select	IRAMSZ[2:0] Size 0x3 2KB 0x2 4KB 0x1 8KB Other reserved	0x1	R/W	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
IRAM Size Select Register (MISC_IRAMSZ) (S1C17622)	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	–	reserved	–	–	–	0x1 when being read.
		D3	–	reserved	–	–	–	0 when being read.
		D2–0	IRAMSZ[2:0]	IRAM size select	IRAMSZ[2:0]: 0x3 → Size: 2KB; 0x2 → 4KB; Other → reserved	0x1	R/W	
IRAM Size Select Register (MISC_IRAMSZ) (S1C17602)	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	–	reserved	–	–	–	0x2 when being read.
		D3	–	reserved	–	–	–	0 when being read.
		D2–0	IRAMSZ[2:0]	IRAM size select	IRAMSZ[2:0]: 0x7–0x0 → Size: reserved	0x2	R/W	
IRAM Size Select Register (MISC_IRAMSZ) (S1C17621)	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	–	reserved	–	–	–	0x2 when being read.
		D3	–	reserved	–	–	–	0 when being read.
		D2–0	IRAMSZ[2:0]	IRAM size select	IRAMSZ[2:0]: 0x7–0x0 → Size: reserved	0x2	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)

D[7:3] Reserved

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

Selects the size of the internal RAM to be used.

Table 27.4.2 Internal RAM Size Selection

IRAMSZ[2:0]	Internal RAM size			
	S1C17624/604	S1C17622	S1C17602	S1C17621
0x3	2KB	2KB	Reserved	Reserved
0x2	4KB	4KB	Reserved (default)	Reserved (default)
0x1	8KB (default)	Reserved (default)	Reserved	Reserved
Other	Reserved	Reserved	Reserved	Reserved

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) (S1C17624/604/602)	0xffff90 (32 bits)	D31–24	–	Unused (fixed at 0)	0x0	0x0	R	
		D23–0	DBRAM[23:0]	Debug RAM base address	S1C17624/604: 0x1fc0 S1C17602: 0x0fc0	←	R	

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

D[23:0] DBRAM[23:0]: Debug RAM Base Address Bits (S1C17624/604/602)

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

D[23:0] Not used (Undefined) (S1C17622/621)

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)

28 Multiplier/Divider (COPRO)

28.1 Overview

The S1C17624/604/622/602/621 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.

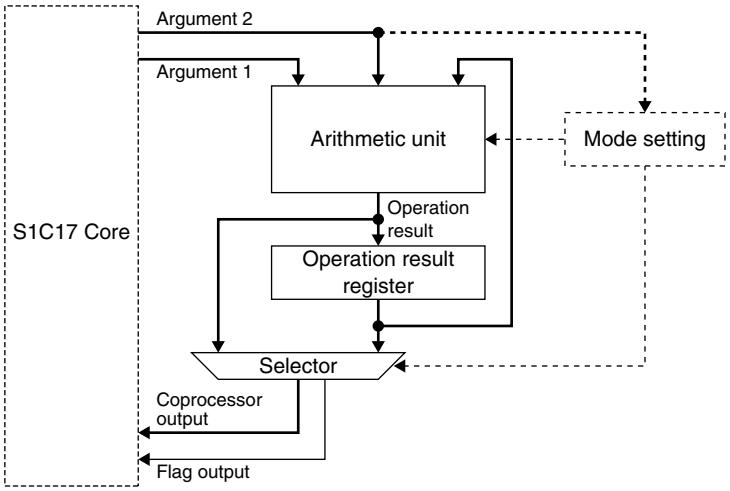


Figure 28.1.1 Multiplier/Divider Block Diagram

28.2 Operation Mode and Output Mode

The Multiplier/divider operates according to the operation mode specified by the application program. As listed in Table 28.2.1, 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)

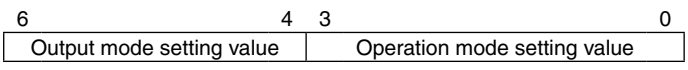


Figure 28.2.1 Mode Setting Register

Table 28.2.1 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

28.3 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.

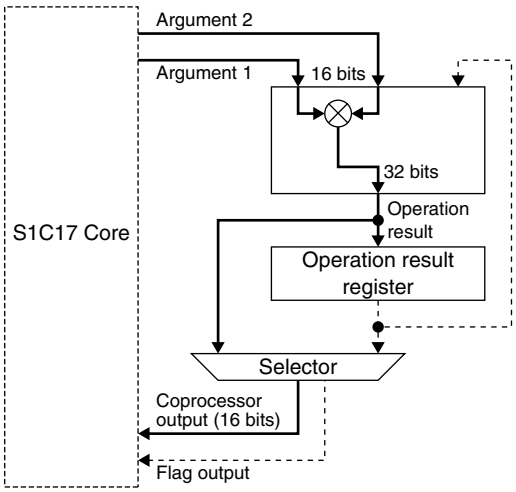


Figure 28.3.1 Data Path in Multiplication Mode

Table 28.3.1 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.
```

28.4 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.

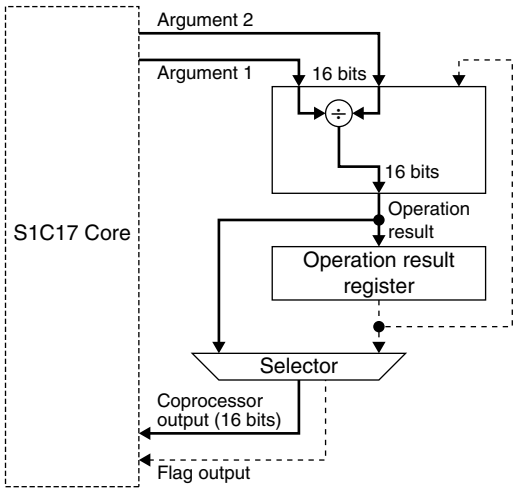


Figure 28.4.1 Data Path in Division Mode

Table 28.4.1 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.
```

28.5 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.

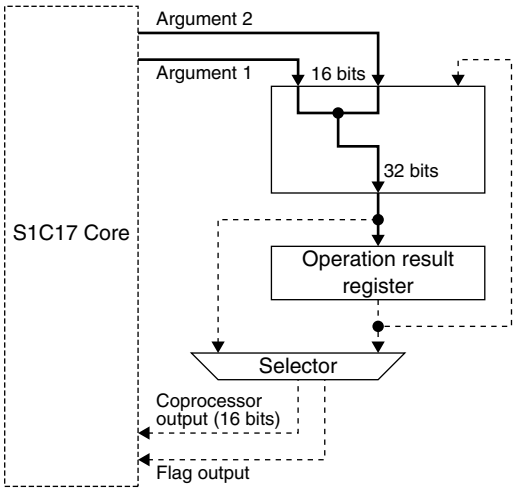


Figure 28.5.1 Data Path in Initialize Mode

Table 28.5.1 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.

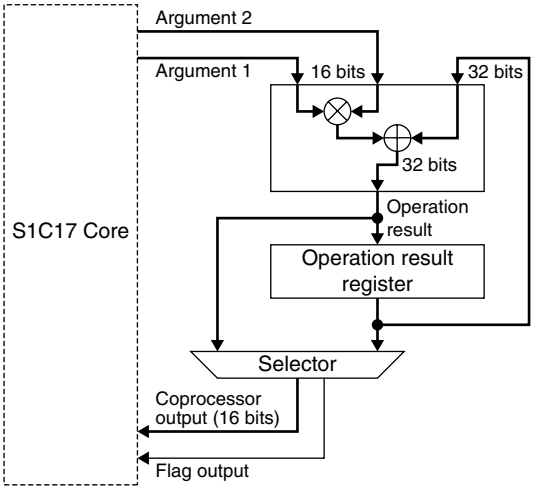


Figure 28.5.2 Data Path in MAC Mode

Table 28.5.2 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 28.5.3 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.

28.6 Reading Operation 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.

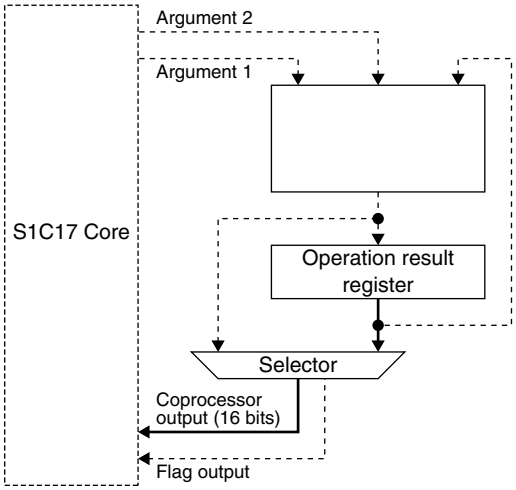


Figure 28.6.1 Data Path in Operation Result Read Mode

Table 28.6.1 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

29 Electrical Characteristics

29.1 Absolute Maximum Ratings

(V_{SS} = 0V)

Item	Symbol	Condition	Rated value	Unit
Power supply voltage	V_{DD}		-0.3 to 4.0	V
Analog power supply voltage	AV_{DD}	$AV_{DD} = V_{DD}$	-0.3 to 4.0	V
LCD power supply voltage	V_{C3}		-0.3 to 6.0	V
Input voltage	V_I		-0.3 to V_{DD} + 0.3	V
Output voltage	V_O		-0.3 to V_{DD} + 0.3	V
High level output current	I_{OH}	1 pin	-5	mA
		Total of all pins	-20	mA
Low level output current	I_{OL}	1 pin	5	mA
		Total of all pins	20	mA
Permissible loss *1	V_O		200	mW
Operating temperature	Ta		-25 to 70	°C
Storage temperature	Tstg		-65 to 150	°C
Soldering temperature/time	Tsol		260°C, 10 seconds (lead section)	–

*1 In case of plastic package

29.2 Recommended Operating Conditions

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Operating power supply voltage	V_{DD}	Normal operation mode	1.8		3.6	V
		Flash programming mode	2.7		3.6	V
Analog power supply voltage	AV_{DD}	$AV_{DD} = V_{DD}$	1.8		3.6	V
Operating frequency	f_{OSC3}	Crystal/ceramic oscillation	0.2		8.2	MHz
	f_{OSC1}	Crystal oscillation		32.768		kHz
Capacitor between V_{SS} and V_{D1}	C_1			0.1		µF
Capacitor between V_{SS} and V_{C1} *1	C_2			0.1		µF
Capacitor between V_{SS} and V_{C2} *1	C_3			0.1		µF
Capacitor between V_{SS} and V_{C3} *1	C_4			0.1		µF
Capacitor between CA and CB *1	C_5			0.1		µF

*1 The capacitors are not required when LCD driver is not used. In this case, leave the V_{C1} to V_{C3}, CA, and CB pins open.

29.3 Current Consumption

S1C17624/604/622 current consumption

Unless otherwise specified: V_{DD} = 1.8 to 3.6V, V_{SS} = 0V, Ta = 25°C, C_1 = 0.1µF, PCKEN[1:0] = 0x3 (ON), VD1MD = 0, FLCYC[2:0] = 0x4 (1 cycle), 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		0.75	2.5	µA
	I_{SLP_RTC} *2	OSC1 = OFF, IOSC = OFF, OSC3 = OFF, RTC = ON		2.2	4.3	µA
Current consumption in HALT mode	I_{HALT1}	OSC1 = 32kHz, IOSC = OFF, OSC3 = OFF, PCKEN[1:0] = 0x0 (OFF)		2.3	4.5	µA
	I_{HALT1_RTC} *2	OSC1 = 32kHz, IOSC = OFF, OSC3 = OFF, PCKEN[1:0] = 0x0 (OFF), RTC = ON		2.3	4.5	µA
	I_{HALT2}	OSC1 = 32kHz, IOSC = OFF, OSC3 = 8MHz (ceramic)		470	500	µA
	I_{HALT3}	OSC1 = 32kHz, IOSC = ON, OSC3 = OFF		200	300	µA
	I_{HALT4}	OSC1 = 32kHz, IOSC = OFF, OSC3 = OFF		3.5	7	
Current consumption during execution *1	I_{EXE1}	OSC1 = 32kHz, IOSC = OFF, OSC3 = OFF, CPU = OSC1		14	24	µA
		OSC1 = 32kHz, IOSC = OFF, OSC3 = OFF, CCLKGR[1:0] = 0x2 (gear ratio 1/4), CPU = OSC1		7	13	µA
	I_{EXE2}	OSC1 = 32kHz, IOSC = OFF, OSC3 = 1MHz (ceramic), CPU = OSC3		400	500	µA
		OSC1 = 32kHz, IOSC = OFF, OSC3 = 8MHz (ceramic), CPU = OSC3		2700	4000	µA
		OSC1 = 32kHz, IOSC = OFF, OSC3 = 8MHz (ceramic), CCLKGR[1:0] = 0x2 (gear ratio 1/4), CPU = OSC3		1300	2000	µA
	I_{EXE3}	OSC1 = 32kHz, IOSC = ON, OSC3 = OFF, CPU = IOSC		1000	1500	µA
	I_{EXE11}	OSC1 = 32kHz, IOSC = OFF, OSC3 = OFF, VD1MD = 1, CPU = OSC1		31	45	µA
	I_{EXE21}	OSC1 = 32kHz, IOSC = OFF, OSC3 = 1MHz (ceramic), VD1MD = 1, CPU = OSC3		690	900	µA
		OSC1 = 32kHz, IOSC = OFF, OSC3 = 8MHz (ceramic), VD1MD = 1, CPU = OSC3		4500	6500	µA
	I_{EXE31}	OSC1 = 32kHz, IOSC = ON, OSC3 = OFF, VD1MD = 1, CPU = IOSC		1600	2400	µA
Current consumption during execution in heavy load protection mode *1	I_{EXE1H}	OSC1 = 32kHz, IOSC = OFF, OSC3 = OFF, CPU = OSC1, HVLD = 1		22	34	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 S1C17624/604 only

S1C17602/621 current consumption

Unless otherwise specified: V_{DD} = 1.8 to 3.6V, V_{SS} = 0V, Ta = 25°C, C_1 = 0.1µF, PCKEN[1:0] = 0x3 (ON), VD1MD = 0, FLCYC[2:0] = 0x4 (1 cycle), CCLKGR[1:0] = 0x0 (gear ratio 1/1)

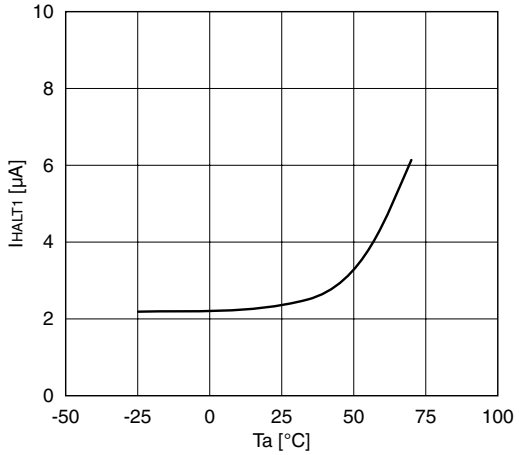
Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Current consumption in SLEEP mode	I_{SLP}	OSC1 = OFF, IOSC = OFF, OSC3 = OFF		0.75	2.5	µA
Current consumption in HALT mode	I_{HALT1}	OSC1 = 32kHz, IOSC = OFF, OSC3 = OFF, PCKEN[1:0] = 0x0 (OFF)		2.5	5.0	µA
	I_{HALT2}	OSC1 = 32kHz, IOSC = OFF, OSC3 = 8MHz (ceramic)		650	900	µA
	I_{HALT3}	OSC1 = 32kHz, IOSC = ON, OSC3 = OFF		270	400	µA
	I_{HALT4}	OSC1 = 32kHz, IOSC = OFF, OSC3 = OFF		4.5	9	
Current consumption during execution *1	I_{EXE1}	OSC1 = 32kHz, IOSC = OFF, OSC3 = OFF, CPU = OSC1		15	25	µA
		OSC1 = 32kHz, IOSC = OFF, OSC3 = OFF, CCLKGR[1:0] = 0x2 (gear ratio 1/4), CPU = OSC1		8	14	µA
	I_{EXE2}	OSC1 = 32kHz, IOSC = OFF, OSC3 = 1MHz (ceramic), CPU = OSC3		410	620	µA
		OSC1 = 32kHz, IOSC = OFF, OSC3 = 8MHz (ceramic), CPU = OSC3		3100	4600	µA
		OSC1 = 32kHz, IOSC = OFF, OSC3 = 8MHz (ceramic), CCLKGR[1:0] = 0x2 (gear ratio 1/4), CPU = OSC3		1500	2200	µA
	I_{EXE3}	OSC1 = 32kHz, IOSC = ON, OSC3 = OFF, CPU = IOSC		1100	1600	µA
	I_{EXE11}	OSC1 = 32kHz, IOSC = OFF, OSC3 = OFF, VD1MD = 1, CPU = OSC1		32	50	µA

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Current consumption during execution *1	I_{EXE21}	OSC1 = 32kHz, IOSC = OFF, OSC3 = 1MHz (ceramic), VD1MD = 1, CPU = OSC3		700	1100	µA
		OSC1 = 32kHz, IOSC = OFF, OSC3 = 8MHz (ceramic), VD1MD = 1, CPU = OSC3		5200	8000	µA
	I_{EXE31}	OSC1 = 32kHz, IOSC = ON, OSC3 = OFF, VD1MD = 1, CPU = IOSC		1900	2700	µA
Current consumption during execution in heavy load protection mode *1	I_{EXE1H}	OSC1 = 32kHz, IOSC = OFF, OSC3 = OFF, CPU = OSC1, HVLD = 1		23	35	µA

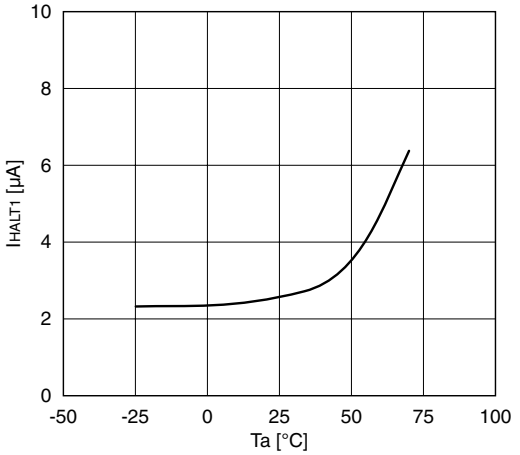
*1 The values of current consumption during execution were measured when a test program consisting of 60.5% ALU instructions, 17% branch instructions, 12% memory read instructions, and 10.5% memory write instructions was executed continuously in the Flash memory.

Current consumption-temperature characteristic in HALT mode (OSC1 operation)

OSC1 = 32.768kHz crystal, IOSC = OFF, OSC3 = OFF, PCKEN[1:0] = 0x0, VD1MD = 0, CCLKGR[1:0] = 0x0, Typ. value



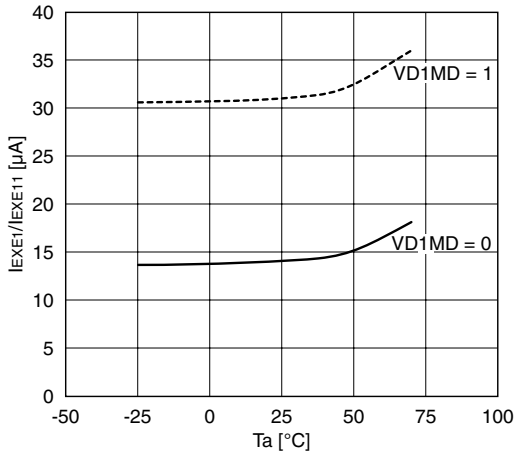
<S1C17624/604/622>



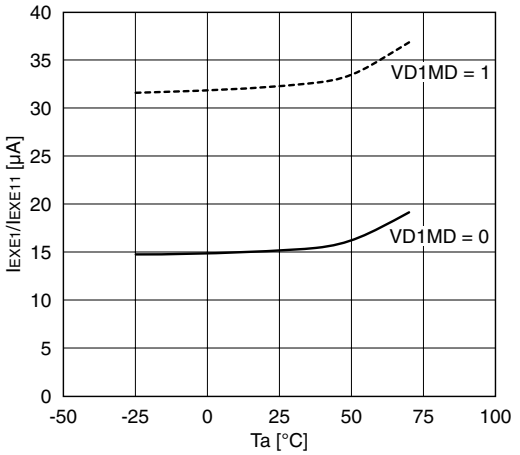
<S1C17602/621>

Current consumption-temperature characteristic during execution with OSC1

OSC1 = 32.768kHz crystal, IOSC = OFF, OSC3 = OFF, PCKEN[1:0] = 0x3, CCLKGR[1:0] = 0x0, Typ. value



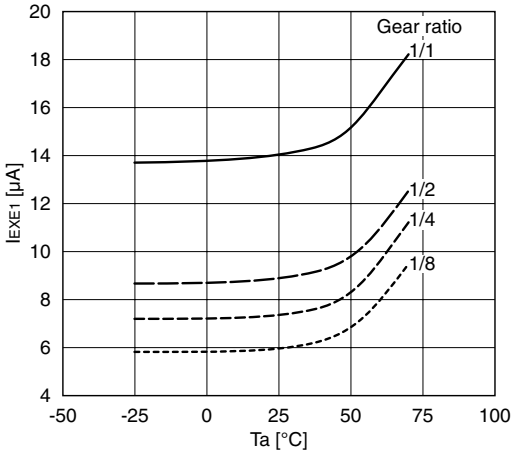
<S1C17624/604/622>



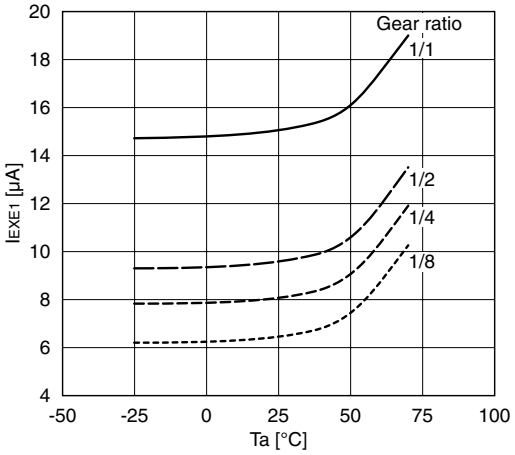
<S1C17602/621>

Current consumption-temperature characteristic during execution with OSC1 + clock gear

