EPSON
EXCEED YOUR VISION

CMOS 16-BIT SINGLE CHIP MICROCONTROLLER

S1C17702
Technical Manual

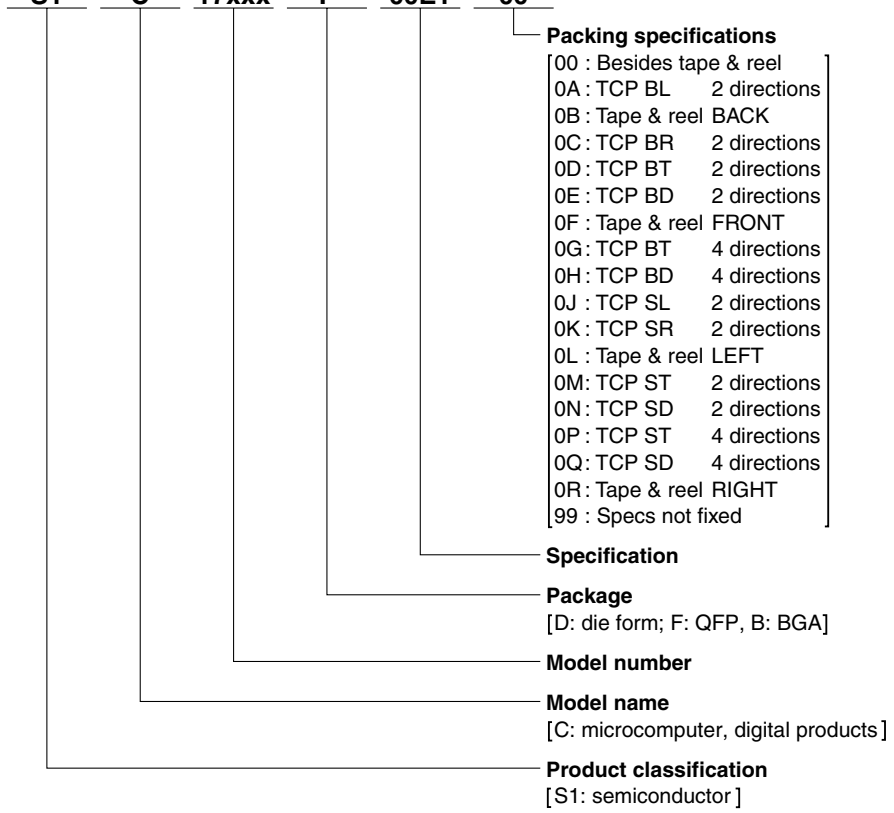
SEIKO EPSON CORPORATION

Rev.1.2

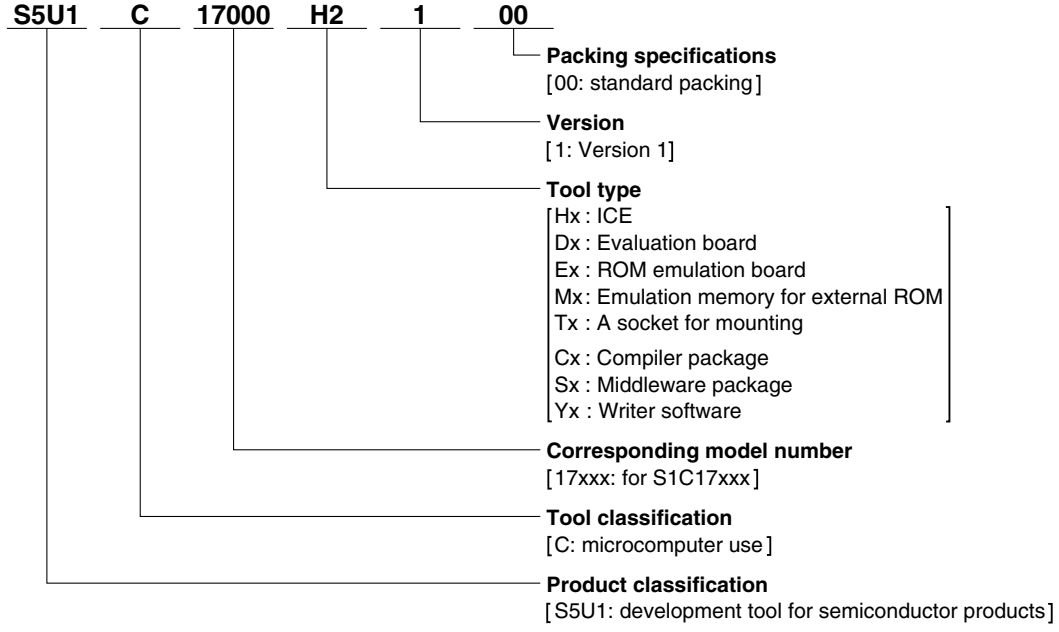
NOTICE

Configuration of product number

Devices



Development tools



– Contents –

1 Overview

The S1C17702 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 incorporates a 128KB of flash memory, a 12KB of RAM, serial interfaces such as UART supporting high-bit rate and IrDA1.0, SPI, and I²C to support various sensors, 8-bit timers, 16-bit timers, PWM & capture timers, a clock timer, a stopwatch timer, a watchdog timer, 28 general input/output ports, max. 72 segments × 32 commons LCD driver and a power supply voltage booster circuit, a power supply voltage detection circuit, 32 kHz and max. 8.2 MHz oscillator circuits, and a 1.8 V voltage regulator.
It allows 8.2 MHz high-speed operation at an operating voltage of just 1.8 V, and executes single instructions using a single clock with 16-bit RISC processing. The product also incorporates a coprocessor for arithmetic functions for multiplication and product-sum operations.
The on-chip ICE function allows onboard programs/deletes of internal flash memory and program debugging and evaluations following connection of the three signal wires to the ICD Mini (S5U1C17001H).
The S1C17702 is ideal for applications (such as remote controllers and sports watches) requiring battery power and sensor interface and for LCD displays of up to 72 × 32 dots.

∗ **This product uses SuperFlash® technology licensed by Silicon Storage Technology, Inc.**

1.1 Features

The main features of the S1C17702 are listed below.

CPU	• Epson original 16-bit RISC CPU core S1C17 • 16-bit × 16-bit + 32-bit product-sum processor • 16-bit ÷ 16-bit divider
IOSC oscillator circuit	• 2.7 MHz (typ.)
OSC3 oscillator circuit	• Crystal oscillator circuit or ceramic oscillator circuit, 8.2 MHz (max.)
OSC1 oscillator circuit	• Crystal oscillator circuit 32.786 kHz (typ.)
Internal flash memory	• 128K bytes (for both instructions and data) • Allows 1,000 overwrites (min.) • Read/program protection function • Allows onboard rewriting with the ICD Mini (S5U1C17001H) debug tool and self-rewriting via software.
Internal RAM	• 12K bytes
Internal display RAM	• 576 bytes
Input/output port	• Max. 28-bit general purpose input/output (shared with peripheral circuit input/output pins)
Serial interface	• SPI (master/slave) 1ch. • I^2C (master) 1ch. • UART (460,800 bps, IrDA1.0 compatible) 2ch. • Remote controller (REMC) 1ch.
Timer	• 8-bit timer (T8F) 2ch. • 16-bit timer (T16) 3ch. • PWM& capture timer (T16E) 2ch. • Clock timer (CT) 1ch. • Stopwatch timer (SWT) 1ch. • Watchdog timer (WDT) 1ch. • 8-bit OSC1 timer (T8OSC1) 1ch.
LCD driver	• 72 SEG × 32 COM or 88 SEG × 16 COM (1/5 bias) • Internal booster power supply circuit (16-step programmable contrast)
Power supply voltage detection (SVD) circuit	• 15-value programmable (1.8 V to 3.2 V)
Interrupt	• Reset • NMI • Programmable interrupt × 18 (8 levels)
Power supply voltage	• 1.8 V to 3.6 V (for normal operations, internal regulator-based 1.8 V low-power operations) • 2.7 V to 3.6 V (for flash erase/writing, internal 2.5 V operations) • Internal constant-voltage circuit (2-step programmable operating voltage)
Operating temperature	• -25°C to 70°C

Current consumption

- SLEEP mode: 1.2 µA typ. (OSC1 = OFF, IOSC = OFF, OSC3 = OFF)
- HALT mode: 2.7 µA typ. (OSC1 = 32 kHz, IOSC = OFF, OSC3 = OFF, PCKEN = 0x0, LCD OFF)
  9.7 µA typ. (OSC1 = 32 kHz, IOSC = OFF, OSC3 = OFF, PCKEN = 0x0, LCD ON (All on, contrast max.))
- When operating: 16 µA typ. (OSC1 = 32 kHz, IOSC = OFF, OSC3 = OFF, LCD OFF)
  450 µA typ. (OSC1 = OFF, IOSC = OFF, OSC3 = 1 MHz ceramic oscillator)

Configuration as shipped

- QFP21-176 pin package (26 mm × 26 mm, pin pitch: 0.5 mm)
- VFBGA8H-181 package (8 mm × 8 mm, ball pitch: 0.5 mm)
- VFBGA10H-180 package (10 mm × 10 mm, ball pitch: 0.65 mm)
- Chip

1.2 Block Diagram

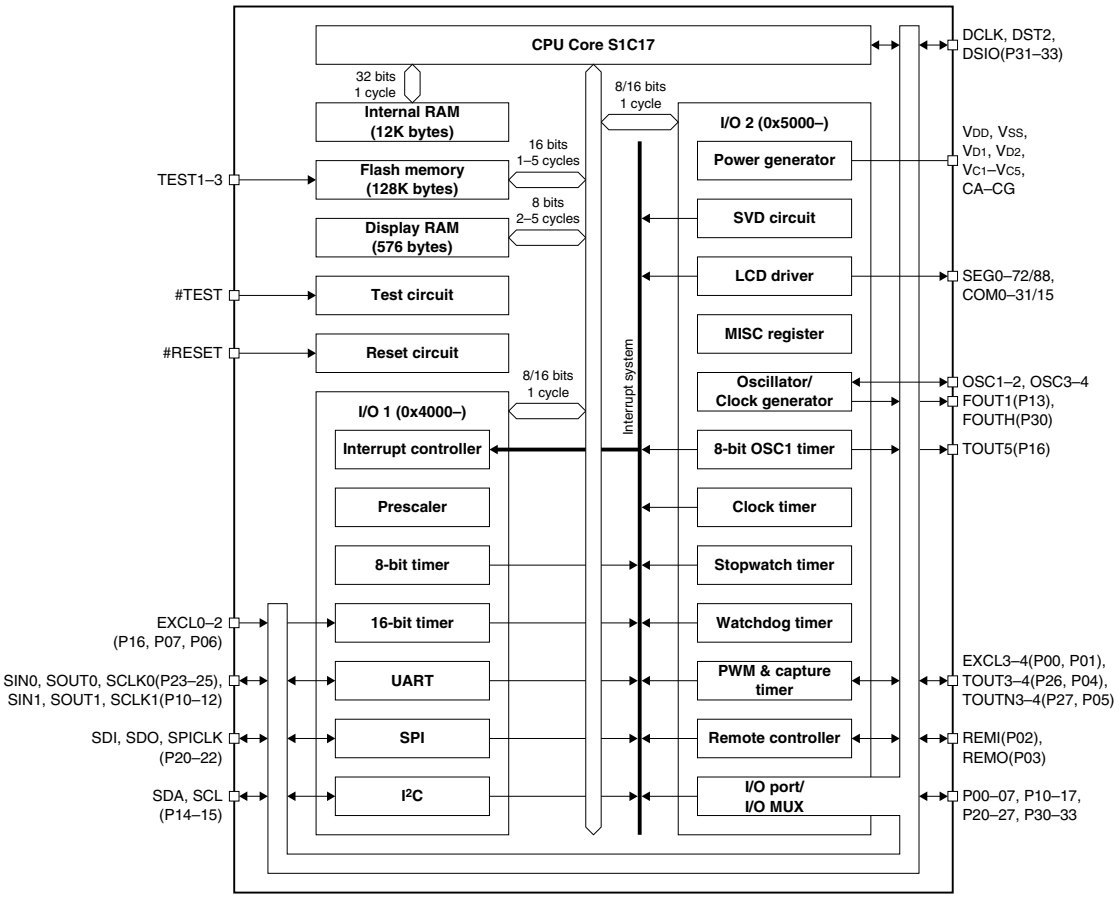


Figure 1.2.1: Block diagram

1.3 Pins

1.3.1 Pinout Diagram

QFP21-176 pin

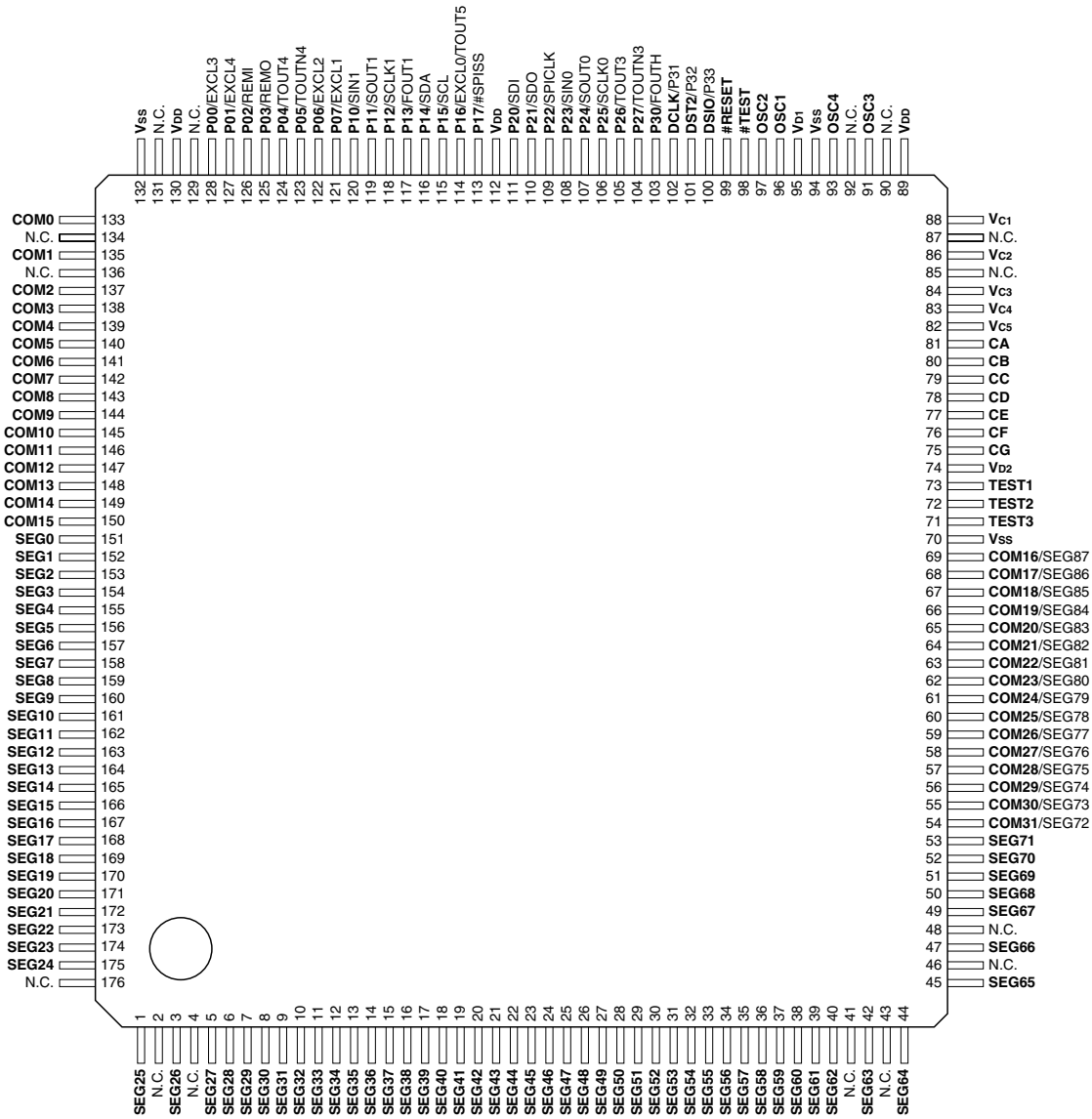
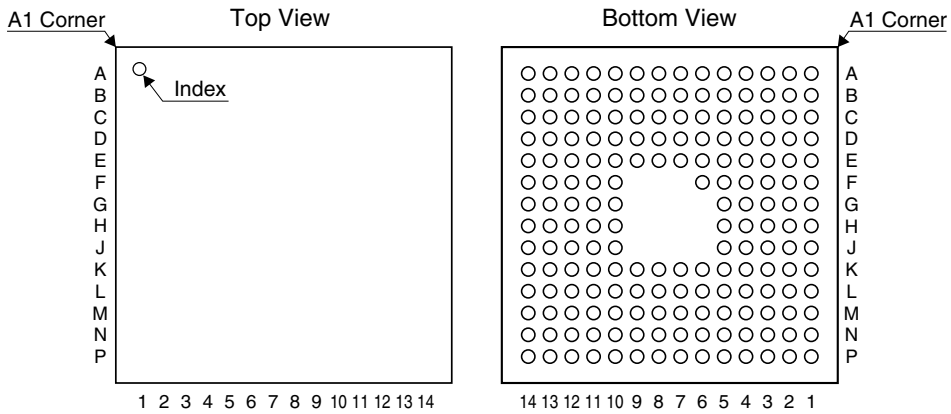


Figure 1.3.1.1: Pinout diagram (QFP21-176 pin)

VFBGA8H-181

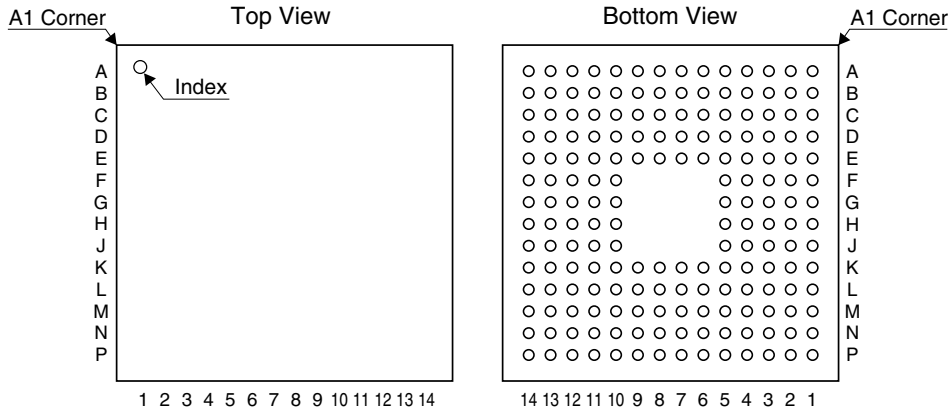


	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
A	N.C.	SEG22	SEG20	SEG17	SEG14	SEG9	SEG6	SEG0	COM15	COM9	COM5	COM2	COM0	N.C.	A
B	SEG25	SEG24	SEG21	SEG18	SEG15	SEG10	SEG7	SEG1	COM13	COM8	COM4	COM1	VDD	VSS	B
C	SEG27	SEG26	SEG23	SEG19	SEG16	SEG11	SEG8	SEG2	COM12	COM7	COM3	P00 EXCL3	P01 EXCL4	P04 TOUT4	C
D	SEG31	SEG30	SEG29	SEG28	SEG13	SEG12	SEG5	SEG3	COM11	COM6	P02 REMI	P03 REMO	P06 EXCL2	P05 TOUTN4	D
E	SEG35	SEG34	SEG33	SEG32	VSS	VSS	SEG4	COM14	COM10	VSS	P07 EXCL1	P12 SCLK1	P14 SDA	P13 FOUT1	E
F	SEG39	SEG38	SEG37	SEG36	VSS	VSS				VSS	P10 SIN1	P16 EXCL0 TOUT5	P15 SCL	P17 #SPISS	F
G	SEG44	SEG43	SEG42	SEG41	SEG40					VSS	P11 SOUT1	P21 SDO	VDD	P20 SDI	G
H	SEG45	SEG46	SEG47	SEG48	VSS					VSS	P25 SCLK0	P24 SOUT0	P23 SIN0	P22 SPICLK	H
J	SEG49	SEG50	SEG51	SEG52	SEG56					VSS	DCLK P31	P30 FOUTH	P26 TOUT3	P27 TOUTN3	J
K	SEG53	SEG54	SEG55	SEG57	COM30 SEG73	VSS	VSS	VSS	VSS	VSS	VC4	DST2 P32	#RESET	DSIO P33	K
L	SEG58	SEG59	SEG60	SEG68	COM29 SEG74	COM25 SEG78	COM23 SEG80	VSS	CE	CD	VC5	#TEST	OSC2	OSC1	L
M	SEG61	SEG62	SEG67	SEG69	COM28 SEG75	COM24 SEG79	COM20 SEG83	VSS	TEST1	VD2	CA	VC1	VD1	VSS	M
N	SEG63	SEG64	SEG66	SEG71	COM27 SEG76	COM21 SEG82	COM19 SEG84	COM17 SEG86	TEST3	CG	CC	VC2	OSC4	VDD	N
P	N.C.	SEG65	SEG70	COM31 SEG72	COM26 SEG77	COM22 SEG81	COM18 SEG85	COM16 SEG87	TEST2	CF	CB	VC3	OSC3	N.C.	P
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	

Top View

Figure 1.3.1.2: Pinout diagram (VFBGA8H-181)

VFBGA10H-180

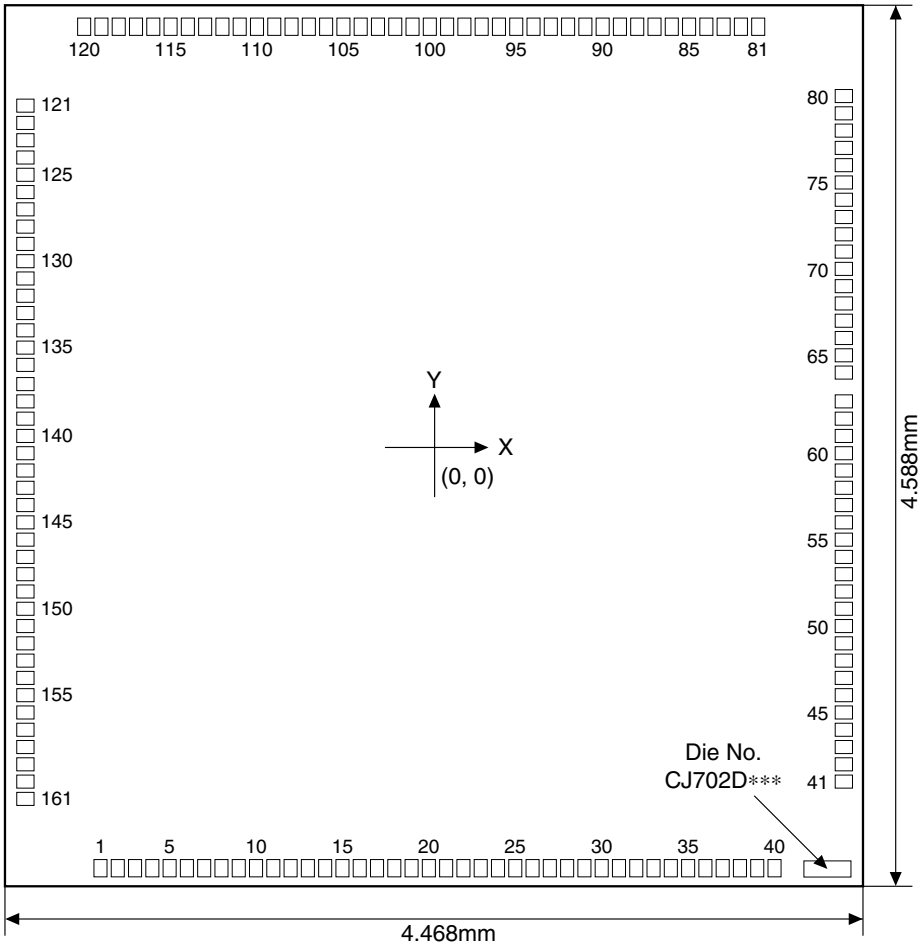


	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
A	N.C.	SEG22	SEG20	SEG17	SEG14	SEG9	SEG6	SEG0	COM15	COM9	COM5	COM2	COM0	N.C.	A
B	SEG25	SEG24	SEG21	SEG18	SEG15	SEG10	SEG7	SEG1	COM13	COM8	COM4	COM1	VDD	VSS	B
C	SEG27	SEG26	SEG23	SEG19	SEG16	SEG11	SEG8	SEG2	COM12	COM7	COM3	P00 EXCL3	P01 EXCL4	P04 TOUT4	C
D	SEG31	SEG30	SEG29	SEG28	SEG13	SEG12	SEG5	SEG3	COM11	COM6	P02 REMI	P03 REMO	P06 EXCL2	P05 TOUTN4	D
E	SEG35	SEG34	SEG33	SEG32	VSS	VSS	SEG4	COM14	COM10	VSS	P07 EXCL1	P12 SCLK1	P14 SDA	P13 FOUT1	E
F	SEG39	SEG38	SEG37	SEG36	VSS					VSS	P10 SIN1	P16 EXCL0 TOUT5	P15 SCL	P17 #SPISS	F
G	SEG44	SEG43	SEG42	SEG41	SEG40					VSS	P11 SOUT1	P21 SDO	VDD	P20 SDI	G
H	SEG45	SEG46	SEG47	SEG48	VSS					VSS	P25 SCLK0	P24 SOUT0	P23 SIN0	P22 SPICLK	H
J	SEG49	SEG50	SEG51	SEG52	SEG56					VSS	DCLK P31	P30 FOUTH	P26 TOUT3	P27 TOUTN3	J
K	SEG53	SEG54	SEG55	SEG57	COM30 SEG73	VSS	VSS	VSS	VSS	VSS	VC4	DST2 P32	#RESET	DSIO P33	K
L	SEG58	SEG59	SEG60	SEG68	COM29 SEG74	COM25 SEG78	COM23 SEG80	VSS	CE	CD	VC5	#TEST	OSC2	OSC1	L
M	SEG61	SEG62	SEG67	SEG69	COM28 SEG75	COM24 SEG79	COM20 SEG83	VSS	TEST1	VD2	CA	VC1	VD1	VSS	M
N	SEG63	SEG64	SEG66	SEG71	COM27 SEG76	COM21 SEG82	COM19 SEG84	COM17 SEG86	TEST3	CG	CC	VC2	OSC4	VDD	N
P	N.C.	SEG65	SEG70	COM31 SEG72	COM26 SEG77	COM22 SEG81	COM18 SEG85	COM16 SEG87	TEST2	CF	CB	VC3	OSC3	N.C.	P
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	

Top View

Figure 1.3.1.3: Pinout diagram (VFBGA10H-180)

Chip



(Pad opening) No. 1 to 40, 81 to 120: 68 × 90 µm
No. 41 to 80, 121 to 161: 90 × 68 µm

Figure 1.3.1.3: Pad layout diagram

Table 1.3.1.1: Pad coordinates

Pad No.	Pad name	X (mm)	Y(mm)	Pad No.	Pad name	X (mm)	Y(mm)
1	SEG25	-1.740	-2.191	81	V_{DD}	1.755	2.191
2	SEG26	-1.650	-2.191	82	OSC3	1.665	2.191
3	SEG27	-1.560	-2.191	83	OSC4	1.575	2.191
4	SEG28	-1.470	-2.191	84	V_{SS}	1.485	2.191
5	SEG29	-1.380	-2.191	85	V_{D1}	1.395	2.191
6	SEG30	-1.290	-2.191	86	OSC1	1.235	2.191
7	SEG31	-1.200	-2.191	87	OSC2	1.145	2.191
8	SEG32	-1.110	-2.191	88	#TEST	1.055	2.191
9	SEG33	-1.020	-2.191	89	#RESET	0.965	2.191
10	SEG34	-0.930	-2.191	90	DSIO/P33	0.875	2.191
11	SEG35	-0.840	-2.191	91	DST2/P32	0.785	2.191
12	SEG36	-0.750	-2.191	92	DCLK/P31	0.695	2.191
13	SEG37	-0.660	-2.191	93	P30/FOUTH	0.605	2.191
14	SEG38	-0.570	-2.191	94	P27/TOUTN3	0.515	2.191
15	SEG39	-0.480	-2.191	95	P26/TOUT3	0.425	2.191
16	SEG40	-0.390	-2.191	96	P25/SCLK0	0.335	2.191
17	SEG41	-0.300	-2.191	97	P24/SOUT0	0.245	2.191
18	SEG42	-0.210	-2.191	98	P23/SIN0	0.155	2.191
19	SEG43	-0.120	-2.191	99	P22/SPICLK	0.065	2.191
20	SEG44	-0.030	-2.191	100	P21/SDO	-0.025	2.191
21	SEG45	0.060	-2.191	101	P20/SDI	-0.115	2.191

Pad No.	Pad name	X (mm)	Y(mm)	Pad No.	Pad name	X (mm)	Y(mm)
22	SEG46	0.150	-2.191	102	V_{DD}	-0.205	2.191
23	SEG47	0.240	-2.191	103	P17/#SPISS	-0.295	2.191
24	SEG48	0.330	-2.191	104	P16/EXCL0/TOUT5	-0.385	2.191
25	SEG49	0.420	-2.191	105	P15/SCL	-0.475	2.191
26	SEG50	0.510	-2.191	106	P14/SDA	-0.565	2.191
27	SEG51	0.600	-2.191	107	P13/FOUT1	-0.655	2.191
28	SEG52	0.690	-2.191	108	P12/SCLK1	-0.745	2.191
29	SEG53	0.780	-2.191	109	P11/SOUT1	-0.835	2.191
30	SEG54	0.870	-2.191	110	P10/SIN1	-0.925	2.191
31	SEG55	0.960	-2.191	111	P07/EXCL1	-1.015	2.191
32	SEG56	1.050	-2.191	112	P06/EXCL2	-1.105	2.191
33	SEG57	1.140	-2.191	113	P05/TOUTN4	-1.195	2.191
34	SEG58	1.230	-2.191	114	P04/TOUT4	-1.285	2.191
35	SEG59	1.320	-2.191	115	P03/REMO	-1.375	2.191
36	SEG60	1.410	-2.191	116	P02/REMI	-1.465	2.191
37	SEG61	1.500	-2.191	117	P01/EXCL4	-1.555	2.191
38	SEG62	1.590	-2.191	118	P00/EXCL3	-1.645	2.191
39	SEG63	1.680	-2.191	119	V_{DD}	-1.735	2.191
40	SEG64	1.770	-2.191	120	V_{SS}	-1.825	2.191
41	SEG65	2.131	-1.740	121	COM0	-2.131	1.780
42	SEG66	2.131	-1.650	122	COM1	-2.131	1.690
43	SEG67	2.131	-1.560	123	COM2	-2.131	1.600
44	SEG68	2.131	-1.470	124	COM3	-2.131	1.510
45	SEG69	2.131	-1.380	125	COM4	-2.131	1.420
46	SEG70	2.131	-1.290	126	COM5	-2.131	1.330
47	SEG71	2.131	-1.200	127	COM6	-2.131	1.240
48	COM31	2.131	-1.100	128	COM7	-2.131	1.150
49	COM30	2.131	-1.010	129	COM8	-2.131	1.060
50	COM29	2.131	-0.920	130	COM9	-2.131	0.970
51	COM28	2.131	-0.830	131	COM10	-2.131	0.880
52	COM27	2.131	-0.740	132	COM11	-2.131	0.790
53	COM26	2.131	-0.650	133	COM12	-2.131	0.700
54	COM25	2.131	-0.560	134	COM13	-2.131	0.610
55	COM24	2.131	-0.470	135	COM14	-2.131	0.520
56	COM23	2.131	-0.380	136	COM15	-2.131	0.430
57	COM22	2.131	-0.290	137	SEG0	-2.131	0.330
58	COM21	2.131	-0.200	138	SEG1	-2.131	0.240
59	COM20	2.131	-0.110	139	SEG2	-2.131	0.150
60	COM19	2.131	-0.020	140	SEG3	-2.131	0.060
61	COM18	2.131	0.070	141	SEG4	-2.131	-0.030
62	COM17	2.131	0.160	142	SEG5	-2.131	-0.120
63	COM16	2.131	0.250	143	SEG6	-2.131	-0.210
64	V_{SS}	2.131	0.400	144	SEG7	-2.131	-0.300
65	TEST3	2.131	0.490	145	SEG8	-2.131	-0.390
66	TEST2	2.131	0.580	146	SEG9	-2.131	-0.480
67	TEST1	2.131	0.670	147	SEG10	-2.131	-0.570
68	V_{D2}	2.131	0.760	148	SEG11	-2.131	-0.660
69	CG	2.131	0.850	149	SEG12	-2.131	-0.750
70	CF	2.131	0.940	150	SEG13	-2.131	-0.840
71	CE	2.131	1.030	151	SEG14	-2.131	-0.930
72	CD	2.131	1.120	152	SEG15	-2.131	-1.020
73	CC	2.131	1.210	153	SEG16	-2.131	-1.110
74	CB	2.131	1.300	154	SEG17	-2.131	-1.200
75	CA	2.131	1.390	155	SEG18	-2.131	-1.290
76	V_{C5}	2.131	1.480	156	SEG19	-2.131	-1.380
77	V_{C4}	2.131	1.570	157	SEG20	-2.131	-1.470
78	V_{C3}	2.131	1.660	158	SEG21	-2.131	-1.560
79	V_{C2}	2.131	1.750	159	SEG22	-2.131	-1.650
80	V_{C1}	2.131	1.840	160	SEG23	-2.131	-1.740
–	–	–	–	161	SEG24	-2.131	-1.830