OSC1 = 32.768kHz crystal, IOSC = OFF, OSC3 = OFF, PCKEN[1:0] = 0x3, VD1MD = 0, Typ. value



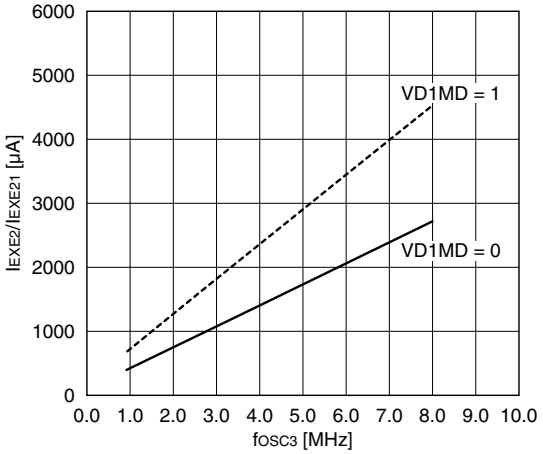
<S1C17624/604/622>



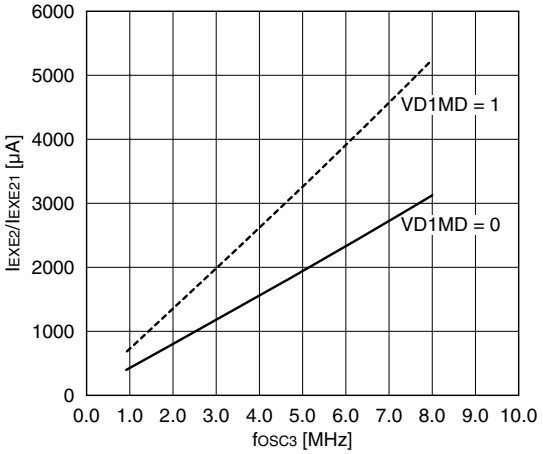
<S1C17602/621>

Current consumption-frequency characteristic during execution with OSC3

OSC3 = crystal/ceramic, IOSC = OFF, OSC1 = OFF, PCKEN[1:0] = 0x3, CCLKGR[1:0] = 0x0, Ta = 25°C, Typ. value



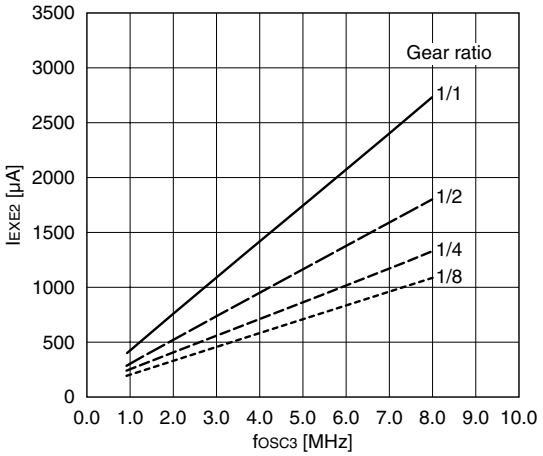
<S1C17624/604/622>



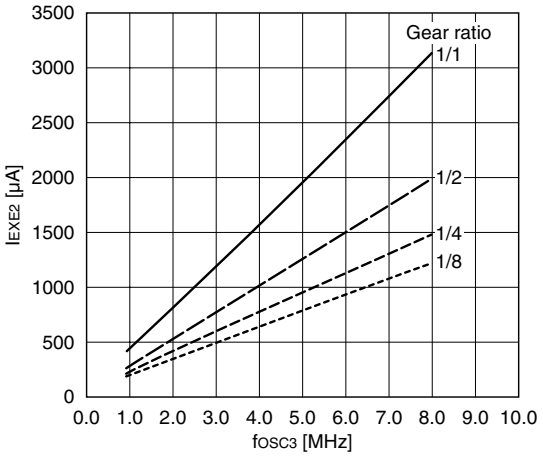
<S1C17602/621>

Current consumption-frequency characteristic during execution with OSC3 + clock gear

OSC3 = crystal/ceramic, IOSC = OFF, OSC1 = OFF, PCKEN[1:0] = 0x3, VD1MD = 0, Ta = 25°C, Typ. value



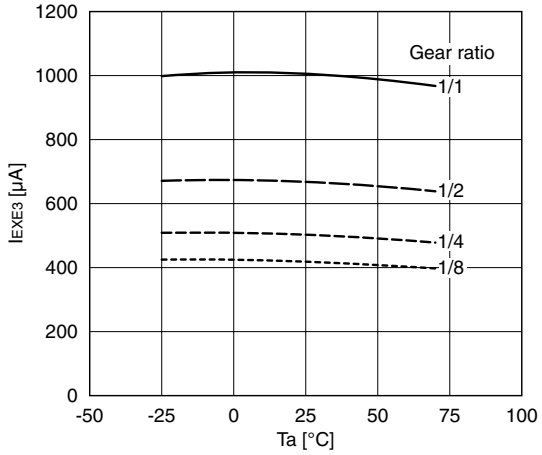
<S1C17624/604/622>



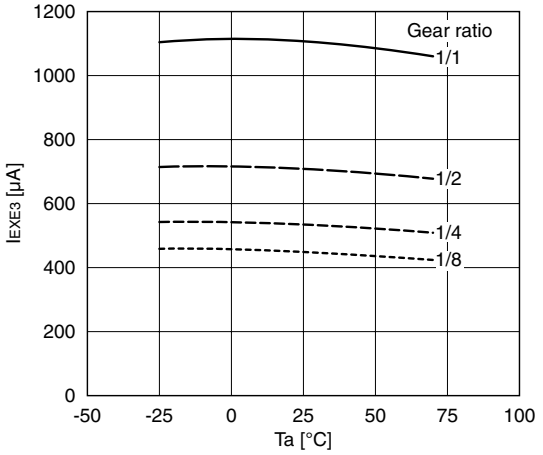
<S1C17602/621>

Current consumption-temperature characteristic during execution with IOSC + clock gear

IOSC = internal oscillator, OSC1 = OFF, OSC3 = OFF, PCKEN[1:0] = 0x3, VD1MD = 0, Typ. value



<S1C17624/604/622>



<S1C17602/621>

29.4 Oscillation Characteristics

Oscillation characteristics change depending on conditions (board pattern, components used, etc.). Use the following characteristics as reference values. See Appendix E for recommended resonators.

OSC1 crystal oscillation

Unless otherwise specified: V_{DD} = 1.8 to 3.6V, V_{SS} = 0V, Ta = 25°C, C_{G1} = 25pF external, C_{D1} = built-in, R_{f1} = built-in

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Oscillation start time*1	t_{sta}				3	s
Built-in drain capacitance	C_{D1}	In case of the chip		10		pF

*1 Crystal resonator = MC-146: manufactured by EPSON TOYOCOM (R_1 = 65kΩ Max., C_L = 12.5pF)

OSC3 crystal oscillation

Unless otherwise specified: V_{DD} = 1.8 to 3.6V, V_{SS} = 0V, Ta = 25°C, R_{F3} = 1MΩ

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Oscillation start time *1	t_{sta}				20	ms

*1 Crystal resonator = MA-406: manufactured by EPSON TOYOCOM (R_1 = 150Ω, C_L = 10pF)

OSC3 ceramic oscillation

Unless otherwise specified: V_{DD} = 1.8 to 3.6V, V_{SS} = 0V, Ta = 25°C, R_{F3} = 1MΩ

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

*1 Ceramic resonator = CSTR4M00G53095-R0: manufactured by Murata Manufacturing Co., Ltd (C_{G3} = C_{D3} = 15pF built-in)

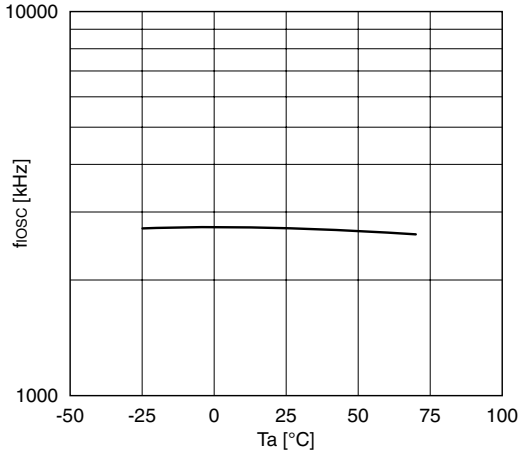
IOSC oscillation

Unless otherwise specified: V_{DD} = 1.8 to 3.6V, V_{SS} = 0V, Ta = -25 to 70°C

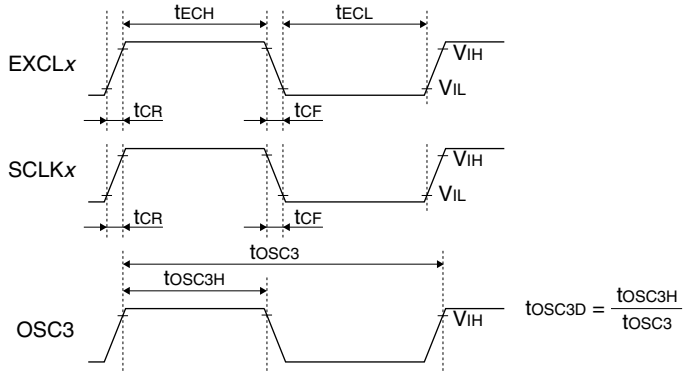
Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Oscillation start time	t_{sta}				5	µs
Oscillation frequency	f_{IOSC}	Normal operation mode	2.16	2.70	3.24	MHz

IOSC oscillation frequency-temperature characteristic

Typ. value



29.5 External Clock Input Characteristics



Unless otherwise specified: V_{DD} = 1.8 to 3.6V, V_{SS} = 0V, V_{IH} = 0.8V_{DD}, V_{IL} = 0.2V_{DD}, Ta = -25 to 70°C

Item	Symbol	Min.	Typ.	Max.	Unit
EXCL0/1/2/3 input High pulse width	t_{ECH}	2/f_{SYS}			s
EXCL5/6 input High pulse width		60			ns
EXCL0/1/2/3 input Low pulse width	t_{ECL}	2/f_{SYS}			s
EXCL5/6 input Low pulse width		60			ns
UART transfer rate	R_U			460800	bps
UART transfer rate (IrDA mode)	R_{UIrDA}			115200	bps
Input rise time	t_{CR}			80	ns
Input fall time	t_{CF}			80	ns
OSC3 clock cycle time	t_{OSC3}	125			ns
OSC3 clock input duty	t_{OSC3D}	46		54	%

∗ f_{SYS}: System operating clock frequency

29.6 Input/Output Pin Characteristics

S1C17624/604/622

Unless otherwise specified: V_{DD} = 1.8 to 3.6V, V_{SS} = 0V, Ta = -25 to 70°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
High level input voltage	V_{IH}	P*xx*	0.8V_{DD}		V_{DD}	V
Low level input voltage	V_{IL}	P*xx*	0		0.2V_{DD}	V
High level Schmitt input threshold voltage	V_{T1+}	#RESET	0.5V_{DD}		0.9V_{DD}	V
Low level Schmitt input threshold voltage	V_{T1-}	#RESET	0.1V_{DD}		0.5V_{DD}	V
High level Schmitt input threshold voltage ∗1	V_{T2+}	P*xx*	0.5V_{DD}		0.9V_{DD}	V
Low level Schmitt input threshold voltage ∗1	V_{T2-}	P*xx*	0.1V_{DD}		0.5V_{DD}	V
Hysteresis voltage	ΔV_T	P*xx*, #RESET	0.1			V

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
High level output current	I_{OH}	P*xx*, V_{OH} = 0.9V_{DD}			-0.5	mA
Low level output current	I_{OL}	P*xx*, V_{OL} = 0.1V_{DD}	0.5			mA
Leakage current	I_{LEAK}	P*xx*, #RESET	-100		100	nA
Input pull-up resistance	R_{IN}	P*xx*, #RESET	100		500	kΩ
Pin capacitance	C_{IN}	P*xx*, V_{IN} = 0V, f = 1MHz, Ta = 25°C			15	pF
Reset low pulse width	t_{SR}	V_{IH} = 0.8V_{DD}, V_{IL} = 0.2V_{DD}	100			µs
Operating power voltage	V_{SR}		1.8			V
Power-on reset time	t_{PSR}		1.0			ms

*1 When Schmitt input is enabled

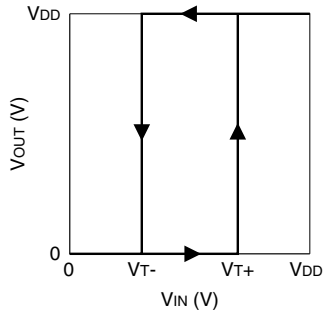
S1C17602/621

Unless otherwise specified: V_{DD} = 1.8 to 3.6V, V_{SS} = 0V, Ta = -25 to 70°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
High level input voltage	V_{IH}	P*xx*	0.8V_{DD}		V_{DD}	V
Low level input voltage	V_{IL}	P*xx*	0		0.2V_{DD}	V
High level Schmitt input threshold voltage	V_{T1+}	#RESET	0.5V_{DD}		0.9V_{DD}	V
Low level Schmitt input threshold voltage	V_{T1-}	#RESET	0.1V_{DD}		0.5V_{DD}	V
High level Schmitt input threshold voltage *1	V_{T2+}	P*xx*	0.5V_{DD}		0.9V_{DD}	V
Low level Schmitt input threshold voltage *1	V_{T2-}	P*xx*	0.1V_{DD}		0.5V_{DD}	V
High level output current	I_{OH}	P*xx*, V_{OH} = 0.9V_{DD}			-0.5	mA
Low level output current	I_{OL}	P*xx*, V_{OL} = 0.1V_{DD}	0.5			mA
Leakage current	I_{LEAK}	P*xx*, #RESET	-1		1	µA
Input pull-up resistance	R_{IN}	P*xx*, #RESET	100		500	kΩ
Pin capacitance	C_{IN}	P*xx*, V_{IN} = 0V, f = 1MHz, Ta = 25°C			15	pF
Reset low pulse width	t_{SR}	V_{IH} = 0.8V_{DD}, V_{IL} = 0.2V_{DD}	100			µs
Operating power voltage	V_{SR}		1.8			V
Power-on reset time	t_{PSR}		1.0			ms

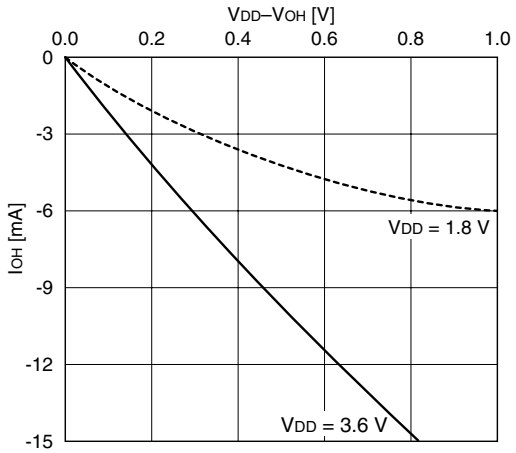
*1 When Schmitt input is enabled

Schmitt input threshold voltage



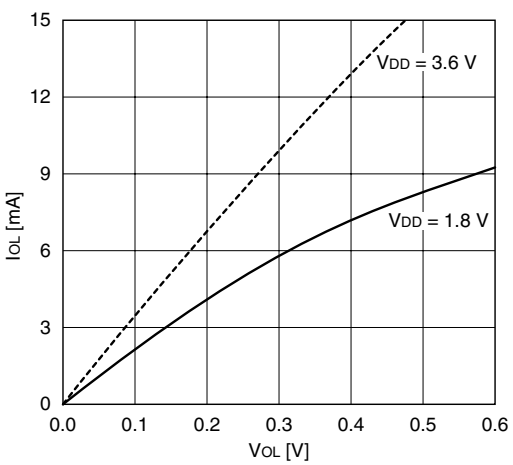
High-level output current characteristic

Ta = 70°C, Max. value

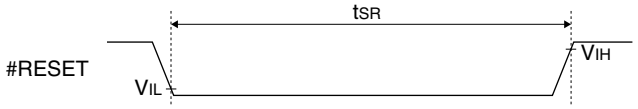


Low-level output current characteristic

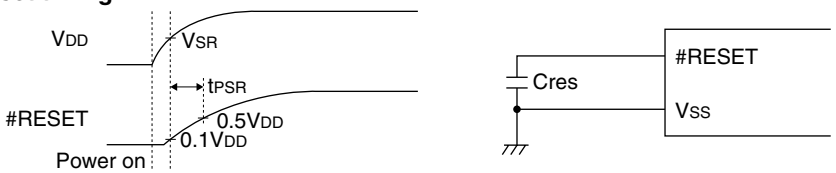
Ta = 70°C, Min. value



Reset pulse

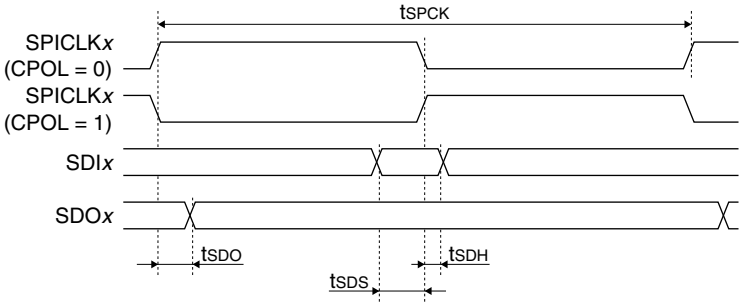


Power-on reset timing



Note: Be sure to set the #RESET pin to 0.1 V_{DD} or less when performing a power-on reset after the power is turned off.

29.7 SPI Characteristics



Master mode

Unless otherwise specified: V_{DD} = 1.8 to 3.6V, V_{SS} = 0V, Ta = -25 to 70°C

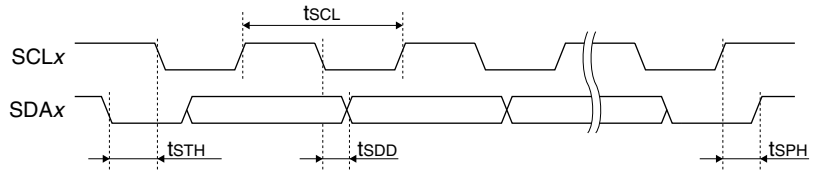
Item	Symbol	Min.	Typ.	Max.	Unit
SPICLK*x* cycle time	t_{SPCK}	500			ns
SDI*x* setup time	t_{SDS}	120			ns
SDI*x* hold time	t_{SDH}	10			ns
SDO*x* output delay time	t_{SDO}			20	ns

Slave mode

Unless otherwise specified: V_{DD} = 1.8 to 3.6V, V_{SS} = 0V, Ta = -25 to 70°C

Item	Symbol	Min.	Typ.	Max.	Unit
SPICLK*x* cycle time	t_{SPCK}	500			ns
SDI*x* setup time	t_{SDS}	10			ns
SDI*x* hold time	t_{SDH}	10			ns
SDO*x* output delay time	t_{SDO}			130	ns

29.8 I²C Characteristics



Unless otherwise specified: V_{DD} = 1.8 to 3.6V, V_{SS} = 0V, Ta = -25 to 70°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

29.9 LCD Driver Characteristics

The typical values in the following LCD driver characteristics varies depending on the panel load (panel size, drive duty, number of display pixels and display contents), so evaluate them by connecting to the actually used LCD panel.