1.3.2 Pin Descriptions

Table 1.3.2.1: Pin descriptions

Pin/ball No.		Name	I/O	Default status	Function
QFP	VFBGA				
1, 3, 5 to 40, 42, 44 to 45, 47, 49 to 53	*2	**SEG25 to 71**	O	O (L)	LCD segment output pin
54 to 69	*3	**COM31 to 16**/ SEG72 to 87	O	O (L)	LCD common output pin*1/LCD segment output pin
70	*4	**VSS**	–	–	Power supply pin (GND)
71	N9	**TEST3**	–	–	Flash test pin (open during normal operations)
72	P9	**TEST2**	–	–	Flash test pin (fixed to High during normal operations)
73	M9	**TEST1**	–	–	Flash test pin (open during normal operations)
74	M10	**VD2**	–	–	LCD circuit power supply booster output pin
75	N10	**CG**	–	–	Power supply voltage booster capacitor connector pin
76	P10	**CF**	–	–	Power supply voltage booster capacitor connector pin
77	L9	**CE**	–	–	LCD booster capacitor connector pin
78	L10	**CD**	–	–	LCD booster capacitor connector pin
79	N11	**CC**	–	–	LCD booster capacitor connector pin
80	P11	**CB**	–	–	LCD booster capacitor connector pin
81	M11	**CA**	–	–	LCD booster capacitor connector pin
82	L11	**VC5**	–	–	LCD circuit drive voltage output pin
83	K11	**VC4**	–	–	LCD circuit drive voltage output pin
84	P12	**VC3**	–	–	LCD circuit drive voltage output pin
86	N12	**VC2**	–	–	LCD circuit drive voltage output pin
88	M12	**VC1**	–	–	LCD circuit drive voltage output pin
89	N14	**VDD**	–	–	Power supply pin (+)
91	P13	**OSC3**	I	I	OSC3 oscillator input pin
93	N13	**OSC4**	O	O	OSC3 oscillator output pin
94	*4	**VSS**	–	–	Power supply pin (GND)
95	M13	**VD1**	–	–	Internal logic and oscillator circuit voltage regulator output pin
96	L14	**OSC1**	I	I	OSC1 oscillator input pin
97	L13	**OSC2**	O	O	OSC1 oscillator output pin
98	L12	**#TEST**	I	I (Pull-UP)	Test pin (fixed to High during normal operations)
99	K13	**#RESET**	I	I (Pull-UP)	Initial set input pin
100	K14	**DSIO**/P33	I/O	I (Pull-UP)	On-chip debugger data input/output pin*1/input/output port pin
101	K12	**DST2**/P32	I/O	O (L)	On-chip debugger status output pin*1/input/output port pin
102	J11	**DCLK**/P31	I/O	O (H)	On-chip debugger clock output pin*1/input/output port pin
103	J12	**P30**/FOUTH	I/O	I (Pull-UP)	Input/output port pin/OSC3 dividing clock output pin
104	J14	**P27**/TOUTN3	I/O	I (Pull-UP)	Input/output port pin*1/T16E Ch.0 PWM signal inverted output pin
105	J13	**P26**/TOUT3	I/O	I (Pull-UP)	Input/output port pin*1/T16E Ch.0 PWM signal output pin
106	H11	**P25**/SCLK0	I/O	I (Pull-UP)	Input/output port pin*1/UART Ch.0 clock input pin
107	H12	**P24**/SOUT0	I/O	I (Pull-UP)	Input/output port pin*1/UART Ch.0 data output pin
108	H13	**P23**/SIN0	I/O	I (Pull-UP)	Input/output port pin*1/UART Ch.0 data input pin
109	H14	**P22**/SPICLK	I/O	I (Pull-UP)	Input/output port pin*1/SPI clock input/output pin
110	G12	**P21**/SDO	I/O	I (Pull-UP)	Input/output port pin*1/SPI data output pin
111	G14	**P20**/SDI	I/O	I (Pull-UP)	Input/output port pin*1/SPI data input pin
112	G13	**VDD**	–	–	Power supply pin (+)
113	F14	**P17**/#SPISS	I/O	I (Pull-UP)	Input/output port pin (with interrupt)*1/SPI slave select input pin
114	F12	**P16**/EXCL0/ TOUT5	I/O	I (Pull-UP)	Input/output port pin (with interrupt)/T16 Ch.0 external clock input pin T8OSC1 PWM signal output pin
115	F13	**P15**/SCL	I/O	I (Pull-UP)	Input/output port pin (with interrupt)*1/I²C clock output pin
116	E13	**P14**/SDA	I/O	I (Pull-UP)	Input/output port pin (with interrupt)*1/I²C data input/output pin
117	E14	**P13**/FOUT1	I/O	I (Pull-UP)	Input/output port pin (with interrupt)*1/OSC1 clock output pin
118	E12	**P12**/SCLK1	I/O	I (Pull-UP)	Input/output port pin (with interrupt)*1/UART Ch.1 clock input pin
119	G11	**P11**/SOUT1	I/O	I (Pull-UP)	Input/output port pin (with interrupt)*1/UART Ch.1 data output pin
120	F11	**P10**/SIN1	I/O	I (Pull-UP)	Input/output port pin (with interrupt)*1/UART Ch.1 data input pin
121	E11	**P07**/EXCL1	I/O	I (Pull-UP)	Input/output port pin (with interrupt)/T16 Ch.1 external clock input pin
122	D13	**P06**/EXCL2	I/O	I (Pull-UP)	Input/output port pin (with interrupt)/T16 Ch.2 external clock input pin
123	D14	**P05**/TOUTN4	I/O	I (Pull-UP)	Input/output port pin (with interrupt)*1/T16E Ch.1 PWM signal inverted output pin
124	C14	**P04**/TOUT4	I/O	I (Pull-UP)	Input/output port pin (with interrupt)*1/T16E Ch.1 PWM signal output pin
125	D12	**P03**/REMO	I/O	I (Pull-UP)	Input/output port pin (with interrupt)*1/Remote output pin
126	D11	**P02**/REMI	I/O	I (Pull-UP)	Input/output port pin (with interrupt)*1/Remote input pin

Pin/ball No.		Name	I/O	Default status	Function
QFP	VFBGA				
127	C13	**P01**/EXCL4	I/O	I (Pull-UP)	Input/output port pin (with interrupt)/T16E Ch.1 external clock input pin
128	C12	**P00**/EXCL3	I/O	I (Pull-UP)	Input/output port pin (with interrupt)/T16E Ch.0 external clock input pin
130	B13	**VDD**	–	–	Power supply pin (+)
132	*4	**VSS**	–	–	Power supply pin (GND)
133, 135, 137 to 150	*3	**COM0 to 15**	O	O (L)	LCD common output pin
151 to 175	*2	**SEG0 to 24**	O	O (L)	LCD segment output pin

*1: Default function settings

*2: SEG0 to 71 ball numbers

SEG No.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Ball No.	A8	B8	C8	D8	E7	D7	A7	B7	C7	A6	B6	C6	D6	D5	A5	B5
SEG No.	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Ball No.	C5	A4	B4	C4	A3	B3	A2	C3	B2	B1	C2	C1	D4	D3	D2	D1
SEG No.	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
Ball No..	E4	E3	E2	E1	F4	F3	F2	F1	G5	G4	G3	G2	G1	H1	H2	H3
SEG No.	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
Ball No.	H4	J1	J2	J3	J4	K1	K2	K3	J5	K4	L1	L2	L3	M1	M2	N1
SEG No.	64	65	66	67	68	69	70	71								
Ball No.	N2	P2	N3	M3	L4	M4	P3	N4								

*3: COM0 to 31 ball numbers

COM No.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Ball No.	A13	B12	A12	C11	B11	A11	D10	C10	B10	A10	E9	D9	C9	B9	E8	A9
COM No.	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Ball No.	P8	N8	P7	N7	M7	N6	P6	L7	M6	L6	P5	N5	M5	L5	K5	P4

*4: VSS ball numbers
B14, E5, E6, E10, F5, F6(note), F10, G10, H5, H10, J10, K6, K7, K8, K9, K10, L8, M8, M14
(note) The F6 pin does not exist in the VFBGA10H-180 package.

2 CPU

The S1C17702 uses an S1C17 core as the core processor.
The S1C17 core is an original Seiko Epson 16-bit RISC processor.
It features low power consumption, high-speed operation, wide address space, main instruction single-clock execution, and gate-saving design. It is ideal for use in controllers or sequencers, in which 8-bit CPUs are widely used.
For detailed information on the S1C17 core, refer to the *S1C17 Family S1C17 Core Manual*.

2.1 S1C17 Core Features

Processor type

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

Instruction set

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

Register set

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

Memory space, buses

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

Interrupt

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

Power saving

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

Coprocessor interface

- 16-bit × 16-bit + 32-bit product-sum processor
- 16-bit ÷ 16-bit divider

2.2 CPU Registers

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

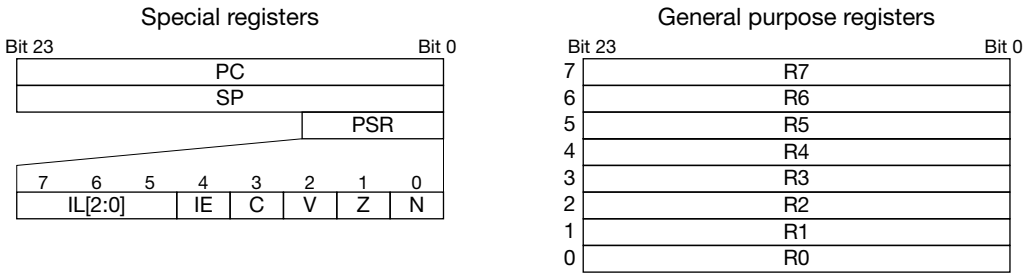


Figure 2.2.1: Registers

2.3 Instruction Set

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

Table 2.3.1: S1C17 core instruction list

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

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

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

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

The codes used in this table are explained below.

Table 2.3.2: Code meanings

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

2.4 Vector Table

The vector table contains the vectors (handler routine start addresses) for interrupt handler routines. When an interrupt occurs, the S1C17 core reads the vector corresponding to the interrupt and executes that handler routine. The boot address for starting program execution must be written at the top of the vector table after resetting.
The S1C17702 vector table starts from address 0x8000. The vector table base address can be read from the TTBR (vector table base register) at address 0xffff80.
Table 2.4.1 shows the S1C17702 vector table.

Table 2.4.1: Vector table

Vector No./ Software interrupt No.	Vector address	Hardware interrupt name	Hardware interrupt factor	Priority
0 (0x00)	TTBR + 0x00	Reset	• Low input to #RESET pin • Watchdog timer overflow *2	1
1 (0x01)	TTBR + 0x04	Irregular address interrupt	Memory access instruction	2
–	(0xfffc00)	Debug interrupt	brk instruction etc.	3
2 (0x02)	TTBR + 0x08	NMI	Watchdog timer overflow *2	4
3 (0x03)	TTBR + 0x0c	reserved	–	–
4 (0x04)	TTBR + 0x10	P0 port interrupt	P00 to P07 port input	High *1
5 (0x05)	TTBR + 0x14	P1 port interrupt	P10 to P17 port input	↑
6 (0x06)	TTBR + 0x18	Stopwatch timer interrupt	• Timer 100 Hz signal • Timer 10 Hz signal • Timer 1 Hz signal	
7 (0x07)	TTBR + 0x1c	Clock timer interrupt	• Timer 32 Hz signal • Timer 8 Hz signal • Timer 2 Hz signal • Timer 1 Hz signal	
8 (0x08)	TTBR + 0x20	8-bit OSC1 timer interrupt	Compare match	
9 (0x09)	TTBR + 0x24	SVD interrupt	Detect power supply voltage drop	
10 (0x0a)	TTBR + 0x28	LCD interrupt	Frame signal	
11 (0x0b)	TTBR + 0x2c	PWM & capture timer Ch.0 interrupt	• Compare A • Compare B	
12 (0x0c)	TTBR + 0x30	8-bit timer Ch.0 & Ch.1 interrupt	• Ch.0 underflow • Ch.1 underflow	
13 (0x0d)	TTBR + 0x34	16-bit timer Ch.0 interrupt	Timer underflow	
14 (0x0e)	TTBR + 0x38	16-bit timer Ch.1 interrupt	Timer underflow	
15 (0x0f)	TTBR + 0x3c	16-bit timer Ch.2 interrupt	Timer underflow	
16 (0x10)	TTBR + 0x40	UART Ch.0 interrupt	• Transmit buffer empty • Receive buffer full • Receive error	
17 (0x11)	TTBR + 0x44	UART Ch.1 interrupt	• Transmit buffer empty • Receive buffer full • Receive error	
18 (0x12)	TTBR + 0x48	SPI interrupt	• Transmit buffer empty • Receive buffer full	
19 (0x13)	TTBR + 0x4c	I²C interrupt	• Transmit buffer empty • Receive buffer full	
20 (0x14)	TTBR + 0x50	Remote controller interrupt	• Data length counter underflow • Input rising edge detection • Input falling edge detection	
21 (0x15)	TTBR + 0x54	PWM & capture timer Ch.1 interrupt	• Compare A • Compare B	
22 (0x16) : 31 (0x1f)	TTBR + 0x58 : TTBR + 0x7c	reserved : reserved	– : –	↓ Low *1

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

The base (top) address for the vector table for writing interrupt vectors can be set using the MISC_TTBRL and MISC_TTBRH registers (0x5328 and 0x532a). "TTBR" in Table 2.4.1 indicates the values set for these registers. The MISC_TTBRL and MISC_TTBRH registers are set to the 0x8000 address after initial resetting. This means only the reset vector must be written to the above address, even when changing the vector table location. Bits 7 to 0 in the MISC_TTBRL register are fixed to 0; the initial address of the vector table normally starts from the 256 byte boundary.

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

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

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

2.5 PSR Readout

The S1C17702 incorporates a PSR register (0x532c) for reading out the contents of the PSR (Processor Status Register) in the S1C17 core. Reading out the contents of this register makes it possible to check the contents of the PSR using application software. Note that data cannot be written to the PSR.

0x532c: PSR Register (MISC_PSR)

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

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

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

D4 PSRIE: PSR Interrupt Enable (IE) Bit

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

D3 PSRC: PSR Carry (C) Flag

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

D2 PSRV: PSR Overflow (V) Flag

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

D1 PSRZ: PSR Zero (Z) Flag

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

D0 PSRN: PSR Negative (N) Flag

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

2.6 Processor Information

The S1C17702 contains a processor ID register (0xffff84) to allow specification of the CPU core type by the application software.

0xffff84: Processor ID Register (IDIR)

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

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

3 Memory Map and Bus Control

Figure 3.1 shows the S1C17702 memory map.

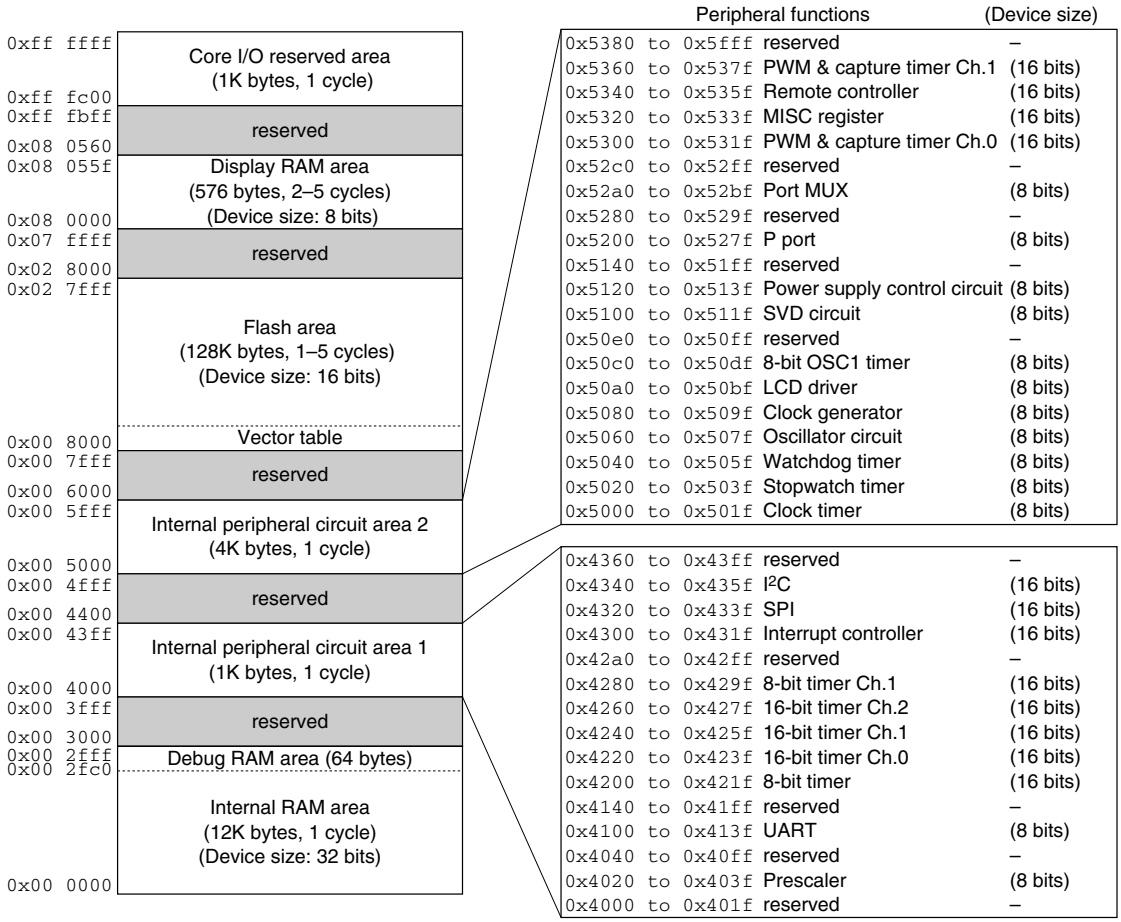


Figure 3.1: S1C17702 memory map

3.1 Bus Cycle

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

Table 3.1.1: Bus access numbers

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

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

Bus cycle calculation example

Number of bus cycles when accessing Display RAM area (8-bit device, 2 cycles) from CPU using 16-bit read/program instruction:
2 cycles × 2 bus accesses = 4 CCLK cycles

3.1.1 Access Size Restrictions

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

Flash memory

Only 16-bit program instructions can be used for flash memory programming. No particular restrictions apply for data reads.

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

3.1.2 Instruction Execution Cycle Restrictions

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

- If a instruction is executed for the flash area accessing flash area, display RAM area, and internal peripheral circuit area 2 (0x5000 onward) data
- If a instruction is executed for an internal RAM area accessing internal RAM area data

3.2 Flash Area

3.2.1 Internal Flash Memory

The 128K-byte area from 0x8000 to 0x27fff contains flash memory (4K bytes × 32 sectors) enabling data or application programs to be written. Address 0x8000 is defined as the vector table base address. The vector table (see "2.4 Vector Table") must be placed at the start of this area. The vector table base address can be modified with the MISC_TTBRL/MISC_TTBRH registers (0x5328 and 0x532a).
Flash memory is read in 1 to 5 cycles.

3.2.2 Flash Memory Programming

The S1C17702 supports onboard flash memory programming, allowing programs or data to be written by a debugger via the ICD (e.g. S5U1C17001H). Self-writing is also possible via software. Programming uses 16-bit units. For specific information on flash memory programming, refer to "Appendix B: Flash Memory Programming."
Data can be deleted either using chip erase or sector erase. Refer to the table below for detailed information on the correspondence between addresses and sectors used for sector erase.

Note: Debuggers support chip erase only. Sector erase is not possible using debuggers.

Table 3.2.2.1: Correspondence between memory addresses and flash sectors

S1C17702 address	Flash sector number
0x17000 to 0x17fff	15
0x16000 to 0x16fff	14
0x15000 to 0x15fff	13
0x14000 to 0x14fff	12
0x13000 to 0x13fff	11
0x12000 to 0x12fff	10
0x11000 to 0x11fff	9
0x10000 to 0x10fff	8
0x0f000 to 0x0ffff	7
0x0e000 to 0x0efff	6
0x0d000 to 0x0dfff	5
0x0c000 to 0x0cfff	4
0x0b000 to 0x0bfff	3
0x0a000 to 0x0afff	2
0x09000 to 0x09fff	1
0x08000 to 0x08fff	0

S1C17702 address	Flash sector number
0x27000 to 0x27fff	31
0x26000 to 0x26fff	30
0x25000 to 0x25fff	29
0x24000 to 0x24fff	28
0x23000 to 0x23fff	27
0x22000 to 0x22fff	26
0x21000 to 0x21fff	25
0x20000 to 0x20fff	24
0x1f000 to 0x1ffff	23
0x1e000 to 0x1efff	22
0x1d000 to 0x1dfff	21
0x1c000 to 0x1cfff	20
0x1b000 to 0x1bfff	19
0x1a000 to 0x1afff	18
0x19000 to 0x19fff	17
0x18000 to 0x18fff	16

Note: The last 32 bits (0x27ffc to 0x27fff) of sector 31 are reserved by the system as protect bits. Be careful to ensure that data other than protection settings is not written to these bits.

3.2.3 Protect Bits

Write-protect and data read-protect can be set in the respective 16K-byte areas to protect internal flash memory contents.
Write-protect prevents data writing to the set area.
Data read-protect prevents data reading from the set area (the value read is always 0x0000). Note that CPU instruction fetch operations are not protected.
This setting uses the protect bits listed below. To set protection, program the protect bit corresponding to the area to be set to 0.

0x27ffc–0x27ffe: Flash Protect Bits

Address	Bit	Function	Setting				Init.	R/W	Remarks
0x27ffc (16 bits)	D15–8	reserved	–				–	–	
	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 0x0c000–0x0ffff	1	Writable	0	Protected	1	R/W	
	D0	Flash write-protect bit for 0x08000–0x0bfff	1	Writable	0	Protected	1	R/W	
0x27ffe (16 bits)	D15–4	reserved	–				–	–	
	D7	Flash data-read-protect bit for 0x24000–0x27fff	1	Readable	0	Protected	1	R/W	
	D6	Flash data-read-protect bit for 0x20000–0x23fff	1	Readable	0	Protected	1	R/W	
	D5	Flash data-read-protect bit for 0x1c000–0x1ffff	1	Readable	0	Protected	1	R/W	
	D4	Flash data-read-protect bit for 0x18000–0x1bfff	1	Readable	0	Protected	1	R/W	
	D3	Flash data-read-protect bit for 0x14000–0x17fff	1	Readable	0	Protected	1	R/W	
	D2	Flash data-read-protect bit for 0x10000–0x13fff	1	Readable	0	Protected	1	R/W	
	D1	Flash data-read-protect bit for 0x0c000–0x0ffff	1	Readable	0	Protected	1	R/W	
	D0	reserved	1				1	R/W	Must be set to 1.

Note: • Do not place the area set for data read-protect in the .data or .rodata sections.
• D0 of 0x27ffe must always be set to 1. The program cannot be booted if this is set to 0.

3.2.4 Flash Controller Access Control

The S1C17702 internal flash memory is accessed via a dedicated flash controller. The MISC register is used for setting access to this controller.

Flash controller read access cycle settings

Set the optimum read access cycles using FLCYC[2:0] (D[2:0]/MISC_FL register) to suit the CCLK frequency to ensure that data is read correctly from the flash memory.

0x5320: FLASHC/SRAMC Control Register (MISC_FL)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
FLASHC/ SRAMC Control Register (MISC_FL)	**0x5320** (16 bits)	D15–10	–	reserved	–		–	–	0 when being read.
		D9–8	**SRCYC[1:0]**	SRAMC access cycle	SRCYC[1:0]	Access cycle	0x3	R/W	
					0x3	5 cycles			
					0x2	4 cycles			
					0x1	3 cycles			
					0x0	2 cycles			
		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 Setup Bits

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

Table 3.2.4.1: Flash controller read access cycle settings

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

(Default: 0x3)

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

3.3 Internal RAM Area

3.3.1 Internal RAM

RAM exists in a 12-Kbyte area from address 0x0 to 0x2fff. This RAM can be accessed in one cycle for reading or writing. In addition to storing variables, it can also be used to copy instruction codes and execute them rapidly in RAM.

Note: The last 64 bytes of the internal RAM (0x2fc0 to 0x2fff) are reserved for on-chip debugging. This area should not be accessed by application programs when using debug functions (for example, during application development).
It can be used for applications in mass-produced products that do not require debugging.

0x5326: IRAM Size Select Register (MISC_IRAMSZ)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
IRAM Size Select Register (MISC_IRAMSZ)	**0x5326** (16 bits)	D15–9	–	reserved	–	–	–	0 when being read.
		D8	**DBADR**	Debug base address select	1 0x0 / 0 0xfffc00	0	R/W	
		D7–2	–	reserved	–	–	–	0 when being read.
		D1–0	**IRAMSZ[1:0]**	IRAM size select	IRAMSZ[1:0] / Read cycle: 0x3 reserved; 0x2 reserved; 0x1 reserved; 0x0 reserved	0x0	R/W	

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

Table 3.3.1.1: Internal RAM size selection

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

(Default: 0x0)

Note: • The IRAM Size Select Register is write-protected. The write-protection must be overridden by writing 0x96 to the MISC Protect Register (0x5324). Note that MISC Protect Register (0x5324) should normally be set to a value other than 0x96, except when writing to the IRAM Size Select Register. Unnecessary programs may result in system malfunctions.
• Do not alter IRAMSZ[1:0] (D[1:0]/MISC_IRAMSZ register) from the default value.

3.4 Display RAM Area

3.4.1 Display RAM

The 576-byte area from address 0x80000 to 0x8055f is assigned as an 8-bit device by the display RAM for internal LCD driver. This RAM is accessed in 2 to 5 cycles. Areas not used for display can be used for general purposes. For specific information on the display memory, refer to “22.5 Display Memory”.

3.4.2 SRAM Controller Access Control

The S1C17702 display RAM is accessed via a dedicated SRAM controller. The MISC register is used to set access to this controller.

SRAM controller access cycle settings

Set the optimum access cycles using SRCYC[1:0] (D[9:8]/MISC_FL register) to suit the CCLK frequency to ensure that data is read or written correctly to or from the display RAM.

0x5320: FLASHC/SRAMC Control Register (MISC_FL)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
FLASHC/ SRAMC Control Register (MISC_FL)	**0x5320** (16 bits)	D15–10	**–**	reserved	–		–	–	0 when being read.
		D9–8	**SRCYC[1:0]**	SRAMC access cycle	SRCYC[1:0]	Access cycle	0x3	R/W	
					0x3	5 cycles			
					0x2	4 cycles			
					0x1	3 cycles			
					0x0	2 cycles			
		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[9:8] SRCYC[1:0]: SRAMC Access Cycle Setup Bits

Sets the number of access cycles for the SRAM (display RAM) controller.

Table 3.4.2.1: SRAM controller access cycle settings

SRCYC[1:0]	Access cycles	CCLK frequency
0x3	5 cycles	8.2 MHz max.
0x2	4 cycles	8.2 MHz max.
0x1	3 cycles	8.2 MHz max.
0x0	2 cycles	6 MHz max.

(Default: 0x3)

3.5 Internal Peripheral Circuit Area

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

3.5.1 Internal Peripheral Circuit Area 1 (0x4000 onward)

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

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

3.5.2 Internal Peripheral Circuit Area 2 (0x5000 onward)

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

- Clock timer (CT, 8-bit device)
- Stopwatch timer (SWT, 8-bit device)
- Watchdog timer (WDT, 8-bit device)
- Oscillator circuit (OSC, 8-bit device)
- Clock generator (CLG, 8-bit device)
- LCD driver (LCD, 8-bit device)
- 8-bit OSC1 timer (T8OSC1, 8-bit device)
- SVD circuit (SVD, 8-bit device)
- Power supply circuit (VD1, 8-bit device)
- Input/output port & port MUX (P, 8-bit device)
- PWM & capture timer (T16E, 16-bit device)
- MISC register (MISC, 16-bit device)
- Remote controller (REMC, 16-bit device)

3.5.3 I/O Map

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

Table 3.5.3.1: I/O map (internal peripheral circuit area 1)

Peripheral circuit	Address	Register name		Function
Prescaler (8-bit device)	0x4020	PSC_CTL	Prescaler Control Register	Prescaler start/stop control
	0x4021 to 0x403f	–	–	Reserved
UART (with IrDA) Ch.0 (8-bit device)	0x4100	UART_ST0	UART Ch.0 Status Register	Transfer, buffer, and error status display
	0x4101	UART_TXD0	UART Ch.0 Transmit Data Register	Transmission data
	0x4102	UART_RXD0	UART Ch.0 Receive Data Register	Received data
	0x4103	UART_MOD0	UART Ch.0 Mode Register	Transfer data format setting
	0x4104	UART_CTL0	UART Ch.0 Control Register	Data transfer control
	0x4105	UART_EXP0	UART Ch.0 Expansion Register	IrDA mode setting
	0x4106 to 0x411f	–	–	Reserved
UART (with IrDA) Ch.1 (8-bit device)	0x4120	UART_ST1	UART Ch.1 Status Register	Transfer, buffer, and error status display
	0x4121	UART_TXD1	UART Ch.1 Transmit Data Register	Transmission data
	0x4122	UART_RXD1	UART Ch.1 Receive Data Register	Received data
	0x4123	UART_MOD1	UART Ch.1 Mode Register	Transfer data format setting
	0x4124	UART_CTL1	UART Ch.1 Control Register	Data transfer control
	0x4125	UART_EXP1	UART Ch.1 Expansion Register	IrDA mode setting
	0x4126 to 0x413f	–	–	Reserved
8-bit timer (with F mode) Ch.0 (16-bit device)	0x4200	T8F_CLK	8-bit Timer Input Clock Select Register	Prescaler output clock selection
	0x4202	T8F_TR	8-bit Timer Reload Data Register	Reload data setting
	0x4204	T8F_TC	8-bit Timer Counter Data Register	Counter data
	0x4206	T8F_CTL	8-bit Timer Control Register	Timer mode setting and timer RUN/STOP
	0x4208	T8F_INT	8-bit Timer Interrupt Control Register	Interrupt control
	0x420a to 0x421f	–	–	Reserved
16-bit timer Ch.0 (16-bit device)	0x4220	T16_CLK0	16-bit Timer Ch.0 Input Clock Select Register	Prescaler output clock selection
	0x4222	T16_TR0	16-bit Timer Ch.0 Reload Data Register	Reload data setting
	0x 24	T16_TC0	16-bit Timer Ch.0 Counter Data Register	Counter data
	0x4226	T16_CTL0	16-bit Timer Ch.0 Control Register	Timer mode setting and timer RUN/STOP
	0x4228	T16_INT0	16-bit Timer Ch.0 Interrupt Control Register	Interrupt control
	0x422a to 0x423f	–	–	Reserved
16-bit timer Ch.1 (16-bit device)	0x4240	T16_CLK1	16-bit Timer Ch.1 Input Clock Select Register	Prescaler output clock selection
	0x4242	T16_TR1	16-bit Timer Ch.1 Reload Data Register	Reload data setting
	0x4244	T16_TC1	16-bit Timer Ch.1 Counter Data Register	Counter data
	0x4246	T16_CTL1	16-bit Timer Ch.1 Control Register	Timer mode setting and timer RUN/STOP
	0x4248	T16_INT1	16-bit Timer Ch.1 Interrupt Control Register	Interrupt control
	0x424a to 0x425f	–	–	Reserved
16-bit timer Ch.2 (16-bit device)	0x4260	T16_CLK2	16-bit Timer Ch.2 Input Clock Select Register	Prescaler output clock selection
	0x4262	T16_TR2	16-bit Timer Ch.2 Reload Data Register	Reload data setting
	0x4264	T16_TC2	16-bit Timer Ch.2 Counter Data Register	Counter data
	0x4266	T16_CTL2	16-bit Timer Ch.2 Control Register	Timer mode setting and timer RUN/STOP
	0x4268	T16_INT2	16-bit Timer Ch.2 Interrupt Control Register	Interrupt control
	0x426a to 0x427f	–	–	Reserved
8-bit timer (with F mode) Ch.1 (16-bit device)	0x4280	T8F_CLK1	8-bit Timer Ch.1 Input Clock Select Register	Prescaler output clock selection
	0x4282	T8F_TR1	8-bit Timer Ch.1 Reload Data Register	Reload data setting
	0x4284	T8F_TC1	8-bit Timer Ch.1 Counter Data Register	Counter data
	0x4286	T8F_CTL1	8-bit Timer Ch.1 Control Register	Timer mode setting and timer RUN/STOP
	0x4288	T8F_INT1	8-bit Timer Ch.1 Interrupt Control Register	Interrupt control
	0x428a to 0x429f	–	–	Reserved
Interrupt controller (16-bit device)	0x4300 to 0x4304	–	–	Reserved
	0x4306	ITC_LV0	Interrupt Level Setup Register 0	P0/P1 interrupt level setting
	0x4308	ITC_LV1	Interrupt Level Setup Register 1	SWT/CT interrupt level setting
	0x430a	ITC_LV2	Interrupt Level Setup Register 2	T8OSC1/SVD interrupt level setting
	0x430c	ITC_LV3	Interrupt Level Setup Register 3	LCD/T16E Ch.0 interrupt level setting
	0x430e	ITC_LV4	Interrupt Level Setup Register 4	T8F/T16 Ch.0 interrupt level setting
	0x4310	ITC_LV5	Interrupt Level Setup Register 5	T16 Ch.1/Ch.2 interrupt level setting
	0x4312	ITC_LV6	Interrupt Level Setup Register 6	UART Ch.0/Ch.1 interrupt level setting
	0x4314	ITC_LV7	Interrupt Level Setup Register 7	SPI/I^2C interrupt level setting
	0x4316	ITC_LV8	Interrupt Level Setup Register 8	REMC/T16E Ch.1 interrupt level setting
	0x4318 to 0x431f	–	–	Reserved

Peripheral circuit	Address	Register name		Function
SPI (16-bit device)	0x4320	SPI_ST	SPI Status Register	Transfer and buffer status display
	0x4322	SPI_TXD	SPI Transmit Data Register	Transmission data
	0x4324	SPI_RXD	SPI Receive Data Register	Received data
	0x4326	SPI_CTL	SPI Control Register	SPI mode and data transfer permission setting
	0x4328 to 0x433f	–	–	Reserved
I²C (16-bit device)	0x4340	I2C_EN	I²C Enable Register	I²C module enable
	0x4342	I2C_CTL	I²C Control Register	I²C control and transfer status display
	0x4344	I2C_DAT	I²C Data Register	Transfer data
	0x4346	I2C_ICTL	I²C Interrupt Control Register	I²C interrupt control
	0x4348 to 0x435f	–	–	Reserved

Table 3.5.3.2: I/O map (internal peripheral circuit area 2)

Peripheral circuit	Address	Register name		Function
Clock timer (8-bit device)	0x5000	CT_CTL	Clock Timer Control Register	Timer reset and RUN/STOP control
	0x5001	CT_CNT	Clock Timer Counter Register	Counter data
	0x5002	CT_IMSK	Clock Timer Interrupt Mask Register	Interrupt mask setting
	0x5003	CT_IFLG	Clock Timer Interrupt Flag Register	Interrupt occurrence status display/reset
	0x5004 to 0x501f	–	–	Reserved
Stopwatch timer (8-bit device)	0x5020	SWT_CTL	Stopwatch Timer Control Register	Timer reset and RUN/STOP control
	0x5021	SWT_BCNT	Stopwatch Timer BCD Counter Register	BCD Counter data
	0x5022	SWT_IMSK	Stopwatch Timer Interrupt Mask Register	Interrupt mask setting
	0x5023	SWT_IFLG	Stopwatch Timer Interrupt Flag Register	Interrupt occurrence status display/reset
	0x5024 to 0x503f	–	–	Reserved
Watchdog timer (8-bit device)	0x5040	WDT_CTL	Watchdog Timer Control Register	Timer reset and RUN/STOP control
	0x5041	WDT_ST	Watchdog Timer Status Register	Timer mode setting and NMI status display
	0x5042 to 0x505f	–	–	Reserved
Oscillator circuit (8-bit device)	0x5060	OSC_SRC	Clock Source Select Register	Clock source selection
	0x5061	OSC_CTL	Oscillation Control Register	Oscillation control
	0x5062	OSC_NFEN	Noise Filter Enable Register	Noise filter ON/OFF
	0x5063	OSC_LCLK	LCD Clock Setup Register	LCD clock setting
	0x5064	OSC_FOUT	FOUT Control Register	Clock external output control
	0x5065	OSC_T8OSC1	T8OSC1 Clock Control Register	8-bit OSC1 timer clock setting
	0x5066 to 0x507f	–	–	Reserved
Clock generator (8-bit device)	0x5080	CLG_PCLK	PCLK Control Register	PCLK feed control
	0x5081	CLG_CCLK	CCLK Control Register	CCLK division ratio setting
	0x5082 to 0x509f	–	–	Reserved
LCD driver (8-bit device)	0x50a0	LCD_DCTL	LCD Display Control Register	LCD display control
	0x50a1	LCD_CADJ	LCD Contrast Adjust Register	Contrast control
	0x50a2	LCD_CCTL	LCD Clock Control Register	LCD clock duty selection
	0x50a3	LCD_VREG	LCD Voltage Regulator Control Register	LCD driver voltage regulator control
	0x50a4	LCD_PWR	LCD Power Voltage Booster Control Register	LCD power supply voltage booster circuit control
	0x50a5	LCD_IMSK	LCD Interrupt Mask Register	Interrupt mask setting
	0x50a6	LCD_IFLG	LCD Interrupt Flag Register	Interrupt occurrence status display/reset
	0x50a7 to 0x50bf	–	–	Reserved
8-bit OSC1 timer (8-bit device)	0x50c0	T8OSC1_CTL	8-bit OSC1 Timer Control Register	Timer mode setting and timer RUN/STOP
	0x50c1	T8OSC1_CNT	8-bit OSC1 Timer Counter Data Register	Counter data
	0x50c2	T8OSC1_CMP	8-bit OSC1 Timer Compare Data Register	Compare data setting
	0x50c3	T8OSC1_IMSK	8-bit OSC1 Timer Interrupt Mask Register	Interrupt mask setting
	0x50c4	T8OSC1_IFLG	8-bit OSC1 Timer Interrupt Flag Register	Interrupt occurrence status display/reset
	0x50c5	T8OSC1_DUTY	8-bit OSC1 Timer PWM Duty Data Register	PWM output data setting
	0x50c6 to 0x50df	–	–	Reserved
SVD circuit (8-bit device)	0x5100	SVD_EN	SVD Enable Register	SVD operation permission/prohibition
	0x5101	SVD_CMP	SVD Compare Voltage Register	Comparison voltage setting
	0x5102	SVD_RSLT	SVD Detection Result Register	Voltage detection results
	0x5103	SVD_IMSK	SVD Interrupt Mask Register	Interrupt mask setting
	0x5104	SVD_IFLG	SVD Interrupt Flag Register	Interrupt occurrence status display/reset
	0x5105 to 0x511f	–	–	Reserved
Power supply circuit (8-bit device)	0x5120	VD1_CTL	V_{D1} Control Register	V_{D1} voltage and load protection control
	0x5121 to 0x513f	–	–	Reserved

Peripheral circuit	Address	Register name		Function
P port & port MUX (8-bit device)	0x5200	P0_IN	P0 Port Input Data Register	P0 port input data
	0x5201	P0_OUT	P0 Port Output Data Register	P0 port output data
	0x5202	P0_OEN	P0 Port Output Enable Register	P0 port output enable
	0x5203	P0_PU	P0 Port Pull-up Control Register	P0 port pull-up control
	0x5204	P0_SM	P0 Port Schmitt Trigger Control Register	P0 port Schmitt trigger control
	0x5205	P0_IMSK	P0 Port Interrupt Mask Register	P0 port interrupt mask setting
	0x5206	P0_EDGE	P0 Port Interrupt Edge Select Register	P0 port interrupt edge selection
	0x5207	P0_IFLG	P0 Port Interrupt Flag Register	P0 port interrupt occurrence status display/ reset
	0x5208	P0_CHAT	P0 Port Chattering Filter Control Register	P0 port chattering filter control
	0x5209	P0_KRST	P0 Port Key-Entry Reset Configuration Register	P0 port key entry reset setting
	0x520a	P0_IEN	P0 Port Input Enable Register	P0 port input enable
	0x520b to 0x520f	–	–	Reserved
	0x5210	P1_IN	P1 Port Input Data Register	P1 port input data
	0x5211	P1_OUT	P1 Port Output Data Register	P1 port output data
	0x5212	P1_OEN	P1 Port Output Enable Register	P1 port output enable
	0x5213	P1_PU	P1 Port Pull-up Control Register	P1 port pull-up control
	0x5214	P1_SM	P1 Port Schmitt Trigger Control Register	P1 port Schmitt trigger control
	0x5215	P1_IMSK	P1 Port Interrupt Mask Register	P1 port interrupt mask setting
	0x5216	P1_EDGE	P1 Port Interrupt Edge Select Register	P1 port interrupt edge selection
	0x5217	P1_IFLG	P1 Port Interrupt Flag Register	P1 port interrupt occurrence status display/ reset
	0x5218	P1_CHAT	P1 Port Chattering Filter Control Register	P1 port chattering filter control
	0x5219	–	–	Reserved
	0x521a	P1_IEN	P1 Port Input Enable Register	P1 port input enable
	0x521b to 0x521f	–	–	Reserved
	0x5220	P2_IN	P2 Port Input Data Register	P2 port input data
	0x5221	P2_OUT	P2 Port Output Data Register	P2 port output data
	0x5222	P2_OEN	P2 Port Output Enable Register	P2 port output enable
	0x5223	P2_PU	P2 Port Pull-up Control Register	P2 port pull-up control
	0x5224	P2_SM	P2 Port Schmitt Trigger Control Register	P2 port Schmitt trigger control
	0x5225 to 0x5229	–	–	Reserved
	0x522a	P2_IEN	P2 Port Input Enable Register	P2 port input enable
	0x522b to 0x522f	–	–	Reserved
	0x5230	P3_IN	P3 Port Input Data Register	P3 port input data
	0x5231	P3_OUT	P3 Port Output Data Register	P3 port output data
	0x5232	P3_OEN	P3 Port Output Enable Register	P3 port output enable
	0x5233	P3_PU	P3 Port Pull-up Control Register	P3 port pull-up control
	0x5234	P3_SM	P3 Port Schmitt Trigger Control Register	P3 port Schmitt trigger control
	0x5235 to 0x5239	–	–	Reserved
	0x523a	P3_IEN	P3 Port Input Enable Register	P3 port input enable
	0x523b to 0x527f	–	–	Reserved
	0x52a0	P0_PMUX	P0 Port Function Select Register	P0 port function selection
	0x52a1	P1_PMUX	P1 Port Function Select Register	P1 port function selection
	0x52a2	P2_PMUX	P2 Port Function Select Register	P2 port function selection
	0x52a3	P3_PMUX	P3 Port Function Select Register	P3 port function selection
	0x52a4 to 0x52bf	–	–	Reserved
PWM & capture timer Ch.0 (16-bit device)	0x5300	T16E_CA0	PWM Timer Ch.0 Compare Data A Register	Compare data A setting
	0x5302	T16E_CB0	PWM Timer Ch.0 Compare Data B Register	Compare data B setting
	0x5304	T16E_TC0	PWM Timer Ch.0 Counter Data Register	Counter data
	0x5306	T16E_CTL0	PWM Timer Ch.0 Control Register	Timer mode setting and timer RUN/STOP
	0x5308	T16E_CLK0	PWM Timer Ch.0 Input Clock Select Register	Prescaler output clock selection
	0x530a	T16E_IMSK0	PWM Timer Ch.0 Interrupt Mask Register	Interrupt mask setting
	0x530c	T16E_IFLG0	PWM Timer Ch.0 Interrupt Flag Register	Interrupt occurrence status display/reset
	0x530e to 0x531f	–	–	Reserved
MISC register (16-bit device)	0x5320	MISC_FL	FLASHC/SRAMC Control Register	FLASHC/SRAMC access condition setting
	0x5322	MISC_OSC1	OSC1 Peripheral Control Register	OSC1 operation peripheral function setting for debugging
	0x5324	MISC_PROT	MISC Protect Register	MISC register write protection
	0x5326	MISC_IRAMSZ	IRAM Size Select Register	IRAM size selection
	0x5328	MISC_TTBRL	Vector Table Address Low Register	Vector table address setting
	0x532a	MISC_TTBRH	Vector Table Address High Register	
	0x532c	MISC_PSR	PSR Register	S1C17 core PSR readout
	0x532e to 0x533f	–	–	Reserved
Remote controller (16-bit device)	0x5340	REMC_CFG	REMC Configuration Register	Clock transfer control
	0x5342	REMC_CAR	REMC Carrier Length Setup Register	Carrier H/L section length setting
	0x5344	REMC_LCNT	REMC Length Counter Register	Transfer bit and transfer data length setting
	0x5346	REMC_INT	REMC Interrupt Control Register	Interrupt control
	0x5348 to 0x535f	–	–	Reserved

Peripheral circuit	Address	Register name		Function
PWM & capture timer Ch.1 (16-bit device)	0x5360	T16E_CA1	PWM Timer Ch.1 Compare Data A Register	Compare data A setting
	0x5362	T16E_CB1	PWM Timer Ch.1 Compare Data B Register	Compare data B setting
	0x5364	T16E_TC1	PWM Timer Ch.1 Counter Data Register	Counter data
	0x5366	T16E_CTL1	PWM Timer Ch.1 Control Register	Timer mode setting and timer RUN/STOP
	0x5368	T16E_CLK1	PWM Timer Ch.1 Input Clock Select Register	Prescaler output clock selection
	0x536a	T16E_IMSK1	PWM Timer Ch.1 Interrupt Mask Register	Interrupt mask setting
	0x536c	T16E_IFLG1	PWM Timer Ch.1 Interrupt Flag Register	Interrupt occurrence status display/reset
	0x536e to 0x537f	–	–	Reserved

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

3.6 Core I/O Reserved Area

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

Table 3.6.1: I/O map (Core I/O reserved area)

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

See “2.5 Processor Information” for more information on IDIR and “24. On-chip Debugger (DBG)” for more information on other registers.
This area incorporates S1C17 core registers, in addition to those described above. For more information on these registers, refer to the *S1C17 Core Manual*.

4 Power Supply

4.1 Power Supply Voltage

The S1C17702 operation power supply voltages are given below.

Normal operation: 1.8 V to 3.6 V
Flash memory programming: 2.7 V to 3.6 V

Supply voltages within the range to V_{DD} pin with the V_{SS} pin as GND.

The S1C17702 QFP21-176 pin package has three V_{DD} pins and three V_{SS} pins. The VFBGA8H-181 package has three V_{DD} pins and 19 V_{SS} pins. The VFBGA10H-180 package has three V_{DD} pins and 18 V_{SS} pins. In either case, all must be connected to the + power supply and GND rather than left open.

4.2 Internal Power Supply Circuit

The S1C17702 includes a power supply circuit, as shown in Figure 4.2.1, which generates all the voltages required for internal circuits within the IC. Broadly speaking, the power supply circuit is divided into three sections.

Table 4.2.1: Power supply circuit

Circuit	Power supply circuit	Output voltage
Oscillator circuit, internal circuit	Internal voltage regulator	V_{D1}
LCD voltage regulator	Power supply voltage booster circuit	V_{DD} or V_{D2}
LCD driver	LCD voltage regulator	V_{C1} to V_{C5}

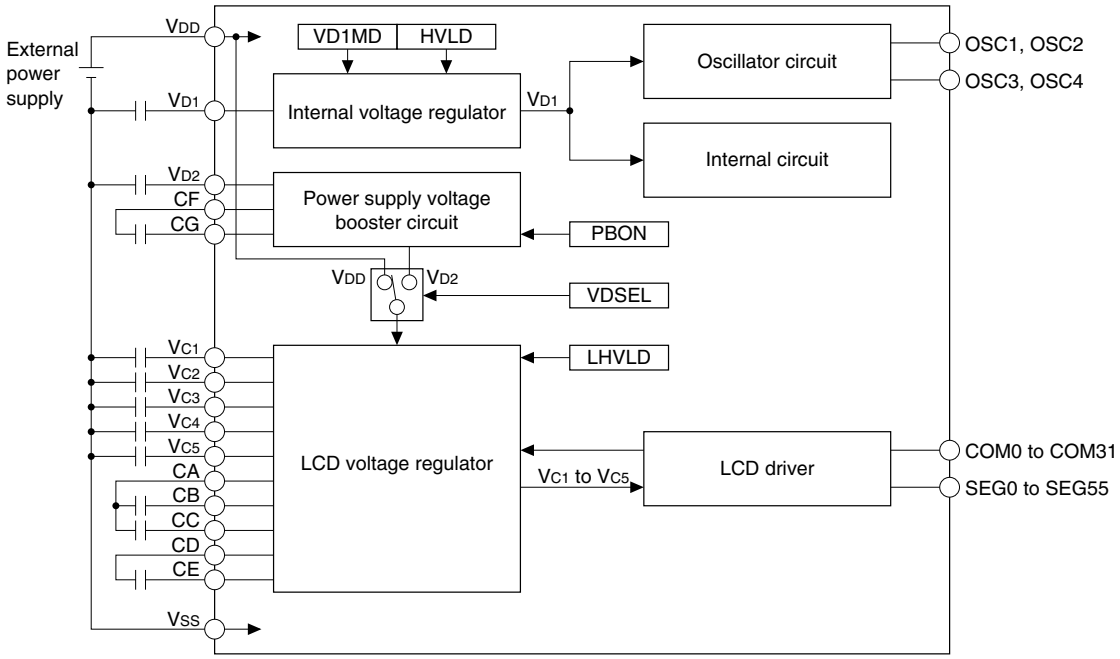


Figure 4.2.1: Power supply circuit configuration

Note: Never use the output from pins V_{D1}, V_{D2}, or V_{C1} to V_{C5} to drive external circuits.

Internal voltage regulator

The internal voltage regulator generates voltage V_{D1} to operate internal logic and oscillator circuits. The voltage of V_{D1} can be switched via the program and is set to 1.8 V for normal operations and 2.5 V for flash memory programming.

Power supply voltage booster circuit

The power supply voltage booster circuit generates voltage V_{D2} for operating LCD voltage regulators. The LCD voltage regulator power supply can be selected to be V_{DD} or V_{D2} to suit the power supply voltage V_{DD}.

Table 4.2.1: LCD voltage regulator power supply

Power supply voltage V_{DD}	LCD voltage regulator power supply
1.8 to 2.5V	V_{D2} ($\approx V_{DD} \times 2$)
2.5 to 3.6V	V_{DD}

LCD voltage regulator

The LCD voltage regulator generates 1/5 bias voltages V_{C1}, V_{C2}, V_{C3}, V_{C4}, and V_{C5} for driving LCDs.

In the S1C17702, these LCD drive voltages are fed to the internal LCD driver to drive the LCD panels connected to the common/segment pins.

Note: Voltages V_{C1} to V_{C5} cannot be obtained correctly if VDD is used as the power supply for LCD voltage regulators when 2.5 V or less.

4.3 Power Supply Circuit Control

The various power supply circuits can be controlled via software to ensure that correct operating voltages within the chip are generated to suit the power supply voltage and operating mode and to minimize consumption current.

Operating mode switching

The S1C17702 features two operating modes:

1. Normal operating mode
   Normal operating mode for running application programs.
   V_{DD} = 1.8 V to 3.6 V, internal operating voltage V_{D1} = 1.8 V

2. Flash erase/programming mode
   Operating mode for deleting and writing program/data to flash memory.
   V_{DD} = 2.7 V to 3.6 V, internal operating voltage V_{D1} = 2.5 V

The voltage V_{D1} must be switched as described to suit the operating mode above. This is done using VD1MD (D0/VD1_CTL register). VD1MD is normally used with the default setting 0 (V_{D1} = 1.8 V). VD1MD is set to 1 for flash memory erase/programming.

* **VD1MD**: Flash Erase/Program Mode Bit in the V_{D1} Control (VD1_CTL) Register (D0/0x5120)

Note: An interval of 5 ms (max) is required for the internal operating voltage to stabilize after switching the operating mode. Start flash memory programming only after this stabilization time has elapsed.

LCD power supply control

The LCD voltage regulators must be operated at a voltage of at least 2.5 V to ensure that the correct LCD drive voltages V_{C1} to V_{C5} are generated. When the 1.8 V to 2.5 V power supply (V_{DD}) is used, V_{DD} is approximately doubled by the power supply voltage booster circuit, and the output voltage V_{D2} is used to drive the LCD voltage regulator. In this case, set PBON (D0/LCD_PWR register) to 1 and turn on the power supply voltage booster circuit. To drive the LCD voltage regulator using V_{D2} generated, set VDSEL (D1/LCD_PWR register) to 1. PBON must be set to 1 before switching to the V_{D2} voltage.

* **PBON**: Power Voltage Booster Control Bit in the LCD Power Voltage Booster Control (LCD_PWR) Register (D0/0x50a4)
* **VDSEL**: Regulator Power Source Select Bit in the LCD Power Voltage Booster Control (LCD_PWR) Register (D1/0x50a4)

V_{DD} is used to drive the LCD voltage regulator when using a power supply (V_{DD}) of 2.5 V or more. The power supply voltage booster circuit is switched off to minimize current consumption. In this case, PBON and VDSEL should both be left at the default value of 0.

Note: An interval of approximately 1 ms is required for the output voltage V_{D2} to stabilize after you turn on the power supply voltage booster circuit. Do not switch the LCD voltage regulator power supply to V_{D2} until this time has elapsed.

LCD drive voltages V_{C1} to V_{C5} are fed to the LCD driver if DSPC[1:0] (D[1:0]/LCD_DCTL register) is set to a value other than 0x0 (display off).

* **DSPC[1:0]**: LCD Display Control Bits in the LCD Display Control (LCD_DCTL) Register (D[1:0]/0x50a0)

If the internal LCD driver is not used, turn off the power supply voltage booster circuit and LCD voltage regulator to minimize current consumption. PBON, VDSEL, and DSPC[1:0] should all be 0 (default).

Power supply control bit settings list

Table 4.3.1 lists the power supply control bit settings for different conditions.

Table 4.3.1: Power supply control bit settings list

Condition			Control bit			
Operating mode	V_{DD}	LCD driver	VD1MD	PBON	VDSEL	DSPC[1:0]
Normal operation	1.8 to 2.5 V	Used	0	1	1	Other than 0x0
	2.5 to 3.6 V	Used	0	0	0	Other than 0x0
	1.8 to 3.6 V	Not used	0	0	0	0x0
Flash erase/ programming	1.8 to 2.7 V	–	(Not to be used)			
	2.7 to 3.6 V	Used	1	0	0	Other than 0x0
	2.7 to 3.6 V	Not used	1	0	0	0x0

For specific information on DSPC[1:0] settings, refer to "0x50a0: LCD Display Control Register (LCD_DCTL)" in section 22.8.

4.4 Heavy Load Protection Function

The internal constant-voltage and LCD voltage regulators include heavy load protection functions that can be set via software to ensure stable operations and LCD display, even when the power supply voltage fluctuates due to external loads.
The internal voltage regulator is switched to heavy load protection mode by writing 1 to HVLD (D5/VD1_CTL register), stabilizing V_{D1} output.

∗ **HVLD**: V_{D1} Heavy Load Protection Mode Bit in the V_{D1} Control (VD1_CTL) Register (D5/0x5120)

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 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.)

Note: Always cancel heavy load protection mode after the processing that causes unstable operations has finished. When executing an unstable processing repeatedly, the program must maintain heavy load protection mode while it repeating the processing.

The LCD voltage regulator is switched to heavy load protection mode by writing 1 to LHVLD (D4/LCD_VREG register), stabilizing the V_{C1} to V_{C5} output. Make this setting if you observe brightness fluctuations on the LCD display.

∗ **LHVLD**: LCD Heavy Load Protection Mode Bit in the LCD Voltage Regulator Control (LCD_VREG) Register (D4/0x50a3)

Note: Current consumption will be higher in heavy load protection mode than normal operations. Avoid setting heavy load protection via software unless necessary.

4.5 Control Register Details

Table 4.5.1: Power supply control register list

Address	Register		Function
0x5120	VD1_CTL	V_{D1} Control Register	V_{D1} voltage and heavy load protection control
0x50a3	LCD_VREG	LCD Voltage Regulator Control Register	LCD driver voltage regulator control
0x50a4	LCD_PWR	LCD Power Voltage Booster Control Register	LCD power supply voltage booster circuit control

The individual power supply control registers are described below. These are all 8-bit registers.

Note: When writing data to registers, always program 0 to those bits indicated as "Reserved." Avoid writing 1.

0x5120: 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/program 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 voltage regulator to heavy load protection mode.
1 (R/W): Heavy load protection on
0 (R/W): Heavy load protection off (default)

The internal voltage regulator is switched to heavy load protection mode by writing 1 to HVLD, stabilizing the 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 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.)

Note: • Always cancel heavy load protection mode after the processing that causes unstable operations has finished. When executing an unstable processing repeatedly, the program must maintain heavy load protection mode while it repeating the processing.

• Since it increases current consumption, avoid setting heavy load protection mode unless necessary.

D[4:1] Reserved

D0 VD1MD: Flash Erase/Program Mode Bit

Selects the internal operating voltage VD1 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)

VD1MD is normally used with the default setting of 0 (V_{D1} = 1.8 V). Set VD1MD to 1 for flash memory erase and programming.

Note: An interval of 5 ms (max) is required for the internal operating voltage to stabilize after switching the operating mode. Start flash memory programming after this stabilization time has elapsed.

0x50a3: 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	
		D3–0	–	reserved	–	–	–	0 when being read.

D[7:5] Reserved

D4 LHVLD: LCD Heavy Load Protection Mode Bit

Sets the LCD voltage regulator to heavy load protection mode.
1 (R/W): Heavy load protection on
0 (R/W): Heavy load protection off (default)

The LCD voltage regulator is switched to heavy load protection mode by writing 1 to LHVLD, stabilizing the V_{C1} to V_{C5} output. Make this setting if you observe brightness fluctuations on the LCD display.
Since it increases current consumption, avoid setting heavy load protection mode unless necessary.

D[3:0] Reserved

0x50a4: LCD Power Voltage Booster Control Register (LCD_PWR)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
LCD Power Voltage Booster Control Register (LCD_PWR)	0x50a4 (8 bits)	D7–2	–	reserved	–	–	–	0 when being read.
		D1	VDSEL	Regulator power source select	1 VD2 0 VDD	0	R/W	
		D0	PBON	Power voltage booster control	1 On 0 Off	0	R/W	

D[7:2] Reserved

D1 VDSEL: Regulator Power Source Select Bit

Selects the LCD voltage regulator power supply voltage.
1 (R/W): V_{D2}
0 (R/W): V_{DD} (default)

When using the 1.8 V to 2.5 V power supply (V_{DD}), program 1 to VDSEL to drive the LCD voltage regulator using V_{D2} generated by the power supply voltage booster circuit. 1 must be written to PBON (D0) before switching on the power supply voltage booster circuit.
When using a power supply (V_{DD}) of 2.5 V or more, set VDSEL to 0 to drive the LCD voltage regulator using V_{DD}. The power supply voltage booster circuit is switched off in this case to minimize current consumption.

D0 PBON: Power Voltage Booster Control Bit

Controls the power supply voltage booster circuit.
1 (R/W): On
0 (R/W): Off (default)

When the 1.8 V to 2.5 V power supply (V_{DD}) is used, program 1 to PBON to turn on the power supply voltage booster circuit. The power supply voltage booster circuit doubles V_{DD} to generate V_{D2} to drive the LCD voltage regulator. VDSEL (D1) must also be set to 1 to drive the LCD voltage regulator using V_{D2}. When using a power supply (V_{DD}) of 2.5 V or more, there is no need to generate V_{D2}. The power supply voltage booster circuit should be switched off to minimize current consumption.

Note: An interval of approximately 1 ms is required for the output voltage V_{D2} to stabilize after you turn on the power supply voltage booster circuit. Do not switch the LCD voltage regulator power supply to V_{D2} until this time has elapsed.

4.6 Precautions

- Never use the output from pins V_{D1}, V_{D2}, and V_{C1} to V_{C5} to drive external circuits.
- The correct V_{C1} to V_{C5} voltages will not be obtained if you use V_{DD} as the power supply for LCD voltage regulators when 2.5 V or less.
- An interval of 5 ms (max.) is required for the internal operating voltage to stabilize after switching the operating mode to Flash erase/programming mode. Start flash memory programming after this stabilization time has elapsed.
- An interval of approximately 1 ms is required for the output voltage V_{D2} to stabilize after you turn on the power supply voltage booster circuit. Do not switch the LCD voltage regulator power supply to V_{D2} until this time has elapsed.
- Always cancel heavy load protection mode after the processing that causes unstable operations has finished. When executing an unstable processing repeatedly, the program must maintain heavy load protection mode while it repeating the processing.
- Current consumption will be higher in heavy load protection mode than for normal operation. Avoid setting heavy load protection mode via software unless necessary.

5 Initial Reset

5.1 Initial Reset Factors

Shown below are the three different initial reset factors for initializing S1C17702 internal circuits.

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

Figure 5.1.1 illustrates the initial reset circuit configuration.

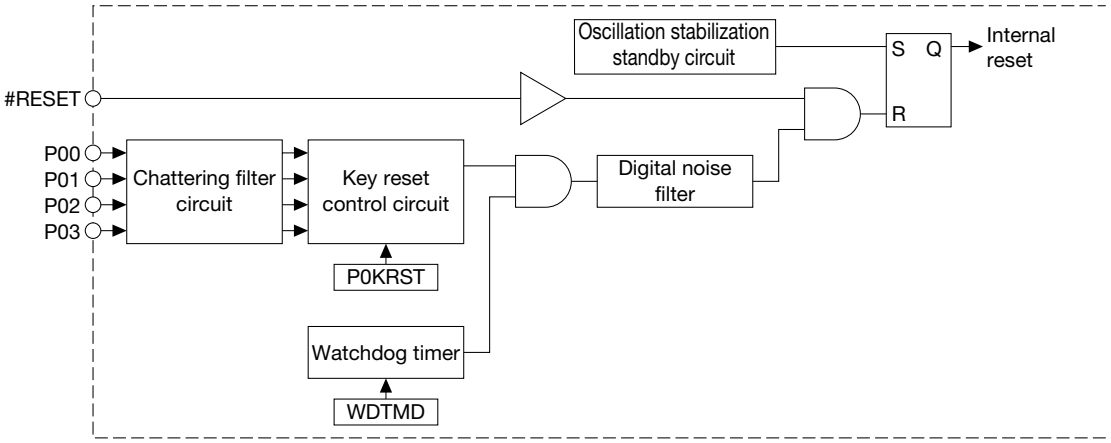


Figure 5.1.1: Initial reset circuit configuration

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

5.1.1 #RESET pin

Initial resetting is possible by inputting external Low level to the #RESET pin.
To initialize the S1C17702 reliably, the #RESET pin must be maintained at Low level for at least the specified duration after the power supply voltage rises. (Refer to "26.6 AC Characteristics.")
Initial resetting is cleared if the #RESET input changes from Low to High, and the CPU begins reset interrupt processing.
The #RESET pin incorporates a pull-up resistance.

5.1.2 P0 Port Key-Entry Reset

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

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

Table 5.1.2.1: P0 port key-entry reset settings

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

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

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

5.1.3 Reset by Watchdog Timer

The S1C17702 incorporates a watchdog timer to detect runaway CPU. If the watchdog timer is not reset by software every 4 seconds (with this failure indicating a runaway CPU), the timer overflows, generating an NMI or reset. A reset is generated by writing "1" to WDTMD (D1/WDT_ST register). (NMI is generated if WDTMD is 0.)

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

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

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

5.2 Initial Reset Sequence

CPU startup waits for the oscillation stabilization standby time (64/fiosc seconds*) to expire after resetting is cleared via the #RESET pin at power-on. Figure 5.2.1 illustrates the sequence of operations after clearing the initial reset. The CPU starts up in sync with the IOSC (internal CR oscillator circuit) clock after the reset is cleared.

*fiosc: IOSC clock frequency

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

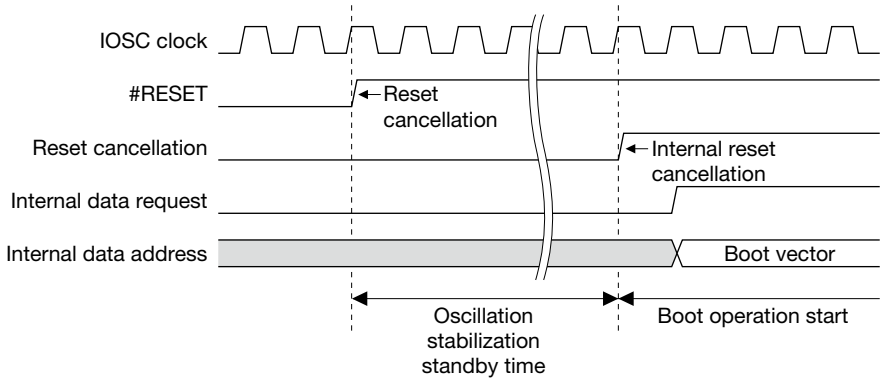


Figure 5.2.1: Sequence of operations after initial reset cancellation

5.3 Initial Settings at Initial Resetting

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

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

The internal RAM and display memory should be initialized via software, since they are not initialized by initial resetting.

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

6 Interrupt Controller

6.1 ITC Configuration

The S1C17702 features the following 18 different types of interrupts:

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

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

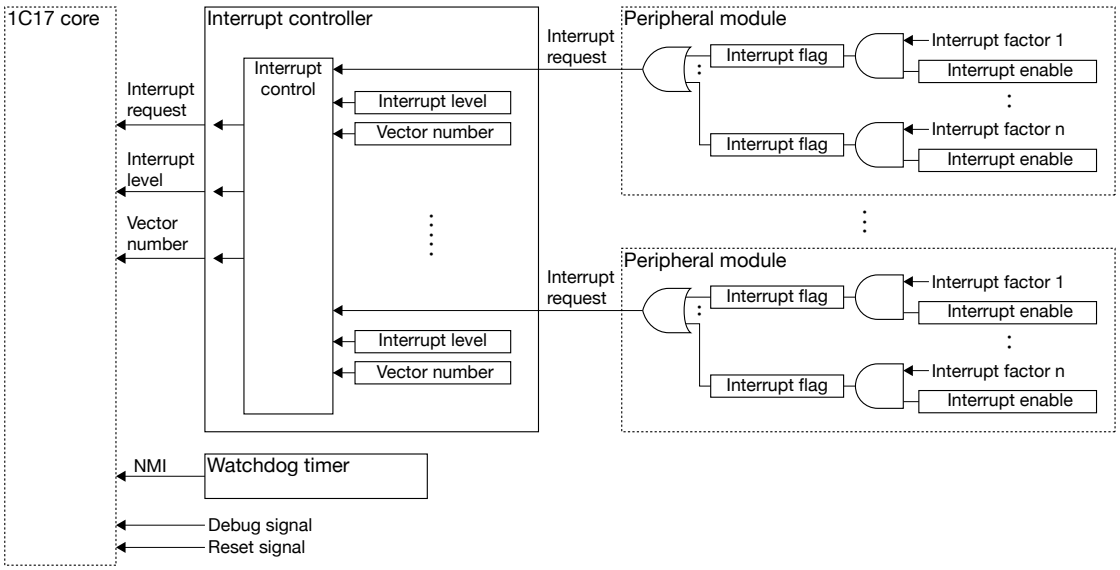


Figure 6.1.1: Interrupt system

6.2 Vector Table

The vector table contains the vectors (handler routine start addresses) for interrupt handler routines. When an interrupt occurs, the S1C17 core reads the vector corresponding to the interrupt and executes that handler routine. The base (top) address for the vector table can be set using the MISC_TTBRL and MISC_TTBRH registers (0x5328 and 0x532a) (See "2.4 Vector Table"). "TTBR" in Table 6.2.1 indicates the values set for these registers. The MISC_TTBRL and MISC_TTBRH registers are set to the 0x8000 address after initial resetting. Table 6.2.1 shows the S1C17702 vector table.

Table 6.2.1: Vector table

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

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

Vector numbers 4 to 21 are assigned maskable interrupts supported by the S1C17702.

6.3 Maskable Interrupt Control

6.3.1 Peripheral Module Interrupt Control Bit

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

6.3.2 ITC Interrupt Request Processing

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

Table 6.3.2.1: Interrupt level setting bits

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

If interrupt requests are input to the ITC simultaneously from multiple peripheral modules, the ITC outputs the interrupt request with the highest priority to the S1C17 core in accordance with the following conditions.

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

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

6.3.3 S1C17 Core Interrupt Processing

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

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

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

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

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

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

6.4 NMI

The S1C17702 can generate NMIs (non-maskable interrupts) using the watchdog timer. The vector number for NMIs is 2, and the vector address is set in the vector table initial address + 8 bytes. These interrupts take precedence over other interrupt factors and are accepted unconditionally by the S1C17 core.

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

6.5 Software Interrupts

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

6.6 HALT and SLEEP Mode Cancellation

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

Note:
- **If the CPU is able to receive interrupts when the HALT or SLEEP modes have been cleared by an interrupt request for the CPU from the ITC, processing branches to an interrupt handler routine immediately after cancellation. In all other cases, commands are executed following the halt and slp commands.**
- **HALT or SLEEP mode clearing due to interrupt requests cannot be masked (prohibited) using ITC interrupt level settings.**

For more information, refer to "C.1 Clock Control Power Saving" in Appendix C.

6.7 Control Register Details

Table 6.7.1: ITC registers

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

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

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

0x4306: Interrupt Level Setup Register 0 (ITC_LV0)

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

D[15:11] Reserved

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

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

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

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

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

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

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

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

D[7:3] Reserved

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

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

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

0x4308: Interrupt Level Setup Register 1 (ITC_LV1)

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

D[15:11] Reserved

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

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

D[7:3] Reserved

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

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

0x430a: Interrupt Level Setup Register 2 (ITC_LV2)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Interrupt Level Setup Register 2 (ITC_LV2)	**0x430a** (16 bits)	D15–11	**–**	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	

D[15:11] Reserved

D[10:8] ILV5[2:0]: SVD Interrupt Level Bits

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

D[7:3] Reserved

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

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

0x430c: Interrupt Level Setup Register 3 (ITC_LV3)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Interrupt Level Setup Register 3 (ITC_LV3)	**0x430c** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**ILV7[2:0]**	T16E Ch.0 interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**ILV6[2:0]**	LCD interrupt level	0 to 7	0x0	R/W	

D[15:11] Reserved

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

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

D[7:3] Reserved

D[2:0] ILV6[2:0]: LCD Interrupt Level Bits

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

0x430e: Interrupt Level Setup Register 4 (ITC_LV4)

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

D[15:11] Reserved

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

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

D[7:3] Reserved

D[2:0] ILV8[2:0]: 8-bit Timer Interrupt Level Bits

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

0x4310: Interrupt Level Setup Register 5 (ITC_LV5)

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

D[15:11] Reserved

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

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

D[7:3] Reserved

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

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

0x4312: Interrupt Level Setup Register 6 (ITC_LV6)

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

D[15:11] Reserved

D[10:8] ILV13[2:0]: UART Ch.1 Interrupt Level Bits

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

D[7:3] Reserved

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

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

0x4314: Interrupt Level Setup Register 7 (ITC_LV7)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Interrupt Level Setup Register 7 (ITC_LV7)	**0x4314** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**ILV15[2:0]**	I²C interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**ILV14[2:0]**	SPI interrupt level	0 to 7	0x0	R/W	

D[15:11] Reserved

D[10:8] ILV15[2:0]: I²C Interrupt Level Bits

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

D[7:3] Reserved

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

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

0x4316: Interrupt Level Setup Register 8 (ITC_LV8)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Interrupt Level Setup Register 8 (ITC_LV8)	**0x4316** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**ILV17[2:0]**	T16E 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	

D[15:11] Reserved

D[10:8] ILV17[2:0]: PWM & Capture Timer Ch.1 Interrupt Level Bits

Set the PWM & capture timer Ch.1 interrupt level (0 to 7). (Default: 0)
Refer to the discussion of ITC_LV0 register (0x4306) ILV1[2:0] (D[10:8]).

D[7:3] Reserved

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

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

6.8 Precautions

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

7 Oscillator Circuit (OSC)

7.1 OSC Module Configuration

The S1C17702 contains three internal oscillator circuits (IOSC, OSC3, and OSC1). The IOSC and OSC3 oscillator circuits generate the main clock for high-speed operation of the S1C17 core and peripheral circuits. The OSC1 oscillator circuit generates a sub-clock for timer and low-power operations.
The IOSC clock is selected as the system clock after initial resetting.
Oscillator circuit on/off switching and system clock selection (between IOSC/OSC3 and OSC1) is controlled by software. External clock output is also possible.
Figure 7.1.1 illustrates the clock system and OSC module configuration.