LCD drive voltage

Unless otherwise specified: V_{DD} = 2.5 to 3.6V, V_{SS} = 0V, Ta = 25°C, C_2–C_5 = 0.1µF, Checker pattern displayed, No panel load, VCSEL = 1 (V_{C2} reference voltage)

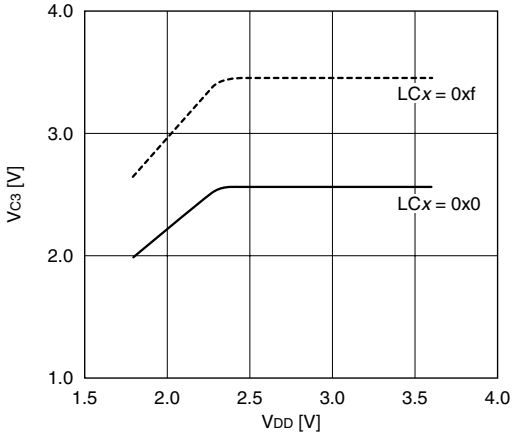
Item	Symbol	Condition		Min.	Typ.	Max.	Unit
LCD drive voltage (V_{C2} reference voltage)	V_{C1}	Connect 1MΩ load resistor between V_{SS} and V_{C1}		0.324 × V_{C3}(Typ.)		0.350 × V_{C3}(Typ.)	V
	V_{C2}	Connect 1MΩ load resistor between V_{SS} and V_{C2}		0.649 × V_{C3}(Typ.)		0.701 × V_{C3}(Typ.)	V
	V_{C3}	Connect 1MΩ load resistor between V_{SS} and V_{C3}	LC[3:0] = 0x0		2.56		V
			LC[3:0] = 0x1		2.62		V
			LC[3:0] = 0x2		2.68		V
			LC[3:0] = 0x3		2.74		V
			LC[3:0] = 0x4		2.80		V
			LC[3:0] = 0x5		2.86		V
			LC[3:0] = 0x6		2.92		V
			LC[3:0] = 0x7	Typ. × 0.96	2.98	Typ. × 1.04	V
			LC[3:0] = 0x8		3.04		V
			LC[3:0] = 0x9		3.10		V
			LC[3:0] = 0xa		3.15		V
			LC[3:0] = 0xb		3.21		V
			LC[3:0] = 0xc		3.27		V
			LC[3:0] = 0xd		3.33		V
			LC[3:0] = 0xe		3.39		V
			LC[3:0] = 0xf		3.45		V

Unless otherwise specified: V_{DD} = 1.8 to 3.6V, V_{SS} = 0V, Ta = 25°C, C_2–C_5 = 0.1µF, Checker pattern displayed, No panel load, VCSEL = 0 (V_{C1} reference voltage)

Item	Symbol	Condition		Min.	Typ.	Max.	Unit
LCD drive voltage (V_{C1} reference voltage)	V_{C1}	Connect 1MΩ load resistor between V_{SS} and V_{C1}		0.333 × V_{C3}(Typ.)		0.360 × V_{C3}(Typ.)	V
	V_{C2}	Connect 1MΩ load resistor between V_{SS} and V_{C2}		0.645 × V_{C3}(Typ.)		0.696 × V_{C3}(Typ.)	V
	V_{C3}	Connect 1MΩ load resistor between V_{SS} and V_{C3}	LC[3:0] = 0x0		2.50		V
			LC[3:0] = 0x1		2.56		V
			LC[3:0] = 0x2		2.61		V
			LC[3:0] = 0x3		2.67		V
			LC[3:0] = 0x4		2.73		V
			LC[3:0] = 0x5		2.79		V
			LC[3:0] = 0x6		2.85		V
			LC[3:0] = 0x7	Typ. × 0.96	2.90	Typ. × 1.04	V
			LC[3:0] = 0x8		2.96		V
			LC[3:0] = 0x9		3.02		V
			LC[3:0] = 0xa		3.08		V
			LC[3:0] = 0xb		3.14		V
			LC[3:0] = 0xc		3.19		V
			LC[3:0] = 0xd		3.25		V
			LC[3:0] = 0xe		3.31		V
			LC[3:0] = 0xf		3.37		V

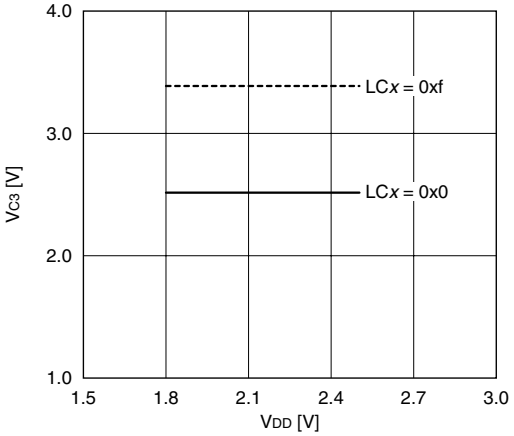
LCD drive voltage-supply voltage characteristic (when V_{C2} reference voltage is selected)

When a 1 MΩ load resistor is connected between V_{SS} and V_{C3} (no panel load), Ta = 25°C, Typ. value



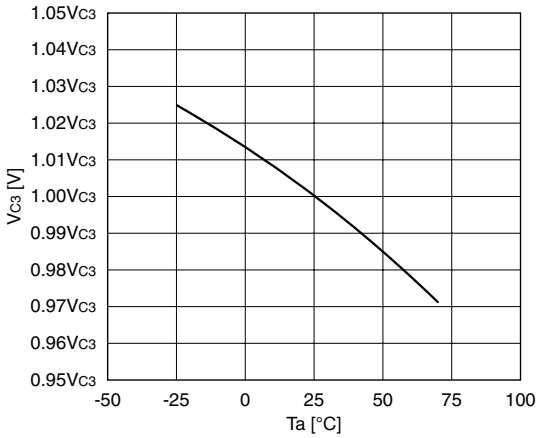
LCD drive voltage-supply voltage characteristic (when V_{C1} reference voltage is selected)

When a 1 MΩ load resistor is connected between V_{SS} and V_{C3} (no panel load), Ta = 25°C, Typ. value



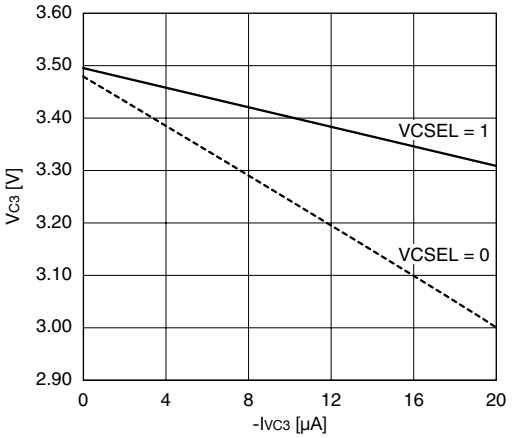
LCD drive voltage-temperature characteristic

Typ. value (V_{C2}, V_{C1} reference voltage)



LCD drive voltage-load characteristic

When a load is connected to the V_{C3} pin only
LC*x* = 0xf, Ta = 25°C, Typ. value



SEG/COM output characteristics

Unless otherwise specified: V_{DD} = 1.8 to 3.6V, V_{SS} = 0V, Ta = -25 to 70°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Segment/Common output current	I_{SEGH}	SEG*xx*, COM*xx*, V_{SEGH} = V_{C3} - 0.1V			-5	µA
	I_{SEGL}	SEG*xx*, COM*xx*, V_{SEGL} = 0.1V	5			µA

S1C17624/604/622 LCD driver circuit current consumption

Unless otherwise specified: V_{DD} = 1.8 to 3.6V, V_{SS} = 0V, Ta = 25°C, C_2–C_5 = 0.1µF, No LCD panel load, PCKEN[1:0] = 0x0 (OFF), FLCYC[2:0] = 0x4 (1 cycle), CCLKGR[1:0] = 0x0 (gear ratio 1/1)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
LCD circuit current with V_{C2} reference voltage *1	I_{LCD2}	DSPC[1:0] = 0x1 (checker pattern), LC[3:0] = 0xf, OSC1 = 32kHz, V_{DD} = 2.5 to 3.6V, VCSEL = 1		1.7	5	µA
Heavy load protection mode LCD circuit current with V_{C2} reference voltage *1	I_{LCD2H}	DSPC[1:0] = 0x1 (checker pattern), LC[3:0] = 0xf, OSC1 = 32kHz, V_{DD} = 2.5 to 3.6V, LHVLD = 1, VCSEL = 1		21	32	µA
LCD circuit current with V_{C1} reference voltage *1	I_{LCD1}	DSPC[1:0] = 0x1 (checker pattern), LC[3:0] = 0xf, OSC1 = 32kHz, V_{DD} = 1.8 to 3.6V, VCSEL = 0		2.6	8	µA
Heavy load protection mode LCD circuit current with V_{C1} reference voltage *1	I_{LCD1H}	DSPC[1:0] = 0x1 (checker pattern), LC[3:0] = 0xf, OSC1 = 32kHz, V_{DD} = 1.8 to 3.6V, LHVLD = 1, VCSEL = 0		13	20	µA

*1 This value is added to the current consumption in HALT mode or current consumption during execution when the LCD circuit is active. Current consumption increases according to the display contents and panel load.

S1C17602/621 LCD driver circuit current consumption

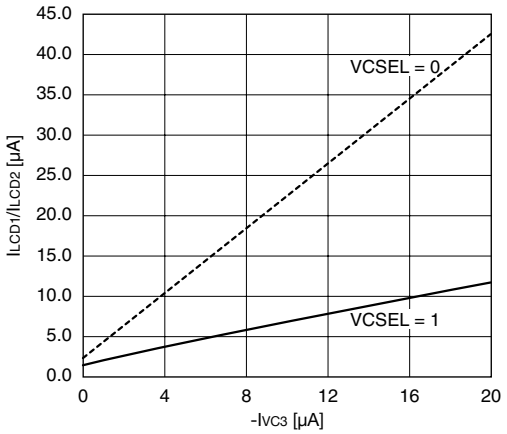
Unless otherwise specified: V_{DD} = 1.8 to 3.6V, V_{SS} = 0V, Ta = 25°C, C_2–C_5 = 0.1µF, No LCD panel load, PCKEN[1:0] = 0x0 (OFF), FLCYC[2:0] = 0x4 (1 cycle), CCLKGR[1:0] = 0x0 (gear ratio 1/1)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
LCD circuit current with V_{C2} reference voltage *1	I_{LCD2}	DSPC[1:0] = 0x1 (checker pattern), LC[3:0] = 0xf, OSC1 = 32kHz, V_{DD} = 2.5 to 3.6V, VCSEL = 1		1	3	µA
Heavy load protection mode LCD circuit current with V_{C2} reference voltage *1	I_{LCD2H}	DSPC[1:0] = 0x1 (checker pattern), LC[3:0] = 0xf, OSC1 = 32kHz, V_{DD} = 2.5 to 3.6V, LHVLD = 1, VCSEL = 1		21	32	µA
LCD circuit current with V_{C1} reference voltage *1	I_{LCD1}	DSPC[1:0] = 0x1 (checker pattern), LC[3:0] = 0xf, OSC1 = 32kHz, V_{DD} = 1.8 to 3.6V, VCSEL = 0		1.5	5	µA
Heavy load protection mode LCD circuit current with V_{C1} reference voltage *1	I_{LCD1H}	DSPC[1:0] = 0x1 (checker pattern), LC[3:0] = 0xf, OSC1 = 32kHz, V_{DD} = 1.8 to 3.6V, LHVLD = 1, VCSEL = 0		13	20	µA

*1 This value is added to the current consumption in HALT mode or current consumption during execution when the LCD circuit is active. Current consumption increases according to the display contents and panel load.

LCD current consumption-load characteristic

When a load is connected to the V_{C3} pin only
V_{DD} = 3.6V, Ta = 25°C, Typ. value



29.10 A/D Converter Characteristics

Analog characteristics

Unless otherwise specified: V_{DD} = AV_{DD} = 1.8 to 3.6V, V_{SS} = 0V, Ta = -25 to 70°C, ADST[2:0] = 0x7 (9 cycles)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Resolution	–			10		bits
A/D conversion clock	f_{ADCLK}		16		2000	kHz
Sampling rate *1	f_{SMP}		0.8		100	ksps
Zero-scale error	E_{ZS}				±3	LSB
Full-scale error	E_{FS}				±3	LSB
Integral linearity error *2	E_{INL}	AV_{DD} = 2.7 to 3.6 V			±1.5	LSB
		AV_{DD} = 1.8 to 3.6 V			±2.0	LSB
Differential linearity error	E_{DNL}				±1.0	LSB
Analog input resistance	R_{AIN}				11	kΩ
Analog input capacitance	C_{AIN}				20	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} = 16kHz.

*2 Integral linearity error is measured at the end point line.

A/D converter current consumption

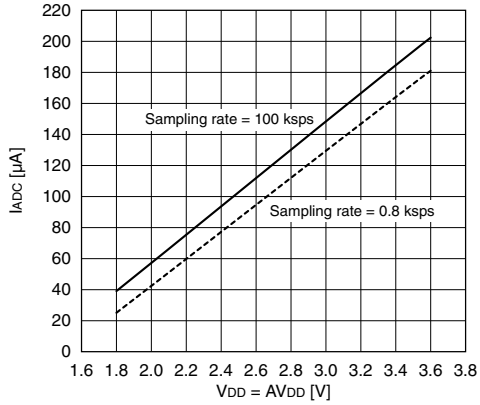
Unless otherwise specified: V_{DD} = AV_{DD} = 1.8 to 3.6V, V_{SS} = 0V, Ta = 25°C, ADST[2:0] = 0x7 (9 cycles), PCKEN[1:0] = 0x3 (ON), A_{IN} = AV_{DD}/2V

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
A/D converter operating current *1	I_{ADC}	V_{DD} = AV_{DD} = 3.6V, sampling rate = 100ksps		200	350	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

A_{IN} = AV_{DD}/2V, Ta = 25°C, Typ. value



29.11 R/F Converter Characteristics

Analog characteristics

Unless otherwise specified: V_{DD} = 1.8 to 3.6V, V_{SS} = 0V, Ta = -25 to 70°C

Item	Symbol	Condition		Min.	Typ.	Max.	Unit
Reference and sensor oscillation frequency *1	f_{RFCLK}			1		4000	kHz
Reference and sensor oscillation frequency IC deviation *2	$\Delta f_{RFCLK}/\Delta IC$	Resistive sensor DC/AC oscillation mode	V_{DD} = 3.6V	-25		25	%
			V_{DD} = 1.8V	-40		40	%
		Capacitive sensor DC oscillation mode	V_{DD} = 3.6V	-25		25	%
			V_{DD} = 1.8V	-50		50	%
Reference resistor and resistive sensor resistance *3	R_{REF}, R_{SEN}	Resistive sensor DC/Capacitive sensor DC oscillation mode		1			kΩ
		Resistive sensor AC oscillation mode		10			kΩ
Reference capacitor and capacitive sensor capacitance *3	C_{REF}, C_{SEN}	Resistive sensor DC/AC oscillation mode		100			pF
		Capacitive sensor DC oscillation mode		100		2000	pF
Time base counter clock frequency	f_{TCCLK}					8.2	MHz
RFIN pin high level Schmitt input voltage	V_{T+}			0.5•V_{DD}		0.9•V_{DD}	V
RFIN pin low level Schmitt input voltage	V_{T-}			0.1•V_{DD}		0.5•V_{DD}	V

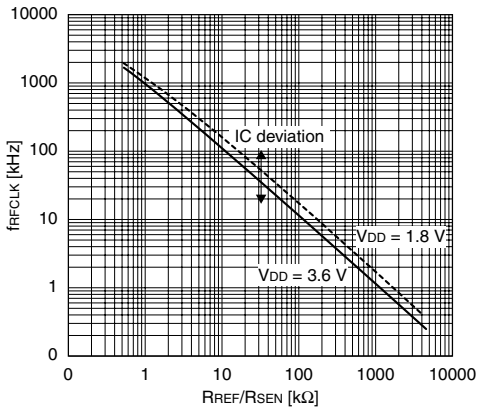
*1 The oscillation frequency/IC deviation characteristic value may increase due to variations in oscillation frequency caused by leakage if the oscillation frequency is 1 kHz or lower.

*2 In these characteristics, unevenness between production lots, and variations in board, resistances and capacitances used in the measurement environment are taken into account (variations in temperature are not included).

*3 The CR oscillation can be performed if the resistance or capacitance is out of the range shown in the table (see characteristic curves), note, however, that the oscillation frequency/IC deviation characteristic value may increase due to parasitic elements on the board and those in the IC.

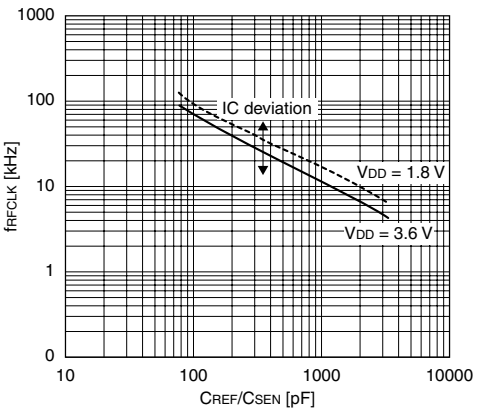
RFC reference/sensor oscillation frequency-resistance characteristic (Resistive sensor DC/AC oscillation mode)

C_{REF}/C_{SEN} = 1000pF, Ta = 25°C, Typ. value



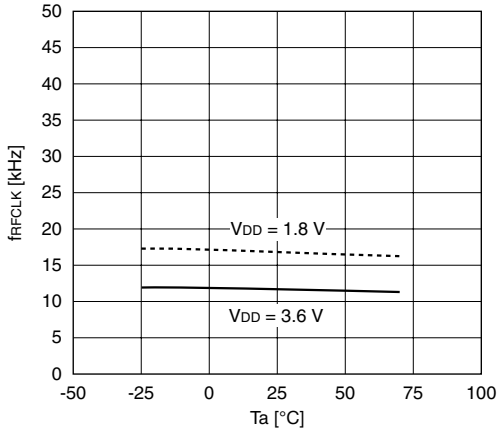
RFC reference/sensor oscillation frequency-capacitance characteristic (Capacitive sensor DC oscillation mode)

R_{REF}/R_{SEN} = 100kΩ, Ta = 25°C, Typ. value



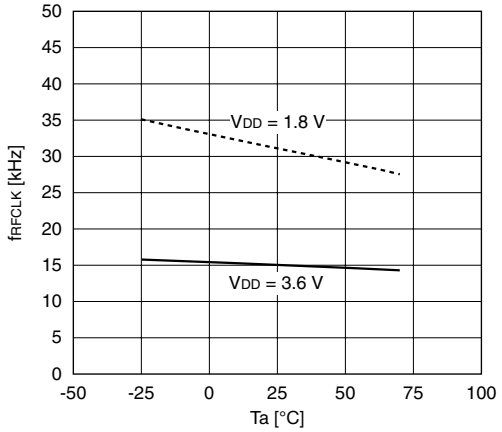
RFC reference/sensor oscillation frequency-temperature characteristic (Resistive sensor DC/AC oscillation mode)

RREF/RSEN = 100kΩ, CREF/CSEN = 1000pF, Typ. value



RFC reference/sensor oscillation frequency-temperature characteristic (Capacitive sensor DC oscillation mode)

RREF/RSEN = 100kΩ, CREF/CSEN = 1000pF, Typ. value



S1C17624/604/622 R/F converter current consumption

Unless otherwise specified: VDD = 3.6V, VSS = 0V, Ta = 25°C, PCKEN[1:0] = 0x0 (OFF), CREF/CSEN = 1000pF, RREF/RSEN = 100kΩ, TCCLK = 8MHz

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
R/F converter operating current *1	IRFC	Resistive sensor DC/AC oscillation mode		110	160	µA
		Capacitive sensor DC oscillation mode		120	180	µA

*1 This value is added to the current consumption in HALT mode or current consumption during execution when the R/F converter is active. Current consumption depends on the VDD voltage, reference capacitance/sensor capacitance, and reference/sensor oscillation frequency.

S1C17602/621 R/F converter current consumption

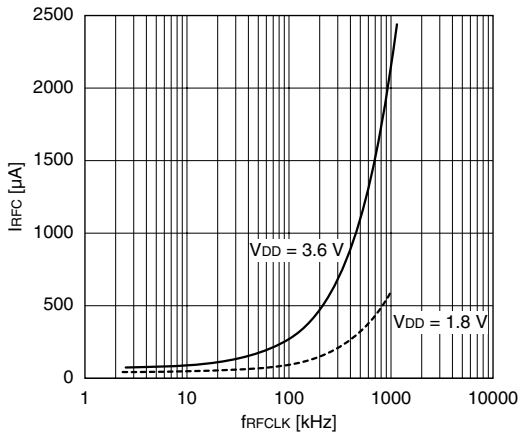
Unless otherwise specified: VDD = 3.6V, VSS = 0V, Ta = 25°C, PCKEN[1:0] = 0x0 (OFF), CREF/CSEN = 1000pF, RREF/RSEN = 100kΩ, TCCLK = 8MHz

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
R/F converter operating current *1	IRFC	Resistive sensor DC/AC oscillation mode		140	200	µA
		Capacitive sensor DC oscillation mode		150	220	µA

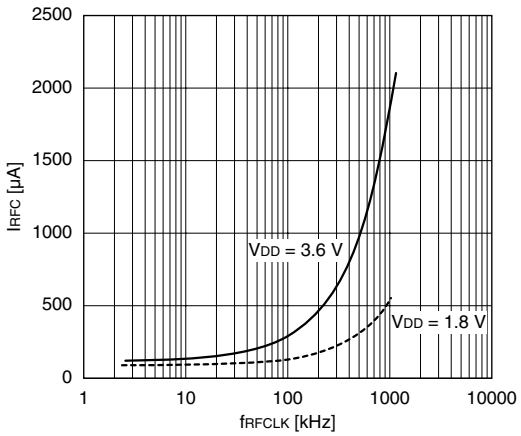
*1 This value is added to the current consumption in HALT mode or current consumption during execution when the R/F converter is active. Current consumption depends on the VDD voltage, reference capacitance/sensor capacitance, and reference/sensor oscillation frequency.

RFC reference/sensor oscillation current consumption-frequency characteristic (Resistive sensor DC/AC oscillation mode)

CREF/CSEN = 1000pF, Typ. value



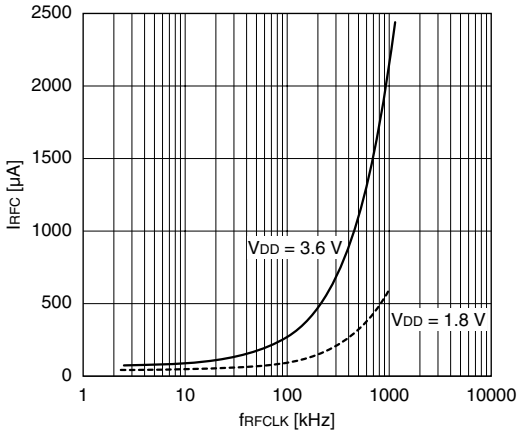
<S1C17624/604/622>



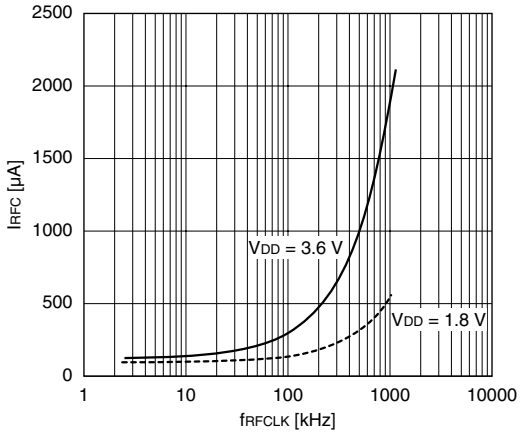
<S1C17602/621>

RFC reference/sensor oscillation current consumption-frequency characteristic (Capacitive sensor DC oscillation mode)

C_{REF}/C_{SEN} = 1000pF, Typ. value



<S1C17624/604/622>



<S1C17602/621>

29.12 SVD Circuit Characteristics

Analog characteristics

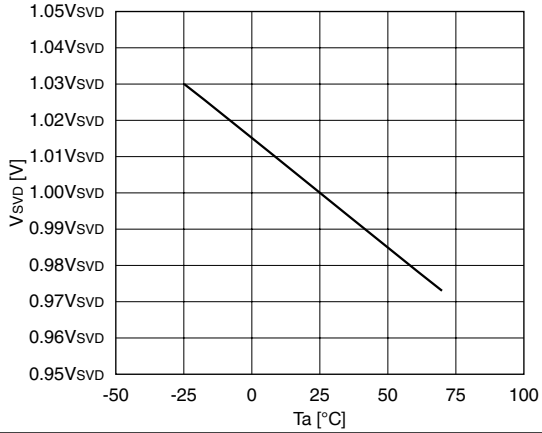
Unless otherwise specified: V_{DD} = 1.8 to 3.6V, V_{SS} = 0V, Ta = 25°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
SVD voltage	V_{SVD}	SVDC[3:0] = 0x0		–		V
		SVDC[3:0] = 0x1	Typ. × 0.96	1.8	Typ. × 1.04	V
		SVDC[3:0] = 0x2		1.9		V
		SVDC[3:0] = 0x3		2.0		V
		SVDC[3:0] = 0x4		2.1		V
		SVDC[3:0] = 0x5		2.2		V
		SVDC[3:0] = 0x6		2.3		V
		SVDC[3:0] = 0x7		2.4		V
		SVDC[3:0] = 0x8		2.5		V
		SVDC[3:0] = 0x9		2.6		V
		SVDC[3:0] = 0xa		2.7		V
		SVDC[3:0] = 0xb		2.8		V
		SVDC[3:0] = 0xc		2.9		V
		SVDC[3:0] = 0xd		3.0		V
		SVDC[3:0] = 0xe		3.1		V
		SVDC[3:0] = 0xf		3.2		V
SVD circuit-enable response time *1	t_{SVDEN}				500	µs
SVD circuit response time *2	t_{SVD}				60	µs

*1 This time is required to obtain stable detection results after SVDEN is altered from 0 to 1.
*2 This time is required to obtain stable detection results after SVDC[3:0] is altered.