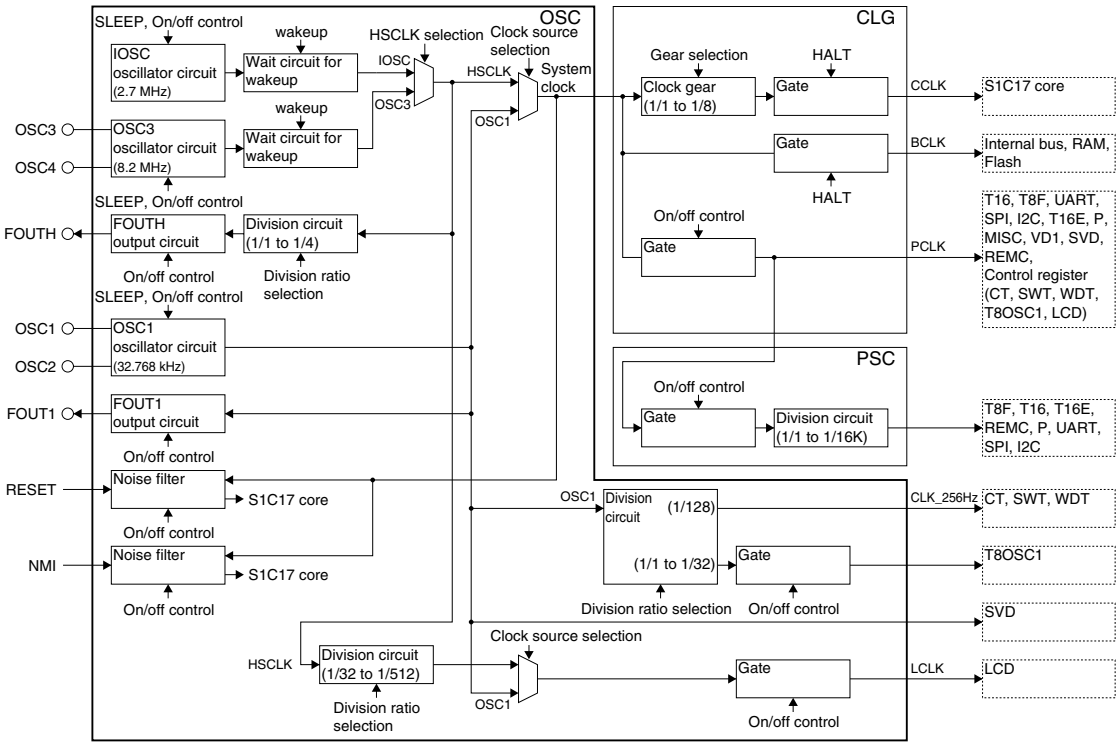


Figure 7.1.1: OSC module configuration

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

7.2 IOSC Oscillator Circuit

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

IOSC oscillation on/off

The IOSC oscillator circuit stops oscillating if IOSCEN (D2/OSC_CTL register) is set to 0 and begins oscillating if set to 1. The IOSC oscillator circuit also stops oscillating in SLEEP mode.

∗ **IOSCEN**: IOSC Enable Bit in the Oscillation Control (OSC_CTL) Register (D2/0x5061)

Following initial resetting, IOSCEN is set to 1 to turn the IOSC oscillator circuit on. Since the IOSC clock is used as the system clock, the S1C17 core begins operating using the IOSC clock.

The S1C17702 also contains an OSC3 oscillator circuit for high-speed clock generation. Selection of IOSC or OSC3 as the high-speed clock can occur after startup. For specific information on selecting the clock, refer to "7.5.1 High-speed Clock (HSCLK) Selection."

Stabilization wait time when IOSC oscillation begins

When using the IOSC clock, the IOSC oscillator circuit incorporates an oscillation stabilization wait timer to prevent malfunctions due to unstable clock operations when IOSC oscillation begins—e.g., when waking from SLEEP, or when the IOSC oscillator circuit is turned on via software. The IOSC clock is not fed to the system until the time set for this timer has elapsed.
Use IOSCWT[1:0] (D[7:6]/OSC_CTL register) to select one of four oscillation stabilization wait times.

∗ **IOSCWT[1:0]**: IOSC Wait Cycle Select Bits in the Oscillation Control (OSC_CTL) Register (D[7:6]/0x5061)

Table 7.2.1: 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 begins operating following the reset only after the duration indicated below (at a maximum) has elapsed.

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

When the system clock is switched to IOSC on waking from SLEEP or immediately after turning on the IOSC oscillator circuit, the IOSC clock is fed to the system only after the duration 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 feed wait time ≤ IOSC oscillation start time (max.) + IOSC oscillation stabilization wait time

7.3 OSC3 Oscillator Circuit

The OSC3 oscillator circuit is a high-precision, high-speed oscillator circuit that uses either a crystal or ceramic oscillator. It can be switched for use with the IOSC oscillator circuit.
Figure 7.3.1 illustrates the OSC3 oscillator circuit configuration.

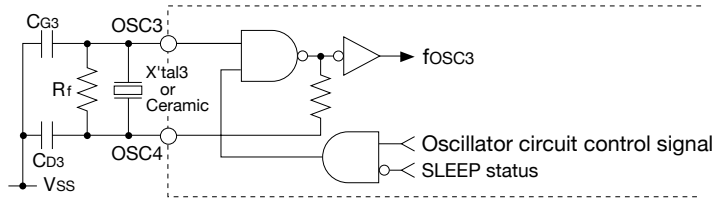


Figure 7.3.1: OSC3 oscillator circuit

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

OSC3 oscillation on/off

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

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

After an initial resetting, OSC3EN is set to 0 and the OSC3 oscillator circuit is halted. The IOSC clock is used as the default high-speed clock. To use the OSC3 clock, the clock must also be switched, in addition to the on/off controls described above. For specific information on switching, see "7.5 System Clock Switching."

Stabilization wait time at start of OSC3 oscillation

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

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

Table 7.3.1: OSC3 oscillation stabilization wait time settings

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

(Default: 0x0)

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

Note: Oscillation stability will vary, depending on the oscillator and other external components. Carefully consider the OSC3 oscillation stabilization wait time before reducing the time. When the system clock is switched to OSC3 immediately after the OSC3 oscillator circuit is switched on, the OSC3 clock is fed to the system only after the duration indicated below (at a maximum) has elapsed.

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

OSC3 external clock input

An external clock can be input to the OSC3 pin. When disabling the external clock supply, make sure the clock signal level is a V_{SS} level (do not stop while the clock signal is in a V_{DD} period). For more information on the input clock waveform, see "27 Electrical Characteristics."

7.4 OSC1 Oscillator Circuit

The OSC1 oscillator circuit is a high-precision, low-speed oscillator circuit that uses a 32.768 kHz crystal oscillator. The OSC1 clock is generally used as the timer operation clock (for the clock timer, stopwatch timer, watchdog timer, and 8-bit OSC1 timer). It reduces power consumption and can be used as the system clock instead of the IOSC or OSC3 clock when no high-speed processing is required.
Figure 7.4.1 illustrates the OSC1 oscillator circuit configuration.

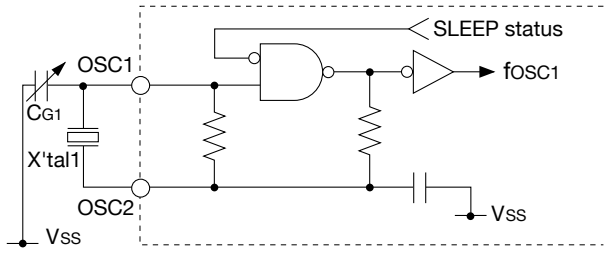


Figure 7.4.1: OSC1 oscillator circuit

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

OSC1 oscillation on/off

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

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

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

Stabilization wait time at start of OSC1 oscillation

The OSC1 oscillator circuit incorporates an oscillation stabilization wait timer (256 cycles) to prevent malfunctions caused by unstable clock operations at the start of OSC1 oscillation—for example, on initial power-up, on waking from SLEEP, or when the OSC1 oscillator circuit is turned on via software. When the system clock is switched to OSC1 immediately after the OSC1 oscillator circuit is turned on, the OSC1 clock is fed to the system only after the duration indicated below (at a maximum) has elapsed.

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

7.5 System Clock Switching

The S1C17702 system clock selector consists of high-speed clock (HSCLK) selection and OSC1 or HSCLK selection, as shown in Figure 7.5.1.

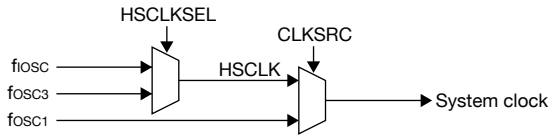


Figure 7.5.1: System clock selector

7.5.1 High-speed Clock (HSCLK) Selection

The S1C17702 incorporates IOSC and OSC3 oscillator circuits to generate the high-speed clock (HSCLK). The IOSC oscillator circuit is turned on following an initial reset, and the S1C17702 begins operating with the IOSC clock selected as HSCLK.
To select OSC3 as HSCLK, turn on the OSC3 oscillator circuit (see section 7.3) and write 1 to HSCLKSEL (D1/OSC_SRC register). To select IOSC as HSCLK, turn on the IOSC oscillator circuit (see section 7.2) and write 0 to HSCLKSEL.

∗ **HSCLKSEL**: High-speed Clock Select Bit in the Clock Source Select (OSC_SRC) Register (D1/0x5060)

Note: Both the IOSC and OSC3 oscillator circuits must be turned on when selecting HSCLK. Otherwise, the system will fail to switch to HSCLK even when HSCLKSEL is written to, and the HSCLKSEL value will remain unchanged. When selecting HSCLK, make sure PCKEN[1:0] (D[1:0]/CLG_PCLK register) is set to 0x3 (on) before writing to HSCLKSEL.

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

7.5.2 System Clock (OSC1 or HSCLK) Selection

The S1C17702 incorporates an OSC1 oscillator circuit to generate the low-speed clock. Either OSC1 or HSCLK can be selected as the system clock. Operations begin with HSCLK selected after an initial reset.
To select OSC1 as the system clock, turn on the OSC1 oscillator circuit (see section 7.4) and write 1 to CLKSRC (D0/OSC_SRC register). To select HSCLK as the system clock, write 0 to CLKSRC with HSCLK running.
When switch the system clock between the HSCLK and OSC1 clocks, it takes switching time from one cycle of HSCLK (min.) to one cycle of OSC1 (max.).

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

Oscillator circuits other than those selected as the system clock can be shut down to reduce consumption current, provided they are not used as operating clocks for peripheral circuits.

Note: • 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.
Table 7.5.2.1 shows the combinations of clock operating states and register settings enabling system clock (OSC1 or HSCLK) selection.

Table 7.5.2.1: System clock (OSC1/HSCLK) switching 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

• The oscillator circuit selected as the system clock cannot be turned off.
• Continuous write/read access to CLKSRC is prohibited. At least one command unrelated to CLKSRC access must be inserted between the write and read commands.

7.6 LCD Clock Control

The OSC module incorporates an LCD clock generator for generating the LCD driver operating clock (LCLK). For specific information on the LCD driver, see "22. LCD Driver LCD."

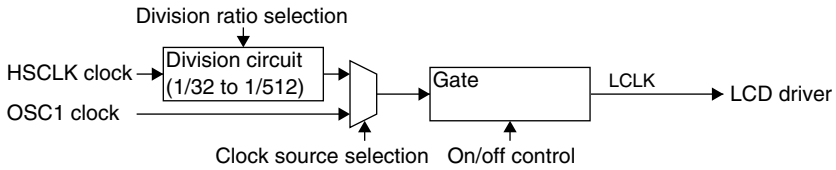


Figure 7.6.1: LCD clock generator

Clock source selection

Use LCKSRC (D1/OSC_LCLK register) to select whether OSC1 or HSCLK is used to generate the LCD clock. OSC1 is selected when LCKSRC is 1 (default), while HSCLK is selected when set to 0.

∗ **LCKSRC**: LCD Clock Source Select Bit in the LCD Clock Setup (OSC_LCLK) Register (D1/0x5063)

Clock division ratio selection

OSC1 clock

No division ratio needs to be selected if OSC1 has been selected for the clock source. The OSC1 clock (Typ 32.768 kHz) is sent to the LCD driver unchanged.

HSCLK clock

If HSCLK has been selected for the clock source, use LCKDV[2:0] (D[4:2]/OSC_LCLK register) to select the division ratio.

∗ **LCKDV[2:0]**: LCD Clock Division Ratio Select Bits in the LCD Clock Setup (OSC_LCLK) Register (D[4:2]/0x5063)

Table 7.6.1: LCD clock division ratio selection

LCKDV[2:0]	Division ratio
0x7 to 0x5	Reserved
0x4	HSCLK-1/512
0x3	HSCLK-1/256
0x2	HSCLK-1/128
0x1	HSCLK-1/64
0x0	HSCLK-1/32

(Default: 0x0)

Clock feed control

Clock feed to the LCD driver is controlled using LCKEN (D0/OSC_LCLK register).
The LCKEN default setting is 0, which stops the clock feed. Setting LCKEN to 1 sends the clock generated as above to the LCD driver.

∗ **LCKEN**: LCD Clock Enable Bit in the LCD Clock Setup (OSC_LCLK) Register (D0/0x5063)

Note: Be sure to wait at least one LCLK clock cycle before setting LCKEN to 0 to stop the LCLK clock feed immediately after setting DSPC[1:0] (D[1:0]/LCD_DCTL register) to 0x0 (display off). When resuming display with no wait time inserted after the LCLK feed is disabled, wait at least one LCLK clock cycle until DSPC[1:0] is set to a value other than 0x0 after LCKEN is set to 1 to resume clock feed.

∗ **DSPC[1:0]**: LCD Display Control Bits in the LCD Display Control (LCD_DCTL) Register (D[1:0]/0x50a0)

7.7 8-bit OSC1 Timer Clock Control

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

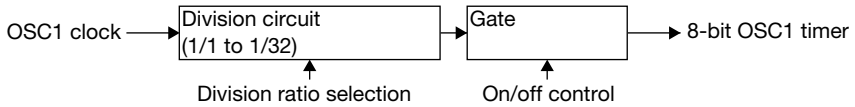


Figure 7.7.1: 8-bit OSC1 timer clock control circuit

Clock division ratio selection

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

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

Table 7.7.1: T8OSC1 clock division ratio selection

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

(Default: 0x0)

Clock feed control

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

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

7.8 Clock External Output (FOUTH, FOUT1)

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

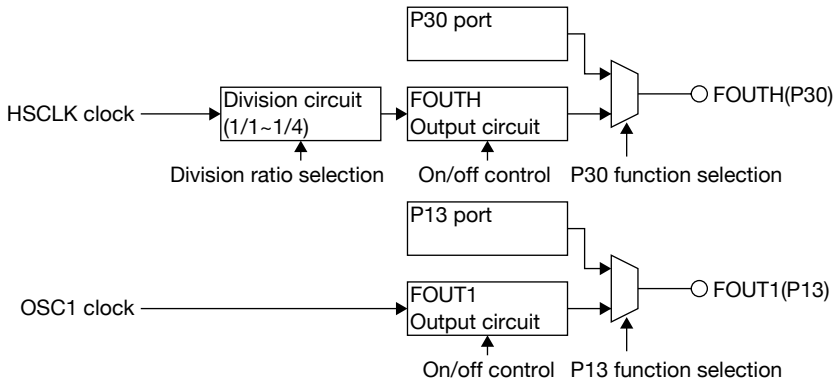


Figure 7.8.1: Clock output circuit

FOUTH output

FOUTH is the HSCLK division clock.

Output pin setting

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

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

FOUTH clock frequency selection

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

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

Table 7.8.1: FOUTH clock division ratio selection

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

(Default: 0x0)

Clock output control

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

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

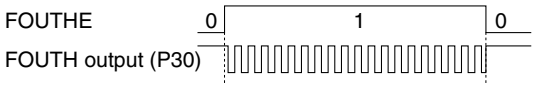


Figure 7.8.2: FOUTH output

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

FOUT1 output

FOUT1 is the OSC1 clock.

Output pin setting

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

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

Clock output control

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

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

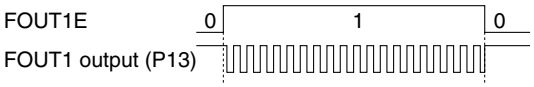


Figure 7.8.3: FOUT1 output

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

7.9 RESET and NMI Input Noise Filters

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

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

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

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

Note:
- **The RESET input noise filter should normally be enabled.**
- **The S1C17702 does not feature external NMI input pins, but the watchdog timer NMI request signal passes through these filters.**

7.10 Control Register Details

Table 7.10.1 OSC register list

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

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

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

0x5060: Clock Source Select Register (OSC_SRC)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Clock Source Select Register (OSC_SRC)	0x5060 (8 bits)	D7–2	–	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 IOSC and OSC3 oscillator circuits must be turned on when selecting HSCLK. Otherwise, the system will not switch to HSCLK, even when HSCLKSEL is written to, and the HSCLKSEL value will remain unchanged. When selecting HSCLK, make sure PCKEN[1:0] (D[1:0]/CLG_PCLK register) is set to 0x3 (on) before writing to HSCLKSEL.

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

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 power consumption.

Note: • 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.
Table 7.10.2 shows the combinations of clock operating states and register settings enabling system clock (OSC1 or HSCLK) selection.

Table 7.10.2: System clock (OSC1/HSCLK) switching 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

• The oscillator circuit selected as the system clock cannot be turned off.
• Continuous write/read access to CLKSRC is prohibited. At least one command unrelated to CLKSRC access must be inserted between the write and read commands.

0x5061: Oscillation Control Register (OSC_CTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Oscillation Control Register (OSC_CTL)	**0x5061** (8 bits)	D7–6	**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 when IOSC oscillation begins.

The IOSC clock is not fed to the system immediately after IOSC oscillation starts—e.g., when power is first turned on, when waking from SLEEP, or the IOSC oscillator circuit is switched on via software, until the time set here has elapsed.

Table 7.10.3: 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)

Since this is set to 64 cycles (IOSC clock) after an initial reset, the CPU begins operating following the reset only after the duration indicated below (at a maximum) has elapsed.

CPU operation start time on an initial reset ≤ IOSC oscillation start time (max.) + IOSC oscillation stabilization wait time (64 cycles)

When the system clock is switched to IOSC on waking from SLEEP or immediately after turning on the IOSC oscillator circuit, the IOSC clock is fed to the system only after the duration 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 feed 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 timer is set to prevent malfunctions due to unstable clock operation at the start of OSC3 oscillation.

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

Table 7.10.4: OSC3 oscillation stabilization wait time settings

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

(Default: 0x0)

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

Note: Note: Oscillation stability will vary, depending on the oscillator and other external components. Carefully consider the OSC3 oscillation stabilization wait time before reducing the time. When the system clock is switched to OSC3 immediately after the OSC3 oscillator circuit is turned on, the OSC3 clock is fed to the system only after the duration indicated below (at a maximum) has elapsed.

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

D3 Reserved

D2 IOSCEN: IOSC Enable Bit

Permits or prevents IOSC oscillator circuit operations.
1 (R/W): Permitted (on) (default)
0 (R/W): Prohibited (off)

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

D1 OSC1EN: OSC1 Enable Bit

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

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

• When the system clock is switched to OSC1 immediately after the OSC1 oscillator circuit is turned on, the OSC1 clock is fed to the system only after the duration indicated below (at a maximum) has elapsed.

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

D0 OSC3EN: OSC3 Enable Bit

Permits or prohibits OSC3 oscillator circuit operation.
1 (R/W): Permitted (on) (default)
0 (R/W): Prohibited (off)

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

0x5062: Noise Filter Enable Register (OSC_NFEN)

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

D[7:2] Reserved

D1 RSTFE: Reset Noise Filter Enable Bit

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

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

D0 NMIFE: NMI Noise Filter Enable Bit

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

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

Note: The S1C17702 does not feature external NMI input pins, but the watchdog timer NMI request signal passes through these filters.

0x5063: LCD Clock Setup Register (OSC_LCLK)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
LCD Clock Setup 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: HSCLK-1/512 0x3: HSCLK-1/256 0x2: HSCLK-1/128 0x1: HSCLK-1/64 0x0: HSCLK-1/32	0x0	R/W	
		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

Select the division ratio here when HSCLK has been selected for the LCD clock source.

Table 7.10.5: LCD clock division ratio selection

LCKDV[2:0]	Division ratio
0x7 to 0x5	Reserved
0x4	HSCLK-1/512
0x3	HSCLK-1/256
0x2	HSCLK-1/128
0x1	HSCLK-1/64
0x0	HSCLK-1/32

(Default: 0x0)

No division ratio needs to be selected if OSC1 has been selected for 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

D0 LCKEN: LCD Clock Enable Bit

Permits or prevents the LCD clock feed to the LCD driver.
1 (R/W): Permitted (on)
0 (R/W): Prohibited (off) (default)

The LCKEN default setting is 0, which stops the clock feed. Setting LCKEN to 1 sends the clock selected as above to the LCD driver. If no LCD display is required, stop the clock feed to minimize current consumption.

0x5064: FOUT Control Register (OSC_FOUT)

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

D[7:4] Reserved

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

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

Table 7.10.6: FOUTH clock division ratio selection

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

(Default: 0x0)

D1 FOUTHE: FOUTH Output Enable Bit

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

1 (R/W): Permitted (on)

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

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

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

D0 FOUT1E: FOUT1 Output Enable Bit

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

1 (R/W): Permitted (on)

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

Setting FOUT1E to 1 outputs the FOUT1 clock from the FOUT1 pin. Setting it to 0 stops the output. The FOUT1 output pin is combined with the P13 port. This functions as the P13 port pin by default, so the pin function should be changed by writing 1 to P13MUX (D3/P1_PMUX register) if use is required for FOUT1 output.

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

0x5065: T8OSC1 Clock Control Register (OSC_T8OSC1)

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

D[7:4] Reserved

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

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

Table 7.10.7: T8OSC1 clock division ratio selection

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

(Default: 0x0)

D0 T8O1CE: T8OSC1 Clock Output Enable Bit

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

1 (R/W): Permitted (on)

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

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

7.11 Precautions

- OSC3 oscillation stability will vary, depending on the oscillator and other external components. Carefully consider the OSC3 oscillation stabilization wait time before reducing the time. When the system clock is switched to OSC3 immediately after the OSC3 oscillator circuit is turned on, the OSC3 clock is fed to the system only after the duration indicated below (at a maximum) has elapsed.

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

- When the system clock is switched to OSC1 immediately after the OSC1 oscillator circuit is turned on, the OSC1 clock is fed to the system only after the duration indicated below (at a maximum) has elapsed.

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

- The IOSC oscillator circuit cannot be stopped if the IOSC clock is being used as the system clock.
- The OSC3 oscillator circuit cannot be stopped if the OSC3 clock is being used as the system clock.
- The OSC1 oscillator circuit cannot be stopped if the OSC1 clock is being used as the system clock.
- Since the FOUTH/FOUT1 signal is not synchronized with FOUTHE/FOUT1E writing, switching output on or off will generate certain hazards.
- Both the IOSC and OSC3 oscillator circuits must be turned on when selecting HSCLK. Otherwise, the system will fail to switch to HSCLK even when HSCLKSEL is written to, and the HSCLKSEL value will remain unchanged. When selecting HSCLK, make sure PCKEN[1:0] (D[1:0]/CLG_PCLK register) is set to 0x3 (on) before writing to HSCLKSEL.
- 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.
  Table 7.11.1 shows the combinations of clock operating states and register settings enabling system clock (OSC1 or HSCLK) selection.

Table 7.11.1: System clock (OSC1/HSCLK) switching 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

- The oscillator circuit selected as the system clock cannot be turned off.
- Continuous write/read access to CLKSRC is prohibited. At least one command unrelated to CLKSRC access must be inserted between the write and read commands.
- Be sure to wait at least one LCLK clock cycle before setting LCKEN (D0/OSC_LCLK register) to 0 to stop the LCLK clock feed immediately after setting DSPC[1:0] (D[1:0]/LCD_DCTL register) to 0x0 (display off). When resuming display with no wait time inserted after the LCLK feed is disabled, wait at least one LCLK clock cycle until DSPC[1:0] is set to a value other than 0x0 after LCKEN is set to 1 to resume clock feed.

8 Clock Generator (CLG)

8.1 Clock Generator Configuration

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

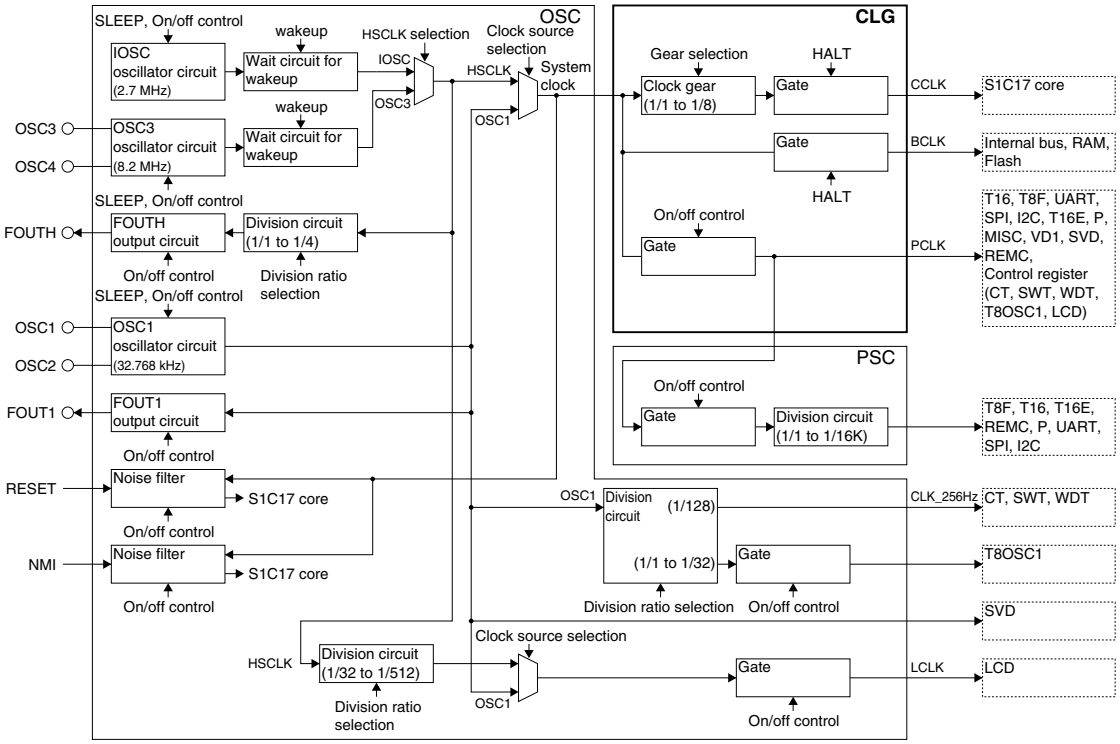


Figure 8.1.1: CLG module configuration

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

8.2 CPU Core Clock (CCLK) Control

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

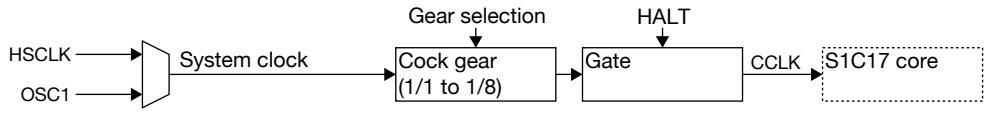


Figure 8.2.1: CCLK feed system

Clock gear settings

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

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

Table 8.2.1: CCLK gear ratio selection

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

(Default: 0x0)

Clock feed control

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

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

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

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

8.3 Peripheral Module Clock (PCLK) Control

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

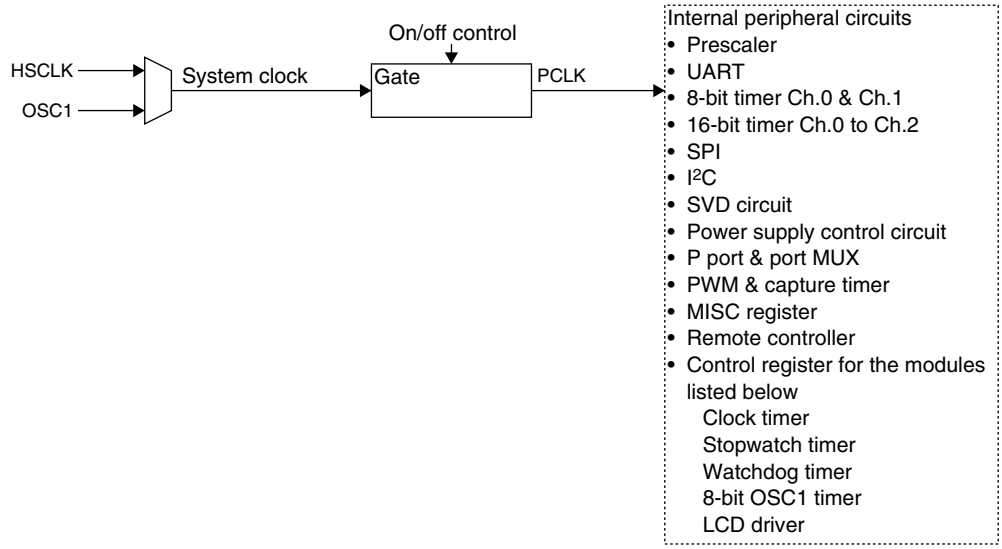


Figure 8.3.1: Peripheral module clock control circuit

Clock feed control

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

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

Table 8.3.1: PCLK control

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

(Default: 0x3)

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

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

Peripheral modules not operating on PCLK

With the exception of control register access, the following peripheral modules operate using a clock other than PCLK. Thus, PCLK is not required after setting the control register and initiating operations.

OSC1 peripheral module

The clock timer, stopwatch timer, watchdog timer, and 8-bit OSC1 timer operate using the OSC1 division clock. Stopping PCLK prevents read/program access to/from the control register, but operations will continue.

LCD driver

The LCD driver operates using the OSC1 clock or HSCLK division clock. Stopping PCLK prevents read/program access to/from the control register, but operations will continue.
PCLK is not required to access the display memory.

8.4 Control Register Details

Table 8.4.1 CLG register list

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

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

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

0x5080: PCLK Control Register (CLG_PCLK)

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

D[7:2] Reserved

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

Permits or prohibits clock (PCLK) feed to internal peripheral modules.

Table 8.4.2: PCLK control

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

(Default: 0x3)

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

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

Since the following peripheral modules are not operated using PCLK except for control register access, PCLK is not required after setting the control register to start operations.
- Clock timer
- Stopwatch timer
- Watchdog timer
- 8-bit OSC1 timer
- LCD driver

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

0x5081: CCLK Control Register (CLG_CCLK)

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

D[7:2] Reserved

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

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

Table 8.4.3: CCLK gear ratio selection

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

(Default: 0x0)

8.5 Precautions

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

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

Since the following peripheral modules are not operated using PCLK except for control register access, PCLK is not required after setting the control register to start operations.
- Clock timer
- Stopwatch timer
- Watchdog timer
- 8-bit OSC1 timer
- LCD driver

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

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

9 Prescaler (PSC)

9.1 Prescaler Configuration

The S1C17702 incorporates a prescaler to generate a clock for timer operations. The prescaler generates 15 different frequencies by dividing the PCLK clock fed from the clock generator into 1/1 to 1/16K. The peripheral modules to which the clock is fed include clock selection registers enabling selection of one as a count or operation clock.

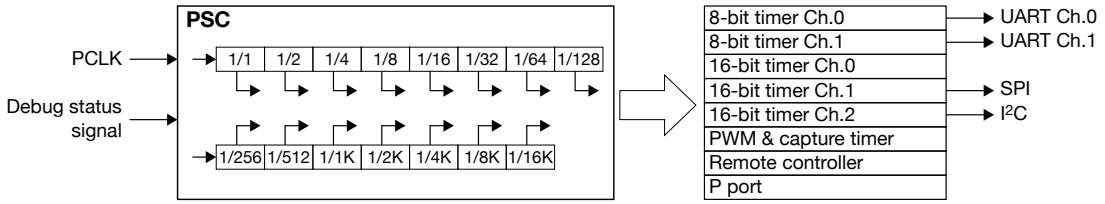


Figure 9.1.1: Prescaler

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

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

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

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

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

9.2 Control Register Details

Table 9.2.1: Prescaler register

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

The prescaler register is an 8-bit register.

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

0x4020: Prescaler Control Register (PSC_CTL)

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

D[7:2] Reserved

D1 PRUND: Prescaler Run/Stop Setting Bit for Debug Mode

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

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

D0 PRUN: Prescaler Run/Stop Control Bit

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

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

9.3 Precautions

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

10 Input/Output Port (P)

10.1 Input/Output Port Configuration

The S1C17702 includes 28 input/output ports (P0[7:0], P1[7:0], P2[7:0], P3[3:0]) to allow software switching of input/output direction. These share internal peripheral module input/output pins (with certain exceptions), but pins not used for peripheral modules can be used as general purpose input/output ports.
Figure 10.1.1 illustrates the input/output port configuration.

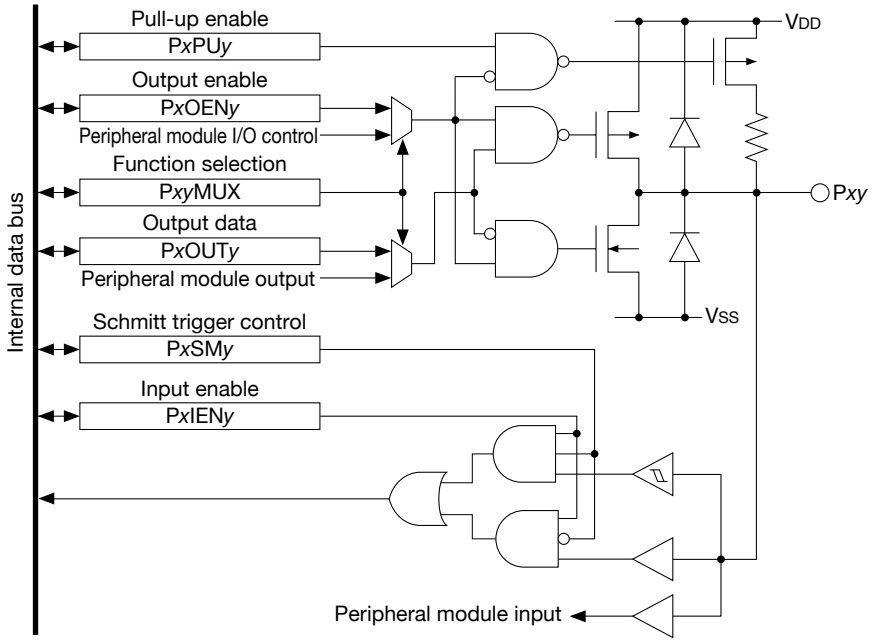


Figure 10.1.1: Input/output port configuration

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

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

10.2 Input/Output Pin Function Selection (Port MUX)

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

Table 10.2.1: Input/output pin function selection

Pin function 1 P*xx*MUX = 0	Pin function 2 P*xx*MUX = 1	Port function selection bit	Control register
P00/EXCL3 (T16E)	–	–	–
P01/EXCL4 (T16E)	–	–	–
P02	REMI (REMC)	P02MUX (D2)	P0 Port Function Select (P0_PMUX) Register (0x52a0)
P03	REMO (REMC)	P03MUX (D3)	
P04	TOUT4 (T16E)	P04MUX (D4)	
P05	TOUTN4 (T16E)	P05MUX (D5)	
P06/EXCL2 (T16)	–	–	–
P07/EXCL1 (T16)	–	–	–
P10	SIN1 (UART)	P10MUX (D0)	P1 Port Function Select (P1_PMUX) Register (0x52a1)
P11	SOUT1 (UART)	P11MUX (D1)	
P12	SCLK1 (UART)	P12MUX (D2)	
P13	FOUT1 (OSC)	P13MUX (D3)	
P14	SDA (I2C)	P14MUX (D4)	
P15	SCL (I2C)	P15MUX (D5)	
P16/EXCL0 (T16)	TOUT5 (T8OSC1)	P16MUX (D6)	
P17	#SPISS (SPI)	P17MUX (D7)	
P20	SDI (SPI)	P20MUX (D0)	P2 Port Function Select (P2_PMUX) Register (0x52a2)
P21	SDO (SPI)	P21MUX (D1)	
P22	SPICLK (SPI)	P22MUX (D2)	
P23	SIN0 (UART)	P23MUX (D3)	
P24	SOUT0 (UART)	P24MUX (D4)	
P25	SCLK0 (UART)	P25MUX (D5)	
P26	TOUT3 (T16E)	P26MUX (D6)	
P27	TOUTN3 (T16E)	P27MUX (D7)	
P30	FOUTH (OSC)	P30MUX (D0)	P3 Port Function Select (P3_PMUX) Register (0x52a3)
DCLK (DBG)	P31	P31MUX (D1)	
DST2 (DBG)	P32	P32MUX (D2)	
DSIO (DBG)	P33	P33MUX (D3)	

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

Pins P00, P01, P06, P07, and P16 can also be used as 16-bit timer external clock input pins by setting them to input mode. They are simultaneously set to function as general purpose input ports.

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

10.3 Data Input/Output

Data input/output control

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

* P0OEN[7:0]: P0[7:0] Port Output Enable Bits in the P0 Port Output Enable (P0_OEN) Register (D[7:0]/0x5202)
* P1OEN[7:0]: P1[7:0] Port Output Enable Bits in the P1 Port Output Enable (P1_OEN) Register (D[7:0]/0x5212)
* P2OEN[7:0]: P2[7:0] Port Output Enable Bits in the P2 Port Output Enable (P2_OEN) Register (D[7:0]/0x5222)
* P3OEN[3:0]: P3[3:0] Port Output Enable Bits in the P3 Port Output Enable (P3_OEN) Register (D[3:0]/0x5232)

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

Table 10.3.1: Data input/output table

P*x*OEN[7:0] output control	P*x*IEN[7:0] input control	P*x*PU[7:0] pull-up control	Port status
0	1	0	Functions as input port (pull-up off). Port pin (external input signal) value can be read out from P*x*IN[7:0] (input data). Output is prohibited.
0	1	1	Functions as input port (pull-up on). (Default) Port pin (external input signal) value can be read out from P*x*IN[7:0] (input data). Output is prohibited.
1	0	1 or 0	Functions as output port (pull-up off). Input is prohibited. The value read from P*x*IN[7:0] (input data) is 0.
1	1	1 or 0	Functions as output port (pull-up off). Input is also permitted. The port pin value (output value) can be read out from P*x*IN[7:0] (input data).
0	0	0	Pin switches to high-impedance (pull-up off). Output/input is prohibited. The value read from P*x*IN[7:0] (input data) is 0.
0	0	1	Pin switches to high-impedance (pull-up on). Output/input is prohibited. The value read from P*x*IN[7:0] (input data) is 0.

The input/output direction of ports with peripheral module function selected is controlled by the peripheral module. P*x*OEN[7:0] and P*x*IEN[7:0] settings are ignored.

Data input

To input the port pin state and read out the value, set P*x*IEN[7:0] to 1 (default) and permit input.

To input an external signal, P*x*OEN[7:0] should also be set to 0 (default). This will switch the input/output port to high-impedance and function as an input port (input mode). The port will be pulled up if pull-up is enabled by the P*x*_PU register.
In input mode, the input pin state can be read out directly from P*x*IN[7:0] (P*x*_IN register). The value read will be 1 when the input pin is at High (HV_{DD}) level and 0 when it is at Low (V_{SS}) level.

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

The port pin state will be input if P*x*IEN[7:0] is 1, even when output is permitted (P*x*OEN[7:0] = 1) (output mode). In this case, the value actually output can be read out from P*x*IN[7:0].

If P*x*IEN[7:0] is set to 0, input will be prohibited, and the value read out from P*x*IN[7:0] is 0.

Data output

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

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

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

10.4 Pull-up Control

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

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

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

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

A delay will occur in the waveform 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 by the internal pull-up resistor. An appropriate wait time must be set for the input/output port loading. The wait time set should be a value not less than that calculated from the following equation.

Wait time = $R_{IN} \times (C_{IN}$ + load capacitance on board) $\times 1.6$ [s]

R_{IN}: pull-up resistance maximum value
C_{IN}: pin capacitance maximum value

10.5 Input Interface Level

The input/output port input interface level can be selected individually for each bit using PxSM[7:0] (Px_SM register).

- **P0SM[7:0]**: P0[7:0] Port Schmitt Trigger Input Enable Bits in the P0 Port Schmitt Trigger Control (P0_SM) Register (D[7:0]/0x5204)
- **P1SM[7:0]**: P1[7:0] Port Schmitt Trigger Input Enable Bits in the P1 Port Schmitt Trigger Control (P1_SM) Register (D[7:0]/0x5214)
- **P2SM[7:0]**: P2[7:0] Port Schmitt Trigger Input Enable Bits in the P2 Port Schmitt Trigger Control (P2_SM) Register (D[7:0]/0x5224)
- **P3SM[3:0]**: P3[3:0] Port Schmitt Trigger Input Enable Bits in the P3 Port Schmitt Trigger Control (P3_SM) Register (D[3:0]/0x5234)

Setting PxSM[7:0] to 1 (default) selects CMOS Schmitt level; setting to 0 selects CMOS level.

10.6 P0 and P1 Port Chattering Filter Function

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

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

Table 10.6.1: Chattering filter function settings

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

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

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

10.7 Port Input Interrupt

Ports P0 and P1 include input interrupt functions.
Select which of the 16 ports are to be used for interrupts based on requirements. You can also select whether interrupts are generated for either the rising edge or falling edge of input signals.
Figure 10.7.1 illustrates the port input interrupt circuit configuration.

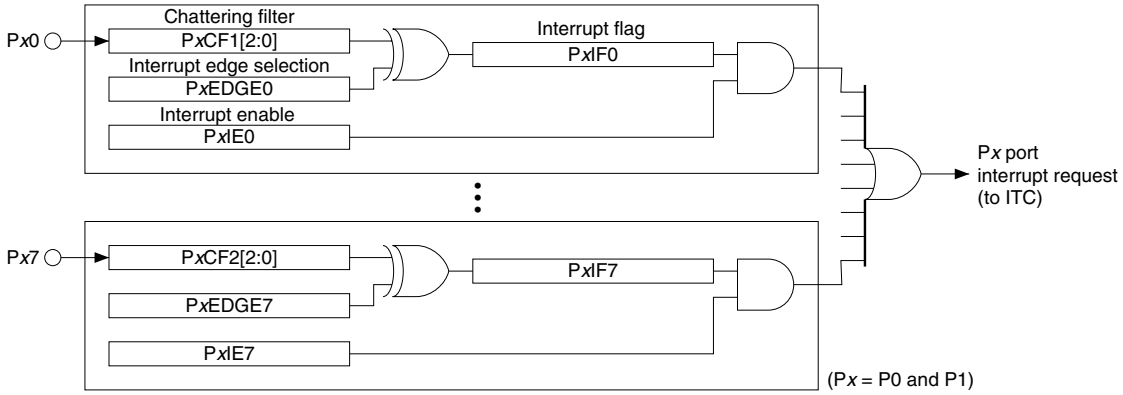


Figure 10.7.1: Port input interrupt circuit configuration

Interrupt port selection

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

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

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

Interrupt edge selection

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

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

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

Interrupt flags

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

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

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

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

Interrupt vector

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

Table 10.7.1: Port interrupt vectors

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

Other interrupt settings

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

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

10.8 Control Register Details

Table 10.8.1: Input/output port control register list

Address	Register name		Function
0x5200	P0_IN	P0 Port Input Data Register	P0 port input data
0x5201	P0_OUT	P0 Port Output Data Register	P0 port output data
0x5202	P0_OEN	P0 Port Output Enable Register	P0 port output enable
0x5203	P0_PU	P0 Port Pull-up Control Register	P0 port pull-up control
0x5204	P0_SM	P0 Port Schmitt Trigger Control Register	P0 port Schmitt trigger control
0x5205	P0_IMSK	P0 Port Interrupt Mask Register	P0 port interrupt mask setting
0x5206	P0_EDGE	P0 Port Interrupt Edge Select Register	P0 port interrupt edge selection
0x5207	P0_IFLG	P0 Port Interrupt Flag Register	P0 port interrupt occurrence status display/reset
0x5208	P0_CHAT	P0 Port Chattering Filter Control Register	P0 port chattering filter control
0x5209	P0_KRST	P0 Port Key-Entry Reset Configuration Register	P0 port key entry reset setting
0x520a	P0_IEN	P0 Port Input Enable Register	P0 port input enable
0x5210	P1_IN	P1 Port Input Data Register	P1 port input data
0x5211	P1_OUT	P1 Port Output Data Register	P1 port output data
0x5212	P1_OEN	P1 Port Output Enable Register	P1 port output enable
0x5213	P1_PU	P1 Port Pull-up Control Register	P1 port pull-up control
0x5214	P1_SM	P1 Port Schmitt Trigger Control Register	P1 port Schmitt trigger control
0x5215	P1_IMSK	P1 Port Interrupt Mask Register	P1 port interrupt mask setting
0x5216	P1_EDGE	P1 Port Interrupt Edge Select Register	P1 port interrupt edge selection
0x5217	P1_IFLG	P1 Port Interrupt Flag Register	P1 port interrupt occurrence status display/reset
0x5218	P1_CHAT	P1 Port Chattering Filter Control Register	P1 port chattering filter control
0x521a	P1_IEN	P1 Port Input Enable Register	P1 port input enable
0x5220	P2_IN	P2 Port Input Data Register	P2 port input data
0x5221	P2_OUT	P2 Port Output Data Register	P2 port output data
0x5222	P2_OEN	P2 Output Enable Register	P2 port output enable
0x5223	P2_PU	P2 Port Pull-up Control Register	P2 port pull-up control
0x5224	P2_SM	P2 Port Schmitt Trigger Control Register	P2 port Schmitt trigger control
0x522a	P2_IEN	P2 Port Input Enable Register	P2 port input enable
0x5230	P3_IN	P3 Port Input Data Register	P3 port input data
0x5231	P3_OUT	P3 Port Output Data Register	P3 port output data
0x5232	P3_OEN	P3 Port Output Enable Register	P3 port output enable
0x5233	P3_PU	P3 Port Pull-up Control Register	P3 port pull-up control
0x5234	P3_SM	P3 Port Schmitt Trigger Control Register	P3 port Schmitt trigger control
0x523a	P3_IEN	P3 Port Input Enable Register	P3 port input enable
0x52a0	P0_PMUX	P0 Port Function Select Register	P0 port function selection
0x52a1	P1_PMUX	P1 Port Function Select Register	P1 port function selection
0x52a2	P2_PMUX	P2 Port Function Select Register	P2 port function selection
0x52a3	P3_PMUX	P3 Port Function Select Register	P3 port function selection

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

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

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

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

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

D[7:0] P*x*IN[7:0]: P*x*[7:0] Port Input Data Bits (P3 port is P3IN[3:0])

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

1(R): High level

0(R): Low level

P*x*IN[7:0] correspond directly to the P*x*[7:0] pins. The pin voltage level can be read out when input is permitted (P*x*IEN[7:0] = 1) (even when output is permitted (P*x*OEN[7:0] = 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 prohibited (P*x*IEN[7:0] = 0).

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

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

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

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

D[7:0] P*x*OUT[7:0]: P*x*[7:0] Port Output Data Bits (P3 port is P3OUT[3:0])

Set the data to be output from the port pin.

1(R/W): High level

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

P*x*OUT [7:0] correspond directly to the P*x*[7:0] pins. The data written will be output unchanged from the port pins when output is permitted (P*x*OEN[7:0] = 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 prohibited (P*x*OEN[7:0] = 0) (the pin state is unaffected).

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

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

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

D[7:0] P*x*OEN[7:0]: P*x*[7:0] Port Output Enable Bits (P3 port is P3OEN[3:0])

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

1(R/W): Permitted

0(R/W): Prohibited (default)

P*x*OEN[7:0] are output enable bits that correspond directly to P*x*[7:0] ports. Setting to 1 permits output and outputs the corresponding P*x*OUT[7:0] setting from the port pin. Output is prohibited when set to 0, and the port pin is set to high-impedance. The peripheral module determines whether output is permitted or prohibited when the port is used for peripheral modules.

Refer to Table 10.3.1 for more information on input/output states for ports, including settings other than for the P*x*OEN register.

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

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

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

D[7:0] P*x*PU[7:0]: P*x*[7:0] Port Pull-up Enable Bits (P3 port is P3PU[3:0])

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

P*x*PU[7:0] are the pull-up control bits that correspond directly to the P*x*[7:0] ports. Setting to 1 enables pull-up resistance and pulls up the port pin when output is prohibited (P*x*OEN[7:0] = 0). When setting to 0, the pin is not pulled up.
When output is permitted (P*x*OEN[7:0] = 1), the P*x*PU[7:0] setting is disabled, and the pin is not pulled up.
Input/output ports that are not used should be set with pull-up enabled.
This pull-up setting is also enabled for ports for which the peripheral module input function is 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 by the internal pull-up resistor. An appropriate wait time must be set for the input/output port loading. The wait time set should be a value not less than that calculated from the following equation.

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

0x5204/0x5214/0x5224/0x5234: 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 (0xff)	R/W	
P1 Port Schmitt Trigger Control Register (P1_SM)	**0x5214** (8 bits)	D7–0	**P1SM[7:0]**	P1[7:0] port Schmitt trigger input enable	1	Enable (Schmitt)	0	Disable (CMOS)	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	Disable (CMOS)	1 (0xff)	R/W	
P3 Port Schmitt Trigger Control Register (P3_SM)	**0x5234** (8 bits)	D7–4	**–**	reserved	–				–	–	0 when being read.
		D3–0	**P3SM[3:0]**	P3[3:0] port Schmitt trigger input enable	1	Enable (Schmitt)	0	Disable (CMOS)	1 (0xff)	R/W	

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

D[7:0] P*x*SM[7:0]: P*x*[7:0] Port Schmitt Trigger Input Enable Bits (P3 port is P3SM[3:0])

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

P*x*SM[7:0] are Schmitt input control bits that correspond directly to the P*x*[7:0] ports. Setting to 1 enables the Schmitt input buffer, and setting to 0 uses the CMOS level input buffer.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

These are interrupt flags indicating the interrupt factor occurrence status.

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

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

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

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

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

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

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

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

D7 Reserved

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

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

D3 Reserved

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

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

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

Table 10.8.2: 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 OSC3 = 2 MHz and PCLK = OSC3)

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

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

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

D[7:2] Reserved

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

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

Table 10.8.3: P0 port key entry input reset settings

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

(Default: 0x0)

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

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

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

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

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

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

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

D[7:0] P*x*IEN[7:0]: P*x*[7:0] Port Input Enable Bits (P3 port is P3IEN[3:0])

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

P*x*IEN[7:0] are input enable bits that correspond directly to the P*x*[7:0] ports. Setting to 1 permits input and reads the corresponding port pin input or output signal level from the P*x*_IN register. Setting to 0 prohibits input.
Refer to Table 10.3.1 for more information on port input/output states, including settings other than for the P*x*IEN register.

0x52a0: P0 Port Function Select Register (P0_PMUX)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
P0 Port Function Select Register (P0_PMUX)	0x52a0 (8 bits)	D7–6	–	reserved	–				–	–	0 when being read.
		D5	P05MUX	P05 port function select	1	TOUTN4	0	P05	0	R/W	
		D4	P04MUX	P04 port function select	1	TOUT4	0	P04	0	R/W	
		D3	P03MUX	P03 port function select	1	REMO	0	P03	0	R/W	
		D2	P02MUX	P02 port function select	1	REMI	0	P02	0	R/W	
		D1–0	–	reserved	–				–	–	0 when being read.

The P02 to P05 input/output port pins are shared with the peripheral module pins. This register is used to select how the pins are used.

D[7:6] Reserved

D5 P05MUX: P05 Port Function Select Bit

1 (R/W): TOUTN4 (T16E Ch.1)
0 (R/W): P05 port (default)

D4 P04MUX: P04 Port Function Select Bit

1 (R/W): TOUT4 (T16E Ch.1)
0 (R/W): P04 port (default)

D3 P03MUX: P03 Port Function Select Bit

1 (R/W): REMO (REMC)
0 (R/W): P03 port (default)

D2 P02MUX: P02 Port Function Select Bit

1 (R/W): REMI (REMC)
0 (R/W): P02 port (default)

D[1:0] Reserved

0x52a1: P1 Port Function Select Register (P1_PMUX)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
P1 Port Function Select Register (P1_PMUX)	0x52a1 (8 bits)	D7	**P17MUX**	P17 port function select	1	#SPISS	0	P17	0	R/W	
		D6	**P16MUX**	P16 port function select	1	TOUT5	0	P16/EXCL0	0	R/W	
		D5	**P15MUX**	P15 port function select	1	SCL	0	P15	0	R/W	
		D4	**P14MUX**	P14 port function select	1	SDA	0	P14	0	R/W	
		D3	**P13MUX**	P13 port function select	1	FOUT1	0	P13	0	R/W	
		D2	**P12MUX**	P12 port function select	1	SCLK1	0	P12	0	R/W	
		D1	**P11MUX**	P11 port function select	1	SOUT1	0	P11	0	R/W	
		D0	**P10MUX**	P10 port function select	1	SIN1	0	P10	0	R/W	

The P10 to P17 input/output port pins are shared with the peripheral module pins. This register is used to select how the pins are used.

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

D6 P16MUX: P16 Port Function Select Bit
1 (R/W): TOUT5 (T8OSC1)
0 (R/W): P16 port/EXCL0 (T16) (default)

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

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

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

D2 P12MUX: P12 Port Function Select Bit
1 (R/W): SCLK1 (UART Ch.1)
0 (R/W): P12 port (default)

D1 P11MUX: P11 Port Function Select Bit
1 (R/W): SOUT1 (UART Ch.1)
0 (R/W): P11 port (default)

D0 P10MUX: P13 Port Function Select Bit
1 (R/W): SIN1 (UART Ch.1)
0 (R/W): P10 port (default)

0x52a2: P2 Port Function Select Register (P2_PMUX)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
P2 Port Function Select Register (P2_PMUX)	0x52a2 (8 bits)	D7	**P27MUX**	P27 port function select	1	TOUTN3	0	P27	0	R/W	
		D6	**P26MUX**	P26 port function select	1	TOUT3	0	P26	0	R/W	
		D5	**P25MUX**	P25 port function select	1	SCLK0	0	P25	0	R/W	
		D4	**P24MUX**	P24 port function select	1	SOUT0	0	P24	0	R/W	
		D3	**P23MUX**	P23 port function select	1	SIN0	0	P23	0	R/W	
		D2	**P22MUX**	P22 port function select	1	SPICLK	0	P22	0	R/W	
		D1	**P21MUX**	P21 port function select	1	SDO	0	P21	0	R/W	
		D0	**P20MUX**	P20 port function select	1	SDI	0	P20	0	R/W	

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

D7 **P27MUX: P27 Port Function Select Bit**
1 (R/W): TOUTN3 (T16E Ch.0)
0 (R/W): P27 port (default)

D6 **P26MUX: P26 Port Function Select Bit**
1 (R/W): TOUT3 (T16E Ch.0)
0 (R/W): P26 port (default)

D5 **P25MUX: P25 Port Function Select Bit**
1 (R/W): SCLK0 (UART Ch.0)
0 (R/W): P25 port (default)

D4 **P24MUX: P24 Port Function Select Bit**
1 (R/W): SOUT0 (UART Ch.0)
0 (R/W): P24 port (default)

D3 **P23MUX: P23 Port Function Select Bit**
1 (R/W): SIN0 (UART Ch.0)
0 (R/W): P23 port (default)

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

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

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

0x52a3: P3 Port Function Select Register (P3_PMUX)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
P3 Port Function Select Register (P3_PMUX)	0x52a3 (8 bits)	D7–4	–	reserved	–				–	–	0 when being read.
		D3	P33MUX	P33 port function select	1	P33	0	DSIO	0	R/W	
		D2	P32MUX	P32 port function select	1	P32	0	DST2	0	R/W	
		D1	P31MUX	P31 port function select	1	P31	0	DCLK	0	R/W	
		D0	P30MUX	P30 port function select	1	FOUTH	0	P30	0	R/W	

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

D[7:4] Reserved

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

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

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

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

10.9 Precautions

Operation clock

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

Pull-up

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

P0 and P1 port interrupts

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

P0/P1 Port chattering filter circuit

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

P0 port key-entry reset

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

11 16-bit Timer (T16)

11.1 16-bit Timer Overview

The S1C17702 incorporates a 3-channel 16-bit timer (T16).
The 16-bit timer consists of a 16-bit presettable down counter and a 16-bit reload data register holding the preset values. The timer counts down from the initial value set in the reload data register and outputs an underflow signal when the counter underflows. The underflow signal is used to generate an interrupt and an internal serial interface clock. The underflow cycle can be programmed by selecting the prescaler clock and reload data, enabling the application program to obtain time intervals and serial transfer speeds as required.
The timer also combines an event counter function via the input/output port pins and the external input signal pulse width measurement function.
Figure 11.1.1 illustrates the 16-bit timer configuration.

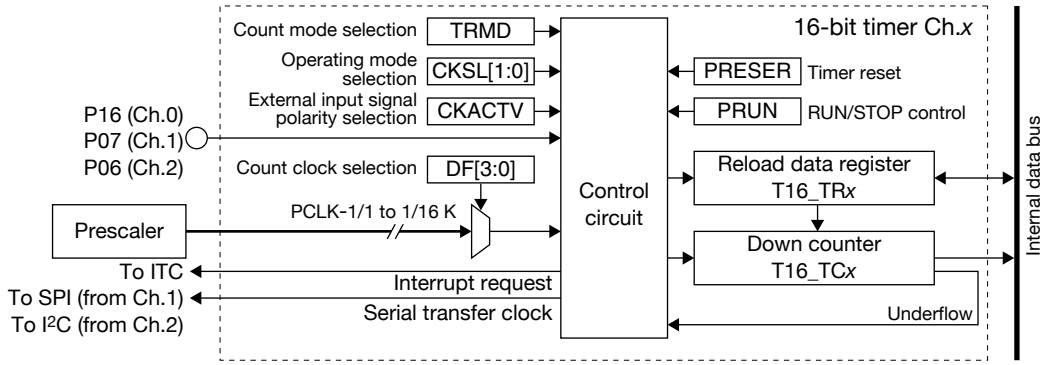


Figure 11.1.1: 16-bit timer configuration (1-channel)

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

11.2 16-bit Timer Operating Modes

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

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

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

Table 11.2.1: Operating mode selection

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

(Default: 0x0)

11.2.1 Internal Clock Mode

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

Count clock selection

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

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

Table 11.2.1.1: Count clock selection

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

(Default: 0x0)

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

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

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

11.2.2 External Clock Mode

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

External clock input port

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

Table 11.2.2.1: External clock input port

Timer channel	Input signal name	Input/output port pin
Ch.0	EXCL0	P16
Ch.1	EXCL1	P07
Ch.2	EXCL2	P06

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

The P16, P07, and P06 ports used by 16-bit timer Ch.1 and Ch.2 incorporate chattering filter circuits and can also be used as EXCL*x* inputs. For instructions on controlling chattering filter circuits, see "10.7 P0/P1 Port Chattering Filter Function."

Signal polarity selection

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

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

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

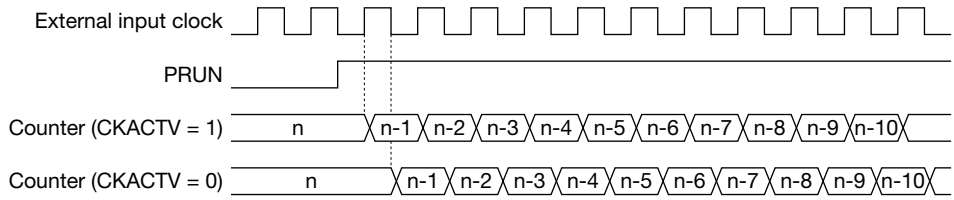


Figure 11.2.2.1: External clock mode count

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

11.2.3 Pulse Width Measurement Mode

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

Pulse input port

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

Count clock selection

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

Signal polarity selection

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

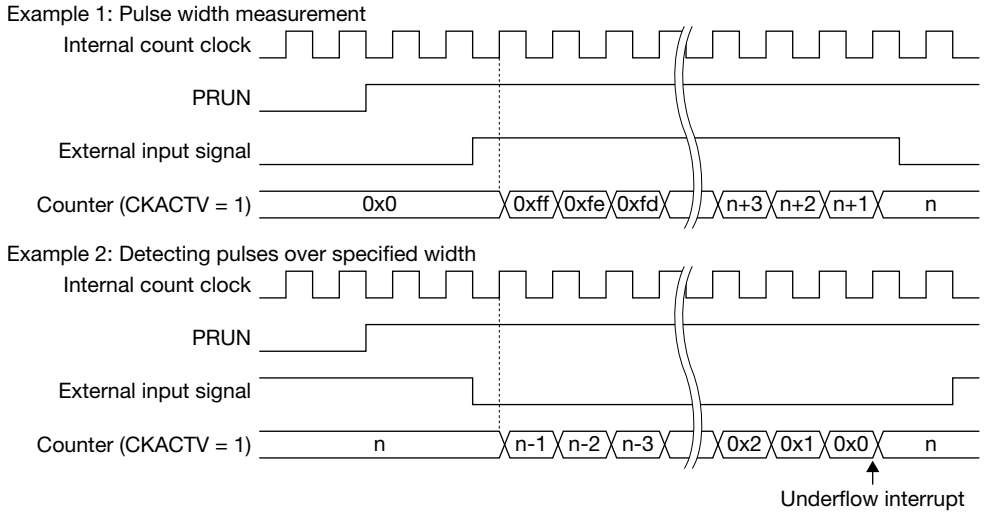


Figure 11.2.3.1: Pulse width measurement mode count operation

11.3 Count Mode

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

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

Repeat mode (TRMD = 0, default)

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

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

One-shot mode (TRMD = 1)

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

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

11.4 16-bit Timer Reload Register and Underflow Cycle

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

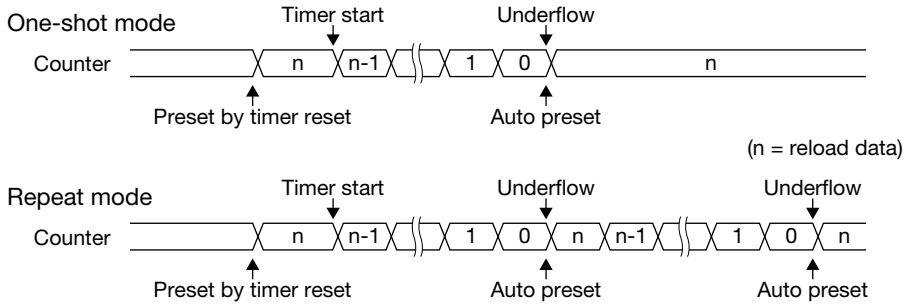


Figure 11.4.1: Preset timing

The underflow cycle can be calculated as follows:

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

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

11.5 16-bit Timer Reset

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

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

11.6 16-bit Timer RUN/STOP Control

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

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

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

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

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

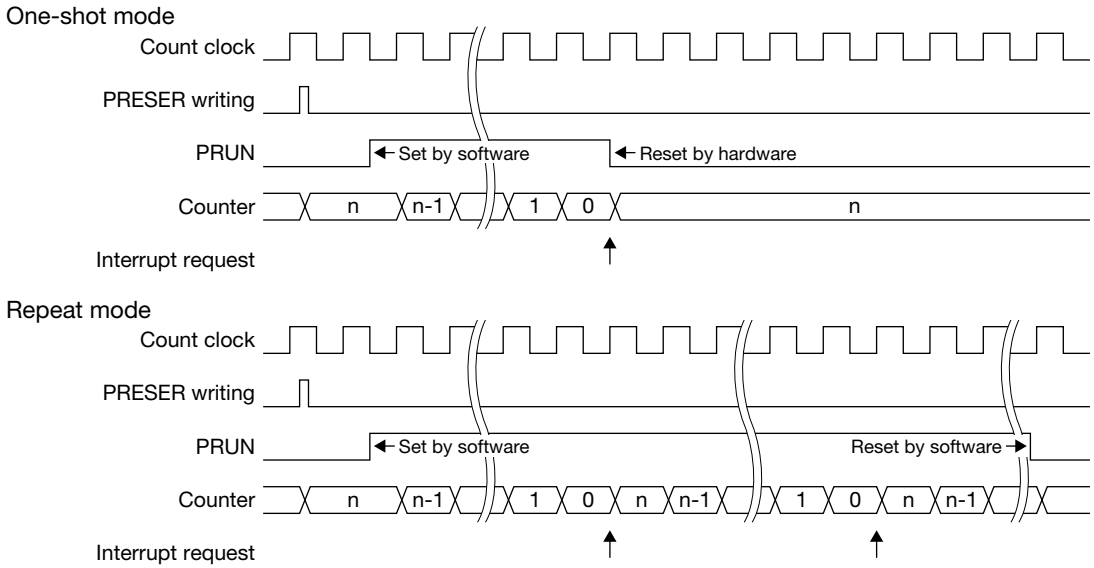


Figure 11.6.1: Count operation

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

11.7 16-bit Timer Output Signal

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

The clock generated is sent to the internal serial interface, as shown below.
16-bit timer Ch.1 output clock → SPI
16-bit timer Ch.2 output clock → I²C

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

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

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

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

11.8 16-bit Timer Interrupts

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

Underflow interrupt

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

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

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

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

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

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

Interrupt vectors

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

Table 11.8.2: Timer interrupt vectors

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

Other interrupt settings

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

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

11.9 Control Register Details

Table 11.9.1: 16-bit timer register list

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

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

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

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

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
16-bit Timer Ch.x Input Clock Select Register (T16_CLKx)	0x4220 0x4240 0x4260 (16 bits)	D15–4	–	reserved	–		–	–	0 when being read.
		D3–0	DF[3:0]	Timer input clock select (Prescaler output clock)	DF[3:0]	Clock	0x0	R/W	
					0xf	reserved			
					0xe	PCLK-1/16384			
					0xd	PCLK-1/8192			
					0xc	PCLK-1/4096			
					0xb	PCLK-1/2048			
					0xa	PCLK-1/1024			
					0x9	PCLK-1/512			
					0x8	PCLK-1/256			
					0x7	PCLK-1/128			
					0x6	PCLK-1/64			
					0x5	PCLK-1/32			
					0x4	PCLK-1/16			
					0x3	PCLK-1/8			
					0x2	PCLK-1/4			
					0x1	PCLK-1/2			
					0x0	PCLK-1/1			

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

D[15:4] Reserved

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

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

Table 11.9.2: Count clock selection

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

(Default: 0x0)

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

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

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
16-bit Timer Ch.x Reload Data Register (T16_TRx)	**0x4222** **0x4242** **0x4262** (16 bits)	D15–0	**TR[15:0]**	16-bit timer reload data TR15 = MSB TR0 = LSB	0x0 to 0xffff	0x0	R/W	

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

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

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

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

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

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

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
16-bit Timer Ch.x Counter Data Register (T16_TCx)	**0x4224** **0x4244** **0x4264** (16 bits)	D15–0	**TC[15:0]**	16-bit timer counter data TC15 = MSB TC0 = LSB	0x0 to 0xffff	0xffff	R	

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

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

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

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

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

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
16-bit Timer Ch.x Control Register (T16_CTLx)	0x4226 0x4246 0x4266 (16 bits)	D15–11	–	reserved	–	–	–	0 when being read.
		D10	CKACTV	External clock active level select	1 High 0 Low	1	R/W	
		D9–8	CKSL[1:0]	Input clock and pulse width measurement mode select	CKSL[1:0] Mode 0x3 reserved 0x2 Pulse width 0x1 External clock 0x0 Internal clock	0x0	R/W	
		D7–5	–	reserved	–	–	–	0 when being read.
		D4	TRMD	Count mode select	1 One shot 0 Repeat	0	R/W	
		D3–2	–	reserved	–	–	–	0 when being read.
		D1	PRESER	Timer reset	1 Reset 0 Ignored	0	W	
		D0	PRUN	Timer run/stop control	1 Run 0 Stop	0	R/W	

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

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

D[15:11] Reserved

D10 CKACTV: External Clock Active Level Select Bit

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

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

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

Select the 16-bit timer operating mode.

Table 11.9.3: Operating mode selection

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

(Default: 0x0)

Internal clock mode uses the prescaler output clock as the count clock. The timer counts down from the initial value set in the reload data register and outputs an underflow signal when the counter underflows. The underflow signal is used to generate an interrupt and an internal serial interface clock. The time until underflow occurs can be finely programmed by selecting the prescaler clock and initial counter value, allowing its use for serial transfer clock generation and sporadic time measurement.

External clock mode uses the clock and pulses input via the input/output ports (Ch.0: P16, Ch.1: P07, Ch.2: P06) as a count clock and can also be used as an event counter. Timer operations other than the input clock are the same as for internal clock mode.

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

D[7:5] Reserved

D4 TRMD: Count Mode Select Bit

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

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

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

D[3:2] Reserved

D1 PRESER: Timer Reset Bit

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

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

D0 PRUN: Timer Run/Stop Control Bit

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

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

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

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
16-bit Timer Ch.x Interrupt Control Register (T16_INTx)	0x4228 0x4248 0x4268 (16 bits)	D15–9	–	reserved	–	–	–	0 when being read.
		D8	T16IE	16-bit timer interrupt enable	1 Enable 0 Disable	0	R/W	
		D7–1	–	reserved	–	–	–	0 when being read.
		D0	T16IF	16-bit timer interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.

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

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

D[15:9] Reserved

D8 T16IE: 16-bit Timer Interrupt Enable Bit

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

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

D[7:1] Reserved

D0 T16IF: 16-bit Timer Interrupt Flag

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

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

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

11.10 Precautions

- The prescaler must run before the 16-bit timer.
- Set the count clock and count mode only while the 16-bit timer count is stopped.
- To prevent interrupt recurrences, the T16 module interrupt flag T16IF (D0/T16_INT*x* register) must be reset within the interrupt handler routine following a 16-bit timer interrupt.

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

- To prevent unwanted interrupts, reset T16IF before permitting 16-bit timer interrupts with T16IE (D8/T16_INT*x* register).

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

12 8-bit Timer (T8F)

12.1 8-bit Timer Overview

The S1C17702 incorporates an 8-bit timer (two channels) with Fine mode.
The 8-bit timer consists of an 8-bit presettable down counter and an 8-bit reload data register holding the preset values. The timer counts down from the initial value set in the reload data register and outputs an underflow signal when the counter underflows. The underflow signal is used to generate an interrupt and UART clock. The underflow cycle can be programmed by selecting the prescaler clock and reload data, enabling the application program to obtain time intervals and serial transfer speeds as required. Fine mode provides a function that minimizes transfer rate errors.
Figure 12.1.1 illustrates the 8-bit timer configuration.

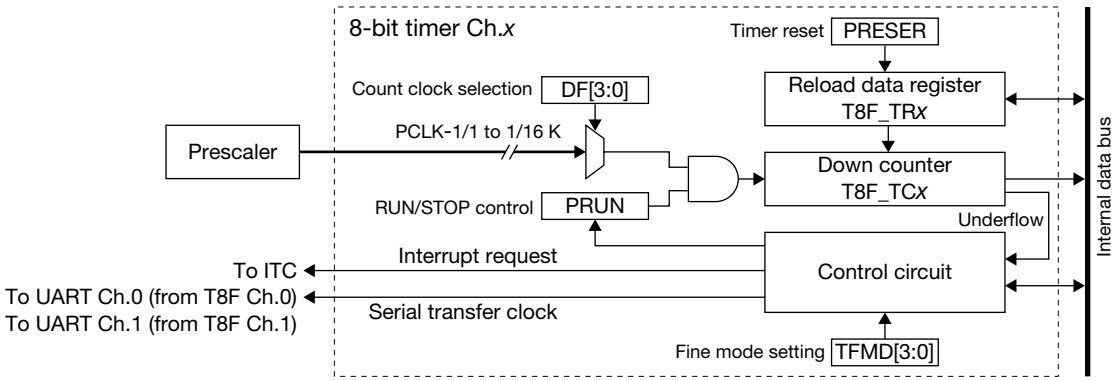


Figure 12.1.1: 8-bit timer configuration (One channel)

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

12.2 8-bit Timer Count Mode

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

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

Repeat mode (TRMD = 0, default)

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

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

One-shot mode (TRMD = 1)

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

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

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

12.3 Count Clock

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

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

Table 12.3.1: Count clock selection

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

(Default: 0x0)

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

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

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

12.4 8-bit Timer Reload Register and Underflow Cycle

The reload data register T8F_TR*x* (0x4202/0x4282) is used to set the initial value for the down counter.
The initial counter value set in the reload data register is preset to the down counter if the 8-bit timer is reset or the counter underflows. If the 8-bit timer is started after resetting, the timer counts down from the reload value (initial value). This means this reload value and the input clock frequency 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.