SVD voltage-ambient temperature characteristic

SVDC[3:0] = 0xf, Typ. value



SVD circuit current consumption

Unless otherwise specified: V_{DD} = 1.8 to 3.6V, V_{SS} = 0V, Ta = 25°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
SVD circuit current *1	I_{SVD}	V_{DD} = 3.6V, SVDC[3:0] = 0x1		8	15	µA

*1 This value is added to the current consumption during execution or current consumption during execution in heavy load protection mode when the SVD circuit is active.

29.13 Flash Memory Characteristics

Analog characteristics

Unless otherwise specified: V_{DD} = 2.7 to 3.6V (VD1MD = 1), V_{SS} = 0V, Ta = -25 to 70°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Erase time *1	t_{SE}	Erase 4K bytes			25	ms
Programming time *1	t_{BP}	Program 16 bits			20	µs
Erase/program count *2	C_{FEP}		1000			times

*1 Data transfer and data verification are included and erase/program start control time is not included.

*2 The erase/program count assumes that “erasing + programming” or “overwrite programming” is one count and the programmed data is guaranteed to be retained for 10 years.

Flash memory current consumption

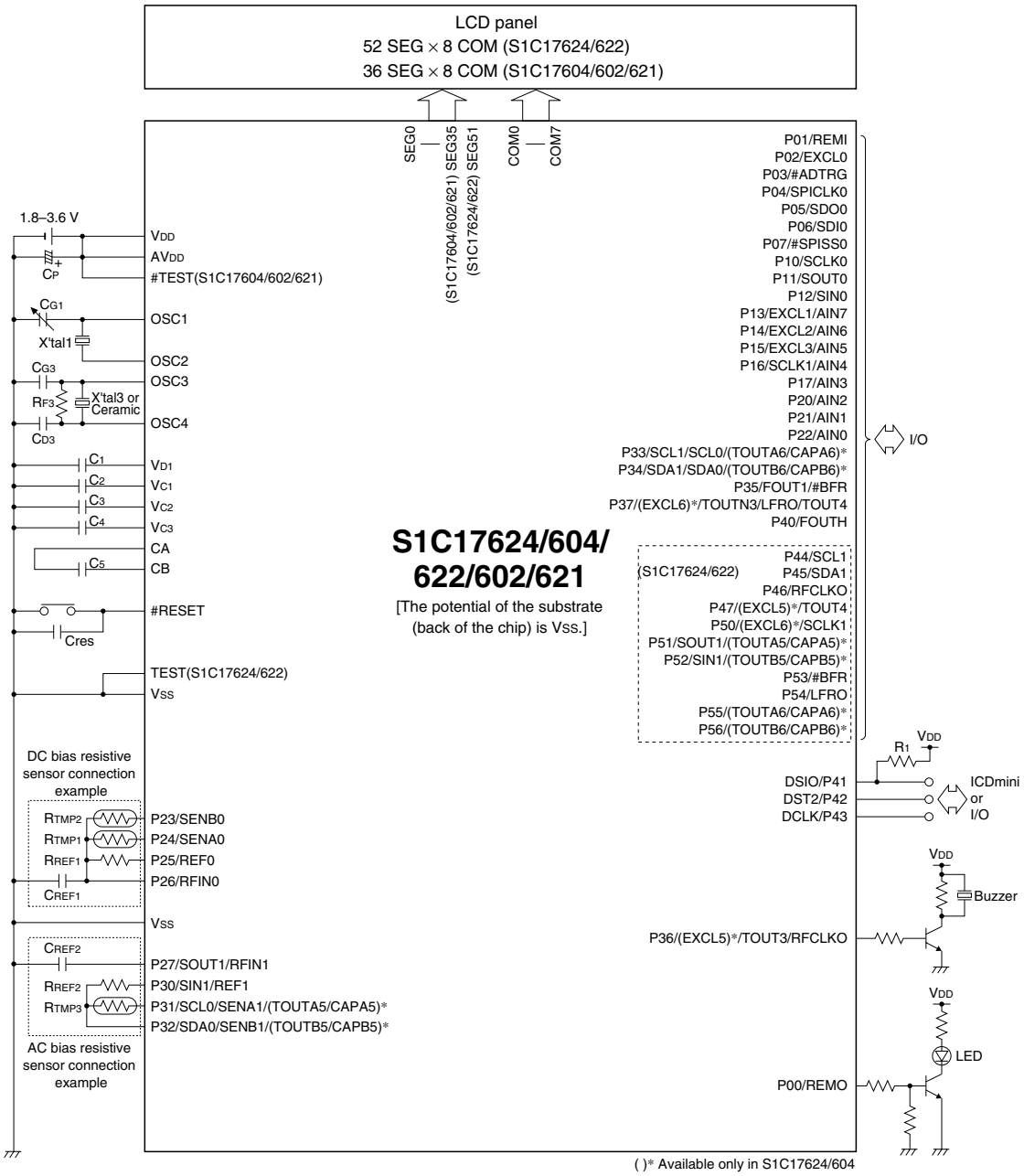
Unless otherwise specified: V_{DD} = 2.7 to 3.6V (VD1MD = 1), V_{SS} = 0V, Ta = 25°C, FLCYC[2:0] = 0x4 (1 cycle), CCLKGR[1:0] = 0x0 (gear ratio 1/1)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Flash memory erasing current *1	I_{FERS}	When CPU runs with 8MHz, VD1MD = 1		7	14	mA
Flash memory programming current *2	I_{FPRG}	When CPU runs with 8MHz, VD1MD = 1		7	14	mA

*1 This value is added to the current consumption during execution when the Flash memory is being erased in self-programming mode.

*2 This value is added to the current consumption during execution when the Flash memory is being programmed in self-programming mode.

30 Basic External Connection Diagram



( )* Available only in S1C17624/604

Example of external parts

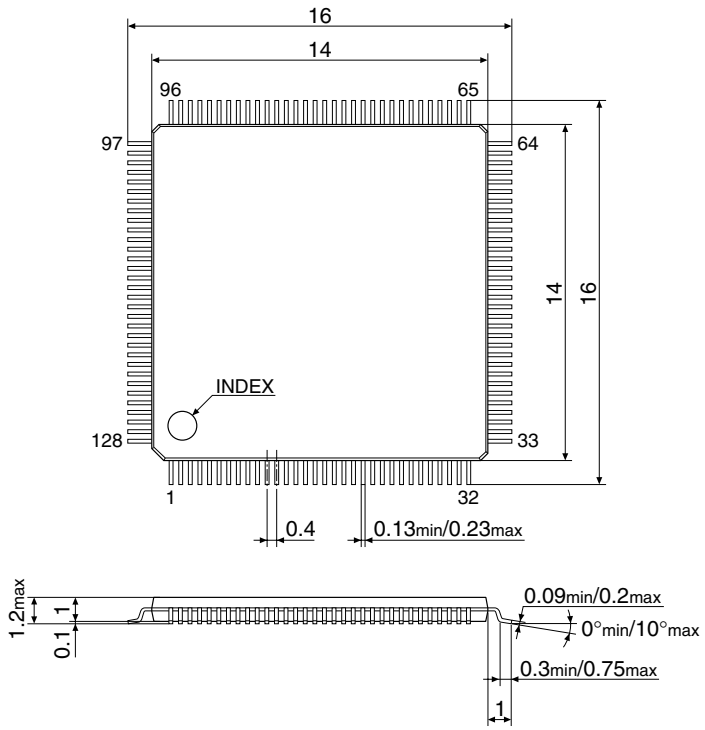
Symbol	Name	Recommended value
X'tal1	Crystal resonator	32.768 kHz
C_{G1}	Trimmer capacitor or fixed capacitor	0 to 25 pF
X'tal3	Crystal oscillator	0.2 to 8 MHz
Ceramic	Ceramic resonator	0.2 to 8 MHz
C_{G3}	Gate capacitor	15 to 30 pF
C_{D3}	Drain capacitor	15 to 30 pF
R_{F3}	Feedback resistor	1 MΩ
Cres	Power-on reset capacitor	0.47 µF

Symbol	Name	Recommended value
C_P	Bypass capacitor	0.47 µF
C_1	V_{D1} stabilizing capacitor	0.1 µF
C_2–C_4	V_{C1}–V_{C3} stabilizing capacitors	0.1 µF
C_5	Booster capacitor	0.1 µF
R_1	DSIO pull-up resistor	10 kΩ
C_{REF1-2}	Reference capacitors	–
R_{REF1-2}	Reference resistors	–
R_{TMP1-3}	Resistive sensors	–

31 Package

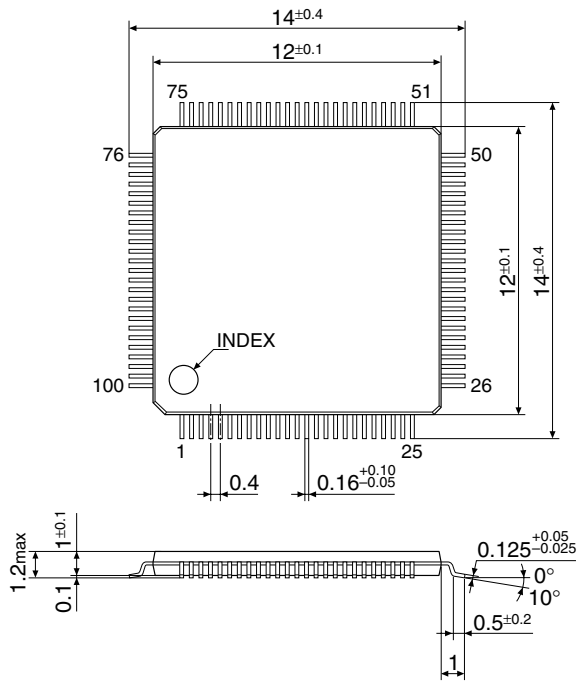
TQFP15-128pin package

(Unit: mm)



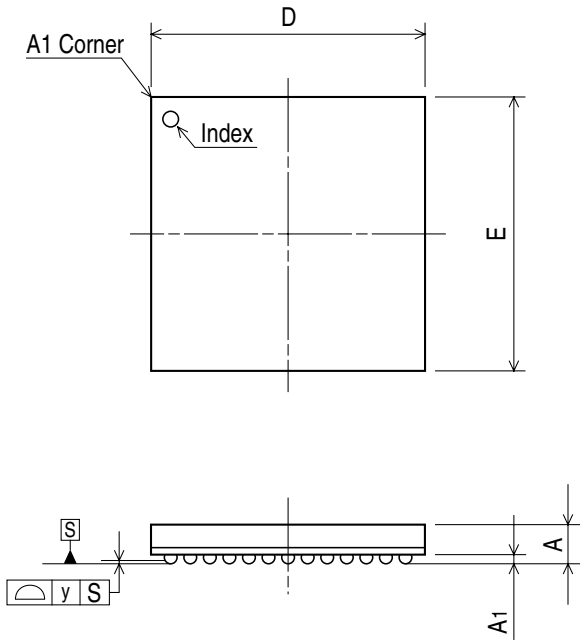
TQFP14-100pin package

(Unit: mm)

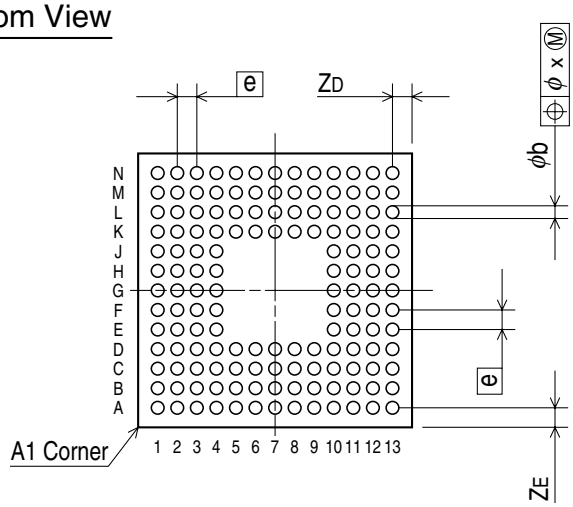


VFBGA7H-144 package

Top View



Bottom View



Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
D	–	7	–
E	–	7	–
A	–	–	1.0
A_1	–	0.23	–
e	–	0.5	–
b	0.26	–	0.36
X	–	–	0.08
y	–	–	0.1
Z_D	–	0.5	–
Z_E	–	0.5	–

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	PSC_CTL	Prescaler Control Register	Controls prescalers.
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.
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.
Fine mode 8-bit timer Ch. 0 (16-bit device)	0x4200	T8F_CLK0	T8F Ch.0 Count Clock Select Register	Selects a count clock.
	0x4202	T8F_TR0	T8F Ch.0 Reload Data Register	Sets reload data.
	0x4204	T8F_TC0	T8F Ch.0 Counter Data Register	Counter data
	0x4206	T8F_CTL0	T8F Ch.0 Control Register	Sets the timer mode and starts/stops the timer.
	0x4208	T8F_INT0	T8F 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 8-bit timer Ch. 1 (16-bit device)	0x4280	T8F_CLK1	T8F Ch.1 Count Clock Select Register	Selects a count clock.
	0x4282	T8F_TR1	T8F Ch.1 Reload Data Register	Sets reload data.
	0x4284	T8F_TC1	T8F Ch.1 Counter Data Register	Counter data
	0x4286	T8F_CTL1	T8F Ch.1 Control Register	Sets the timer mode and starts/stops the timer.
	0x4288	T8F_INT1	T8F 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/RTC interrupt levels.
	0x430a	ITC_LV2	Interrupt Level Setup Register 2	Sets the T8OSC1 and SVD interrupt levels.
	0x430c	ITC_LV3	Interrupt Level Setup Register 3	Sets the LCD/T16A2 Ch.0 and T16E Ch.0 interrupt levels.
	0x430e	ITC_LV4	Interrupt Level Setup Register 4	Sets the T8F Ch.0&1 and T16 Ch.0 interrupt levels.
	0x4310	ITC_LV5	Interrupt Level Setup Register 5	Sets the T16 Ch.1 and T16 Ch.2 interrupt levels.
	0x4312	ITC_LV6	Interrupt Level Setup Register 6	Sets the UART Ch.0 and I2CS/UART 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 and T16A2 Ch.1 interrupt levels.
	0x4318	ITC_LV9	Interrupt Level Setup Register 9	Sets the ADC10 and RFC interrupt levels.
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.

Peripheral	Address	Register name		Function
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.

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) (LCD, T8OSC1, SVD, RFC, T16A2, RTC)	0x5060	OSC_SRC	Clock Source Select Register	Selects the clock source.
	0x5061	OSC_CTL	Oscillation Control Register	Controls oscillation.
	0x5062	OSC_NFEN	Noise Filter Enable Register	Enables/disables noise filters.
	0x5063	OSC_LCLK	LCD Clock Select Register	Selects the LCD clock.
	0x5064	OSC_FOUT	FOUT Control Register	Controls FOUTH/FOUT1 clock outputs.
	0x5065	OSC_T8OSC1	T8OSC1 Clock Control Register	Controls the count clock.
	0x5066	OSC_SVD	SVD Clock Control Register	Selects the SVD operating clock.
	0x5067	OSC_RFC	RFC Clock Control Register	Selects the RFC operating clock.
	0x5068	T16A_CLK0	T16A Clock Control Register Ch.0	Controls the T16A2 Ch.0 clock. (S1C17624/604)
	0x5069	T16A_CLK1	T16A Clock Control Register Ch.1	Controls the T16A2 Ch.1 clock. (S1C17624/604)
	0x506e	RTC_CC	RTC Clock Control Register	Controls the RTC clock source.
	0x5080	CLG_PCLK	PCLK Control Register	Controls the PCLK supply.
	0x5081	CLG_CCLK	CCLK Control Register	Configures the CCLK division ratio.
LCD driver (8-bit device)	0x50a0	LCD_DCTL	LCD Display Control Register	Controls the LCD display.
	0x50a1	LCD_CADJ	LCD Contrast Adjustment Register	Controls the contrast.
	0x50a2	LCD_CCTL	LCD Clock Control Register	Controls the LCD drive duty.
	0x50a3	LCD_VREG	LCD Voltage Regulator Control Register	Controls the LCD drive voltage regulator.
	0x50a5	LCD_IMSK	LCD Interrupt Mask Register	Enables/disables interrupts.
	0x50a6	LCD_IFLG	LCD Interrupt Flag Register	Indicates/resets interrupt occurrence status.
8-bit OSC1 timer (8-bit device)	0x50c0	T8OSC1_CTL	T8OSC1 Control Register	Sets the timer mode and starts/stops the timer.
	0x50c1	T8OSC1_CNT	T8OSC1 Counter Data Register	Counter data
	0x50c2	T8OSC1_CMP	T8OSC1 Compare Data Register	Sets compare data.
	0x50c3	T8OSC1_IMSK	T8OSC1 Interrupt Mask Register	Sets the interrupt mask.
	0x50c4	T8OSC1_IFLG	T8OSC1 Interrupt Flag Register	Indicates and reset interrupt occurrence status.
	0x50c5	T8OSC1_DUTY	T8OSC1 PWM Duty Data Register	Sets data for PWM output.
SVD circuit (8-bit device)	0x5100	SVD_EN	SVD Enable Register	Enables/disables the SVD operation.
	0x5101	SVD_CMP	SVD Comparison Voltage Register	Sets the comparison voltage.
	0x5102	SVD_RSLT	SVD Detection Result Register	Voltage detection results
	0x5103	SVD_IMSK	SVD Interrupt Mask Register	Enables/disables interrupts.
	0x5104	SVD_IFLG	SVD Interrupt Flag Register	Indicates/resets interrupt occurrence status.
Power generator (8-bit device)	0x5120	VD1_CTL	V_{D1} Control Register	Controls the V_{D1} voltage and heavy load protection mode.
Real-time clock (8-bit device) (S1C17624/604)	0x5140	RTC_INTSTAT	RTC Interrupt Status Register	Indicates RTC interrupt status.
	0x5141	RTC_INTMODE	RTC Interrupt Mode Register	Sets up RTC interrupt modes.
	0x5142	RTC_CNTL0	RTC Control 0 Register	Controls the RTC.
	0x5143	RTC_CNTL1	RTC Control 1 Register	
	0x5144	RTC_SEC	RTC Second Register	Second counter data
	0x5145	RTC_MIN	RTC Minute Register	Minute counter data
	0x5146	RTC_HOUR	RTC Hour Register	Hour counter data
	0x5147	RTC_DAY	RTC Day Register	Day counter data
	0x5148	RTC_MONTH	RTC Month Register	Month counter data
	0x5149	RTC_YEAR	RTC Year Register	Year counter data
	0x514a	RTC_WEEK	RTC Days of Week Register	Days of week counter data

Peripheral	Address	Register name		Function
P port & port MUX (8-bit device)	0x5200	P0_IN	P0 Port Input Data Register	P0 port input data
	0x5201	P0_OUT	P0 Port Output Data Register	P0 port output data
	0x5202	P0_OEN	P0 Port Output Enable Register	Enables P0 port outputs.
	0x5203	P0_PU	P0 Port Pull-up Control Register	Controls the P0 port pull-up resistor.
	0x5204	P0_SM	P0 Port Schmitt Trigger Control Register	Controls the P0 port Schmitt trigger input.
	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.
	0x5214	P1_SM	P1 Port Schmitt Trigger Control Register	Controls the P1 port Schmitt trigger input.
	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.
	0x5224	P2_SM	P2 Port Schmitt Trigger Control Register	Controls the P2 port Schmitt trigger input.
	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.
	0x5234	P3_SM	P3 Port Schmitt Trigger Control Register	Controls the P3 port Schmitt trigger input.
	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.
	0x5244	P4_SM	P4 Port Schmitt Trigger Control Register	Controls the P4 port Schmitt trigger input.
	0x524a	P4_IEN	P4 Port Input Enable Register	Enables P4 port inputs.
	0x5250	P5_IN	P5 Port Input Data Register	P5 port input data (S1C17624/622)
	0x5251	P5_OUT	P5 Port Output Data Register	P5 port output data (S1C17624/622)
	0x5252	P5_OEN	P5 Port Output Enable Register	Enables P5 port outputs. (S1C17624/622)
	0x5253	P5_PU	P5 Port Pull-up Control Register	Controls the P5 port pull-up resistor. (S1C17624/622)
	0x5254	P5_SM	P5 Port Schmitt Trigger Control Register	Controls the P5 port Schmitt trigger input. (S1C17624/622)
	0x525a	P5_IEN	P5 Port Input Enable Register	Enables P5 port inputs. (S1C17624/622)
	0x52a0	P00_03PMUX	P0[3:0] Port Function Select Register	Selects the P0[3:0] port functions.
	0x52a1	P04_07PMUX	P0[7:4] Port Function Select Register	Selects the P0[7:4] port functions.
	0x52a2	P10_13PMUX	P1[3:0] Port Function Select Register	Selects the P1[3:0] port functions.
	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_47PMUX	P4[7:4] Port Function Select Register	Selects the P4[7:4] port functions. (S1C17624/622)
	0x52aa	P50_53PMUX	P5[3:0] Port Function Select Register	Selects the P5[3:0] port functions. (S1C17624/622)
	0x52ab	P54_56PMUX	P5[6:4] Port Function Select Register	Selects the P5[6:4] port functions. (S1C17624/622)

Peripheral	Address	Register name		Function
16-bit PWM timer (T16E) Ch.0 (16-bit device)	0x5300	T16E_CA0	T16E Ch.0 Compare Data A Register	Sets compare data A.
	0x5302	T16E_CB0	T16E Ch.0 Compare Data B Register	Sets compare data B.
	0x5304	T16E_TC0	T16E Ch.0 Counter Data Register	Counter data
	0x5306	T16E_CTL0	T16E Ch.0 Control Register	Sets the timer mode and starts/stops the timer.
	0x5308	T16E_DF0	T16E Ch.0 Clock Division Ratio Select Register	Selects the count clock.
	0x530a	T16E_IMSK0	T16E Ch.0 Interrupt Mask Register	Sets the interrupt mask.
	0x530c	T16E_IFLG0	T16E Ch.0 Interrupt Flag Register	Indicates and reset interrupt occurrence status.
MISC registers (16-bit device)	0x5320	MISC_FL	FLASHC Control Register	Sets FLASHC access condition.
	0x5322	MISC_OSC1	OSC1 Peripheral Control Register	Enables peripheral operations in debug mode (except PCLK).
	0x5324	MISC_PROT	MISC Protect Register	Enables writing to the MISC registers.
	0x5326	MISC_IRAMSZ	IRAM Size Select Register	Selects the IRAM size.
	0x5328	MISC_TTBRL	Vector Table Address Low Register	Sets vector table address.
	0x532a	MISC_TTBRH	Vector Table Address High Register	
	0x532c	MISC_PSR	PSR Register	Indicates the S1C17 Core PSR values.
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	ADC_DIV	A/D Clock Control Register	Controls A/D converter clock.
R/F converter (16-bit device)	0x53a0	RFC_CTL	RFC Control Register	Controls R/F converter.
	0x53a2	RFC_TRG	RFC Oscillation Trigger Register	Controls oscillations.
	0x53a4	RFC_MCL	RFC Measurement Counter Low Register	Measurement counter data
	0x53a6	RFC_MCH	RFC Measurement Counter High Register	
	0x53a8	RFC_TCL	RFC Time Base Counter Low Register	Time base counter data
	0x53aa	RFC_TCH	RFC Time Base Counter High Register	
	0x53ac	RFC_IMSK	RFC Interrupt Mask Register	Enables/disables interrupts.
	0x53ae	RFC_IFLG	RFC Interrupt Flag Register	Indicates/resets interrupt occurrence status.
16-bit PWM timer (T16A2) Ch.0 (16-bit device) (S1C17624/604)	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.
16-bit PWM timer (T16A2) Ch.1 (16-bit device) (S1C17624/604)	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.