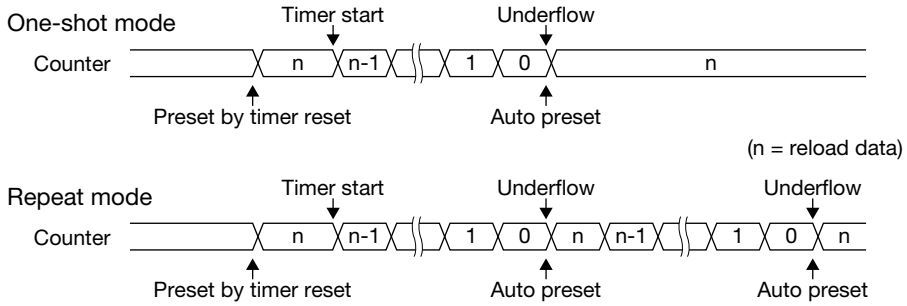


Figure 12.4.1: Preset timing

The underflow cycle can be calculated as follows:

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

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

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

12.5 8-bit Timer Reset

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

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

12.6 8-bit Timer RUN/STOP Control

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

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

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

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

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

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

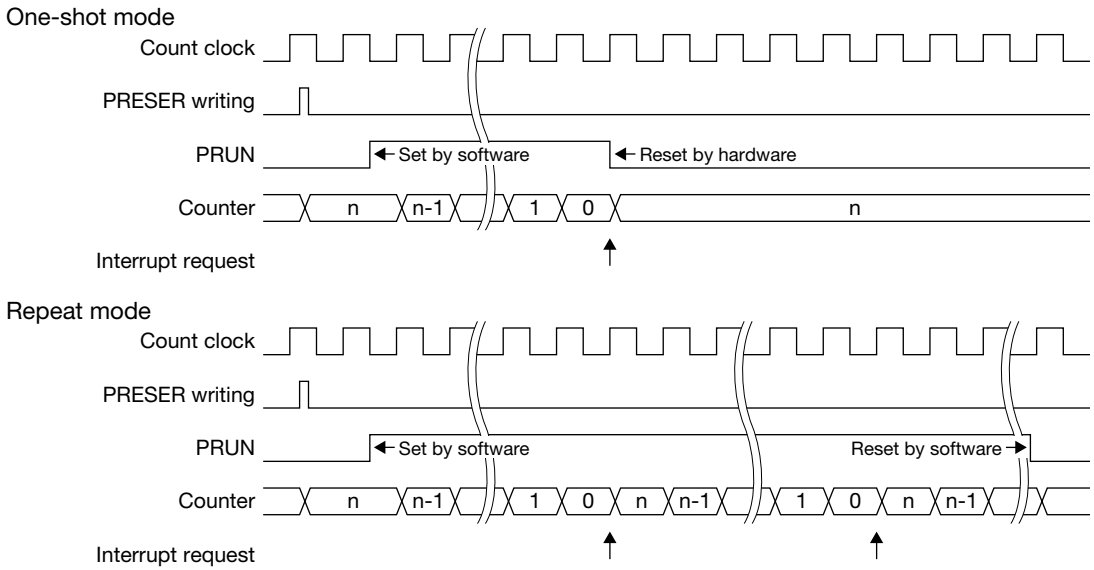


Figure 12.6.1: Count operation

12.7 8-bit Timer Output Signal

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

The underflow pulses are also used to generate the serial transfer clock and are transmitted to the UART.
Use the following equations to calculate the reload data register value for obtaining the desired transfer rate.
8-bit timer Ch.0 output clock → UART Ch.0
8-bit timer Ch.1 output clock → UART Ch.1

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

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

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

12.8 Fine Mode

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

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

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

Table 12.8.1: Delay patterns specified by TFMD[3:0]

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

D: Indicates the insertion of a delay cycle.

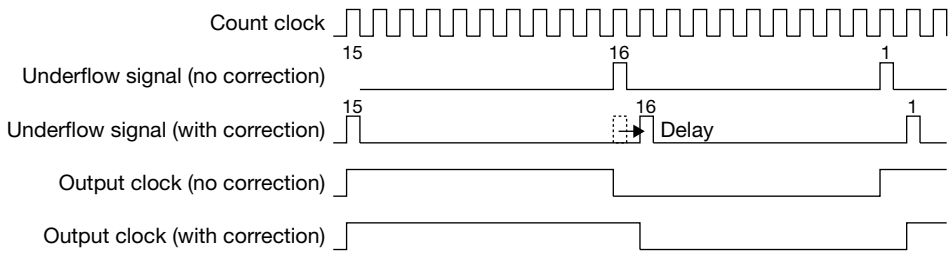


Figure 12.8.1: Delay cycle insertion in Fine mode

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

12.9 8-bit Timer Interrupts

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

Underflow interrupt

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

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

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

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

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

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

Interrupt vectors

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

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

Other interrupt settings

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

12.10 Control Register Details

Table 12.10.1: 8-bit timer register list

Address	Register name		Function
0x4200	T8F_CLK	8-bit Timer Input Clock Select Register	Prescaler output clock selection
0x4202	T8F_TR	8-bit Timer Reload Data Register	Reload data setting
0x4204	T8F_TC	8-bit Timer Counter Data Register	Counter data
0x4206	T8F_CTL	8-bit Timer Control Register	Timer mode setting and timer RUN/STOP
0x4208	T8F_INT	8-bit Timer Interrupt Control Register	Interrupt control
0x4280	T8F_CLK1	8-bit Timer Ch.1 Input Clock Select Register	Prescaler output clock selection
0x4282	T8F_TR1	8-bit Timer Ch.1 Reload Data Register	Reload data setting
0x4284	T8F_TC1	8-bit Timer Ch.1 Counter Data Register	Counter data
0x4286	T8F_CTL1	8-bit Timer Ch.1 Control Register	Timer mode setting and timer RUN/STOP
0x4288	T8F_INT1	8-bit Timer Ch.1 Interrupt Control Register	Interrupt control

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

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

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

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
8-bit Timer Ch.*x* Input Clock Select Register (T8F_CLK*x*)	0x4200 0x4280 (16 bits)	D15–4	–	reserved	–		–	–	0 when being read.
		D3–0	DF[3:0]	8-bit timer input clock select (Prescaler output clock)	DF[3:0]	Clock	0x0	R/W	
					0xf	reserved			
					0xe	PCLK-1/16384			
					0xd	PCLK-1/8192			
					0xc	PCLK-1/4096			
					0xb	PCLK-1/2048			
					0xa	PCLK-1/1024			
					0x9	PCLK-1/512			
					0x8	PCLK-1/256			
					0x7	PCLK-1/128			
					0x6	PCLK-1/64			
					0x5	PCLK-1/32			
					0x4	PCLK-1/16			
					0x3	PCLK-1/8			
					0x2	PCLK-1/4			
					0x1	PCLK-1/2			
					0x0	PCLK-1/1			

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

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

D[15:4] Reserved

D[3:0] DF[3:0]: 8-bit Timer Input Clock Select Bits

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

Table 12.10.2: Count clock selection

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

(Default: 0x0)

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

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

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

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

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

D[15:8] Reserved

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

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

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

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

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

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

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

D[15:8] Reserved

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

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

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
8-bit Timer Ch.*x*	**0x4206**	D15–12	**–**	reserved	–	–	–	0 when being read.
Control Register (T8F_CTL*x*)	**0x4286**	D11–8	**TFMD[3:0]**	Fine mode setup	0x0 to 0xf	0x0	R/W	Set a number of times to insert delay into a 16-underflow period.
		D7–5	**–**	reserved	–	–	–	0 when being read.
		D4	**TRMD**	Count mode select	1 One shot 0 Repeat	0	R/W	
		D3–2	**–**	reserved	–	–	–	0 when being read.
		D1	**PRESER**	Timer reset	1 Reset 0 Ignored	0	W	
		D0	**PRUN**	Timer run/stop control	1 Run 0 Stop	0	R/W	

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

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

D[15:12] Reserved

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

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

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

Table 12.10.3: Delay patterns specified by TFMD[3:0]

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

D: Indicates the insertion of a delay cycle.

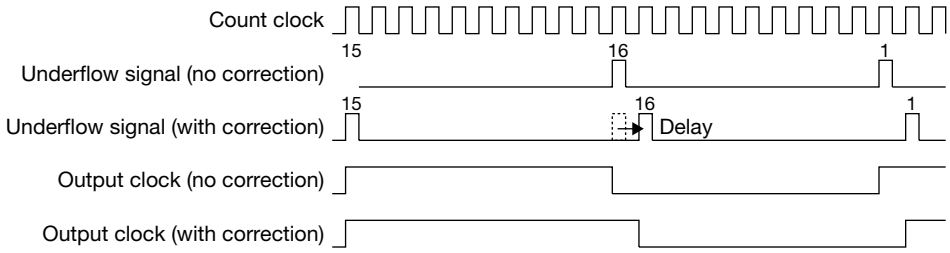


Figure 12.10.1: Delay cycle insertion in Fine mode

D[7:5] Reserved

D4 TRMD: Count Mode Select Bit

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

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

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

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

D[3:2] Reserved

D1 PRESER: Timer Reset Bit

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

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

D0 PRUN: Timer Run/Stop Control Bit

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

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

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

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

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

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

D[15:9] Reserved

D8 T8IE: 8-bit Timer Interrupt Enable Bit

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

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

D[7:1] Reserved

D0 T8IF: 8-bit Timer Interrupt Flag

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

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

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

12.11 Precautions

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

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

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

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

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

13 PWM & Capture Timer (T16E)

13.1 PWM & Capture Timer Overview

The S1C17702 incorporates a double-channel PWM & capture timer.
Figure 13.1.1 illustrates the PWM & capture timer configuration.

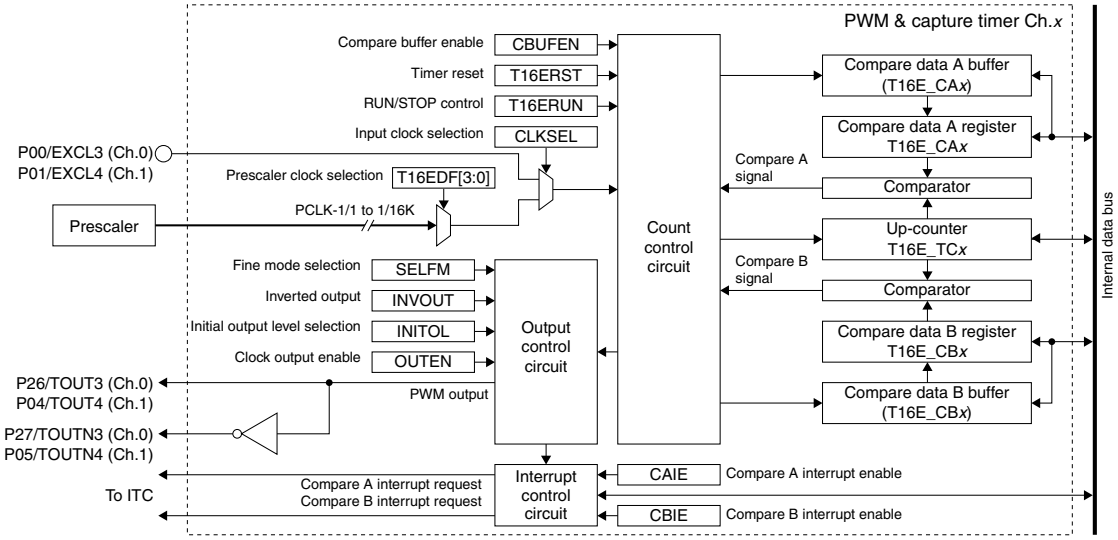


Figure 13.1.1: PWM & capture timer configuration

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

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

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

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

Note: The PWM & capture timer (T16E) modules for the two channels have the same functions, except for control register addresses. For this reason, the description in this section applies to all PWM & capture timer channels. The "*x*" in the register name indicates the channel number (0 or 1). Register addresses are indicated either as "Ch.0" or "Ch.1".
E.g.: T16E_CTL*x* register (0x5306/0x5366)
Ch.0: T16_CTL0 register (0x5306)
Ch.1: T16_CTL1 register (0x5366)

13.2 PWM & Capture Timer Operating Modes

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

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

∗ **CLKSEL**: Input Clock Select Bit in the PWM Timer Ch.*x* Control (T16E_CTL*x*) Register (D3/0x5306/0x5366)

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

Internal clock mode

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

∗ **T16EDF[3:0]**: Timer Input Clock Select Bits in the PWM Timer Ch.*x* Input Clock Select (T16E_CLK*x*) Register (D[3:0]/0x5308/0x5368)

Table 13.2.1: Prescaler clock selection

T16EDF[3:0]	Prescaler output clock	T16EDF[3:0]	Prescaler output clock
0xf	Reserved	0x7	PCLK-1/128
0xe	PCLK-1/16384	0x6	PCLK-1/64
0xd	PCLK-1/8192	0x5	PCLK-1/32
0xc	PCLK-1/4096	0x4	PCLK-1/16
0xb	PCLK-1/2048	0x3	PCLK-1/8
0xa	PCLK-1/1024	0x2	PCLK-1/4
0x9	PCLK-1/512	0x1	PCLK-1/2
0x8	PCLK-1/256	0x0	PCLK-1/1

(Default: 0x0)

Note: • **The prescaler must run before operating the PWM & capture timer in internal clock mode.**

• **Make sure the PWM & capture timer count is halted before changing count clock settings.**

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

External clock mode

In external clock mode, channel 0 uses the clock or pulse input via the P00 (EXCL3) port, while channel 1 uses the clock or pulse input via the P01 (EXCL4) port as the count clock. They can therefore also be used as an event counter. Timer operations other than input clock are the same as for internal clock mode.
The input/output port used for external clock/pulse input should be set to input mode (the default value). No pin function needs to be selected. While the input/output port functions as a general input, the input signal is also sent to the PWM & capture timer.

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

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

13.3 Setting and Resetting Counter Value

The PWM & capture timer counter can be reset to 0 by writing 1 to the T16ERST bit (D1/T16E_CTL*x* register).

∗ **T16ERST**: Timer Reset Bit in the PWM Timer Ch.*x* Control (T16E_CTL*x*) Register (D1/0x5306/0x5366)

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

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

∗ **T16ETC[15:0]**: Counter Data in the PWM Timer Ch.*x* Counter Data (T16E_TC*x*) Register (D[15:0]/0x5304/0x5364)

13.4 Compare Data Settings

Compare data register/buffer selection

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

* **CBUFEN**: Comparison Buffer Enable Bit in the PWM Timer Control (T16E_CTL*x*) Register (D5/0x5306/0x5366)

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

Compare data writing

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

* **T16ECA[15:0]**: Compare Data A in the PWM Timer Ch.*x* Compare Data A (T16E_CA*x*) Register (D[15:0]/0x5300/0x5360)
* **T16ECB[15:0]**: Compare Data B in the PWM Timer Ch.*x* Compare Data B (T16E_CB*x*) Register (D[15:0]/0x5302/0x5362)

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

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

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

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

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

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

13.5 PWM & Capture Timer RUN/STOP Control

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

The PWM & capture timer includes T16ERUN (D0/T16E_CTL*x* register) to control Run/Stop.

* **T16ERUN**: Timer Run/Stop Control Bit in the PWM Timer Ch.*x* Control (T16E_CTL*x*) Register (D0/0x5306/0x5366)

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

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

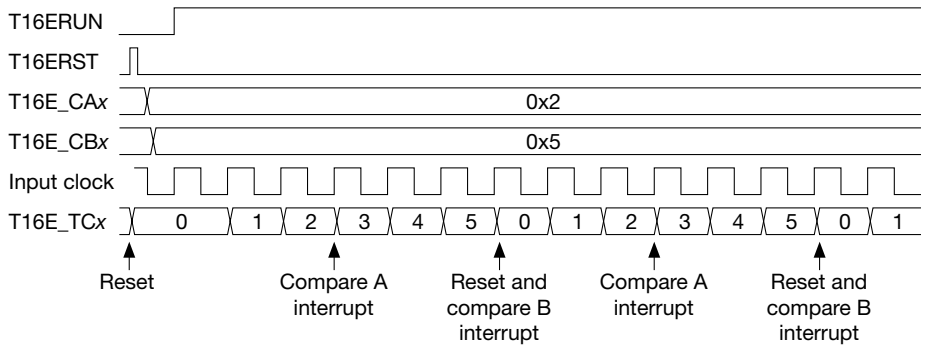


Figure 13.5.1: Basic counter operation timing

13.6 Clock Output Control

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

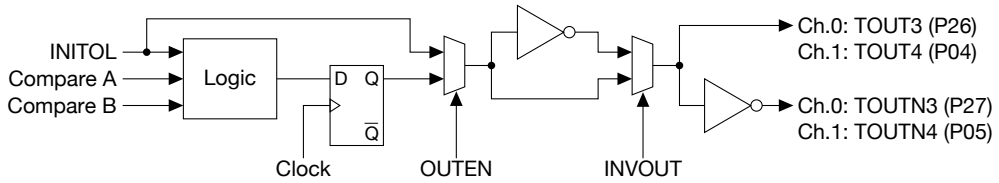


Figure 13.6.1: PWM & capture timer clock output circuit

Initial output level settings

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

∗ **INITOL**: Initial Output Level Select Bit in the PWM Timer Ch.*x* Control (T16E_CTL*x*) Register (D8/0x5306/0x5366)

The TOUT*x* initial output level is Low (TOUTN*x* output is High) when INITOL is 0 (the default value). Setting to 1 switches the initial output level to High (TOUTN*x* output is Low).

Output signal polarity selection

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

∗ **INVOUT**: Inverse Output Control Bit in the PWM Timer Ch.*x* Control (T16E_CTL*x*) Register (D4/0x5306/0x5366)

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

Output pin settings

The TOUTx/TOUTN*x* signal generated here can be output from the following pins and can provide a programmable clock and PWM signal to external devices.
Ch.0: TOUT3 output → TOUT3 (P26) pin, TOUTN3 output → TOUTN3 (P27) pin
Ch.1: TOUT4 output → TOUT4 (P04) pin, TOUTN4 output → TOUTN4 (P05) pin

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

Table 13.6.1: Initial output level

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

Clock output start

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

∗ **OUTEN**: Clock Output Enable Bit in the PWM Timer Ch.*x* Control (T16E_CTL*x*) Register (D2/0x5306/0x5366)

Figure 13.6.2 illustrates the output waveform.

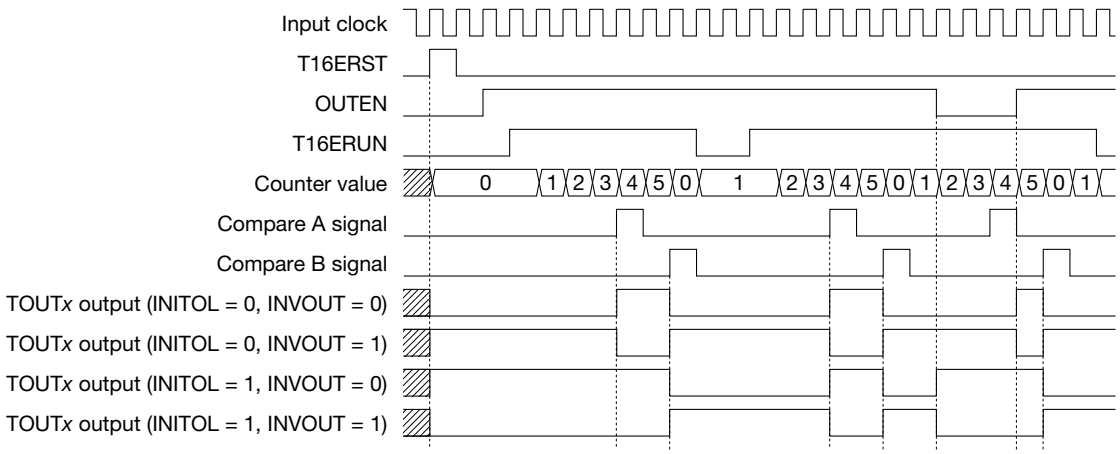


Figure 13.6.2: PWM & capture timer output waveform

When INVOUT = 0 (Active High)

The timer outputs Low level (initial output level at output start) until the counter matches the compare data A set in the T16E_CA*x* register (0x5300/0x5360). When the counter reaches the next compare data A value, the output pin switches to High level, and a compare A interrupt factor is generated. If the counter subsequently counts up to compare data B set in the T16E_CB*x* register (0x5302/0x5362), the counter is reset and the output pin is returned to the Low level. A compare B interrupt factor is also generated at the same time.
The TOUTN*x* pins output the inverted signals described above.

When INVOUT = 1 (Active Low)

The timer outputs High level (inverted value of the initial output level at output start) until the counter matches the compare data A set in the T16E_CA*x* register (0x5300/0x5360). When the counter reaches the next compare data A value, the output pin switches to Low level, and a compare A interrupt factor is generated. If the counter subsequently counts up to compare data B set in the T16E_CB*x* register (0x5302/0x5362), the counter is reset and the output pin is returned to the High level. A compare B interrupt factor is also generated at the same time.
The TOUTN*x* pins output the inverted signals described above.

Clock output Fine mode settings

With the default settings, the clock output changes at the input clock rising edge if the counter value matches the compare data A.

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

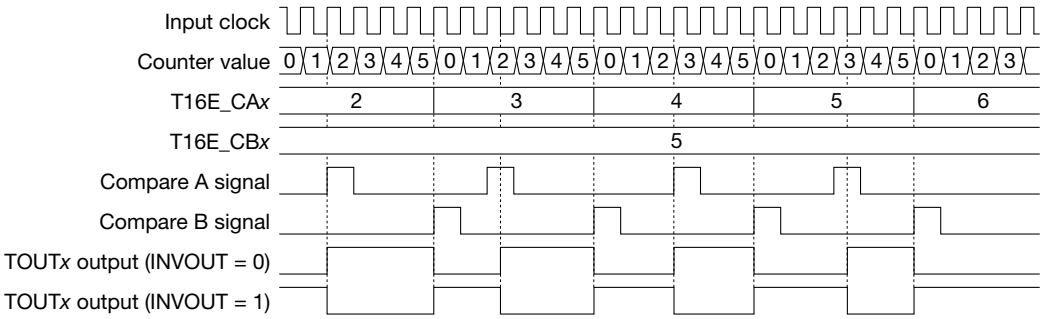


Figure 13.6.3: Fine mode clock output

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

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

∗ **SELFM**: Fine Mode Select Bit in the PWM Timer Ch.*x* Control (T16E_CTL*x*) Register (D6/0x5306/0x5366)

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

Precautions

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

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

13.7 PWM & Capture Timer Interrupts

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

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

Compare A match interrupt

This interrupt request is generated when the counter matches the compare data A register setting during counting. It sets the interrupt flag CAIF (D0/T16E_IFLG*x* register) within the T16E module to 1.

∗ **CAIF**: Compare A Interrupt Flag in the PWM Timer Ch.*x* Interrupt Flag (T16E_IFLG*x*) Register (D0/0x530c/0x536c)

To use this interrupt, set CAIE (D0/T16E_IMSK*x* register) to 1. If CAIE is set to 0 (default), CAIF is not set to 1, and the interrupt request for this factor is not sent to the ITC.

∗ **CAIE**: Compare A Interrupt Enable Bit in the PWM Timer Ch.*x* Interrupt Mask (T16E_IMSK*x*) Register (D0/0x530a/0x536a)

If CAIF is set to 1, the T16E module outputs an interrupt request to the ITC. An interrupt is generated if the ITC and S1C17 core interrupt conditions are satisfied.
CAIF should be read and checked within the PWM & capture timer interrupt handler routine to determine whether the PWM & capture timer interrupt is attributable to compare A matching.

Compare B match interrupt

This interrupt request is generated when the counter matches the compare data B register setting during counting. It sets the interrupt flag CBIF (D1/T16E_IFLG*x* register) within the T16E module to 1.

∗ **CBIF**: Compare B Interrupt Flag in the PWM Timer Ch.*x* Interrupt Flag (T16E_IFLG*x*) Register (D1/0x530c/0x536c)

To use this interrupt, set CBIE (D1/T16E_IMSK*x* register) to 1. If CBIE is set to 0 (default), CBIF is not set to 1, and the interrupt request for this factor is not sent to the ITC.

∗ **CBIE**: Compare B Interrupt Enable Bit in the PWM Timer Ch.*x* Interrupt Mask (T16E_IMSK*x*) Register (D1/0x530a/0x536a)

If CAIF is set to 1, the T16E module outputs an interrupt request to the ITC. An interrupt is generated if the ITC and S1C17 core interrupt conditions are satisfied.
CAIF should be read and checked within the PWM & capture timer interrupt handler routine to determine whether the PWM & capture timer interrupt is attributable to compare A matching.

Note: • To prevent interrupt recurrences, the T16E module interrupt flags CAIF and CBIF must be reset within the interrupt handler routine following a PWM & capture timer interrupt.

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

Interrupt vectors

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

Table 13.7.1: PWM & capture timer interrupt vectors

Timer channel	Vector number	Vector address
T16E Ch.0	11 (0x0b)	TTBR + 0x2c
T16E Ch.1	21 (0x15)	TTBR + 0x54

Other interrupt settings

The ITC allows the priority of PWM & capture timer interrupts to be set between level 0 (the default value) and level 7 for each channel. To generate actual interrupts, the PSR (S1C17 core internal processor status register) IE (interrupt enable) bit must be set to 1.

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

13.8 Control Register Details

Table 13.8.1: PWM & capture timer register list

Address	Register name		Function
0x5300	T16E_CA0	PWM Timer Ch.0 Compare Data A Register	Compare data A setting
0x5302	T16E_CB0	PWM Timer Ch.0 Compare Data B Register	Compare data B setting
0x5304	T16E_TC0	PWM Timer Ch.0 Counter Data Register	Counter data
0x5306	T16E_CTL0	PWM Timer Ch.0 Control Register	Timer mode setting and timer RUN/STOP
0x5308	T16E_CLK0	PWM Timer Ch.0 Input Clock Select Register	Prescaler output clock selection
0x530a	T16E_IMSK0	PWM Timer Ch.0 Interrupt Mask Register	Interrupt mask setting
0x530c	T16E_IFLG0	PWM Timer Ch.0 Interrupt Flag Register	Interrupt occurrence status display/resetting
0x5360	T16E_CA1	PWM Timer Ch.1 Compare Data A Register	Compare data A setting
0x5362	T16E_CB1	PWM Timer Ch.1 Compare Data B Register	Compare data B setting
0x5364	T16E_TC1	PWM Timer Ch.1 Counter Data Register	Counter data
0x5366	T16E_CTL1	PWM Timer Ch.1 Control Register	Timer mode setting and timer RUN/STOP
0x5368	T16E_CLK1	PWM Timer Ch.1 Input Clock Select Register	Prescaler output clock selection
0x536a	T16E_IMSK1	PWM Timer Ch.1 Interrupt Mask Register	Interrupt mask setting
0x536c	T16E_IFLG1	PWM Timer Ch.1 Interrupt Flag Register	Interrupt occurrence status display/resetting

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

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

0x5300/0x5360: PWM Timer Ch.*x* Compare Data A Register (T16E_CA*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
PWM Timer Compare Data A Register (T16E_CA)	**0x5300 0x5360** (16 bits)	D15–0	**T16ECA[15:0]**	Compare data A T16ECA15 = MSB T16ECA0 = LSB	0x0 to 0xffff	0x0	R/W	

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

0x5300: PWM Timer Ch.0 Compare Data A Register (T16E_CA0)
0x5360: PWM Timer Ch.1 Compare Data A Register (T16E_CA1)

D[15:0] T16ECA[15:0]: Compare Data A

Sets the PWM & capture timer compare data A. (Default: 0x0)

When CBUFEN (D5/T16E_CTL*x* register) is set to 0, this register can be used to directly read from or directly program to the compare data A register.

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

The data set is compared against the counter data, and a compare A interrupt factor is generated if the contents match. The timer output waveform changes at the same time (rising when INVOUT (D4/T16E_CTL*x* register) = 0 and trailing when INVOUT = 1). These processes do not affect the counter data or the count process.

0x5302/0x5362: PWM Timer Ch.*x* Compare Data B Register (T16E_CB*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
PWM Timer Compare Data B Register (T16E_CB)	**0x5302 0x5362** (16 bits)	D15–0	**T16ECB[15:0]**	Compare data B T16ECB15 = MSB T16ECB0 = LSB	0x0 to 0xffff	0x0	R/W	

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

0x5302: PWM Timer Ch.0 Compare Data A Register (T16E_CB0)
0x5362: PWM Timer Ch.1 Compare Data A Register (T16E_CB1)

D[15:0] T16ECB[15:0]: Compare Data B

Sets the PWM & capture timer compare data B. (Default: 0x0)

When CBUFEN (D5/T16E_CTL*x* register) is set to 0, this register can be used to directly read from or directly program to the compare data B register.

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

The data set is compared against the counter data, and a compare B interrupt factor is generated if the contents match. The timer output waveform changes at the same time (rising when INVOUT (D4/T16E_CTL*x* register) = 0 and trailing when INVOUT = 1). The counter is reset to 0.

0x5304/0x5364: PWM Timer Ch.*x* Counter Data Register (T16E_TC*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
PWM Timer Ch.*x* Counter Data Register (T16E_TC*x*)	**0x5304 0x5364** (16 bits)	D15–0	**T16ETC[15:0]**	Counter data T16ETC15 = MSB T16ETC0 = LSB	0x0 to 0xffff	0x0	R/W	

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

0x5304: PWM Timer Ch.0 Counter Data Register (T16E_TC0)
0x5364: PWM Timer Ch.1 Counter Data Register (T16E_TC1)

D[15:0] T16ETC[15:0]: Counter Data

Counter data can be read out. (Default: 0x0)
The counter value can also be set by writing data to this register.

0x5306/0x5366: PWM Timer Ch.*x* Control Register (T16E_CTL*x*)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
PWM Timer Ch.*x* Control Register (T16E_CTL*x*)	0x5306 0x5366 (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	

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

0x5306: PWM Timer Ch.0 Control Register (T16E_CTL0)
0x5366: PWM Timer Ch.1 Control Register (T16E_CTL1)

D[15:9] Reserved

D8 INITOL: Initial Output Level Bit

Sets the timer output initial 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 (D2). Note that this level will be inverted when INVOUT (D4) is 1.

D7 Reserved

D6 SELFM: Fine Mode Select Bit

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

When SELFM is set to 1, the clock output is set to Fine mode, and the output clock duty becomes adjustable in input clock half-cycle steps.
When SELFM is set to 0, normal clock output is used.

D5 CBUFEN: Comparison Buffer Enable Bit

Permits and prevents writing to the compare data buffer.
1 (R/W): Permitted
0 (R/W): Prohibited (default)

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

D4 INVOUT: Inverse Output Control Bit

Selects the timer output signal polarity.
1 (R/W): Inverted (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 a TOUTx output active Low signal (Off level = High). When INVOUT is 0, an active High signal (Off level = Low) is generated.
Writing 1 to this bit also inverts the initial output level set by INITOL (D8). The signal level above is inverted for TOUTNx output.

D3 CLKSEL: Input Clock Select Bit

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

Writing 0 to CLKSEL selects the internal clock (prescaler output) as the timer input clock. Writing 1 selects an external clock (a clock entered from EXCL3 (P00) pin for Ch.0 and a clock from EXCL4 (P01) pin for Ch.1), and functions as an event counter.

The input/output port used for external clock/pulse input should be set to input mode (the default value). No pin function needs to be selected. While the input/output port functions as a general input, the input signal is also sent to the PWM & capture timer.

D2 OUTEN: Clock Output Enable Bit

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

Writing 1 to OUTEN outputs the TOUTx/TOUTNx signal from the corresponding output pin.
Ch.0: TOUT3 output → TOUT3 (P26) pin, TOUTN3 output → TOUTN3 (P27) pin
Ch.1: TOUT4 output → TOUT4 (P04) pin, TOUTN4 output → TOUTN4 (P05) pin

Writing 0 to OUTEN stops the output, and switches to the Off level corresponding to the settings for INVOUT (D4) and INITOL (D8). The above pins must be set to TOUTx/TOUTNx output using the port function selection register before outputting the TOUTx/TOUTNx signals.

D1 T16ERST: Timer Reset Bit

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

Writing 1 to T16ERST resets the PWM & capture timer counter.

D0 T16ERUN: Timer Run/Stop Control Bit

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

The PWM & capture timer starts the count when T16ERUN is written as 1 and stops when written as 0. The counter data is retained when stopped until the subsequent reset or run. Counting can be resumed when switched from Stop to Run from the data retained.

0x5308/0x5368: PWM Timer Ch.*x* Input Clock Select Register (T16E_CLK*x*)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
PWM Timer CH.*x* Input Clock Select Register (T16E_CLK*x*)	**0x5308** (16 bits) / **0x5368**	D15–4	**–**	reserved	–		–	–	0 when being read.
		D3–0	**T16EDF[3:0]**	Timer input clock select (Prescaler output clock)	T16EDF[3:0]	Clock	0x0	R/W	
					0xf	reserved			
					0xe	PCLK-1/16384			
					0xd	PCLK-1/8192			
					0xc	PCLK-1/4096			
					0xb	PCLK-1/2048			
					0xa	PCLK-1/1024			
					0x9	PCLK-1/512			
					0x8	PCLK-1/256			
					0x7	PCLK-1/128			
					0x6	PCLK-1/64			
					0x5	PCLK-1/32			
					0x4	PCLK-1/16			
					0x3	PCLK-1/8			
					0x2	PCLK-1/4			
					0x1	PCLK-1/2			
					0x0	PCLK-1/1			

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

0x5308: PWM Timer Ch.0 Input Clock Select Register (T16E_CLK0)
0x5368: PWM Timer Ch.1 Input Clock Select Register (T16E_CLK1)

D[15:4] Reserved

D[3:0] T16EDF[3:0]: Timer Input Clock Select Bits

Select the PWM & capture timer count clock from the 15 different prescaler output clocks.

Table 13.8.2: Count clock selection

T16EDF[3:0]	Prescaler output clock	T16EDF[3:0]	Prescaler output clock
0xf	Reserved	0x7	PCLK-1/128
0xe	PCLK-1/16384	0x6	PCLK-1/64
0xd	PCLK-1/8192	0x5	PCLK-1/32
0xc	PCLK-1/4096	0x4	PCLK-1/16
0xb	PCLK-1/2048	0x3	PCLK-1/8
0xa	PCLK-1/1024	0x2	PCLK-1/4
0x9	PCLK-1/512	0x1	PCLK-1/2
0x8	PCLK-1/256	0x0	PCLK-1/1

(Default: 0x0)

Note: Make sure the PWM & capture timer count is halted before changing count clock settings.

0x530a/0x536a: PWM Timer Ch.*x* Interrupt Mask Registers (T16E_IMSK*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
PWM Timer Ch.*x* Interrupt Mask Register (T16E_IMSK*x*)	0x530a 0x536a (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	

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

0x530a: PWM Timer Ch.0 Interrupt Mask Register (T16E_IMSK0)
0x536a: PWM Timer Ch.1 Interrupt Mask Register (T16E_IMSK1)

D[15:2] Reserved

D1 CBIE: Compare B Interrupt Enable Bit

Permits or prohibits compare B match interrupts.
1 (R/W): Interrupt permitted
0 (R/W): Interrupt prohibited (default)

Setting CBIE to 1 permits compare B interrupt requests to the ITC. Setting it to 0 prohibits interrupts.