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.

0x4100–0x4105 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	–	reserved	–	–	–	0 when being read.
		D6	FER	Framing error flag	1 Error 0 Normal	0	R/W	Reset by writing 1.
		D5	PER	Parity error flag	1 Error 0 Normal	0	R/W	
		D4	OER	Overrun error flag	1 Error 0 Normal	0	R/W	
		D3	RD2B	Second byte receive flag	1 Ready 0 Empty	0	R	
		D2	TRBS	Transmit busy flag	1 Busy 0 Idle	0	R	Shift register status
		D1	RDRY	Receive data ready flag	1 Ready 0 Empty	0	R	
		D0	TDBE	Transmit data buffer empty flag	1 Empty 0 Not empty	1	R	
UART Ch.0 Transmit Data Register (UART_TXD0)	0x4101 (8 bits)	D7–0	TXD[7:0]	Transmit data TXD7(6) = MSB TXD0 = LSB	0x0 to 0xff (0x7f)	0x0	R/W	
UART Ch.0 Receive Data Register (UART_RXD0)	0x4102 (8 bits)	D7–0	RXD[7:0]	Receive data in the receive data buffer RXD7(6) = MSB RXD0 = LSB	0x0 to 0xff (0x7f)	0x0	R	Older data in the buffer is read out first.
UART Ch.0 Mode Register (UART_MOD0)	0x4103 (8 bits)	D7–5	–	reserved	–	–	–	0 when being read.
		D4	CHLN	Character length 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	SSCK	Input clock select	1 External 0 Internal	0	R/W	
UART Ch.0 Control Register (UART_CTL0)	0x4104 (8 bits)	D7	–	reserved	–	–	–	0 when being read.
		D6	REIEN	Receive error int. enable	1 Enable 0 Disable	0	R/W	
		D5	RIEN	Receive buffer full int. enable	1 Enable 0 Disable	0	R/W	
		D4	TIEN	Transmit buffer empty int. enable	1 Enable 0 Disable	0	R/W	
		D3–2	–	reserved	–	–	–	0 when being read.
		D1	RBFI	Receive buffer full int. condition 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	–	reserved	–	–	–	0 when being read.
		D6–4	IRCLK[2:0]	IrDA receive detection clock division ratio select	IRCLK[2:0] / Division ratio: 0x7 1/128 0x6 1/64 0x5 1/32 0x4 1/16 0x3 1/8 0x2 1/4 0x1 1/2 0x0 1/1	0x0	R/W	Source clock = PCLK
		D3–1	–	reserved	–	–	–	0 when being read.
		D0	IRMD	IrDA mode select	1 On 0 Off	0	R/W	

0x4120–0x4125 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	–	reserved	–	–	–	0 when being read.
		D6	FER	Framing error flag	1 Error 0 Normal	0	R/W	Reset by writing 1.
		D5	PER	Parity error flag	1 Error 0 Normal	0	R/W	
		D4	OER	Overrun error flag	1 Error 0 Normal	0	R/W	
		D3	RD2B	Second byte receive flag	1 Ready 0 Empty	0	R	
		D2	TRBS	Transmit busy flag	1 Busy 0 Idle	0	R	Shift register status
		D1	RDRY	Receive data ready flag	1 Ready 0 Empty	0	R	
		D0	TDBE	Transmit data buffer empty flag	1 Empty 0 Not empty	1	R	
UART Ch.1 Transmit Data Register (UART_TXD1)	0x4121 (8 bits)	D7–0	TXD[7:0]	Transmit data TXD7(6) = MSB TXD0 = LSB	0x0 to 0xff (0x7f)	0x0	R/W	
UART Ch.1 Receive Data Register (UART_RXD1)	0x4122 (8 bits)	D7–0	RXD[7:0]	Receive data in the receive data buffer RXD7(6) = MSB RXD0 = LSB	0x0 to 0xff (0x7f)	0x0	R	Older data in the buffer is read out first.
UART Ch.1 Mode Register (UART_MOD1)	0x4123 (8 bits)	D7–5	–	reserved	–	–	–	0 when being read.
		D4	CHLN	Character length 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	SSCK	Input clock select	1 External 0 Internal	0	R/W	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
UART Ch.1 Control Register (UART_CTL1)	**0x4124** (8 bits)	D7	**–**	reserved	–	–	–	0 when being read.
		D6	**REIEN**	Receive error int. enable	1 Enable 0 Disable	0	R/W	
		D5	**RIEN**	Receive buffer full int. enable	1 Enable 0 Disable	0	R/W	
		D4	**TIEN**	Transmit buffer empty int. enable	1 Enable 0 Disable	0	R/W	
		D3–2	**–**	reserved	–	–	–	0 when being read.
		D1	**RBFI**	Receive buffer full int. condition setup	1 2 bytes 0 1 byte	0	R/W	
		D0	**RXEN**	UART enable	1 Enable 0 Disable	0	R/W	
UART Ch.1 Expansion Register (UART_EXP1)	**0x4125** (8 bits)	D7	**–**	reserved	–	–	–	0 when being read.
		D6–4	**IRCLK[2:0]**	IrDA receive detection clock division ratio select	IRCLK[2:0] / Division ratio 0x7 1/128 0x6 1/64 0x5 1/32 0x4 1/16 0x3 1/8 0x2 1/4 0x1 1/2 0x0 1/1	0x0	R/W	Source clock = PCLK
		D3–1	**–**	reserved	–	–	–	0 when being read.
		D0	**IRMD**	IrDA mode select	1 On 0 Off	0	R/W	

0x4200–0x4208 Fine Mode 8-bit Timer Ch.0

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T8F Ch.0 Count Clock Select Register (T8F_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
T8F Ch.0 Reload Data Register (T8F_TR0)	**0x4202** (16 bits)	D15–8	**–**	reserved	–	–	–	0 when being read.
		D7–0	**TR[7:0]**	Reload data TR7 = MSB TR0 = LSB	0x0 to 0xff	0x0	R/W	
T8F Ch.0 Counter Data Register (T8F_TC0)	**0x4204** (16 bits)	D15–8	**–**	reserved	–	–	–	0 when being read.
		D7–0	**TC[7:0]**	Counter data TC7 = MSB TC0 = LSB	0x0 to 0xff	0xff	R	
T8F Ch.0 Control Register (T8F_CTL0)	**0x4206** (16 bits)	D15–12	**–**	reserved	–	–	–	0 when being read.
		D11–8	**TFMD[3:0]**	Fine mode setup	0x0 to 0xf	0x0	R/W	Set a number of times to insert delay into a 16-underflow period.
		D7–5	**–**	reserved	–	–	–	0 when being read.
		D4	**TRMD**	Count mode select	1 One shot 0 Repeat	0	R/W	
		D3–2	**–**	reserved	–	–	–	0 when being read.
		D1	**PRESER**	Timer reset	1 Reset 0 Ignored	0	W	
		D0	**PRUN**	Timer run/stop control	1 Run 0 Stop	0	R/W	
T8F Ch.0 Interrupt Control Register (T8F_INT0)	**0x4208** (16 bits)	D15–9	**–**	reserved	–	–	–	0 when being read.
		D8	**T8FIE**	T8F interrupt enable	1 Enable 0 Disable	0	R/W	
		D7–1	**–**	reserved	–	–	–	0 when being read.
		D0	**T8FIF**	T8F 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)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10	**CKACTV**	External clock active level select	1 High / 0 Low	1	R/W	
		D9–8	**CKSL[1:0]**	Operating mode select	CKSL[1:0] / Mode: 0x3 reserved 0x2 Pulse width 0x1 External clock 0x0 Internal clock	0x0	R/W	
		D7–5	**–**	reserved	–	–	–	0 when being read.
		D4	**TRMD**	Count mode select	1 One shot / 0 Repeat	0	R/W	
		D3–2	**–**	reserved	–	–	–	0 when being read.
		D1	**PRESER**	Timer reset	1 Reset / 0 Ignored	0	W	
		D0	**PRUN**	Timer run/stop control	1 Run / 0 Stop	0	R/W	
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	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16 Ch.1 Control Register (T16_CTL1)	**0x4246** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10	**CKACTV**	External clock active level select	1 High / 0 Low	1	R/W	
		D9–8	**CKSL[1:0]**	Operating mode select	CKSL[1:0] / Mode: 0x3 reserved; 0x2 Pulse width; 0x1 External clock; 0x0 Internal clock	0x0	R/W	
		D7–5	**–**	reserved	–	–	–	0 when being read.
		D4	**TRMD**	Count mode select	1 One shot / 0 Repeat	0	R/W	
		D3–2	**–**	reserved	–	–	–	0 when being read.
		D1	**PRESER**	Timer reset	1 Reset / 0 Ignored	0	W	
		D0	**PRUN**	Timer run/stop control	1 Run / 0 Stop	0	R/W	
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)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10	**CKACTV**	External clock active level select	1 High / 0 Low	1	R/W	
		D9–8	**CKSL[1:0]**	Operating mode select	CKSL[1:0] / Mode: 0x3 reserved; 0x2 Pulse width; 0x1 External clock; 0x0 Internal clock	0x0	R/W	
		D7–5	**–**	reserved	–	–	–	0 when being read.
		D4	**TRMD**	Count mode select	1 One shot / 0 Repeat	0	R/W	
		D3–2	**–**	reserved	–	–	–	0 when being read.
		D1	**PRESER**	Timer reset	1 Reset / 0 Ignored	0	W	
		D0	**PRUN**	Timer run/stop control	1 Run / 0 Stop	0	R/W	
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 8-bit Timer Ch.1

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T8F Ch.1 Count Clock Select Register (T8F_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
T8F Ch.1 Reload Data Register (T8F_TR1)	**0x4282** (16 bits)	D15–8	**–**	reserved	–	–	–	0 when being read.
		D7–0	**TR[7:0]**	Reload data TR7 = MSB TR0 = LSB	0x0 to 0xff	0x0	R/W	
T8F Ch.1 Counter Data Register (T8F_TC1)	**0x4284** (16 bits)	D15–8	**–**	reserved	–	–	–	0 when being read.
		D7–0	**TC[7:0]**	Counter data TC7 = MSB TC0 = LSB	0x0 to 0xff	0xff	R	
T8F Ch.1 Control Register (T8F_CTL1)	**0x4286** (16 bits)	D15–12	**–**	reserved	–	–	–	0 when being read.
		D11–8	**TFMD[3:0]**	Fine mode setup	0x0 to 0xf	0x0	R/W	Set a number of times to insert delay into a 16-underflow period.
		D7–5	**–**	reserved	–	–	–	0 when being read.
		D4	**TRMD**	Count mode select	1 One shot 0 Repeat	0	R/W	
		D3–2	**–**	reserved	–	–	–	0 when being read.
		D1	**PRESER**	Timer reset	1 Reset 0 Ignored	0	W	
		D0	**PRUN**	Timer run/stop control	1 Run 0 Stop	0	R/W	
T8F Ch.1 Interrupt Control Register (T8F_INT1)	**0x4288** (16 bits)	D15–9	**–**	reserved	–	–	–	0 when being read.
		D8	**T8FIE**	T8F interrupt enable	1 Enable 0 Disable	0	R/W	
		D7–1	**–**	reserved	–	–	–	0 when being read.
		D0	**T8FIF**	T8F interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.

0x4306–0x4318 Interrupt Controller

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Interrupt Level Setup Register 0 (ITC_LV0)	**0x4306** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**ILV1[2:0]**	P1 interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**ILV0[2:0]**	P0 interrupt level	0 to 7	0x0	R/W	
Interrupt Level Setup Register 1 (ITC_LV1)	**0x4308** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**ILV3[2:0]**	CT/RTC 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)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**ILV5[2:0]**	SVD interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**ILV4[2:0]**	T8OSC1 interrupt level	0 to 7	0x0	R/W	
Interrupt Level Setup Register 3 (ITC_LV3)	**0x430c** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**ILV7[2:0]**	T16E Ch.0 interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**ILV6[2:0]**	LCD/T16A2 Ch.0 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]**	T8F Ch.0/Ch.1 interrupt level	0 to 7	0x0	R/W	
Interrupt Level Setup Register 5 (ITC_LV5)	**0x4310** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**ILV11[2:0]**	T16 Ch.2 interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**ILV10[2:0]**	T16 Ch.1 interrupt level	0 to 7	0x0	R/W	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Interrupt Level Setup Register 6 (ITC_LV6)	**0x4312** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**ILV13[2:0]**	I2CS/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]**	T16A2 Ch.1 interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**ILV16[2:0]**	REMC interrupt level	0 to 7	0x0	R/W	
Interrupt Level Setup Register 9 (ITC_LV9)	**0x4318** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**ILV19[2:0]**	RFC 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	

0x4320–0x4326 SPI Ch.0

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SPI Ch.0 Status Register (SPI_ST0)	**0x4320** (16 bits)	D15–3	**–**	reserved	–	–	–	0 when being read.
		D2	**SPBSY**	Transfer busy flag (master)	1 Busy 0 Idle	0	R	
				ss signal low flag (slave)	1 ss = L 0 ss = H			
		D1	**SPRBF**	Receive data buffer full flag	1 Full 0 Not full	0	R	
		D0	**SPTBE**	Transmit data buffer empty flag	1 Empty 0 Not empty	1	R	
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	
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	

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	

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	
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 (OSC_SRC)	0x5060 (8 bits)	D7–2	–	reserved	–	–	–	0 when being read.
		D1	HSCLKSEL	High-speed clock select	1 OSC3 0 IOSC	0	R/W	
		D0	CLKSRC	System clock source select	1 OSC1 0 HSCLK	0	R/W	
Oscillation Control Register (OSC_CTL)	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 (OSC_NFEN)	0x5062 (8 bits)	D7–2	–	reserved	–	–	–	0 when being read.
		D1	RSTFE	Reset noise filter enable	1 Enable 0 Disable	1	R/W	
		D0	NMIFE	NMI noise filter enable	1 Enable 0 Disable	0	R/W	
FOUT Control Register (OSC_FOUT)	0x5064 (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.
		D3–2	FOUTHD [1:0]	FOUTH clock division ratio select	FOUTHD[1:0] / Division ratio: 0x3 reserved; 0x2 1/4; 0x1 1/2; 0x0 1/1	0x0	R/W	Source clock = HSCLK
		D1	FOUTHE	FOUTH output enable	1 Enable 0 Disable	0	R/W	
		D0	FOUT1E	FOUT1 output enable	1 Enable 0 Disable	0	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	

0x5063, 0x50a0–0x50a6 LCD Driver

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
LCD Clock Select Register (OSC_LCLK)	0x5063 (8 bits)	D7–5	–	reserved	–	–	–	0 when being read.
		D4–2	LCKDV[2:0]	LCD clock division ratio select	LCKDV[2:0] / Division ratio 0x7–0x5 / reserved 0x4 / 1/512 0x3 / 1/256 0x2 / 1/128 0x1 / 1/64 0x0 / 1/32	0x0	R/W	When the clock source is HSCLK
		D1	LCKSRC	LCD clock source select	1 OSC1 0 HSCLK	1	R/W	
		D0	LCKEN	LCD clock enable	1 Enable 0 Disable	0	R/W	
LCD Display Control Register (LCD_DCTL)	0x50a0 (8 bits)	D7	SEGREV	Segment output assignment control	1 Normal 0 Reverse	1	R/W	
		D6	COMREV	Common output assignment control	1 Normal 0 Reverse	1	R/W	
		D5	DSPAR	Display memory area control	1 Area 1 0 Area 0	0	R/W	
		D4	DSPREV	Reverse display control	1 Normal 0 Reverse	1	R/W	
		D3–2	–	reserved	–	–	–	0 when being read.
		D1–0	DSPC[1:0]	LCD display control	DSPC[1:0] / Display 0x3 / All off 0x2 / All on 0x1 / Normal display 0x0 / Display off	0x0	R/W	
LCD Contrast Adjustment Register (LCD_CADJ)	0x50a1 (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.
		D3–0	LC[3:0]	LCD contrast adjustment	LC[3:0] / Display 0xf / Dark : / : 0x0 / Light	0x7	R/W	
LCD Clock Control Register (LCD_CCTL)	0x50a2 (8 bits)	D7–6	FRMCNT[1:0]	Frame frequency control	FRMCNT[1:0] / Division ratio 0x3 / 1/1024 0x2 / 1/680 0x1 / 1/512 0x0 / 1/256	0x1	R/W	Source clock: LCLK
		D5	LFROUT	LFRO output control	1 On 0 Off	0	R/W	
		D4–3	–	reserved	–	–	–	0 when being read.
		D2–0	LDUTY[2:0]	LCD duty select	LDUTY[2:0] / Duty 0x7–0x5 / reserved 0x4 / 1/8 0x3 / 1/4 0x2 / 1/3 0x1 / 1/2 0x0 / Static	0x4	R/W	
LCD Voltage Regulator Control Register (LCD_VREG)	0x50a3 (8 bits)	D7–5	–	reserved	–	–	–	0 when being read.
		D4	LHVLD	LCD heavy load protection mode	1 On 0 Off	0	R/W	
		D3–1	–	reserved	–	–	–	0 when being read.
		D0	VCSEL	Vc reference voltage select	1 Vc2 0 Vc1	0	R/W	
LCD Interrupt Mask Register (LCD_IMSK)	0x50a5 (8 bits)	D7–1	–	reserved	–	–	–	0 when being read.
		D0	FRMIE	Frame signal interrupt enable	1 Enable 0 Disable	0	R/W	
LCD Interrupt Flag Register (LCD_IFLG)	0x50a6 (8 bits)	D7–1	–	reserved	–	–	–	0 when being read.
		D0	FRMIF	Frame signal interrupt flag	1 Occurred 0 Not occurred	0	R/W	Reset by writing 1.

0x5065, 0x50c0–0x50c5 8-bit OSC1 Timer

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T8OSC1 Clock Control Register (OSC_T8OSC1)	0x5065 (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.
		D3–1	T8O1CK [2:0]	T8OSC1 clock division ratio select	T8O1CK[2:0] / Division ratio 0x7–0x6 / reserved 0x5 / 1/32 0x4 / 1/16 0x3 / 1/8 0x2 / 1/4 0x1 / 1/2 0x0 / 1/1	0x0	R/W	Clock source: OSC1
		D0	T8O1CE	Clock enable	1 Enable 0 Disable	0	R/W	
T8OSC1 Control Register (T8OSC1_CTL)	0x50c0 (8 bits)	D7–5	–	reserved	–	–	–	0 when being read.
		D4	T8ORST	Timer reset	1 Reset 0 Ignored	0	W	
		D3–2	–	reserved	–	–	–	
		D1	T8ORMD	Count mode select	1 One shot 0 Repeat	0	R/W	
		D0	T8ORUN	Timer run/stop control	1 Run 0 Stop	0	R/W	
T8OSC1 Counter Data Register (T8OSC1_CNT)	0x50c1 (8 bits)	D7–0	T8OCNT[7:0]	Timer counter data T8OCNT7 = MSB T8OCNT0 = LSB	0x0 to 0xff	0x0	R	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T8OSC1 Compare Data Register (T8OSC1_CMP)	**0x50c2** (8 bits)	D7–0	**T8OCMP[7:0]**	Compare data T8OCMP7 = MSB T8OCMP0 = LSB	0x0 to 0xff	0x0	R/W	
T8OSC1 Interrupt Mask Register (T8OSC1_IMSK)	**0x50c3** (8 bits)	D7–1	**–**	reserved	–	–	–	0 when being read.
		D0	**T8OIE**	T8OSC1 interrupt enable	1 Enable 0 Disable	0	R/W	
T8OSC1 Interrupt Flag Register (T8OSC1_IFLG)	**0x50c4** (8 bits)	D7–1	**–**	reserved	–	–	–	0 when being read.
		D0	**T8OIF**	T8OSC1 interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
T8OSC1 PWM Duty Data Register (T8OSC1_DUTY)	**0x50c5** (8 bits)	D7–0	**T8ODTY[7:0]**	PWM output duty data T8ODTY7 = MSB T8ODTY0 = LSB	0x0 to 0xff	0x0	R/W	

0x5066, 0x5100–0x5104 SVD Circuit

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SVD Clock Control Register (OSC_SVD)	**0x5066** (8 bits)	D7–2	**–**	reserved	–	–	–	0 when being read.
		D1	**SVDSRC**	SVD clock source select	1 OSC1 0 HSCLK/512	1	R/W	
		D0	**SVDCKEN**	SVD clock enable	1 Enable 0 Disable	0	R/W	
SVD Enable Register (SVD_EN)	**0x5100** (8 bits)	D7–1	**–**	reserved	–	–	–	0 when being read.
		D0	**SVDEN**	SVD enable	1 Enable 0 Disable	0	R/W	
SVD Comparison Voltage Register (SVD_CMP)	**0x5101** (8 bits)	D7–4	**–**	reserved	–	–	–	0 when being read.
		D3–0	**SVDC[3:0]**	SVD comparison voltage select	SVDC[3:0] / Voltage: 0xf 3.2 V 0xe 3.1 V 0xd 3.0 V 0xc 2.9 V 0xb 2.8 V 0xa 2.7 V 0x9 2.6 V 0x8 2.5 V 0x7 2.4 V 0x6 2.3 V 0x5 2.2 V 0x4 2.1 V 0x3 2.0 V 0x2 1.9 V 0x1 1.8 V 0x0 reserved	0x0	R/W	
SVD Detection Result Register (SVD_RSLT)	**0x5102** (8 bits)	D7–1	**–**	reserved	–	–	–	0 when being read.
		D0	**SVDDT**	SVD detection result	1 Low 0 Normal	×	R	
SVD Interrupt Mask Register (SVD_IMSK)	**0x5103** (8 bits)	D7–1	**–**	reserved	–	–	–	0 when being read.
		D0	**SVDIE**	SVD interrupt enable	1 Enable 0 Disable	0	R/W	
SVD Interrupt Flag Register (SVD_IFLG)	**0x5104** (8 bits)	D7–1	**–**	reserved	–	–	–	0 when being read.
		D0	**SVDIF**	SVD interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.