D0 CAIE: Compare A Interrupt Enable Bit

Permits or prohibits compare A match interrupts.
1 (R/W): Interrupt permitted
0 (R/W): Interrupt prohibited (default)

Setting CAIE to 1 permits compare A interrupt requests to the ITC. Setting it to 0 prohibits interrupts.

0x530c/0x536c: PWM Timer Ch.*x* Interrupt Flag Registers (T16E_IFLG*x*)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
PWM Timer Ch.*x* Interrupt Flag Register (T16E_IFLG*x*)	0x530c 0x536c (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	

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

0x530c: PWM Timer Ch.0 Interrupt Flag Register (T16E_IFLG0)
0x536c: PWM Timer Ch.1 Interrupt Flag Register (T16E_IFLG1)

D[15:2] Reserved

D1 CBIF: Compare B Interrupt Flag

Interrupt flag indicating the compare B interrupt factor occurrence status.

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

CBIF is the interrupt flag corresponding to compare B interrupts. Setting CBIE (D1/T16E_IMSK*x*) to 1 sets this to 1 when the counter matches the compare data B register setting during counting. A PWM & capture timer interrupt request signal is output to the ITC at the same time. This interrupt request signal generates an interrupt if the ITC and S1C17 core interrupt conditions are satisfied.
CBIF is reset by writing 1.

D0 CAIF: Compare A Interrupt Flag

Interrupt flag indicating the compare A interrupt factor occurrence status.

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

CAIF is the interrupt flag corresponding to compare A interrupts. Setting CAIE (D0/T16E_IMSK*x*) to 1 sets this to 1 when the counter matches the compare data A register setting during counting. A PWM & capture timer interrupt request signal is output to the ITC at the same time. This interrupt request signal generates an interrupt if the ITC and S1C17 core interrupt conditions are satisfied.
CAIF is reset by writing 1.

Note:
- To prevent interrupt recurrences, T16E module interrupt flags CAIF and CBIF must be reset within the interrupt handler routine following a PWM & capture timer interrupt.
- To prevent generating unnecessary interrupts, reset the corresponding CAIF or CBIF before permitting compare A or compare B interrupts from CAIE (D0/T16E_IMSK*x*) or CBIE (D1/T16E_IMSK*x*).

13.9 Precautions

- The prescaler must run before operating the PWM & capture timer.
- Make sure the PWM & capture timer count is halted before changing count clock settings.
- Compare data should be set with A ≥ 0 and B ≥ 1 when using the timer output. The minimum settings are A = 0 and B = 1, and the timer output cycle is half the input clock.
- Setting compare data with A > B (A > B x 2 for Fine mode) generates a compare B match signal only. It does not generate a compare A match signal. In this case, the timer output is fixed at Low (High when INVOUT = 1).
- To prevent interrupt recurrences, the T16E module interrupt flags CAIF (D0/T16E_INTx register) and CBIF (D1/T16E_INTx register) must be reset within the interrupt handler routine following a PWM & capture timer interrupt.
- To prevent generating unnecessary interrupts, reset the corresponding CAIF (D0/T16E_IFLG*x* register) or CBIF (D1/T16E_IFLG*x* register) before permitting compare A or compare B interrupts from CAIE (D0/T16E_IMSK*x* register) or CBIE (D1/T16E_IMSK*x* register).

14 8-bit OSC1 Timer (T8OSC1)

14.1 8-bit OSC1 Timer Overview

The S1C17702 incorporates a single-channel 8-bit OSC1 timer that uses the OSC1 clock as its oscillation source. Figure 14.1.1 illustrates the 8-bit OSC1 timer configuration.

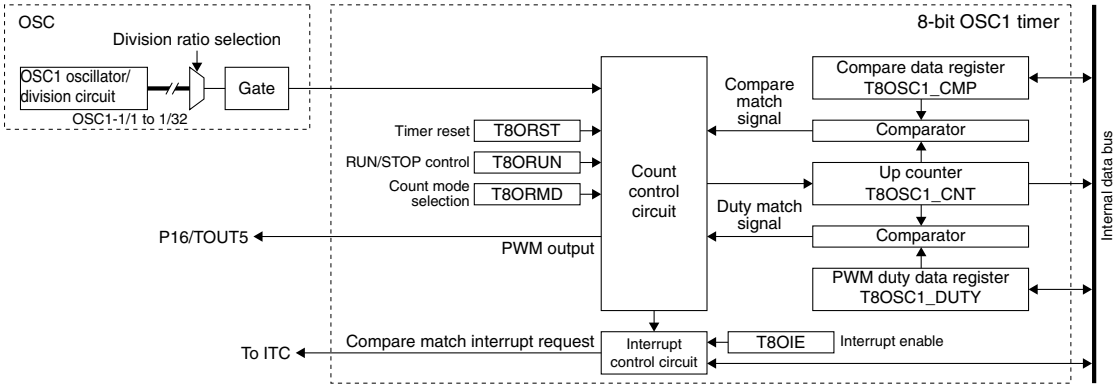


Figure 14.1.1: 8-bit OSC1 timer configuration

The 8-bit OSC1 timer includes an 8-bit up-counter (T8OOSC1_CNT register), an 8-bit compare data register (T8OSC1_CMP register), and an 8-bit PWM duty data register (T8OSC1_DUTY register).

The 8-bit counter can be reset to 0 by software and counts up using the OSC1 division clock (OSC1-1/1 to OSC1-1/32). The count value can be read by software.

The compare data and PWM duty registers store the data used for comparisons against up-counter contents.
If the counter values match the contents of each data register, the comparator outputs a signal to control the interrupts and the PWM output signal. The compare data register can be used to set the interrupt generating and PWM output clock frequencies. The PWM duty data register can be used to set the PWM output clock duty ratio.

14.2 8-bit OSC1 Timer Count Mode

The 8-bit OSC1 timer features two count modes: Repeat mode and One-shot mode. These modes are selected using the T8ORMD bit (D1/T8OSC1_CT register).

∗ **T8ORMD**: Count Mode Select Bit in the 8-bit OSC1 Timer Control (T8OSC1_CTL) Register (D1/0x50c0)

Repeat mode (T8ORMD = 0, default)

Setting T8ORMD to 0 sets the 8-bit OSC1 timer to Repeat mode.

In this mode, once the count starts, the 8-bit OSC1 timer continues running until stopped by the application program. If the counter matches the compare data, the timer resets the counter and continues counting. The interrupt signal is output at the same time. The 8-bit OSC1 timer should be set to this mode to generate periodic interrupts at desired intervals or to perform PWM output.

One-shot mode (T8ORMD = 1)

Setting T8ORMD to 1 sets the 8-bit OSC1 timer to One-shot mode.

In this mode, the 8-bit OSC1 timer stops automatically as soon as the counter matches the compare data.

This means only one interrupt can be generated after the timer starts. Note that the timer resets the counter, then stops after a complete match has occurred. The 8-bit OSC1 timer should be set to this mode to set a specific wait time.

Note:
- **Make sure the 8-bit OSC1 timer count is halted before changing count mode settings.**
- **The counter will not be stopped and one-shot mode will not operate if a compare match arises while the CPU is halted, even when counting is in progress with one-shot mode set.**

14.3 Count Clock

The 8-bit OSC1 timer uses the OSC1 division clock output by the OSC module as the count clock. The OSC module generates 6 different clocks by dividing the OSC1 clock into 1/1 to 1/32 divisions. One of these is selected by T8O1CK[2:0] (D[3:1]/OSC_T8OSC1 register).

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

Table 14.3.1: Count clock selection

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

(Default: 0x0)

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

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

Note: Make sure the 8-bit OSC1 timer count is halted before changing count clock settings.

For detailed information on clock control, refer to "7 Oscillator Circuit (OSC)."

14.4 Resetting 8-bit OSC1 Timer

The 8-bit OSC1 Timer can be reset to 0 by writing 1 to the T8ORS bit (D4/T8OSC1_CTL register).

* **T8ORST**: Timer Reset Bit in the 8-bit OSC1 Timer Control (T8OSC1_CTL) Register (D4/0x50c0)

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

14.5 Compare Data Settings

Compare data is written to T8OCMP[7:0] (D[7:0]/T8OSC1_CMP register).

* **T8OCMP[7:0]**: Compare Data Bits in the 8-bit OSC1 Timer Compare Data (T8OSC1_CMP) Register (D[7:0]/0x50c2)

After initial resetting, the compare data register is set to 0x0.

The timer compares the count data against the compare data register and generates a compare match signal as well as resets the counter if the values are equal. This compare match signal can generate an interrupt.
The compare match cycle can be calculated as follows:

$$\text{Compare match interval} = \frac{\text{CMP} + 1}{\text{clk\_in}} \text{ [s]}$$

$$\text{Compare match cycle} = \frac{\text{clk\_in}}{\text{CMP} + 1} \text{ [Hz]}$$

CMP: Compare data (T8OSC1_CMP register value)
clk_in: 8-bit OSC1 timer count clock frequency

When the 8-bit OSC1 timer is used to generate a PWM signal, the compare data determines the frequency of the output signal. (For a discussion of PWM output, refer to Section 14.8.)

14.6 8-bit OSC1 Timer RUN/STOP Control

Set the following items before starting the 8-bit OSC1 timer.
(1) Set the count mode (One-shot or Repeat). See Section 14.2.
(2) Select the operation clock. See Section 14.3.
(3) If using interrupts, set the interrupt level and permit interrupts for the 8-bit OSC1 timer. See Section 14.7.
(4) Reset the timer. See Section 14.4.
(5) Set the compare data. See Section 14.5.
(6) To output PWM signals, set the PWM duty data. See Section 14.8.

The 8-bit OSC1 timer includes T8ORUN (D0/T8OSC1_CTL register) to control Run/Stop.

∗ **T8ORUN**: Timer Run/Stop Control Bit in the 8-bit OSC1 Timer Control (T8OSC1_CTL) Register (D0/0x50c0)

The timer starts counting when T8ORUN is written as 1. Writing 0 to T8ORUN prevents clock input and stops the count.
This control does not affect the counter data. The counter data is retained even when the count is halted, allowing resumption of the count from that data.
If T8ORUN and T8ORST are written as 1 simultaneously, the timer starts counting after the reset.

If the counter matches the compare data register setting during counting, a compare match signal is output and a compare interrupt factor generated.
Likewise, if the counter matches the compare data B register setting, a compare B match signal is output and a compare B interrupt factor generated. The counter is reset to 0 at the same time.
If interrupts are permitted, an interrupt request is sent to the interrupt controller (ITC).
If One-shot mode is set, the timer stops the count.
If Repeat mode is set, the timer continues to count from 0.

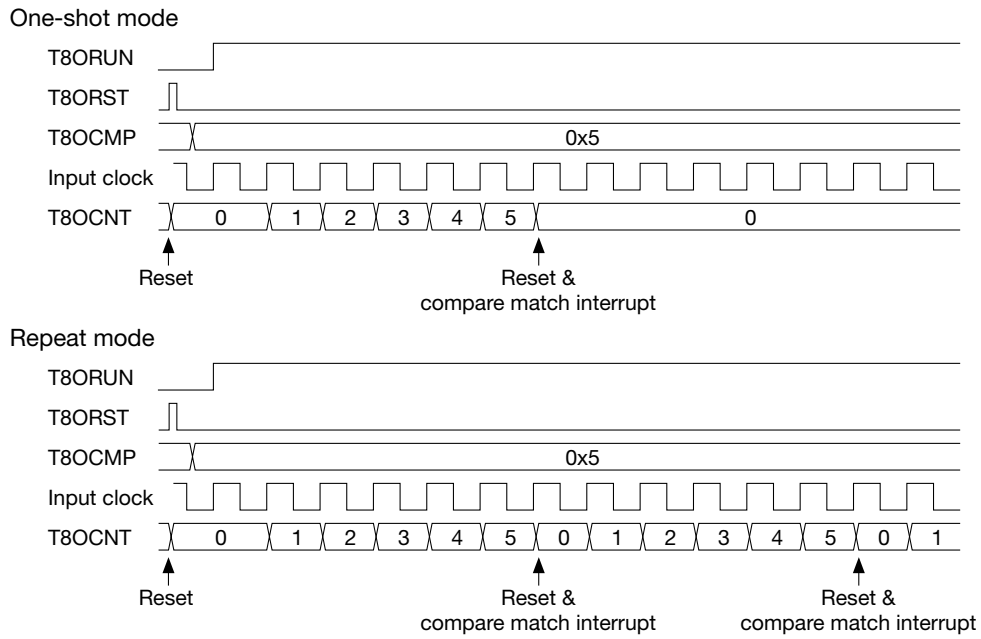


Figure 14.6.1: Basic counter operation timing

14.7 8-bit OSC1 Timer Interrupts

The T8OSC1 module outputs an interrupt request to the interrupt controller (ITC) by compare match.

Compare match interrupt

This interrupt request is generated when the counter matches the compare data register setting during counting. It sets the interrupt flag T8OIF (D0/T8OSC1_IFLG register) within the T8OSC1 module to 1.

∗ **T8OIF**: 8-bit OSC1 Timer Interrupt Flag in the 8-bit OSC1 Timer Interrupt Flag (T8OSC1_IFLG) Register (D0/0x50c4)

To use this interrupt, set T8OIE (D0/T8OSC1_IMSK register) to 1. If T8OIE is set to 0 (default), T8OIE is not set to 1, and the interrupt request for this factor is not sent to the ITC.

∗ **T8OIE**: 8-bit OSC1 Timer Interrupt Enable Bit in the 8-bit OSC1 Timer Interrupt Mask (T8OSC1_IMSK) Register (D0/0x50c3)

If T8OIF is set to 1, the T8OSC1 module outputs an interrupt request to the ITC. This interrupt request signal generates an interrupt if the ITC and S1C17 core interrupt conditions are satisfied.

Note:
- **To prevent interrupt recurrences, the T8OSC1 module interrupt flag T8OIF must be reset within the interrupt handler routine following an 8-bit OSC1 timer interrupt.**
- **To prevent generating unnecessary interrupts, reset the corresponding T8OIF before permitting compare 8-bit OSC1 interrupts from T8OIE.**

Interrupt vectors

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

Vector number: 8 (0x08)
Vector address: TTBR + 0x20

Other interrupt settings

The ITC allows the priority of 8-bit OSC1 timer interrupts to be set between level 0 (the default value) and level 7. To generate actual interrupts, the PSR (S1C17 core internal processor status register) IE (interrupt enable) bit must be set to 1.
For more information on interrupt processing, see “6 Interrupt Controller (ITC).”

14.8 PWM output

The 8-bit OSC1 timer can generate a PWM signal in accordance with the compare data and PWM duty data settings and output it from the TOUT5 (P16) pin.

Output pin setting

The PWM output pin (TOUT5) also acts as a pin (P16) for a general-purpose input/output port. In the default state, this pin is set as a general-purpose input/output port pin. To use it as a PWM output pin, change the function by setting the value 1 in the P16MUX (D6/P1_PMUX register).

∗ **P16MUX**: P16 Port Function Select Bit in the P1 Port Function Select (P1_PMUX) Register (D6/0x52a1)

PWM waveform control

The PWM waveform frequency can be set by the compare data register (0x50c2) (see Section 14.5). The duty ratio can be adjusted by the PWM duty data register (0x50c5).
The timer outputs a Low level signal until the counter value matches the value of the PWM duty data register. When the counter value exceeds the value of the PWM duty data, the output pin changes to High. Once the counter counts up to the compare data register value, the counter is reset and the output pin returns to Low.
Figure 14.8.1 shows the output waveform.

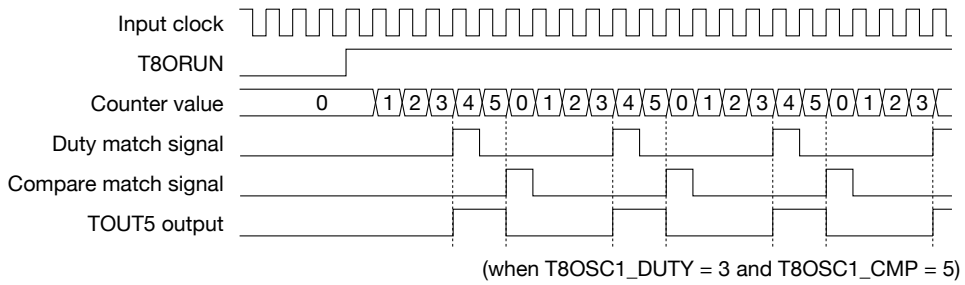


Figure 14.8.1 PWM output waveform

Precautions

(1) When using the timer output, set the following: PWM duty data ≥ 0, compare data ≥ 1. The minimum setting value is 0 for PWM duty data and 1 for compare data. The timer output cycle is 1/2 of the input clock.
(2) When the PWM duty data is set greater than the compare data, only the compare match signal will be generated. No duty match signal will be generated. In that case, the TOUT5 output is fixed to Low.

14.9 Control Register Details

Table 14.9.1: 8-bit OSC1 timer register list

Address	Register name		Function
0x50c0	T8OSC1_CTL	8-bit OSC1 Timer Control Register	Timer mode setting and timer RUN/STOP
0x50c1	T8OSC1_CNT	8-bit OSC1 Timer Counter Data Register	Counter data
0x50c2	T8OSC1_CMP	8-bit OSC1 Timer Compare Data Register	Compare data setting
0x50c3	T8OSC1_IMSK	8-bit OSC1 Timer Interrupt Mask Register	Interrupt mask setting
0x50c4	T8OSC1_IFLG	8-bit OSC1 Timer Interrupt Flag Register	Interrupt occurrence status display/resetting
0x50c5	T8OSC1_DUTY	8-bit OSC1 Timer PWM Duty Data Register	PWM output data setting

The 8-bit OSC1 timer registers are described in detail below. These are 8-bit registers.

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

0x50c0: 8-bit OSC1 Timer Control Register (T8OSC1_CTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
8-bit OSC1 Timer Control Register (T8OSC1_CTL)	0x50c0 (8 bits)	D7–5	–	reserved	–	–	–	0 when being read.
		D4	T8ORST	Timer reset	1 Reset 0 Ignored	0	W	
		D3–2	–	reserved	–	–	–	
		D1	T8ORMD	Count mode select	1 One shot 0 Repeat	0	R/W	
		D0	T8ORUN	Timer run/stop control	1 Run 0 Stop	0	R/W	

D[7:5] Reserved

D4 T8ORST: Timer Reset Bit

Resets the 8-bit OSC1 timer.

1 (W): Reset

0 (W): Disabled

0 (R): Normally 0 when read out (default)

Writing 1 to this bit resets the counter to 0.

D[3:2] Reserved

D1 T8ORMD: Count Mode Select Bit

Selects the 8-bit OSC1 timer count mode.

1 (R/W): One-shot mode

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

Setting T8ORMD to 0 sets the 8-bit OSC1 timer to Repeat mode. In this mode, once the count starts, the 8-bit timer continues to run until stopped by the application. If the counter matches the compare data register value, the timer resets the counter and continues counting. This means the timer periodically outputs a compare match signal. Set the 8-bit OSC1 timer to this mode to generate periodic interrupts at the desired interval or to perform PWM output.

Setting T8ORMD to 1 sets the 8-bit OSC1 timer to One-shot mode. In this mode, the 8-bit OSC1 timer stops automatically when the counter matches the compare data register value. This means an interrupt can be generated only once after the timer has been started. Note that the timer resets the counter and then stops after a compare match has occurred. Set the 8-bit OSC1 timer to this mode to create a specific wait time.

Note: Set the count mode only while the 8-bit OSC1 timer count is stopped.

D0 T8ORUN: Timer Run/Stop Control Bit

Controls the timer RUN/STOP.

1 (R/W): Run

0 (R/W): Stop (default)

The timer starts counting when T8ORUN is written as 1 and stops when written as 0. When the timer is stopped, the counter data is retained until reset or until the next RUN state.

0x50c1: 8-bit OSC1 Timer Counter Data Register (T8OSC1_CNT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
8-bit OSC1 Timer Counter Data Register (T8OSC1_CNT)	**0x50c1** (8 bits)	D7–0	**T8OCNT[7:0]**	Timer counter data T8OCNT7 = MSB T8OCNT0 = LSB	0x0 to 0xff	0x0	R	

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

Reads out the counter data. (Default: 0x0)

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

Note: The correct counter value may not be read out (reading is unstable) if the register is read while counting is underway.

Obtain the counter value by one of the following methods:

- Read the counter value while the counter is halted.
- Read the counter twice in succession. Treat the value as valid if the values read are identical.

0x50c2: 8-bit OSC1 Timer Compare Data Register (T8OSC1_CMP)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
8-bit OSC1 Timer Compare Data Register (T8OSC1_CMP)	**0x50c2** (8 bits)	D7–0	**T8OCMP[7:0]**	Compare data T8OCMP7 = MSB T8OCMP0 = LSB	0x0 to 0xff	0x0	R/W	

D[7:0] T8OCMP[7:0]: Compare Data

Sets the 8-bit OSC1 timer compare data. (Default: 0x0)

The data set is compared against the counter data, and a compare match interrupt factor is generated if the contents match. And the counter is reset to 0.

0x50c3: 8-bit OSC1 Timer Interrupt Mask Register (T8OSC1_IMSK)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
8-bit OSC1 Timer Interrupt Mask Register (T8OSC1_IMSK)	0x50c3 (8 bits)	D7–1	–	reserved	–				–	–	0 when being read.
		D0	T8OIE	8-bit OSC1 timer interrupt enable	1	Enable	0	Disable	0	R/W	

D[7:1] Reserved

D0 T8OIE: 8-bit OSC1 Timer Interrupt Enable Bit

Permits or prohibits compare match interrupts.

1 (R/W): Interrupt permitted

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

Setting T8OIE to 1 permits 8-bit OSC1 timer interrupt requests to the ITC. Setting it to 0 prohibits interrupts.

0x50c4: 8-bit OSC1 Timer Interrupt Flag Register (T8OSC1_IFLG)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
8-bit OSC1 Timer Interrupt Flag Register (T8OSC1_IFLG)	0x50c4 (8 bits)	D7–1	–	reserved	–				–	–	0 when being read.
		D0	T8OIF	8-bit OSC1 timer interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.

D[7:1] Reserved

D0 T8OIF: 8-bit OSC1 Timer Interrupt Flag

Interrupt flag indicating the compare match interrupt factor occurrence status.

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

T8OIF is the T8OSC1 module interrupt flag. Setting T8OIE (D0/T8OSC1_IMSK register) to 1 sets this to 1 when the counter matches the compare data register setting during counting. An 8-bit OSC1 timer interrupt request signal output simultaneously to the ITC generates an interrupt if the ITC and S1C17 core interrupt conditions are met.

T8OIF is reset by writing as 1.

Note:
- **To prevent interrupt recurrences, the T8OSC1 module interrupt flag T8OIF must be reset within the interrupt handler routine following an 8-bit OSC1 timer interrupt.**
- **To prevent generating unnecessary interrupts, reset T8OIF before permitting compare match interrupts using T8OIE (D0/T8OSC1_IMSK register).**

0x50c5: 8-bit OSC1 Timer PWM Duty Data Register (T8OSC1_DUTY)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
8-bit OSC1 Timer PWM Duty Data Register (T8OSC1_DUTY)	**0x50c5** (8 bits)	D7–0	**T8ODTY[7:0]**	PWM output duty data T8ODTY7 = MSB T8ODTY0 = LSB	0x0 to 0xff	0x0	R/W	

D[7:0] T8ODTY[7:0]: PWM Output Duty Data

Sets the data that determines the duty ratio of PWM waveform. (default: 0x0)

The set data is compared against the counter data. If the contents match, the timer output waveform rises. If the counter data matches the compare data, the timer output waveform falls. These processes do not affect the counter data or count process.

14.10 Precautions

- The 8-bit OSC1 timer clock must be output from the OSC module before the 8-bit OSC1 timer begins running.
- Set the count clock and count mode only while the 8-bit OSC1 timer count is stopped.
- To prevent interrupt recurrences, the T8OSC1 module interrupt flag T8OIF (D0/T8OSC1_IFLG register) must be reset within the interrupt handler routine following an 8-bit OSC1 timer interrupt.
- To prevent generating unnecessary interrupts, reset T8OIF (D0/T8OSC1_IFLG register) before permitting compare match interrupts using T8OIE (D0/T8OSC1_IMSK register).
- The correct counter value may not be read out (reading is unstable) if the counter data register is read while counting is underway.
  To obtain the counter value, read the counter data register while the counter is halted or read the counter data register twice in succession. Treat the value as valid if the values read are identical.
- When using the PWM output, set the following: PWM duty data ≥ 0, compare data ≥ 1. The minimum setting value is 0 for PWM duty data and 1 for compare data. The timer output cycle is 1/2 of the input clock.
- When the PWM duty data is set greater than the compare data, only the compare match signal is generated. No duty match signal is generated. In that case, the TOUT5 output is fixed to Low.
- The counter will not be stopped and one-shot mode will not operate if a compare match arises while the CPU is halted, even when counting is in progress with one-shot mode set.

15 Clock Timer (CT)

15.1 Clock Timer Overview

The S1C17702 incorporates a single-channel clock timer that uses the OSC1 clock as its oscillation source.
The clock timer consists of an 8-bit binary counter that uses the 256 Hz signal divided from the OSC1 clock as the input clock and allows data for each bit (128 Hz to 1 Hz) to be read out by software.
The clock timer can also generate interrupts using the 32 Hz, 8 Hz, 2 Hz, and 1 Hz signals.
This clock timer is normally used for various timing functions, such as clocks.
Figure 15.1.1 illustrates the clock timer configuration.

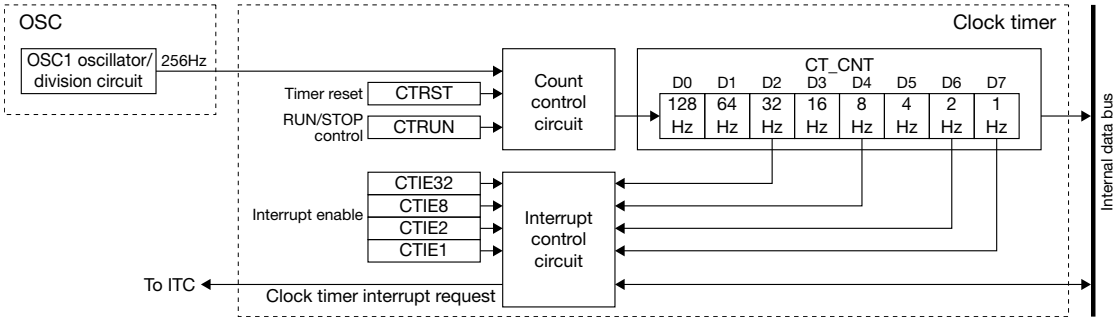


Figure 15.1.1: Clock timer configuration

15.2 Operation Clock

The clock timer uses the 256 Hz clock output by the OSC module as the operation clock.
The OSC module generates this operation clock by dividing the OSC1 clock into 1/128, resulting in a frequency of 256 Hz when the OSC1 clock frequency is 32.768 kHz. The frequency described in this section will vary accordingly for other OSC1 clock frequencies.
The OSC module does not include a 256 Hz clock output control bit. The 256 Hz clock is normally fed to the clock timer when the OSC1 oscillation is on.
For detailed information on OSC1 oscillator circuit control, refer to "7 Oscillator Circuit (OSC)."

15.3 Clock Timer Resetting

Reset the clock timer by writing 1 to the CTRST bit (D4/CT_CTL register). This clears the counter to 0.

* **CTRST**: Clock Timer Reset Bit in the Clock Timer Control (CT_CTL) Register (D4/0x5000)

Apart from this operation, the counter is also cleared by initial resetting.

15.4 Clock Timer RUN/STOP Control

Set the following items before starting the clock timer.
(1) If using interrupts, set the interrupt level and permit interrupts for the clock timer. See Section 15.5.
(2) Reset the timer. See Section 15.3.

The clock timer includes CTRUN (D0/CT_CTL register) to control Run/Stop.

∗ **CTRUN**: Clock Timer Run/Stop Control Bit in the Clock Timer Control (CT_CTL) Register (D0/0x5000)

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

Interrupt factors are generated during counting at the corresponding 32 Hz, 8 Hz, 2 Hz, and 1 Hz signal falling edges. If interrupts are permitted, interrupt requests are sent to the interrupt controller (ITC).

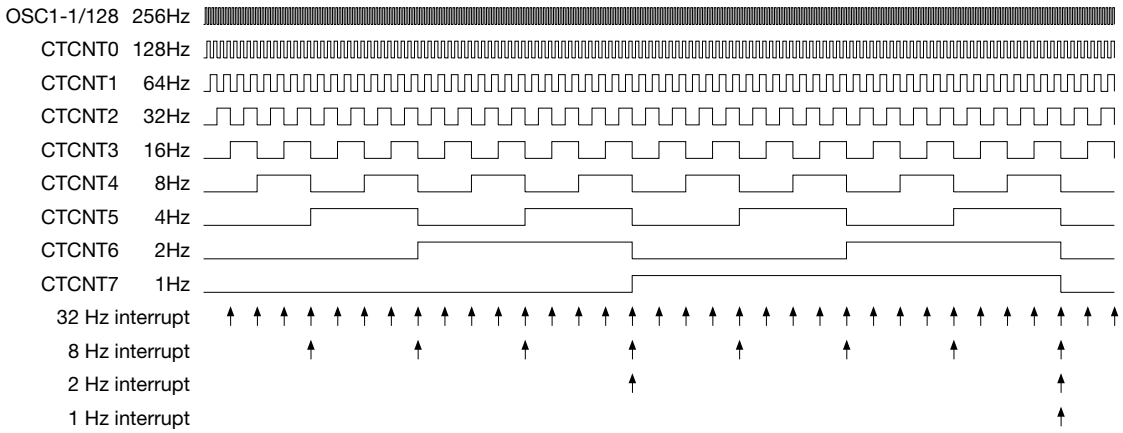


Figure 15.4.1: Clock timer timing chart

Note: The clock timer switches to Run/Stop mode when data is written to CTRUN synchronized with the 256 Hz signal falling edge. When 0 is written to CTRUN, the timer switches to Stop state after counting an additional "+1." 1 is retained for CTRUN reading until the timer actually stops.
Figure 15.4.2 shows the Run/Stop control timing chart.

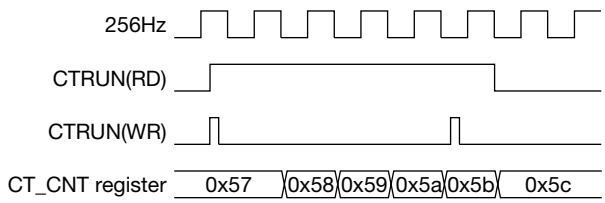


Figure 15.4.2: Run/Stop control timing chart

15.5 Clock Timer Interrupts

The CT module includes functions for generating the following four kinds of interrupts:
32 Hz, 8 Hz, 2 Hz, 1 Hz interrupts

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

32 Hz, 8 Hz, 2 Hz, 1 Hz interrupts

Generated at the 32 Hz, 8 Hz, 2 Hz, and 1 Hz signal falling edges, these interrupt requests set the following interrupt flags in the CT module to 1.

* **CTIF32**: 32 Hz Interrupt Flag in the Clock Timer Interrupt Flag (CT_IFLG) Register (D3/0x5003)
* **CTIF8**: 8 Hz Interrupt Flag in the Clock Timer Interrupt Flag (CT_IFLG) Register (D2/0x5003)
* **CTIF2**: 2 Hz Interrupt Flag in the Clock Timer Interrupt Flag (CT_IFLG) Register (D1/0x5003)
* **CTIF1**: 1 Hz Interrupt Flag in the Clock Timer Interrupt Flag (CT_IFLG) Register (D0/0x5003)

To use these interrupts, set the following interrupt enable bits to 1 for the corresponding interrupt flags. If the interrupt enable bits are set to 0 (default), the interrupt flag will not be set to 1, and the interrupt requests for this factor will not be sent to the ITC.

* **CTIE32**: 32 Hz Interrupt Enable Bit in the Clock Timer Interrupt Mask (CT_IMSK) Register (D3/0x5002)
* **CTIE8**: 8 Hz Interrupt Enable Bit in the Clock Timer Interrupt Mask (CT_IMSK) Register (D2/0x5002)
* **CTIE2**: 2 Hz Interrupt Enable Bit in the Clock Timer Interrupt Mask (CT_IMSK) Register (D1/0x5002)
* **CTIE1**: 1 Hz Interrupt Enable Bit in the Clock Timer Interrupt Mask (CT_IMSK) Register (D0/0x5002)

The CT module outputs an interrupt request to the ITC if the CTIF* is set to 1. This interrupt request signal sets the clock timer interrupt flag inside the ITC to 1 and generates an interrupt if the ITC and S1C17 core interrupt conditions are met.
Check the frequency of a clock timer interrupt by reading CTIF* as part of the clock timer interrupt handler routine.

Note: • To prevent interrupt recurrences, the CT module interrupt flag CTIF*must be reset within the interrupt handler routine following a clock timer interrupt.

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

Interrupt vectors

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

Vector number: 7 (0x07)
Vector address: TTBR + 0x1c

Other interrupt settings

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

15.6 Control Register Details

Table 15.6.1: Clock timer registers list

Address	Register name		Function
0x5000	CT_CTL	Clock Timer Control Register	Timer resetting and Run/Stop control
0x5001	CT_CNT	Clock Timer Counter Register	Counter data
0x5002	CT_IMSK	Clock Timer Interrupt Mask Register	Interrupt mask setting
0x5003	CT_IFLG	Clock Timer Interrupt Flag Register	Interrupt occurrence status display/resetting

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

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

0x5000: Clock Timer Control Register (CT_CTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Clock Timer Control Register (CT_CTL)	**0x5000** (8 bits)	D7–5	**–**	reserved	–	–	–	0 when being read.
		D4	**CTRST**	Clock timer reset	1 Reset 0 Ignored	0	W	
		D3–1	**–**	reserved	–	–	–	
		D0	**CTRUN**	Clock timer run/stop control	1 Run 0 Stop	0	R/W	

D[7:5] Reserved

D4 CTRST: Clock Timer Reset Bit

Resets the clock timer.

1 (W): Reset
0 (W): Disabled
0 (R): Normally 0 when read out (default)

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

D[3:1] Reserved

D0 CTRUN: Clock Timer Run/Stop Control Bit

Controls the clock timer Run/Stop.

1 (R/W): Run
0 (R/W): Stop (default)

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

0x5001: Clock Timer Counter Register (CT_CNT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Clock Timer Counter Register (CT_CNT)	**0x5001** (8 bits)	D7–0	**CTCNT[7:0]**	Clock timer counter value	0x0 to 0xff	0	R	

D[7:0] CTCNT[7:0]: Clock Timer Counter Value

Reads out the counter data. (Default: 0xff)
This register is read-only and cannot be written to.
The bits correspond to various frequencies, as follows:
D7: 1Hz
D6: 2Hz
D5: 4Hz
D4: 8Hz
D3: 16Hz
D2: 32Hz
D1: 64Hz
D0: 128Hz

Note: The correct counter value may not be read out (reading is unstable) if the register is read while counting is underway.
Obtain the counter value by one of the following methods:
- **Read the counter value while the counter is halted.**
- **Read the counter twice in succession. Treat the value as valid if the values read are identical.**

0x5002: Clock Timer Interrupt Mask Register (CT_IMSK)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Clock Timer Interrupt Mask Register (CT_IMSK)	**0x5002** (8 bits)	D7–4	**–**	reserved	–				–	–	0 when being read.
		D3	**CTIE32**	32 Hz interrupt enable	1	Enable	0	Disable	0	R/W	
		D2	**CTIE8**	8 Hz interrupt enable	1	Enable	0	Disable	0	R/W	
		D1	**CTIE2**	2 Hz interrupt enable	1	Enable	0	Disable	0	R/W	
		D0	**CTIE1**	1 Hz interrupt enable	1	Enable	0	Disable	0	R/W	

This register permits or prohibits interrupt requests individually for the clock timer 32 Hz, 8 Hz, 2 Hz, and 1 Hz signals. Setting the CTIE*bit to 1 permits clock timer interrupts for the corresponding frequency signal falling edge, while setting to 0 prohibits interrupts.
To enable interrupt generation, the ITC clock timer interrupt enable bits must also be set to permit interrupts.