0x5120 Power Generator

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
V_{D1} Control Register (VD1_CTL)	**0x5120** (8 bits)	D7–6	**–**	reserved	–	–	–	0 when being read.
		D5	**HVLD**	V_{D1} heavy load protection mode	1 On 0 Off	0	R/W	
		D4	**–**	reserved	–	0	R/W	
		D3–1	**–**	reserved	–	–	–	0 when being read.
		D0	**VD1MD**	Flash erase/programming mode	1 Flash (2.5 V) 0 Norm.(1.8 V)	0	R/W	

0x506e, 0x5140–0x514a Real-time Clock (S1C17624/604)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RTC Clock Control Register (RTC_CC)	**0x506e** (8 bits)	D7–1	**–**	reserved	–	–	–	0 when being read.
		D0	**RTCCE**	RTC clock enable	1 Enable 0 Disable	0	R/W	
RTC Interrupt Status Register (RTC_INTSTAT)	**0x5140** (8 bits)	D7–1	**–**	reserved	–	–	–	0 when being read.
		D0	**RTCIRQ**	Interrupt status	1 Occurred 0 Not occurred	X (0)	R/W	Reset by writing 1.

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RTC Interrupt Mode Register (RTC_INTMODE)	**0x5141** (8 bits)	D7–5	**–**	reserved	–	–	–	0 when being read.
		D4–2	**RTCT[2:0]**	RTC interrupt cycle setup	RTCT[2:0] / Cycle 0x7 reserved 0x6 1/128 second 0x5 1/256 second 0x4 1/512 second 0x3 1 hour 0x2 1 minute 0x1 1 second 0x0 1/64 second	X (0x1)	R/W	
		D1	**RTCIMD**	RTC interrupt mode select	1 Level sense / 0 Edge trigger	X (1)	R/W	
		D0	**RTCIEN**	RTC interrupt enable	1 Enable / 0 Disable	X (0)	R/W	
RTC Control 0 Register (RTC_CNTL0)	**0x5142** (8 bits)	D7–5	**–**	reserved	–	–	–	0 when being read.
		D4	**RTC24H**	24H/12H mode select	1 24H / 0 12H	X (0)	R/W	
		D3	**–**	reserved	–	–	–	0 when being read.
		D2	**RTCADJ**	30-second adjustment	1 Adjust / 0 –	X (0)	R/W	
		D1	**RTCSTP**	Divider run/stop control	1 Stop / 0 Run	X (0)	R/W	
		D0	**RTCRST**	Software reset	1 Reset / 0 –	X (0)	R/W	
RTC Control 1 Register (RTC_CNTL1)	**0x5143** (8 bits)	D7–3	**–**	reserved	–	–	–	0 when being read.
		D2	**RTCRDHLD**	Read buffer enable	1 Enable / 0 Disable	X (0)	R/W	
		D1	**RTCBSY**	Counter busy flag	1 Busy / 0 R/W possible	X (0)	R	
		D0	**RTCHLD**	Counter hold control	1 Hold / 0 Running	X (0)	R/W	
RTC Second Register (RTC_SEC)	**0x5144** (8 bits)	D7	**–**	reserved	–	–	–	0 when being read.
		D6–4	**RTCSH[2:0]**	RTC 10-second counter	0 to 5	X (*)	R/W	
		D3–0	**RTCSL[3:0]**	RTC 1-second counter	0 to 9	X (*)	R/W	
RTC Minute Register (RTC_MIN)	**0x5145** (8 bits)	D7	**–**	reserved	–	–	–	0 when being read.
		D6–4	**RTCMIH[2:0]**	RTC 10-minute counter	0 to 5	X (*)	R/W	
		D3–0	**RTCMIL[3:0]**	RTC 1-minute counter	0 to 9	X (*)	R/W	
RTC Hour Register (RTC_HOUR)	**0x5146** (8 bits)	D7	**–**	reserved	–	–	–	0 when being read.
		D6	**RTCAP**	AM/PM indicator	1 PM / 0 AM	X (*)	R/W	
		D5–4	**RTCHH[1:0]**	RTC 10-hour counter	0 to 2 or 0 to 1	X (*)	R/W	
		D3–0	**RTCHL[3:0]**	RTC 1-hour counter	0 to 9	X (*)	R/W	
RTC Day Register (RTC_DAY)	**0x5147** (8 bits)	D7–6	**–**	reserved	–	–	–	0 when being read.
		D5–4	**RTCDH[1:0]**	RTC 10-day counter	0 to 3	X (*)	R/W	
		D3–0	**RTCDL[3:0]**	RTC 1-day counter	0 to 9	X (*)	R/W	
RTC Month Register (RTC_MONTH)	**0x5148** (8 bits)	D7–5	**–**	reserved	–	–	–	0 when being read.
		D4	**RTCMOH**	RTC 10-month counter	0 to 1	X (*)	R/W	
		D3–0	**RTCMOL[3:0]**	RTC 1-month counter	0 to 9	X (*)	R/W	
RTC Year Register (RTC_YEAR)	**0x5149** (8 bits)	D7–4	**RTCYH[3:0]**	RTC 10-year counter	0 to 9	X (*)	R/W	
		D3–0	**RTCYL[3:0]**	RTC 1-year counter	0 to 9	X (*)	R/W	
RTC Days of Week Register (RTC_WEEK)	**0x514a** (8 bits)	D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**RTCWK[2:0]**	RTC days of week counter	RTCWK[2:0] / Days of week 0x7 – 0x6 Saturday 0x5 Friday 0x4 Thursday 0x3 Wednesday 0x2 Tuesday 0x1 Monday 0x0 Sunday	X (*)	R/W	

Init.: ( ) indicates the value set after a software reset (RTCRST → 1 → 0) is performed.

* Software reset (RTCRST → 1 → 0) does not affect the counter values. This register retains the value set before a software reset is performed.

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)	D7–0	**P0IN[7:0]**	P0[7:0] port input data	1 1 (H) / 0 0 (L)	×	R	
P0 Port Output Data Register (P0_OUT)	**0x5201** (8 bits)	D7–0	**P0OUT[7:0]**	P0[7:0] port output data	1 1 (H) / 0 0 (L)	0	R/W	
P0 Port Output Enable Register (P0_OEN)	**0x5202** (8 bits)	D7–0	**P0OEN[7:0]**	P0[7:0] port output enable	1 Enable / 0 Disable	0	R/W	
P0 Port Pull-up Control Register (P0_PU)	**0x5203** (8 bits)	D7–0	**P0PU[7:0]**	P0[7:0] port pull-up enable	1 Enable / 0 Disable	1 (0xff)	R/W	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
P0 Port Schmitt Trigger Control Register (P0_SM)	**0x5204** (8 bits)	D7–0	**P0SM[7:0]**	P0[7:0] port Schmitt trigger input enable	1 Enable (Schmitt) 0 Disable (CMOS)	1	R/W	
P0 Port Interrupt Mask Register (P0_IMSK)	**0x5205** (8 bits)	D7–0	**P0IE[7:0]**	P0[7:0] port interrupt enable	1 Enable 0 Disable	0	R/W	
P0 Port Interrupt Edge Select Register (P0_EDGE)	**0x5206** (8 bits)	D7–0	**P0EDGE[7:0]**	P0[7:0] port interrupt edge select	1 Falling edge 0 Rising edge	0	R/W	
P0 Port Interrupt Flag Register (P0_IFLG)	**0x5207** (8 bits)	D7–0	**P0IF[7:0]**	P0[7:0] port interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
P0 Port Chattering Filter Control Register (P0_CHAT)	**0x5208** (8 bits)	D7	**–**	reserved	–	–	–	0 when being read.
		D6–4	**P0CF2[2:0]**	P0[7:4] chattering filter time	P0CF2[2:0] / Filter time 0x7 16384/f_{PCLK} 0x6 8192/f_{PCLK} 0x5 4096/f_{PCLK} 0x4 2048/f_{PCLK} 0x3 1024/f_{PCLK} 0x2 512/f_{PCLK} 0x1 256/f_{PCLK} 0x0 None	0x0	R/W	
		D3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**P0CF1[2:0]**	P0[3:0] chattering filter time	P0CF1[2:0] / Filter time 0x7 16384/f_{PCLK} 0x6 8192/f_{PCLK} 0x5 4096/f_{PCLK} 0x4 2048/f_{PCLK} 0x3 1024/f_{PCLK} 0x2 512/f_{PCLK} 0x1 256/f_{PCLK} 0x0 None	0x0	R/W	
P0 Port Key-Entry Reset Configuration Register (P0_KRST)	**0x5209** (8 bits)	D7–2	**–**	reserved	–	–	–	0 when being read.
		D1–0	**P0KRST[1:0]**	P0 port key-entry reset configuration	P0KRST[1:0] / Configuration 0x3 P0[3:0] = 0 0x2 P0[2:0] = 0 0x1 P0[1:0] = 0 0x0 Disable	0x0	R/W	
P0 Port Input Enable Register (P0_IEN)	**0x520a** (8 bits)	D7–0	**P0IEN[7:0]**	P0[7:0] port input enable	1 Enable 0 Disable	1 (0xff)	R/W	
P1 Port Input Data Register (P1_IN)	**0x5210** (8 bits)	D7–0	**P1IN[7:0]**	P1[7:0] port input data	1 1 (H) 0 0 (L)	×	R	
P1 Port Output Data Register (P1_OUT)	**0x5211** (8 bits)	D7–0	**P1OUT[7:0]**	P1[7:0] port output data	1 1 (H) 0 0 (L)	0	R/W	
P1 Port Output Enable Register (P1_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 Schmitt Trigger Control Register (P1_SM)	**0x5214** (8 bits)	D7–6	**P1SM[7:6]**	P1[7:6] port Schmitt trigger input enable	1 Enable (Schmitt) 0 –	1	R	Always enabled
		D5–0	**P1SM[5:0]**	P1[5:0] port Schmitt trigger input enable	1 Enable (Schmitt) 0 Disable (CMOS)	1	R/W	
P1 Port Interrupt Mask Register (P1_IMSK)	**0x5215** (8 bits)	D7–0	**P1IE[7:0]**	P1[7:0] port interrupt enable	1 Enable 0 Disable	0	R/W	
P1 Port Interrupt Edge Select Register (P1_EDGE)	**0x5216** (8 bits)	D7–0	**P1EDGE[7:0]**	P1[7:0] port interrupt edge select	1 Falling edge 0 Rising edge	0	R/W	
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.

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
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	0x0	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	1 (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 Schmitt Trigger Control Register (P2_SM)	**0x5224** (8 bits)	D7–0	**P2SM[7:0]**	P2[7:0] port Schmitt trigger input enable	1: Enable (Schmitt); 0: –	1	R	Always enabled
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	1 (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 Schmitt Trigger Control Register (P3_SM)	**0x5234** (8 bits)	D7–0	**P3SM[7:0]**	P3[7:0] port Schmitt trigger input enable	1: Enable (Schmitt); 0: –	1	R	Always enabled
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	1 (0xff)	R/W	
P4 Port Input Data Register (P4_IN)	**0x5240** (8 bits)	D7–0	**P4IN[7:0]**	P4[7:0] port input data	1: 1 (H); 0: 0 (L)	×	R	D[7:4] =reserved in S1C17604/602/621
P4 Port Output Data Register (P4_OUT)	**0x5241** (8 bits)	D7–0	**P4OUT[7:0]**	P4[7:0] port output data	1: 1 (H); 0: 0 (L)	0	R/W	D[7:4] =reserved in S1C17604/602/621
P4 Port Output Enable Register (P4_OEN)	**0x5242** (8 bits)	D7–0	**P4OEN[7:0]**	P4[7:0] port output enable	1: Enable; 0: Disable	0	R/W	D[7:4] =reserved in S1C17604/602/621

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
P4 Port Pull-up Control Register (P4_PU)	0x5243 (8 bits)	D7–0	P4PU[7:0]	P4[7:0] port pull-up enable	1 Enable 0 Disable	1 (0xff)	R/W	D[7:4] =reserved in S1C17604/602/621
P4 Port Schmitt Trigger Control Register (P4_SM)	0x5244 (8 bits)	D7–4	P4SM[7:4]	P4[7:4] port Schmitt trigger input enable	1 Enable (Schmitt) 0 Disable (CMOS)	1	R/W	D[7:4] = reserved in S1C17604/602/621
		D3–0	P4SM[3:0]	P4[3:0] port Schmitt trigger input enable	1 Enable (Schmitt) 0 –	1	R	Always enabled
P4 Port Input Enable Register (P4_IEN)	0x524a (8 bits)	D7–0	P4IEN[7:0]	P4[7:0] port input enable	1 Enable 0 Disable	1 (0xff)	R/W	D[7:4] =reserved in S1C17604/602/621
P5 Port Input Data Register (P5_IN) (S1C17624/622)	0x5250 (8 bits)	D7	–	reserved	–	–	–	0 when being read.
		D6–0	P5IN[6:0]	P5[6:0] port input data	1 1 (H) 0 0 (L)	×	R	
P5 Port Output Data Register (P5_OUT) (S1C17624/622)	0x5251 (8 bits)	D7	–	reserved	–	–	–	0 when being read.
		D6–0	P5OUT[6:0]	P5[6:0] port output data	1 1 (H) 0 0 (L)	0	R/W	
P5 Port Output Enable Register (P5_OEN) (S1C17624/622)	0x5252 (8 bits)	D7	–	reserved	–	–	–	0 when being read.
		D6–0	P5OEN[6:0]	P5[6:0] port output enable	1 Enable 0 Disable	0	R/W	
P5 Port Pull-up Control Register (P5_PU) (S1C17624/622)	0x5253 (8 bits)	D7	–	reserved	–	–	–	0 when being read.
		D6–0	P5PU[6:0]	P5[6:0] port pull-up enable	1 Enable 0 Disable	1 (0x7f)	R/W	
P5 Port Schmitt Trigger Control Register (P5_SM) (S1C17624/622)	0x5254 (8 bits)	D7	–	reserved	–	–	–	0 when being read.
		D6–3	P5SM[6:3]	P5[6:3] port Schmitt trigger input enable	1 Enable (Schmitt) 0 –	1	R	Always enabled
		D2–0	P5SM[2:0]	P5[2:0] port Schmitt trigger input enable	1 Enable (Schmitt) 0 Disable (CMOS)	1	R/W	
P5 Port Input Enable Register (P5_IEN) (S1C17624/622)	0x525a (8 bits)	D7	–	reserved	–	–	–	0 when being read.
		D6–0	P5IEN[6:0]	P5[6:0] port input enable	1 Enable 0 Disable	1 (0x7f)	R/W	
P0[3:0] Port Function Select Register (P00_03PMUX)	0x52a0 (8 bits)	D7–6	P03MUX[1:0]	P03 port function select	P03MUX[1:0] Function 0x3 reserved 0x2 reserved 0x1 #ADTRG 0x0 P03	0x0	R/W	
		D5–4	P02MUX[1:0]	P02 port function select	P02MUX[1:0] Function 0x3 reserved 0x2 reserved 0x1 reserved 0x0 P02/EXCL0	0x0	R/W	
		D3–2	P01MUX[1:0]	P01 port function select	P01MUX[1:0] Function 0x3 reserved 0x2 reserved 0x1 REMI 0x0 P01	0x0	R/W	
		D1–0	P00MUX[1:0]	P00 port function select	P00MUX[1:0] Function 0x3 reserved 0x2 reserved 0x1 REMO 0x0 P00	0x0	R/W	
P0[7:4] Port Function Select Register (P04_07PMUX)	0x52a1 (8 bits)	D7–6	P07MUX[1:0]	P07 port function select	P07MUX[1:0] Function 0x3 reserved 0x2 reserved 0x1 #SPISS0 0x0 P07	0x0	R/W	
		D5–4	P06MUX[1:0]	P06 port function select	P06MUX[1:0] Function 0x3 reserved 0x2 reserved 0x1 SDI0 0x0 P06	0x0	R/W	
		D3–2	P05MUX[1:0]	P05 port function select	P05MUX[1:0] Function 0x3 reserved 0x2 reserved 0x1 SDO0 0x0 P05	0x0	R/W	
		D1–0	P04MUX[1:0]	P04 port function select	P04MUX[1:0] Function 0x3 reserved 0x2 reserved 0x1 SPICLK0 0x0 P04	0x0	R/W	

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	AIN7			
					0x1	reserved			
					0x0	P13/EXCL1			
		D5–4	**P12MUX[1:0]**	P12 port function select	P12MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	SIN0			
					0x0	P12			
		D3–2	**P11MUX[1:0]**	P11 port function select	P11MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	SOUT0			
					0x0	P11			
		D1–0	**P10MUX[1:0]**	P10 port function select	P10MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	SCLK0			
					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	AIN3			
					0x0	P17			
		D5–4	**P16MUX[1:0]**	P16 port function select	P16MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	AIN4			
					0x1	SCLK1			
					0x0	P16			
		D3–2	**P15MUX[1:0]**	P15 port function select	P15MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	AIN5			
					0x1	reserved			
					0x0	P15/EXCL3			
		D1–0	**P14MUX[1:0]**	P14 port function select	P14MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	AIN6			
					0x1	reserved			
					0x0	P14/EXCL2			
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	SENB0			
					0x0	P23			
		D5–4	**P22MUX[1:0]**	P22 port function select	P22MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	AIN0			
					0x0	P22			
		D3–2	**P21MUX[1:0]**	P21 port function select	P21MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	AIN1			
					0x0	P21			
		D1–0	**P20MUX[1:0]**	P20 port function select	P20MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	AIN2			
					0x0	P20			

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	RFIN1			
					0x1	SOUT1			
					0x0	P27			
		D5–4	**P26MUX[1:0]**	P26 port function select	P26MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	RFIN0			
					0x0	P26			
		D3–2	**P25MUX[1:0]**	P25 port function select	P25MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	REF0			
					0x0	P25			
		D1–0	**P24MUX[1:0]**	P24 port function select	P24MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	SENA0			
					0x0	P24			
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	TOUTA6/CAPA6: S1C17624/604 only
					0x3	TOUTA6/CAPA6			
					0x2	SCL0			
					0x1	SCL1			
					0x0	P33			
		D5–4	**P32MUX[1:0]**	P32 port function select	P32MUX[1:0]	Function	0x0	R/W	TOUTB5/CAPB5: S1C17624/604 only
					0x3	TOUTB5/CAPB5			
					0x2	SENB1			
					0x1	SDA0			
					0x0	P32			
		D3–2	**P31MUX[1:0]**	P31 port function select	P31MUX[1:0]	Function	0x0	R/W	TOUTA5/CAPA5: S1C17624/604 only
					0x3	TOUTA5/CAPA5			
					0x2	SENA1			
					0x1	SCL0			
					0x0	P31			
		D1–0	**P30MUX[1:0]**	P30 port function select	P30MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	REF1			
					0x1	SIN1			
					0x0	P30			
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	EXCL6: S1C17624/604 only (Set EXCL6S to 0 in S1C17624.)
					0x3	TOUT4			
					0x2	LFRO			
					0x1	TOUTN3			
					0x0	P37/EXCL6			
		D5–4	**P36MUX[1:0]**	P36 port function select	P36MUX[1:0]	Function	0x0	R/W	EXCL5: S1C17624/604 only (Set EXCL5S to 0 in S1C17624.)
					0x3	reserved			
					0x2	RFCLKO			
					0x1	TOUT3			
					0x0	P36/EXCL5			
		D3–2	**P35MUX[1:0]**	P35 port function select	P35MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	#BFR			
					0x1	FOUT1			
					0x0	P35			
		D1–0	**P34MUX[1:0]**	P34 port function select	P34MUX[1:0]	Function	0x0	R/W	TOUTB6/CAPB6: S1C17624/604 only
					0x3	TOUTB6/CAPB6			
					0x2	SDA0			
					0x1	SDA1			
					0x0	P34			

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	reserved			
					0x1	P43			
					0x0	DCLK			
		D5–4	**P42MUX[1:0]**	P42 port function select	P42MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	P42			
					0x0	DST2			
		D3–2	**P41MUX[1:0]**	P41 port function select	P41MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	P41			
					0x0	DSIO			
		D1–0	**P40MUX[1:0]**	P40 port function select	P40MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	FOUTH			
					0x0	P40			
P4[7:4] Port Function Select Register (P44_47PMUX) **(S1C17624/622)**	**0x52a9** (8 bits)	D7–6	**P47MUX[1:0]**	P47 port function select	P47MUX[1:0]	Function	0x0	R/W	EXCL5: S1C17624 only (Set EXCL5S to 1.)
					0x3	reserved			
					0x2	reserved			
					0x1	TOUT4			
					0x0	P47/EXCL5			
		D5–4	**P46MUX[1:0]**	P46 port function select	P46MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	RFCLKO			
					0x0	P46			
		D3–2	**P45MUX[1:0]**	P45 port function select	P45MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	SDA1			
					0x0	P45			
		D1–0	**P44MUX[1:0]**	P44 port function select	P44MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	SCL1			
					0x0	P44			
P5[3:0] Port Function Select Register (P50_53PMUX) **(S1C17624/622)**	**0x52aa** (8 bits)	D7–6	**P53MUX[1:0]**	P53 port function select	P53MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	#BFR			
					0x0	P53			
		D5–4	**P52MUX[1:0]**	P52 port function select	P52MUX[1:0]	Function	0x0	R/W	TOUTB5/CAPB5: S1C17624 only
					0x3	reserved			
					0x2	TOUTB5/CAPB5			
					0x1	SIN1			
					0x0	P52			
		D3–2	**P51MUX[1:0]**	P51 port function select	P51MUX[1:0]	Function	0x0	R/W	TOUTA5/CAPA5: S1C17624 only
					0x3	reserved			
					0x2	TOUTA5/CAPA5			
					0x1	SOUT1			
					0x0	P51			
		D1–0	**P50MUX[1:0]**	P50 port function select	P50MUX[1:0]	Function	0x0	R/W	EXCL6: S1C17624 only (Set EXCL6S to 1.)
					0x3	reserved			
					0x2	reserved			
					0x1	SCLK1			
					0x0	P50/EXCL6			
P5[6:4] Port Function Select Register (P54_56PMUX) **S1C17624/622**	**0x52ab** (8 bits)	D7	**EXCL6S**	EXCL6 input select	1 P50/EXCL6	0 P37/EXCL6	0	R/W	S1C17624 only
		D6	**EXCL5S**	EXCL5 input select	1 P47/EXCL5	0 P36/EXCL5	0	R/W	
		D5–4	**P56MUX[1:0]**	P56 port function select	P56MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	TOUTB6/CAPB6			
					0x1	reserved			
					0x0	P56			
		D3–2	**P55MUX[1:0]**	P55 port function select	P55MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	TOUTA6/CAPA6			
					0x1	reserved			
					0x0	P55			
		D1–0	**P54MUX[1:0]**	P54 port function select	P54MUX[1:0]	Function	0x0	R/W	
					0x3	reserved			
					0x2	reserved			
					0x1	LFRO			
					0x0	P54			

0x5300–0x530c 16-bit PWM Timer (T16E) Ch.0

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
T16E Ch.0 Compare Data A Register (T16E_CA0)	0x5300 (16 bits)	D15–0	CA[15:0]	Compare data A CA15 = MSB CA0 = LSB	0x0 to 0xffff	0x0	R/W	
T16E Ch.0 Compare Data B Register (T16E_CB0)	0x5302 (16 bits)	D15–0	CB[15:0]	Compare data B CB15 = MSB CB0 = LSB	0x0 to 0xffff	0x0	R/W	
T16E Ch.0 Counter Data Register (T16E_TC0)	0x5304 (16 bits)	D15–0	TC[15:0]	Counter data TC15 = MSB TC0 = LSB	0x0 to 0xffff	0x0	R/W	
T16E Ch.0 Control Register (T16E_CTL0)	0x5306 (16 bits)	D15–9	–	reserved	–	–	–	0 when being read.
		D8	INITOL	Initial output level	1 High 0 Low	0	R/W	
		D7	–	reserved	–	–	–	0 when being read.
		D6	SELFM	Fine mode select	1 Fine mode 0 Normal mode	0	R/W	
		D5	CBUFEN	Comparison buffer enable	1 Enable 0 Disable	0	R/W	
		D4	INVOUT	Inverse output	1 Invert 0 Normal	0	R/W	
		D3	CLKSEL	Input clock select	1 External 0 Internal	0	R/W	
		D2	OUTEN	Clock output enable	1 Enable 0 Disable	0	R/W	
		D1	T16ERST	Timer reset	1 Reset 0 Ignored	0	W	0 when being read.
		D0	T16ERUN	Timer run/stop control	1 Run 0 Stop	0	R/W	
T16E Ch.0 Clock Division Ratio Select Register (T16E_DF0)	0x5308 (16 bits)	D15–4	–	reserved	–	–	–	0 when being read.
		D3–0	T16EDF[3:0]	Clock division ratio select	T16EDF[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
T16E Ch.*x* Interrupt Mask Register (T16E_IMSK*x*)	0x530a (16 bits)	D15–2	–	reserved	–	–	–	0 when being read.
		D1	CBIE	Compare B interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	CAIE	Compare A interrupt enable	1 Enable 0 Disable	0	R/W	
T16E Ch.*x* Interrupt Flag Register (T16E_IFLG*x*)	0x530c (16 bits)	D15–2	–	reserved	–	–	–	0 when being read.
		D1	CBIF	Compare B interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D0	CAIF	Compare A interrupt flag		0	R/W	

0x4020, 0x5320–0x532c MISC Registers

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Prescaler Control Register (PSC_CTL)	0x4020 (8 bits)	D7–2	–	reserved	–	–	–	0 when being read.
		D1	PRUND	Run/stop select in debug mode	1 Run 0 Stop	0	R/W	
		D0	PRUN	Prescaler run/stop control	1 Run 0 Stop	0	R/W	
FLASHC Control Register (MISC_FL)	0x5320 (16 bits)	D15–10	–	reserved	–	–	–	0 when being read.
		D9–8	–	reserved	–	0x3	–	
		D7–3	–	reserved	–	–	–	0 when being read.
		D2–0	FLCYC[2:0]	FLASHC read access cycle	FLCYC[2:0] / Read cycle: 0x7–0x5 reserved 0x4 1 cycle 0x3 5 cycles 0x2 4 cycles 0x1 3 cycles 0x0 2 cycles	0x3	R/W	
OSC1 Peripheral Control Register (MISC_OSC1)	0x5322 (16 bits)	D15–1	–	reserved	–	–	–	0 when being read.
		D0	O1DBG	Run/stop select in debug mode (except PCLK peripheral circuits)	1 Run 0 Stop	0	R/W	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
MISC Protect Register (MISC_PROT)	**0x5324** (16 bits)	D15–0	**PROT[15:0]**	MISC register write protect	Writing 0x96 removes the write protection of the MISC registers (0x5326–0x532a). Writing another value set the write protection.	0x0	R/W	
IRAM Size Select Register (MISC_IRAMSZ) **(S1C17624/604)**	**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	–	reserved	–	–	–	0x1 when being read.
		D3	–	reserved	–	–	–	0 when being read.
		D2–0	**IRAMSZ[2:0]**	IRAM size select	IRAMSZ[2:0] Size 0x3 2KB 0x2 4KB 0x1 8KB Other reserved	0x1	R/W	
IRAM Size Select Register (MISC_IRAMSZ) **(S1C17622)**	**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	–	reserved	–	–	–	0x1 when being read.
		D3	–	reserved	–	–	–	0 when being read.
		D2–0	**IRAMSZ[2:0]**	IRAM size select	IRAMSZ[2:0] Size 0x3 2KB 0x2 4KB Other reserved	0x1	R/W	
IRAM Size Select Register (MISC_IRAMSZ) **(S1C17602)**	**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	–	reserved	–	–	–	0x2 when being read.
		D3	–	reserved	–	–	–	0 when being read.
		D2–0	**IRAMSZ[2:0]**	IRAM size select	IRAMSZ[2:0] Size 0x7–0x0 reserved	0x2	R/W	
IRAM Size Select Register (MISC_IRAMSZ) **(S1C17621)**	**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	–	reserved	–	–	–	0x2 when being read.
		D3	–	reserved	–	–	–	0 when being read.
		D2–0	**IRAMSZ[2:0]**	IRAM size select	IRAMSZ[2:0] Size 0x7–0x0 reserved	0x2	R/W	
Vector Table Address Low Register (MISC_TTBRL)	**0x5328** (16 bits)	D15–8	**TTBR[15:8]**	Vector table base address A[15:8]	0x0–0xff	0x80	R/W	
		D7–0	**TTBR[7:0]**	Vector table base address A[7:0] (fixed at 0)	0x0	0x0	R	
Vector Table Address High Register (MISC_TTBRH)	**0x532a** (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	**TTBR[23:16]**	Vector table base address A[23:16]	0x0–0xff	0x0	R/W	
PSR Register (MISC_PSR)	**0x532c** (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–5	**PSRIL[2:0]**	PSR interrupt level (IL) bits	0x0 to 0x7	0x0	R	
		D4	**PSRIE**	PSR interrupt enable (IE) bit	1 1 (enable) 0 0 (disable)	0	R	
		D3	**PSRC**	PSR carry (C) flag	1 1 (set) 0 0 (cleared)	0	R	
		D2	**PSRV**	PSR overflow (V) flag	1 1 (set) 0 0 (cleared)	0	R	
		D1	**PSRZ**	PSR zero (Z) flag	1 1 (set) 0 0 (cleared)	0	R	
		D0	**PSRN**	PSR negative (N) flag	1 1 (set) 0 0 (cleared)	0	R	

0x5340–0x5346 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 CGCLK[3:0])	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	

0x5380–0x5386 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 0x7	0x0	R/W	
		D10–8	ADCS[2:0]	Start channel select	0x0 to 0x7	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 0x7	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 (ADC_DIV)	**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

0x5067, 0x53a0–0x53ae R/F Converter

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RFC Clock Control Register (OSC_RFC)	**0x5067** (8 bits)	D7–4	**–**	reserved	–	–	–	0 when being read.
		D3–2	**RFTCKDV [1:0]**	RFC clock division ratio select	RFTCKDV[1:0] / Division ratio 0x3 1/8 0x2 1/4 0x1 1/2 0x0 1/1	0x0	R/W	When the clock source is HSCLK
		D1	**RFTCKSRC**	RFC clock source select	1 OSC1 0 HSCLK	1	R/W	
		D0	**RFTCKEN**	RFC clock enable	1 Enable 0 Disable	0	R/W	
RFC Control Register (RFC_CTL)	**0x53a0** (16 bits)	D15–8	**–**	reserved	–	–	–	0 when being read.
		D7	**CONEN**	Continuous oscillation enable	1 Enable 0 Disable	0	R/W	
		D6	**EVTEN**	Event counter mode enable	1 Enable 0 Disable	0	R/W	
		D5–4	**SMODE[1:0]**	Sensor oscillation mode select	SMODE[1:0] / Sensor 0x3 reserved 0x2 DC capacitive 0x1 AC resistive 0x0 DC resistive	0x0	R/W	
		D3–2	**–**	reserved	–	–	–	0 when being read.
		D1	**CHSEL**	Conversion channel select	1 Ch.1 0 Ch.0	0	R/W	
		D0	**RFCEN**	RFC enable	1 Enable 0 Disable	0	R/W	
RFC Oscillation Trigger Register (RFC_TRG)	**0x53a2** (16 bits)	D15–3	**–**	reserved	–	–	–	0 when being read.
		D2	**SSENB**	Sensor B oscillation control/status	1 Start/Run 0 Stop	0	R/W	
		D1	**SSENA**	Sensor A oscillation control/status	1 Start/Run 0 Stop	0	R/W	
		D0	**SREF**	Reference oscillation control/status	1 Start/Run 0 Stop	0	R/W	
RFC Measurement Counter Low Register (RFC_MCL)	**0x53a4** (16 bits)	D15–0	**MC[15:0]**	Measurement counter low-order 16-bit data	0x0–0xffff	0x0	R/W	
RFC Measurement Counter High Register (RFC_MCH)	**0x53a6** (16 bits)	D15–8	**–**	reserved	–	–	–	0 when being read.
		D7–0	**MC[23:16]**	Measurement counter high-order 8-bit data	0x0–0xff	0x0	R/W	
RFC Time Base Counter Low Register (RFC_TCL)	**0x53a8** (16 bits)	D15–0	**TC[15:0]**	Time base counter low-order 16-bit data	0x0–0xffff	0x0	R/W	
RFC Time Base Counter High Register (RFC_TCH)	**0x53aa** (16 bits)	D15–8	**–**	reserved	–	–	–	0 when being read.
		D7–0	**TC[23:16]**	Time base counter high-order 8-bit data	0x0–0xff	0x0	R/W	

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
RFC Interrupt Mask Register (RFC_IMSK)	0x53ac (16 bits)	D15–5	–	reserved	–	–	–	0 when being read.
		D4	OVTCIE	TC overflow error interrupt enable	1 Enable 0 Disable	0	R/W	
		D3	OVMCIE	MC overflow error interrupt enable	1 Enable 0 Disable	0	R/W	
		D2	ESENBIE	Sensor B oscillation completion interrupt enable	1 Enable 0 Disable	0	R/W	
		D1	ESENAIE	Sensor A oscillation completion interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	EREFIE	Reference oscillation completion interrupt enable	1 Enable 0 Disable	0	R/W	
RFC Interrupt Flag Register (RFC_IFLG)	0x53ae (16 bits)	D15–5	–	reserved	–	–	–	0 when being read.
		D4	OVTCIF	TC overflow error interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D3	OVMCIF	MC overflow error interrupt flag		0	R/W	
		D2	ESENBIF	Sensor B oscillation completion interrupt flag		0	R/W	
		D1	ESENAIF	Sensor A oscillation completion interrupt flag		0	R/W	
		D0	EREFIF	Reference oscillation completion interrupt flag		0	R/W	

0x5068, 0x5400–0x540c 16-bit PWM Timer (T16A2) Ch.0 (S1C17624/604)

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	
		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)	D15–7	–	reserved	–	–	–	0 when being read.
		D6	HCM	Half clock mode enable	1 Enable 0 Disable	0	R/W	
		D5–4	CCABCNT [1:0]	Counter select	CCABCNT[1:0]: Counter Ch. 0x3, 0x2: reserved 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.0 Data Register (T16A_TC0)	0x5402 (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.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 (T16A2) Ch.1 (S1C17624/604)

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	
		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)	D15–7	**–**	reserved	–	–	–	0 when being read.
		D6	**HCM**	Half clock mode enable	1 Enable / 0 Disable	0	R/W	
		D5–4	**CCABCNT [1:0]**	Counter select	CCABCNT[1:0] / Counter Ch. 0x3, 0x2: reserved 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	

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) (S1C17624/604/ 602)	**0xffff90** (32 bits)	D31–24	**–**	Unused (fixed at 0)	0x0				0x0	R	
		D23–0	**DBRAM[23:0]**	Debug RAM base address	S1C17624/604: 0x1fc0 S1C17602: 0x0fc0				←	R	
Debug Control Register (DCR)	**0xffffa0** (8 bits)	D7	**IBE4**	Instruction break #4 enable	1	Enable	0	Disable	0	R/W	
		D6	**IBE3**	Instruction break #3 enable	1	Enable	0	Disable	0	R/W	
		D5	**IBE2**	Instruction break #2 enable	1	Enable	0	Disable	0	R/W	
		D4	**DR**	Debug request flag	1	Occurred	0	Not occurred	0	R/W	Reset by writing 1.
		D3	**IBE1**	Instruction break #1 enable	1	Enable	0	Disable	0	R/W	
		D2	**IBE0**	Instruction break #0 enable	1	Enable	0	Disable	0	R/W	
		D1	**SE**	Single step enable	1	Enable	0	Disable	0	R/W	
		D0	**DM**	Debug mode	1	Debug mode	0	User mode	0	R	
Instruction Break Address Register 1 (IBAR1)	**0xffffb4** (32 bits)	D31–24	**–**	reserved	–				–	–	0 when being read.
		D23–0	**IBAR1[23:0]**	Instruction break address #1 IBAR123 = MSB IBAR10 = LSB	0x0 to 0xffffff				0x0	R/W	
Instruction Break Address Register 2 (IBAR2)	**0xffffb8** (32 bits)	D31–24	**–**	reserved	–				–	–	0 when being read.
		D23–0	**IBAR2[23:0]**	Instruction break address #2 IBAR223 = MSB IBAR20 = LSB	0x0 to 0xffffff				0x0	R/W	
Instruction Break Address Register 3 (IBAR3)	**0xffffbc** (32 bits)	D31–24	**–**	reserved	–				–	–	0 when being read.
		D23–0	**IBAR3[23:0]**	Instruction break address #3 IBAR323 = MSB IBAR30 = LSB	0x0 to 0xffffff				0x0	R/W	
Instruction Break Address Register 4 (IBAR4)	**0xffffd0** (32 bits)	D31–24	**–**	reserved	–				–	–	0 when being read.
		D23–0	**IBAR4[23:0]**	Instruction break address #4 IBAR423 = MSB IBAR40 = LSB	0x0 to 0xffffff				0x0	R/W	

Appendix B Power Saving

Current consumption will vary dramatically, depending on CPU operating mode, operation clock frequency, and the peripheral circuits being operated. Listed below are the control methods for saving power.

B.1 Clock Control Power Saving

This section describes clock systems that can be controlled via software and power-saving control details. For more information on control registers and control methods, refer to the respective module sections.

System SLEEP

- Execute the `slp` instruction
  To stop the entire system, execute the `slp` instruction after setting RTCCE/RTC_CC register to 0. This stops all the oscillator and peripheral circuits. Starting up the CPU from SLEEP mode is therefore limited to start-up using a port.
  If the `slp` instruction is executed when RTCCE = 1 and RTCSTP/RTC_CNTL0 register = 0, the OSC1 oscillator circuit does not stop and the RTC can be operated. In this case, the CPU can reawaken from SLEEP mode by an RTC interrupt as well as a port interrupt.

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
  • 16-bit timer (T16)
  • 8-bit timer (T8F)
  • UART
  • SPI
  • I^2C master (I2CM)
  • I^2C slave (I2CS)
  • 16-bit PWM timer (T16E)
  • I/O port (P)
  • MISC register (MISC)
  • Power generator (VD1)

- Supply voltage detector (SVD)
- IR remote controller (REMC)
- A/D converter (ADC10)
- Interrupt controller (ITC) (PCLK is used only when the register is set.)

PCLK is not required for the peripheral modules/functions shown below.

Peripheral circuits/functions that do not use PCLK
- Clock timer (CT)
- Stopwatch timer (SWT)
- Watchdog timer (WDT)
- 8-bit OSC1 timer (T8OSC1)
- LCD driver (LCD)
- R/F converter (RFC)
- 16-bit PWM timer (T16A2)
- FOUTH/FOUT1 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	Stop	Execute `slp` instruction	1, 2
	Oscillation (system CLK)	Stop	Stop	Stop	Run	Execute `halt` instruction	1, 2, 3, 4
	Oscillation (system CLK)	Stop	Stop	Run	Run	Execute `halt` instruction	1, 2, 3, 4
	Oscillation (system CLK)	Stop	Run (1/1)	Run	Run		
	Oscillation	Oscillation (system CLK)	Stop	Run	Run	Execute `halt` instruction	1, 2, 3, 4
	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 or debug interrupt (ICD forced break).
2. Startup by RTC (S1C17624/604)
   Started up by an RTC interrupt.
3. Startup by OSC1 peripheral circuit
   Started up by a clock timer, stopwatch timer, 8-bit OSC1 timer, or watchdog timer interrupt.
4. Startup by PCLK peripheral circuit
   Started up by a PCLK peripheral circuit interrupt.

B.2 Reducing Power Consumption via Power Supply Control

The available power supply controls are listed below.

Internal voltage regulator

- Setting the internal operating voltage V_{D1} to 2.5 V increases current consumption.
  For normal operations, set V_{D1} to 1.8 V, and switch to 2.5 V only for Flash memory programming.
- Note that turning on internal voltage regulator heavy load protection will increase current consumption.
  Turn off heavy load protection for normal operations. Turn on only if operations are unstable.

LCD voltage regulator

- Setting VCSEL to 0 (V_{C1} reference voltage) will increase current consumption.
  Set VCSEL to 1 (V_{C2} reference voltage) if the power supply voltage V_{DD} is higher than 2.5 V.
- Turning on the LCD voltage regulator heavy load protection will increase current consumption.
  Turn off heavy load protection for normal operations. Turn on only if the display is unstable.
- If no LCD display is being used, turn off the LCD driver.

Power supply voltage detection (SVD) circuit

- Operating the SVD circuit will increase current consumption.
  Turn off power supply voltage detection unless it is required.

Appendix C Mounting Precautions

This section describes various precautions for circuit board design and IC mounting.

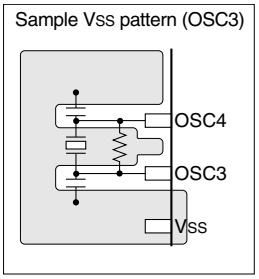
Oscillator circuit

- 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 FOUTH or FOUT1 pin output.
You can check the quality of the OSC3 output waveform via the FOUTH output. Confirm that the frequency is as designed, is free of noise, and has minimal jitter.
You can check the quality of the OSC1 waveform via the FOUT1 output. In particular, enlarge the areas before and after the clock rising and falling edges and take special care to confirm that the regions approximately 100 ns to either side are free of clock or spiking noise.

Failure to observe precautions (1) to (3) adequately may lead to jitter in the OSC3 output and noise in the OSC1 output. Jitter in the OSC3 output will reduce operating frequencies, while noise in the OSC1 output will destabilize timers operated by the OSC1 clock as well as CPU Core operations when the system clock switches to OSC1.

Reset circuit

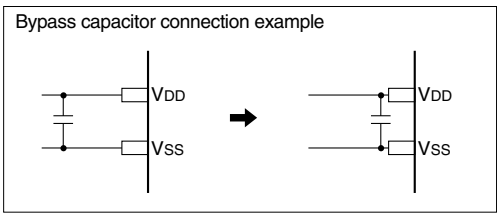
- The reset signal input to the #RESET pin when power is turned on will vary, depending on various factors, such as power supply start-up time, components used, and circuit board patterns. Constants such as capacitance and resistance should be determined through testing with real-world products.
- Components such as capacitors and resistors connected to the #RESET pin should have the shortest connections possible to prevent noise-induced resets.

Power supply circuit

Sudden power supply fluctuations due to noise will cause malfunctions. Consider the following issues.

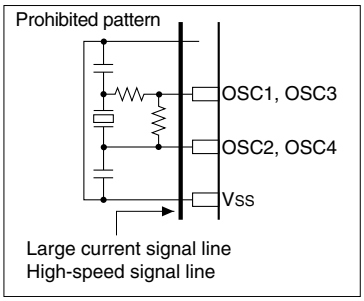
(1) Connections from the power supply to the VDD and VSS pins should be implemented via the shortest, thickest patterns possible.

(2) If a bypass capacitor is connected between VDD and VSS, connections between the VDD and VSS pins should be as short as possible.



Signal line location

- To prevent electromagnetically-induced noise arising from mutual induction, large-current signal lines should not be positioned close to circuits susceptible to noise, such as oscillators.
- Locating signal lines in parallel over significant distances or crossing signal lines operating at high speed will cause malfunctions due to noise generated by mutual interference. Specifically, avoid positioning crossing signal lines operating at high speed close to circuits susceptible to noise, such as oscillators.



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 VDD 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) VDD 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 the OSC1 oscillator circuit is not used, the OSC1 and OSC2 pins should be left open. If the OSC3 oscillator circuit is not used, the OSC3 and OSC4 pins must be configured as I/O port pins. The control registers should be fixed at the initial status (oscillation disabled).

(3) $V_{C1–3}$, CA, CB, SEG*x*, and COM*x* pins
If the LCD driver is not used, these pins should be left open. The control registers should be fixed at the initial status (display off). The unused SEG*x* and COM*x* pins that are not required to connect should be left open even if the LCD driver is used.

Miscellaneous

This product series is manufactured using microscopic processes.
Although it is designed to ensure basic IC reliability meeting EIAJ and MIL standards, minor variations over time may result in electrical damage arising from disturbances in the form of voltages exceeding the absolute maximum rating when mounting the product in addition to physical damage. The following factors can give rise to these variations:

(1) Electromagnetically-induced noise from industrial power supplies used in mounting reflow, reworking after mounting, and individual characteristic evaluation (testing) processes
(2) Electromagnetically-induced noise from a solder iron when soldering

In particular, during soldering, take care to ensure that the soldering iron GND (tip potential) has the same potential as the IC GND.

Appendix D 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_rtc_handler        ; 0x07       0x1c    CT/RTC
.long t8osc1_handler        ; 0x08       0x20    T8OSC1
.long svd_handler           ; 0x09       0x24    SVD
.long lcd_t16a2_0_handler   ; 0x0a       0x28    LCD/T16A2 ch0
.long t16e_0_handler        ; 0x0b       0x2c    T16E ch0
.long t8f_0_1_handler       ; 0x0c       0x30    T8F ch0/ch1
.long t16_0_handler         ; 0x0d       0x34    T16 ch0
.long t16_1_handler         ; 0x0e       0x38    T16 ch1
.long t16_2_handler         ; 0x0f       0x3c    T16 ch2
.long uart_0_handler        ; 0x10       0x40    UART ch0
.long i2cs_uart_1_handler   ; 0x11       0x44    I2CS/UART ch1
.long spi_0_handler         ; 0x12       0x48    SPI ch0
.long i2cm_handler          ; 0x13       0x4c    I2CM
.long remc_handler          ; 0x14       0x50    REMC
.long t16a2_1_handler       ; 0x15       0x54    T16A2 ch1
.long adc10_handler         ; 0x16       0x58    ADC10
.long rfc_handler           ; 0x17       0x5c    RFC
.long int18_handler         ; 0x18       0x60    -
.long int19_handler         ; 0x19       0x64    -
.long int1a_handler         ; 0x1a       0x68    -
.long int1b_handler         ; 0x1b       0x6c    -
.long int1c_handler         ; 0x1c       0x70    -
.long int1d_handler         ; 0x1d       0x74    -
.long int1e_handler         ; 0x1e       0x78    -
.long int1f_handler         ; 0x1f       0x7c    -

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

.text                                                                     ...(3)
.align 1

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

        ; ----- Stack pointer --------------------
        Xld.a   %sp, 0x0fc0     ; S1C17622/602                            ...(4)
        ;Xld.a  %sp, 0x1fc0     ; S1C17624/604

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

        ; FLASHC
        Xld.a   %r0, 0x04       ; 1 cycle access
        ld.b    [%r1], %r0      ; [0x5320] <= 0x04                        ...(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 controller access cycles.
Can be set to 1-cycle access for S1C17624/604/622/602/621.
(See the "Memory Map" 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.

E.1 Recommended Resonators for S1C17624/604/622

(1) OSC1 crystal resonator

Oscillation frequency [kHz]	Manufacturer	Product number
32.768	Epson Toyocom Corporation	MC-146 (SMD)
32.768	Epson Toyocom Corporation	C-002RX (leaded)

(2) OSC3 crystal resonator

Oscillation frequency [MHz]	Manufacturer	Product number
4.0	Epson Toyocom Corporation	MA-406 (SMD)
8.0	Epson Toyocom Corporation	MA-406 (SMD)

(3) OSC3 ceramic resonators

Oscillation frequency [MHz]	Manufacturer	Product number
1.0	Murata Manufacturing Co., Ltd.	CSBLA1M00J58-B0 (leaded)
	Murata Manufacturing Co., Ltd.	CSBFB1M00J58-R1 (SMD)
2.0	Murata Manufacturing Co., Ltd.	CSTCC2M00G56-R0 (SMD)
4.0	Murata Manufacturing Co., Ltd.	CSTLS4M00G56-B0 (leaded)
	Murata Manufacturing Co., Ltd.	CSTCR4M00G55-R0 (SMD)
8.0	Murata Manufacturing Co., Ltd.	CSTLS8M00G56-B0 (leaded)
	Murata Manufacturing Co., Ltd.	CSTCE8M00G55-R0 (SMD)

Note: Please ask the manufacturer to evaluate the resonator mounted on the circuit board.

(4) C_{G3}/C_{D3}/R_{D3}/R_{F3} recommended values

Name	Frequency [MHz]	Product number	C_{G3} [pF]	C_{D3} [pF]	R_{D3} [Ω]	R_{F3} [Ω]
X'tal3	4.0	MA-406 (C_L: 16pF)	27	27	0	1M
	8.0	MA-406 (C_L: 12pF)	18	18	0	1M
Ceramic3	1.0	CSBFB1M00J58-R1	100	100	6.8k	1M
		CSBLA1M00J58-B0	100	100	6.8k	1M
	2.0	CSTCC2M00G56-R0	(47)*	(47)*	0	1M
	4.0	CSTCR4M00G55-R0	(39)*	(39)*	0	1M
		CSTLS4M00G56-B0	(47)*	(47)*	0	1M
	8.0	CSTCE8M00G55-R0	(33)*	(33)*	0	1M
		CSTLS8M00G56-B0	(47)*	(47)*	0	1M

* The values enclosed with ( ) are the built-in capacitances of the resonator.

E.2 Recommended Resonators for S1C17602/621

(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
4.0	Epson Toyocom Corporation	MA-406 (SMD)
8.0	Epson Toyocom Corporation	MA-406 (SMD)

(3) OSC3 ceramic resonators

Oscillation frequency [MHz]	Manufacturer	Product number
4.0	Murata Manufacturing Co., Ltd.	CSTCR4M00G53-R0 (SMD)
	Murata Manufacturing Co., Ltd.	CSTCR4M00G53095-R0 (SMD)
	Murata Manufacturing Co., Ltd.	CSTLS4M00G53095-B0 (leaded)
8.0	Murata Manufacturing Co., Ltd.	CSTLS8M00G53095-B0 (leaded)

Note: Please ask the manufacturer to evaluate the resonator mounted on the circuit board.

(4) C_{G3}/C_{D3}/R_{D3}/R_{F3} recommended values

Name	Frequency [MHz]	Product number	C_{G3} [pF]	C_{D3} [pF]	R_{D3} [Ω]	R_{F3} [Ω]
X'tal3	4.0	MA-406 (C_L: 10pF)	6	6	0	1M
	8.0	MA-406 (C_L: 8pF)	2	2	0	1M
Ceramic3	4.0	CSTCR4M00G53-R0	(15)*	(15)*	0	1M
		CSTCR4M00G53095-R0	(15)*	(15)*	0	1M
		CSTLS4M00G53095-B0	(15)*	(15)*	0	1M
	8.0	CSTLS8M00G53095-B0	(15)*	(15)*	0	1M

* The values enclosed with ( ) are the built-in capacitances of the resonator.

Revision History

Code No.	Page	Contents
411914900	All	New establishment
411914901	Second cover	Notice (Old) No description (New) This product uses SuperFlash® technology licensed from Silicon Storage Technology, Inc.
	1-1	Features: Clock generator - Other (Old) • IOSC control for quick-restart processing from SLEEP mode (New) Deleted
	1-8, 1-11	Pad configuration diagram: Chip (S1C17604/S1C17622) Modified Figures 1.3.2.2 and 1.3.3.2
	7-1	CLG: CLG module overview (Old) • Supports quick-restart processing from SLEEP mode. Turns IOSC on forcibly and switches the system clock to IOSC when SLEEP mode is canceled. (New) Deleted
		CLG: CLG module configuration Modified Figure 7.1.1
	7-7, 7-11	CLG: System clock switching (Old) • When SLEEP mode is canceled, the IOSC oscillator circuit is turned on (IOSCEN = 1) and is ... Canceling HALT mode does not change the clock status configured before the chip entered HALT mode. (New) • Canceling HALT/SLEEP mode does not change the clock status configured before the chip entered HALT/SLEEP mode.
	7-8	CLG: Peripheral module clock control circuit Modified Figure 7.6.1 and Table 7.6.2
		CLG: Peripheral module clock (PCLK) control (Old) No description (New) Note: The interrupt controller (ITC) needs PCLK only when the register is set.
	7-9	CLG: FOUTH/FOUT1 output Modified Figures 7.7.2 and 7.7.3
	7-14, 7-15	CLG: PCLK Control Register (CLG_PCLK) (Old) No description (New) Peripheral modules that use PCLK ... • Interrupt controller (ITC) ... Notes: • ... • The interrupt controller (ITC) needs PCLK only when the register is set.
	9-5, 9-11	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.
	10-4	T8F: T8F output signals (Old) No description (New) Use the following equations to calculate the reload data register value ... TFMD: Fine mode setting (0 to 15)
	12-6	T16E: Precautions on fine mode (Old) No description (New) (3) When fine mode is used, set compare data with B < A / 2 + 0x8000.
	13-6	T16A2: One-shot mode (TRMD = 1) (Old) No description (New) The counter is not cleared to 0 after the count operation is completed in one-shot mode. ... For more information on resetting methods, see Section 13.5.1, “Counter Reset.”
	13-7	T16A2: Counter RUN/STOP control (Old) No description (New) Note: After the T16A_CCA*x* and T16A_CCB*x* registers ... and then run the counter.
	13-8	T16A2: Operation timing in capture mode Modified Figure 13.5.4.2
	13-18	T16A2: T16A Comparator/Capture Ch.*x* A/B Data Registers (T16A_CCA*x*/CCB*x*) (Old) No description (New) Note: After the T16A_CCA*x*/CCBx registers ... and then run the counter. In capture mode (CCAMD = 1)/(CCBMD = 1) When the counter value is captured at the external trigger signal (CAPA*x*/CAPB*x*) edge selected ... thus the captured counter value can be read out in the interrupt handler.

REVISION HISTORY

Code No.	Page	Contents
411914901	18-4, 18-5	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.
	19-2	SPI: SPI clock Modified Figure 19.3.1
		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.
	19-4	SPI: Data transmission timing chart Deleted Figure 19.5.1
		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 19.5.1) ... transmit data bits and the second and following bytes during continuous transfer.
	19-5	SPI: Data transmission/receiving timing chart Modified Figure 19.5.2
		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.
	19-7	SPI: SPI Ch.*x* Transmit Data Register (SPI_TXD*x*) (Old) No description (New) Note: Make sure that SPEN is set to 1 before writing data ... to start data transmission/reception.
	20-2	I2CM: I^{2}C connection example Added Figure 20.2.1
		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.
	20-3	I2CM: Transmit data specifying slave address and transfer direction Modified Figure 20.5.2
	20-4	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.
	20-5	I2CM: End of data transfers (Generating stop condition) (Old) The stop condition generation can be reserved. ... data transfer (including ACK transfer) ends. (New) Before STP can be set to 1, confirm that TBUSY or RBUSY is reset to 0 from 1 (this indicates that ... has finished and the time for the slave device to finish clock stretching has elapsed.
		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.
	20-5, 20-6	I2CM: Timing chart Modified Figures 20.5.6 to 20.5.9
	20-8	I2CM: I^{2}C 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
	20-9	I2CM: I^{2}C 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.
	21-1	I2CS: I2CS Module overview Modified Figure 21.1.1
		I2CS: List of I2CS Pins - SCL1 Modified Table 21.2.1
	21-2	I2CS: I^{2}C connection example Added Figure 21.2.1

Code No.	Page	Contents
411914901	21-2	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) cycles 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.
	21-3	I2CS: Clock stretch function (Old) No description (New) Note that the data setup time ... depending on the I2CS module operating clock (PCLK) frequency.
	21-4	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.
	21-4, 21-5, 21-10	I2CS: Data transmission (Old) No description (New) When the asynchronous address detection function is used, ... after TXEMP has been set to 1.
	21-6	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).
	21-7, 21-8	I2CS: Timing charts Modified Figures 21.5.5 to 21.5.8
	21-9	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.
	21-15	I2CS: I^2C Slave Status Register (I2CS_STAT) - (D0) DA_STOP: Stop Condition Detect Bit (Old) Indicates that a stop condition is detected. ... I^2C 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 I^2C communication process.
		I2CS: I^2C Slave Access Status Register (I2CS_ASTAT) - (D1) SELECTED: I^2C 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.
	24-1	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)
	24-5	ADC: Expression for calculating sampling rate (Old) No description (New) The following shows the relation between sampling time and sampling rate. Sampling rate [sps] = ...
	24-12	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.
	29-5	Electrical characteristics: Oscillation characteristics - OSC3 crystal oscillation (Old) Unless otherwise specified: V_{DD} = 1.8 to 3.6V, V_{SS} = 0V, Ta = 25°C, C_{G3} = C_{D3} = 15pF ... ∗2 The oscillation start time varies ... the crystal resonator used and the C_{G3} and C_{D3} values. (New) Unless otherwise specified: V_{DD} = 1.8 to 3.6V, V_{SS} = 0V, Ta = 25°C, R_{F3} = 1MΩ ... ∗1 Crystal resonator = MA-406: manufactured by EPSON TOYOCOM (R_1 = 150Ω, C_L = 10pF)
		Electrical characteristics: Oscillation characteristics - OSC3 ceramic oscillation (Old) Unless otherwise specified: V_{DD} = 1.8 to 3.6V, V_{SS} = 0V, Ta = 25°C ... ∗2 The oscillation start time varies ... the ceramic resonator used and the C_{G3} and C_{D3} values. (New) Unless otherwise specified: V_{DD} = 1.8 to 3.6V, V_{SS} = 0V, Ta = 25°C, R_{F3} = 1MΩ ... ∗1 Ceramic resonator = CSTR4M00G53095-R0: ... (C_{G3} = C_{D3} = 15pF built-in)
	29-6	Electrical characteristics: Oscillation characteristics - IOSC oscillation frequency-temperature characteristic (Old) f_{IOSC} [Hz] (New) f_{IOSC} [kHz]
	29-9	Electrical characteristics: LCD driver characteristics - LCD drive voltage (V_{C2} reference voltage) (Old) Unless otherwise specified: V_{DD} = 1.8 to 3.6V, ... (New) Unless otherwise specified: V_{DD} = 2.5 to 3.6V, ...
	29-13	Electrical characteristics: S1C17624/604/622 R/F converter current consumption, S1C17602/621 R/F converter current consumption (Old) Unless otherwise specified: ... PCKEN[1:0] = 0x3 (ON), ... R_{REF}/R_{SEN} = 100kΩ (New) Unless otherwise specified: ... PCKEN[1:0] = 0x0 (OFF), ... R_{REF}/R_{SEN} = 100kΩ, TCCLK = 8MHz

REVISION HISTORY

Code No.	Page	Contents
411914901	29-13, 29-14	Electrical characteristics: RFC reference/sensor oscillation current consumption-frequency characteristic (Old) R_{REF}/R_{SEN} = 100kΩ, C_{REF}/C_{SEN} = 1000pF, Typ. value (New) C_{REF}/C_{SEN} = 1000pF, Typ. value
	30-1	Basic external connection diagram: Buzzer circuit and OSC3 oscillator circuit Modified the figure
	AP-B-1	Power saving: System SLEEP (Old) • Execute the slp instruction Execute the slp instruction when the entire system ... startup using a port or RTC (described later). (New) • Execute the slp instruction To stop the entire system, execute the slp instruction ... an RTC interrupt as well as a port interrupt.
	AP-B-2	Power saving: Peripheral clock (PCLK) (Old) Peripheral circuits that use PCLK ... • A/D converter (ADC10) (New) Peripheral circuits that use PCLK ... • A/D converter (ADC10) • Interrupt controller (ITC) (PCLK is used only when the register is set.)
		Power saving: Clock control list Modified Table B.1.1
	AP-B-2	Power saving: HALT and SLEEP mode cancelation methods (CPU startup method) (Old) 1. Startup by port or RTC ... 3. Startup by PCLK peripheral circuit Started up by a PCLK peripheral circuit interrupt. (New) 1. Startup by port ... 2. Startup by RTC (S1C17624/604) ... 4. Startup by PCLK peripheral circuit Started up by a PCLK peripheral circuit interrupt.
	AP-D-1	Initialization routine (Old) Xld.a %sp, 0x0f00 ...(4) (New) Xld.a %sp, 0x0f00 ; S1C17622/602 ...(4) ;Xld.a %sp, 0x1f00 ; S1C17624/604
	AP-E-1	Recommended resonators: Recommended resonators for S1C17624/604/622 - (4) $C_{G3}/C_{D3}/R_{D3}/R_{F3}$ recommended values Modified the table
	AP-E-2	Recommended resonators: Recommended resonators for S1C17602/621 (Old) No description (New) (4) $C_{G3}/C_{D3}/R_{D3}/R_{F3}$ recommended values ... * The values enclosed with ( ) are the built-in capacitances of the resonator.
411914902	3-6	IRAM Size Select Register (MISC_IRAMSZ) (Old) S1C17602: IRAMSZ[2:0] 0x3/0x2 -> 2KB/4KB (New) S1C17602: IRAMSZ[2:0] 0x3/0x2 -> reserved
	3-7	IRAM Size Select Register (MISC_IRAMSZ) (Old) S1C17621: IRAMSZ[2:0] 0x3 -> 2KB (New) S1C17621: IRAMSZ[2:0] 0x3 -> reserved Table 3.3.1.2 Selecting internal RAM Size (Old) S1C17602: IRAMSZ[2:0] 0x3/0x2 -> 2KB/4KB S1C17621: IRAMSZ[2:0] 0x3 -> 2KB (New) S1C17602: IRAMSZ[2:0] 0x3/0x2 -> reserved S1C17621: IRAMSZ[2:0] 0x3 -> reserved Add Notice
	AP-A-23	IRAM Size Select Register (MISC_IRAMSZ) (Old) S1C17602: IRAMSZ[2:0] 0x3/0x2 -> 2KB/4KB S1C17621: IRAMSZ[2:0] 0x3 -> 2KB (New) S1C17602: IRAMSZ[2:0] 0x3/0x2 -> reserved S1C17621: IRAMSZ[2:0] 0x3 -> reserved
411914803	1-1	Features: Clock generator - OSC1 oscillator circuit (Old) Supports an external clock input. (New) Deleted
	3-2	Memory map Mofified Figures 3.2–3.3
	3-5	Memory Map: Protect Bits Modified the Flash Protect Bits (S1C17622/602) and Flash Protect Bits (S1C17621) tables
		Memory map: Protect Bits (Old) Notes: ... • Be sure to set D0 of address 0x27ffe (S1C17624/604), 0x17ffe (S1C17622/602), or 0xfffe (S1C17621) to 1. If it is set to 0, the program cannot be booted. (New) Notes: ... • Be sure to set D0 of address 0x27ffe (S1C17624/604/622) or 0x17ffe (S1C17602/621) to 1. If it is set to 0, the program cannot be booted.

Code No.	Page	Contents
411914803	3-6	Memory map: Embedded RAM (Old) The S1C17624/604 enables the RAM size used to apply restrictions to 8KB, 4KB, or 2KB. The S1C17602 enables the RAM size used to apply restrictions to 4KB or 2KB. For example, when using the S1C17624/604/602 to develop an application for a built-in ROM model ... (New) The S1C17624/604 enables the RAM size used to apply restrictions to 8KB, 4KB, or 2KB. The S1C17622 enables the RAM size used to apply restrictions to 4KB or 2KB. For example, when using the S1C17624/604/622 to develop an application for a built-in ROM model ...
	12-6	T16E: Precautions (Old) No description (New) (4) Be sure to set T16EDF[3:0]/T16E_DFx register to 0x0 (PCLK•1/1) when using fine mode.
	13-15	T16A2: T16A Counter Ch.*x* Control Registers (T16A_CTL*x*) - (D3) CBUFEN: Compare Buffer Enable Bit (Old) Note: Make sure the counter is halted (PRUN = 0) before setting PRESET. (New) Note: Make sure the counter is halted (CLKEN/T16A_CLK*x* register = 0) before setting PRESET.
	27-4	DBG: IRAM Size Select Register (MISC_IRAMSZ) (S1C17602, S1C17621) - (D[2:0]) RAMSZ[2:0]: IRAM Size Select Bits Modified the register tables and Table 27.4.2 (RAMSZ[2:0] settings → reserved)
	AP-D-1	Initialization routine (Old) ; ----- Stack pointer ---------- Xld.a %sp, 0x0f00 ; S1C17622/602 ...(4) ;Xld.a %sp, 0x1f00 ; S1C17624/604 (New) ; ----- Stack pointer ---------- Xld.a %sp, 0x0fc0 ; S1C17622/602 ...(4) ;Xld.a %sp, 0x1fc0 ; S1C17624/604

EPSON

International Sales Operations

AMERICA

EPSON ELECTRONICS AMERICA, INC.
214 Devcon Drive,
San Jose, CA 95112, USA
Phone: +1-800-228-3964 Fax: +1-408-922-0238

EUROPE

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

ASIA

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

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

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

EPSON HONG KONG LTD.
Unit 715-723, 7/F Trade Square, 681 Cheung Sha Wan Road,
Kowloon, Hong Kong
Phone: +852-2585-4600 Fax: +852-2827-4346

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

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

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

SEIKO EPSON CORP.
MICRODEVICES OPERATIONS DIVISION

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

Document Code: 411914903
First Issue April 2010
Revised October 2013 in JAPAN Ⓛ