D[7:4] Reserved

D3 CTIE32: 32 Hz Interrupt Enable Bit
Permits or prohibits 32 Hz signal interrupts.
1 (R/W): Interrupt permitted
0 (R/W): Interrupt prohibited (default)

D2 CTIE8: 8 Hz Interrupt Enable Bit
Permits or prohibits 8 Hz signal interrupts.
1 (R/W): Interrupt permitted
0 (R/W): Interrupt prohibited (default)

D1 CTIE2: 2 Hz Interrupt Enable Bit
Permits or prohibits 2 Hz signal interrupts.
1 (R/W): Interrupt permitted
0 (R/W): Interrupt prohibited (default)

D0 CTIE1: 1 Hz Interrupt Enable Bit
Permits or prohibits 1 Hz signal interrupts.
1 (R/W): Interrupt permitted
0 (R/W): Interrupt prohibited (default)

0x5003: Clock Timer Interrupt Flag Register (CT_IFLG)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Clock Timer Interrupt Flag Register (CT_IFLG)	0x5003 (8 bits)	D7–4	–	reserved	–				–	–	0 when being read.
		D3	CTIF32	32 Hz interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D2	CTIF8	8 Hz interrupt flag					0	R/W	
		D1	CTIF2	2 Hz interrupt flag					0	R/W	
		D0	CTIF1	1 Hz interrupt flag					0	R/W	

This register indicates the occurrence state of interrupt factors due to clock timer 32 Hz, 8 Hz, 2 Hz, and 1 Hz signals. If a clock timer interrupt occurs, identify the interrupt factor (frequency) by reading the interrupt flag in this register. CTIF* are CT module interrupt flags corresponding to the individual 32 Hz, 8 Hz, 2 Hz, and 1 Hz interrupts. It is set to 1 at the falling edge of each signal if CTIE* (CT_IMSK register) is set to 1. The clock timer interrupt request signal is output to the ITC at the same time. This interrupt request signal generates an interrupt if the ITC and S1C17 core interrupt conditions are met.
CTIF* is reset by writing as 1.

Note:
- **To prevent interrupt recurrences, the CT module interrupt flag CTIF*must be reset within the interrupt handler routine following a clock timer interrupt.**
- **To prevent generating unnecessary interrupts, CTIF* must be reset before permitting clock timer interrupts using CTIE.***

D[7:4] Reserved

D3 CTIF32: 32 Hz Interrupt Flag

Interrupt flag indicating the 32 Hz interrupt factor occurrence status.
1(R): Interrupt factor present
0(R): No interrupt factor (default)
1(W): Reset flag
0(W): Disabled

Setting CTIE32 (D3/CT_IMSK register) to 1 sets CTIF32 to 1 at the 32 Hz signal falling edge.

D2 CTIF8: 8 Hz Interrupt Flag

Interrupt flag indicating the 8 Hz interrupt factor occurrence status.
1(R): Interrupt factor present
0(R): No interrupt factor (default)
1(W): Reset flag
0(W): Disabled

Setting CTIE8 (D2/CT_IMSK register) to 1 sets CTIF8 to 1 at the 8 Hz signal falling edge.

D1 CTIF2: 2 Hz Interrupt Flag

Interrupt flag indicating the 2 Hz interrupt factor occurrence status.
1(R): Interrupt factor present
0(R): No interrupt factor (default)
1(W): Reset flag
0(W): Disabled

Setting CTIE2 (D1/CT_IMSK register) to 1 sets CTIF2 to 1 at the 2 Hz signal falling edge.

D0 CTIF1: 1 Hz Interrupt Flag

Interrupt flag indicating the 1 Hz interrupt factor occurrence status.
1(R): Interrupt factor present
0(R): No interrupt factor (default)
1(W): Reset flag
0(W): Disabled

Setting CTIE1 (D0/CT_IMSK register) to 1 sets CTIF1 to 1 at the 1 Hz signal falling edge.

15.7 Precautions

- The OSC1 oscillator circuit must be set to On before operating the clock timer.
- To prevent generating unnecessary interrupts, reset the CT_IFLG register interrupt flag before permitting clock timer interrupts by the CT_IMSK register.
- The clock timer switches to Run/Stop mode when data is written to CTRUN synchronized with the 256 Hz signal falling edge. When 0 is written to CTRUN (D0/CT_CTL register), the timer switches to Stop state after counting an additional "+1." 1 is retained for CTRUN reading until the timer actually stops.
  Figure 15.7.1 shows the Run/Stop control timing chart.

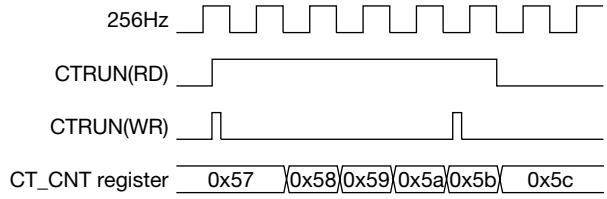


Figure 15.7.1: Run/Stop control timing chart

- Executing the slp instruction will destabilize a running clock timer (CTRUN = 1) during recovery from SLEEP state. When switching to SLEEP state, set the clock timer to STOP (CTRUN = 0) before executing the slp instruction.
- The correct counter value may not be read out (reading is unstable) if the counter register is read while counting is underway.
  Read the counter register while the counter is halted or read the counter register twice in succession. Treat the value as valid if the values read are identical.

16 Stopwatch Timer (SWT)

16.1 Stopwatch Timer Overview

The S1C17702 incorporates a 1/100-second and 1/10-second stopwatch timer. The stopwatch timer consists of a 4-bit 2-stage BCD counter (1/100 and 1/10 second) that uses the 256 Hz signal divided from the OSC1 clock as the input clock and allows count data to be read out by software.
The stopwatch timer can also generate interrupts using the 100 Hz (approximately 100 Hz), 10 Hz (approximately 10 Hz), and 1 Hz signals.
Figure 16.1.1 illustrates the stopwatch timer configuration.

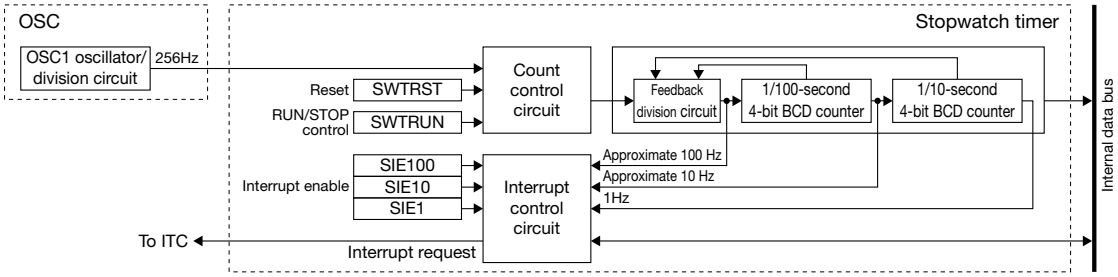


Figure 16.1.1: Stopwatch timer configuration

16.2 BCD Counters

The stopwatch counter consists of 1/100-second and 1/10-second 4-bit BCD counters.
The count value can be read from the SWT_BCNT register.

1/100-second counter

* **BCD100[3:0]**: 1/100 Sec. BCD Counter Value in the Stopwatch Timer BCD Counter (SWT_BCNT) Register (D[3:0]/0x5021)

1/10-second counter

* **BCD10[3:0]**: 1/10 Sec. BCD Counter Value in the Stopwatch Timer BCD Counter (SWT_BCNT) Register (D[7:4]/0x5021)

Count-up Pattern

A feedback division circuit is used to generate 100 Hz, 10 Hz, and 1 Hz signals from the 256 Hz clock. The counter count-up pattern varies as shown in Figure 16.2.1.

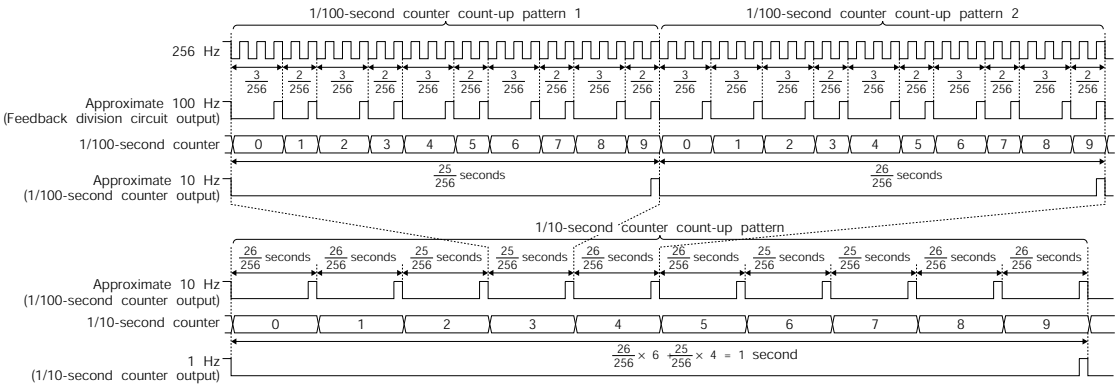


Figure 16.2.1: Stopwatch timer count-up patterns

The feedback division circuit generates an approximate 100 Hz signal at 2/256-second and 3/256-second intervals from the 256 Hz signal fed from the OSC module.

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

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

16.3 Operation Clock

The stopwatch timer uses the 256 Hz clock output by the OSC module as the operation clock.
The OSC module generates this operation clock by dividing the OSC1 clock into 1/128, resulting in a frequency of 256 Hz when the OSC1 clock frequency is 32.768 kHz. The frequency described in this section will vary accordingly for other OSC1 clock frequencies.
The OSC module does not include a 256 Hz clock output control bit. The 256 Hz clock is normally fed to the stopwatch timer when the OSC1 oscillation is on.
For detailed information on OSC1 oscillator circuit control, refer to "7 Oscillator Circuit (OSC)."

16.4 Stopwatch Timer Resetting

Reset the stopwatch timer by writing 1 to the SWTRST bit (D4/SWT_CTL register). This clears the counter to 0.

* **SWTRST**: Stopwatch Timer Reset Bit in the Stopwatch Timer Control (SWT_CTL) Register (D4/0x5020)

Apart from this operation, the counter is also cleared by initial resetting.

16.5 Stopwatch Timer RUN/STOP Control

Set the following items before starting the stopwatch timer.
(1) If using interrupts, set the interrupt level and permit interrupts for the stopwatch timer. See Section 16.6.
(2) Reset the timer. See Section 16.4.

The stopwatch timer includes SWTRUN (D0/SWT_CTL register) to control Run/Stop.

∗ **SWTRUN**: Stopwatch Timer Run/Stop Control Bit in the Stopwatch Timer Control (SWT_CTL) Register (D0/0x5020)

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

Interrupt factors are generated during counting at the corresponding 100 Hz (approximate 100 Hz), 10 Hz (approximate 10 Hz), and 1 Hz signal falling edges. If interrupts are permitted, interrupt requests are sent to the interrupt controller (ITC).

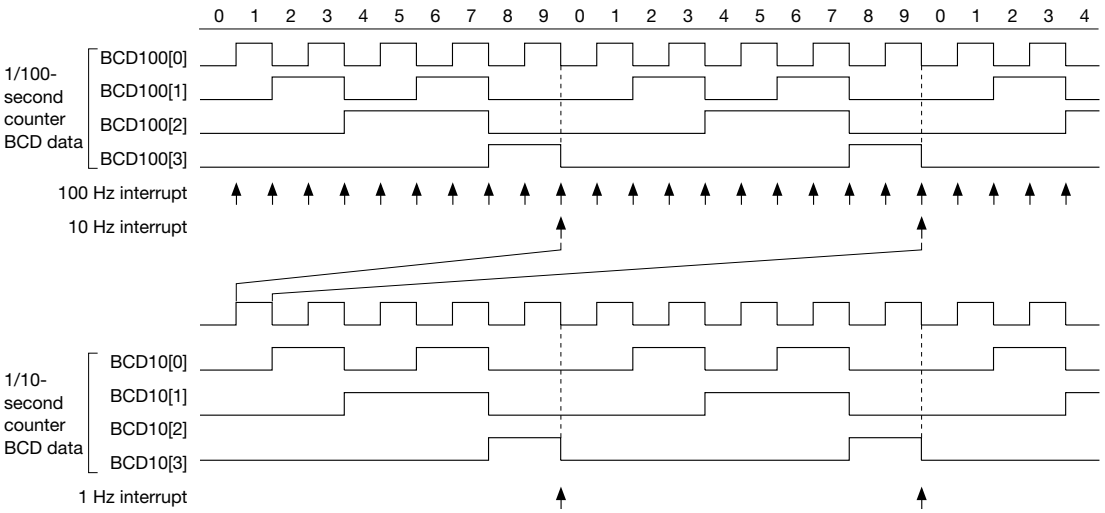


Figure 16.5.1: Stopwatch timer timing chart

Note: The stopwatch timer switches to Run/Stop mode when data is written to SWTRUN synchronized with the 256 Hz signal falling edge. When 0 is written to SWTRUN, the timer switches to Stop state after counting an additional "+1." 1 is retained for SWTRUN reading until the timer actually stops.
Figure 16.5.2 shows the Run/Stop control timing chart.

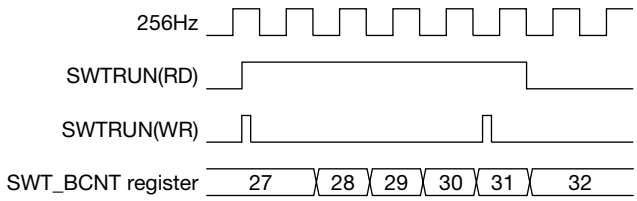


Figure 16.5.2: Run/Stop control timing chart

16.6 Stopwatch Timer Interrupts

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

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

100 Hz, 10 Hz, 1 Hz interrupts

Generated at the 100 Hz (approximate 100 Hz), 10 Hz (approximate 10 Hz), and 1 Hz signal falling edges, these interrupt requests set the following interrupt flags in the SWT module to 1.

* **SIF1**: 1 Hz Interrupt Flag in the Stopwatch Timer Interrupt Flag (SWT_IFLG) Register (D2/0x5023)
* **SIF10**: 10 Hz Interrupt Flag in the Stopwatch Timer Interrupt Flag (SWT_IFLG) Register (D1/0x5023)
* **SIF100**: 100 Hz Interrupt Flag in the Stopwatch Timer Interrupt Flag (SWT_IFLG) Register (D0/0x5023)

To use these interrupts, set the following interrupt enable bits to 1 for the corresponding interrupt flags. If the interrupt enable bits are set to 0 (default), the interrupt flag will not be set to 1, and the interrupt requests for this factor will not be sent to the ITC.

* **SIE1**: 1 Hz Interrupt Enable Bit in the Stopwatch Timer Interrupt Mask (SWT_IMSK) Register (D2/0x5022)
* **SIE10**: 10 Hz Interrupt Enable Bit in the Stopwatch Timer Interrupt Mask (SWT_IMSK) Register (D1/0x5022)
* **SIE100**: 100 Hz Interrupt Enable Bit in the Stopwatch Timer Interrupt Mask (SWT_IMSK) Register (D0/0x5022)

The SWT module outputs an interrupt request to the ITC if the SIF* is set to 1. This interrupt request signal generates an interrupt if the ITC and S1C17 core interrupt conditions are met.
Check the frequency of a stopwatch timer interrupt by reading SIF* as part of the stopwatch timer interrupt handler routine.

Note: • To prevent interrupt recurrences, the SWT module interrupt flag SIF*must be reset within the interrupt handler routine following a stopwatch timer interrupt.
• To prevent generating unnecessary interrupts, reset the corresponding SIF* before permitting stopwatch timer interrupt from SIE*.

Interrupt vectors

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

Vector number: 6 (0x06)
Vector address: TTBR + 0x18

Other interrupt settings

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

16.7 Control Register Details

Table 16.7.1 Stopwatch timer register list

Address	Register name		Function
0x5020	SWT_CTL	Stopwatch Timer Control Register	Timer resetting and Run/Stop control
0x5021	SWT_BCNT	Stopwatch Timer BCD Counter Register	BCD counter data
0x5022	SWT_IMSK	Stopwatch Timer Interrupt Mask Register	Interrupt mask setting
0x5023	SWT_IFLG	Stopwatch Timer Interrupt Flag Register	Interrupt occurrence status display/resetting

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

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

0x5020: Stopwatch Timer Control Register (SWT_CTL)

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

D[7:5] Reserved

D4 SWTRST: Stopwatch Timer Reset Bit

Resets the stopwatch timer.

1 (W): Reset
0 (W): Disabled
0 (R): Normally 0 when read out (default)

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

D[3:1] Reserved

D0 SWTRUN: Stopwatch Timer Run/Stop Control Bit

Controls the stopwatch timer Run/Stop.

1 (R/W): Run
0 (R/W): Stop (default)

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

0x5021: Stopwatch Timer BCD Counter Register (SWT_BCNT)

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

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

Read the 1/10-second counter BCD data. (Default: 0)
This register is read-only and cannot be written to.

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

Read the 1/100-second counter BCD data. (Default: 0)
This register is read-only and cannot be written to.

Note: The correct counter value may not be read out (reading is unstable) if the register is read while counting is underway.
Obtain the counter value by one of the following methods:
- Read the counter value while the counter is halted.
- Read the counter twice in succession. Treat the value as valid if the values read are identical.

0x5022: Stopwatch Timer Interrupt Mask Register (SWT_IMSK)

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

This register permits or prohibits interrupt requests individually for the stopwatch timer 100 Hz, 10 Hz, and 1 Hz signals. Setting the SIE*bit to 1 permits stopwatch timer interrupts for the corresponding frequency signal falling edge, while setting to 0 prohibits interrupts.
To enable interrupt generation, the ITC stopwatch timer interrupt enable bits must also be set to permit interrupts.

D[7:3] Reserved

D2 SIE1: 1 Hz Interrupt Enable Bit

Permits or prohibits 1 Hz signal interrupts.
1 (R/W): Interrupt permitted
0 (R/W): Interrupt prohibited (default)

D1 SIE10: 10 Hz Interrupt Enable Bit

Permits or prohibits 10 Hz signal interrupts.
1 (R/W): Interrupt permitted
0 (R/W): Interrupt prohibited (default)

D0 SIE100: 100 Hz Interrupt Enable Bit

Permits or prohibits 100 Hz signal interrupts.
1 (R/W): Interrupt permitted
0 (R/W): Interrupt prohibited (default)

0x5023: Stopwatch Timer Interrupt Flag Register (SWT_IFLG)

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

This register indicates the occurrence state of interrupt factors due to stopwatch timer 100 Hz, 10 Hz, and 1 Hz signals. If a stopwatch timer interrupt occurs, identify the interrupt factor (frequency) by reading the interrupt flag in this register.
SIF* are SWT module interrupt flags corresponding to the individual 100 Hz, 10 Hz, and 1 Hz interrupts. It is set to 1 at the falling edge of each signal if SIE* (SWT_IMSK register) is set to 1. The stopwatch timer interrupt request signal is output to the ITC at the same time. This interrupt request signal generates an interrupt if the ITC and S1C17 core interrupt conditions are met.
SIF* is reset by writing as 1.

Note:
- **To prevent interrupt recurrences, the SWT module interrupt flag SIF*must be reset within the interrupt handler routine following a stopwatch timer interrupt.**
- **To prevent generating unnecessary interrupts, SIF* must be reset before permitting clock timer interrupts using SIE.***

D[7:3] Reserved

D2 SIF1: 1 Hz Interrupt Flag

Interrupt flag indicating the 1 Hz interrupt factor occurrence status.
1(R): Interrupt factor present
0(R): No interrupt factor (default)
1(W): Reset flag
0(W): Disabled

Setting SIE1 (D2/SWT_IMSK register) to 1 sets SIF1 to 1 at the 1 Hz signal falling edge.

D1 SIF10: 10 Hz Interrupt Flag

Interrupt flag indicating the 10 Hz interrupt factor occurrence status.
1(R): Interrupt factor present
0(R): No interrupt factor (default)
1(W): Reset flag
0(W): Disabled

Setting SIE10 (D1/SWT_IMSK register) to 1 sets SIF10 to 1 at the 10 Hz signal falling edge.

D0 SIF100: 100 Hz Interrupt Flag

Interrupt flag indicating the 100 Hz interrupt factor occurrence status.
1(R): Interrupt factor present
0(R): No interrupt factor (default)
1(W): Reset flag
0(W): Disabled

Setting SIE100 (D0/SWT_IMSK register) to 1 sets SIF100 to 1 at the 100 Hz signal falling edge.

16.8 Precautions

- The OSC1 oscillator circuit must be set to On before operating the stopwatch timer.
- To prevent interrupt recurrences, the SWT_IFLG register interrupt flag must be reset within the interrupt handler routine following a stopwatch timer interrupt.
- To prevent generating unnecessary interrupts, reset the SWT_IFLG register interrupt flag before permitting stopwatch timer interrupts by the SWT_IMSK register.
- The stopwatch timer switches to Run/Stop mode when data is written to SWTRUN (D0/SWT_CTL resister) synchronized with the 256 Hz signal falling edge. When 0 is written to SWTRUN, the timer switches to Stop state after counting an additional "+1." 1 is retained for SWTRUN reading until the timer actually stops.
  Figure 16.8.1 shows the Run/Stop control timing chart.

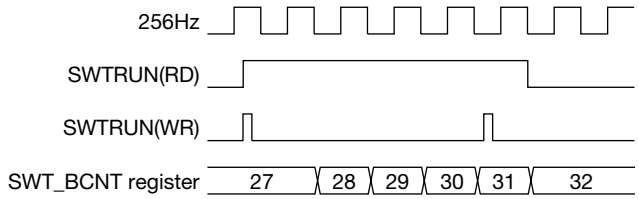


Figure 16.8.1: Run/Stop control timing chart

- Executing the slp instruction will destabilize a running stopwatch timer (SWTRUN = 1) during recovery from SLEEP state. When switching to SLEEP state, set the stopwatch timer to STOP (SWTRUN = 0) before executing the slp instruction.
- The correct counter value may not be read out (reading is unstable) if the counter register is read while counting is underway.
  To obtain the counter value, read the counter register while the counter is halted or read the counter register twice in succession. Treat the value as valid if the values read are identical.

17 Watchdog Timer (WDT)

17.1 Watchdog Timer Overview

The S1C17702 incorporates a watchdog timer that uses the OSC1 oscillator circuit as its oscillation source. The watchdog timer generates an NMI or reset (selectable via software) to the CPU if not reset within 131,072/fosc1 seconds (4 seconds when fosc1 = 32.768 kHz).
Reset the watchdog timer via software within this cycle to prevent NMI/resets, which in turn enables runaway detection for programs that do not pass through the handler routine.
Figure 17.1.1 illustrates the watchdog timer block diagram.

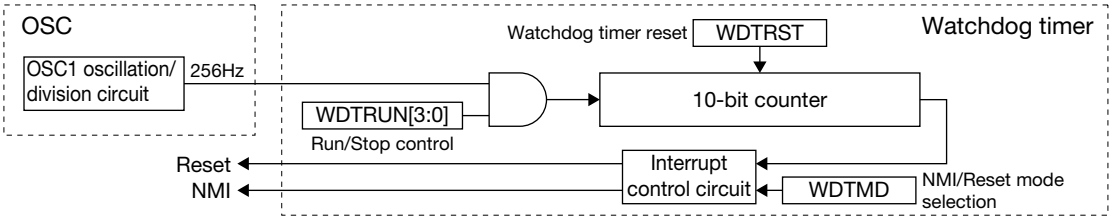


Figure 17.1.1: Watchdog timer block diagram

17.2 Operation Clock

The watchdog timer uses the 256 Hz clock output by the OSC module as the operation clock.
The OSC module generates this operation clock by dividing the OSC1 clock into 1/128, resulting in a frequency of 256 Hz when the OSC1 clock frequency is 32.768 kHz. The frequency described in this section will vary accordingly for other OSC1 clock frequencies.
The OSC module does not include a 256 Hz clock output control bit. The 256 Hz clock is normally fed to the watchdog timer when the OSC1 oscillation is on.
For detailed information on OSC1 oscillator circuit control, refer to "7 Oscillator Circuit (OSC)."

17.3 Watchdog Timer Control

17.3.1 NMI/Reset Mode Selection

WDTMD (D1/WDT_ST register) is used to select whether an NMI signal or a reset signal is output when the watchdog timer has not been reset within the NMI/Reset occurrence cycle.

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

To generate an NMI, set WDTMD to 0 (default). Set to 1 to generate a reset.

17.3.2 Watchdog Timer Run/Stop Control

The watchdog timer starts counting when a value other than 0b1010 is written to WDTRUN[3:0] (D[3:0]/WDT_CTL register) and stops when 0b1010 is written.

* **WDTRUN[3:0]**: Watchdog Timer Run/Stop Control Bits in the Watchdog Timer Control (WDT_CTL) Register (D[3:0]/0x5040)

Initial resetting sets WDTRUN[3:0] to 0b1010 and stops the watchdog timer.
Since an NMI or Reset may be generated immediately after running depending on the counter value, the watchdog timer should also be reset concurrently (before running the watchdog timer), as explained in the following section.

17.3.3 Watchdog Timer Resetting

To reset the watchdog timer, program 1 to WDTRST (D4/WDT_CTL register).

* **WDTRST**: Watchdog Timer Reset Bit in the Watchdog Timer Control (WDT_CTL) Register (D4/0x5040)

A location should be provided for periodically processing the routine for resetting the watchdog timer before an NMI or Reset is generated when using the watchdog timer. Process this routine within 131,072/f_{OSC1} second (4 seconds when f_{OSC1} = 32.768 kHz) cycle.
After resetting, the watchdog timer starts counting with a new NMI/Reset generation cycle.

If the watchdog timer is not reset within the NMI/Reset generation cycle for any reason, the CPU is switched to interrupt processing by NMI or resetting, an interrupt vector is read out, and an interrupt 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 the watchdog timer being reset, WDTST (D0/WDT_ST register) is set to 1.

* **WDTST**: NMI Status Bit in the Watchdog Timer Status (WDT_ST) Register (D0/0x5041)

This bit is provided to confirm that the watchdog timer was the source of the NMI.
The WDTST set to 1 is cleared to 0 by resetting the watchdog timer.

17.3.4 Operation in Standby Mode

HALT mode

The watchdog timer operates in HALT mode, as the clock is fed. HALT mode is therefore cleared by an NMI or Reset if it continues for more than the NMI/Reset cycle. To disable the watchdog timer while in HALT mode, stop the watchdog timer by writing 0b1010 to WDTRUN[3:0] before executing the halt instruction. Reset the watchdog timer before resuming operations after HALT mode is cleared.

SLEEP mode

The clock fed from the OSC module is stopped in SLEEP mode, which also stops the watchdog timer. To prevent generation of an unnecessary NMI or Reset after clearing SLEEP mode, reset the watchdog timer before executing the slp instruction. The watchdog should also be stopped as required using WDTRUN[3:0].

17.4 Control Register Details

Table 17.4.1 Watchdog timer register list

Address	Register name		Function
0x5040	WDT_CTL	Watchdog Timer Control Register	Timer reset and Run/Stop control
0x5041	WDT_ST	Watchdog Timer Status Register	Timer mode setting and NMI status display

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

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

0x5040: Watchdog Timer Control Register (WDT_CTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Watchdog Timer Control Register (WDT_CTL)	0x5040 (8 bits)	D7–5	–	reserved	–	–	–	0 when being read.
		D4	WDTRST	Watchdog timer reset	1 Reset 0 Ignored	0	W	
		D3–0	WDTRUN[3:0]	Watchdog timer run/stop control	Other than 1010 Run 1010 Stop	1010	R/W	

D[7:5] Reserved

D4 WDTRST: Watchdog Timer Reset Bit

Resets the watchdog timer.

1 (W): Reset
0 (W): Disabled
0 (R): Normally 0 when read out (default)

To use the watchdog timer, it must be reset by writing 1 to this bit within the NMI/Reset generation cycle (4 seconds when f_{OSC1} = 32.768 kHz).
This resets the up-counter to 0 and starts counting with a new NMI/Reset generation cycle.

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

Controls the watchdog timer Run/Stop.
Values other than 0b1010 (R/W): Run
0b1010 (R/W): Stop (default)

The watchdog timer must also be reset to prevent generation of an unnecessary NMI or Reset while the watchdog timer operates.

0x5041: Watchdog Timer Status Register (WDT_ST)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Watchdog Timer Status Register (WDT_ST)	**0x5041** (8 bits)	D7–2	**–**	reserved	–				–	–	0 when being read.
		D1	**WDTMD**	NMI/Reset mode select	1	Reset	0	NMI	0	R/W	
		D0	**WDTST**	NMI status	1	NMI occurred	0	Not occurred	0	R	

D[7:2] Reserved

D1 WDTMD: NMI/Reset Mode Select Bit

Selects NMI or Reset generation on counter overflow.
1 (R/W): Reset
0 (R/W): NMI (default)

Setting this bit to 1 outputs a reset signal when the counter overflows. Setting to 0 outputs an NMI signal.

D0 WDTST: NMI Status Bit

Indicates a counter overflow and NMI occurrence.
1 (R): NMI occurred (counter overflow)
0 (R): NMI did not occur (default)

This bit confirms that the watchdog timer was the source of the NMI.
The WDTST set to 1 is cleared to 0 by resetting the watchdog timer.
This is also set by a counter overflow if reset output is selected, but is cleared by initial resetting and cannot be confirmed.

17.5 Precautions

- When the watchdog timer is running, this must be reset by software within a 131,072 f_{OSC1} seconds (4 seconds when f_{OSC1} = 32.768 kHz) cycle.
- The watchdog timer must also be reset to prevent generation of an unnecessary NMI or Reset while the watchdog timer operates.

18 UART

18.1 UART Configuration

The S1C17702 incorporates a single-channel UART. The UART transfers asynchronous data with external devices at a transfer rate of 150 to 460,800 bps (115,200 bps in IrDA mode). It includes a 2-byte receive data buffer and 1-byte transmit data buffer and is capable of full-duplex communications. The transfer clock can be either the internal clock using the timer module or an external clock input via the SCLK (P25) pin. Data length (7 or 8 bits), stop bit length (1 or 2 bits), and parity mode (even, odd, no parity) can be selected via the software. The start bit is fixed at 1 bit. Overrun errors, framing errors, and parity errors can be detected while receiving data. The UART generates three different interrupt types (transmit buffer empty, receive buffer full, and receive error) and enables efficient processing of serial data transfer using interrupt processing.
This UART module also incorporates an RZI modulation/demodulation circuit that enables IrDA 1.0-compatible infrared communications simply by adding basic external circuits.
Figure 18.1.1 illustrates the UART configuration.

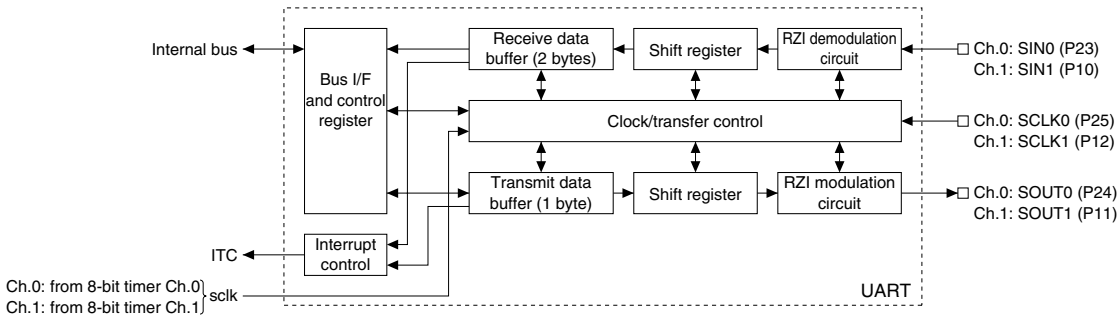


Figure 18.1.1: UART configuration

Note: The UART modules for the two channels have the same functions except for control register addresses. For this reason, the description in this section applies to all UART channels. The "x" in the register name indicates the channel number (0 or 1). Register addresses are indicated either as Ch.0" or "Ch.1".
E.g.: UART_CTLx register (0x4104/0x4124)
Ch.0: UART_CTL0 register (0x4104)
Ch.1: UART_CTL1 register (0x4124)

18.2 UART Pin

Table 18.2.1 lists the UART input/output pins.

Table 18.2.1: UART pin list

Pin name	I/O	Qty	Function
SIN0 (P23)	I	1	UART Ch.0 data input pin Inputs serial data sent from an external device.
SOUT0 (P24)	O	1	UART Ch.0 data output pin Outputs serial data sent to an external device.
SCLK0 (P25)	I	1	UART Ch.0 clock input pin Inputs the external clock when used for the transfer clock.
SIN1 (P10)	I	1	UART Ch.1 data input pin Inputs serial data sent from an external device.
SOUT1 (P11)	O	1	UART Ch.1 data output pin Outputs serial data sent to an external device.
SCLK1 (P12)	I	1	UART Ch.1 clock input pin Inputs the external clock when used for the transfer clock.

The UART input/output pins (SIN*x*, SOUT*x*, SCLK*x*) are shared with general purpose input/output port pins (P23, P24, P25) and are initially set as general purpose input/output port pins. The function must be switched using the P2_PMUX, P1_PMUX register setting to use general purpose input/output port pins as UART input/output pins. Switch the pins to serial interface mode by setting the following control bits to 1.

UART Ch.0

P23 → SIN0

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

P24 → SOUT0

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

P25 → SCLK0 (only when using external clock)

∗ **P25MUX**: P25 Port Function Select Bit in the P2 Port Function Select (P2_PMUX) Register (D5/0x52a2)

UART Ch.1

P10 → SIN1

∗ **P10MUX**: P10 Port Function Select Bit in the P1 Port Function Select (P1_PMUX) Register (D0/0x52a1)

P11 → SOUT1

∗ **P11MUX**: P11 Port Function Select Bit in the P1 Port Function Select (P1_PMUX) Register (D1/0x52a1)

P12 → SCLK1 (only when using external clock)

∗ **P12MUX**: P12 Port Function Select Bit in the P1 Port Function Select (P1_PMUX) Register (D2/0x52a1)

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

18.3 Transfer Clock

The UART transfer clock can be set to internal or external using SSCK (D0/UART_MOD*x* register).

* **SSCK**: Input Clock Select Bit in the UART Ch.*x* Mode (UART_MOD*x*) Register (D0/0x4103/0x4123)

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

* **RXEN**: UART Enable Bit in the UART Control (UART_CTL*x*) Register (D0/0x4104/0x4124)

Internal clock

Setting SSCK to 0 (the default value) selects the internal clock. UART Ch.0 uses the 8-bit timer Ch.0 output clock as the transfer timer, while UART Ch.1 uses the 8-bit timer Ch.1 output clock. Thus, bit timers must be programmed to output a clock suited to the transfer rate.
For more information on 8-bit timer control, see "12 8-bit Timer (T8F)."

External clock

Setting SSCK to 1 selects the external clock. In this case, set P25 to the SCLK pin (see Section 18.2) to input the external clock.

Note:
- **The UART generates a sampling clock that divides the 8-bit timer output into 1/16 divisions. Be careful when setting the transfer rate.**
- **To input the external clock via the SCLK pin, the clock frequency must be less than half of the PCLK and have a duty ratio of 50%.**

18.4 Transfer Data Settings

Set the following conditions to set the transfer data format.
- Data length: 7 or 8 bits
- Start bit: Fixed at 1 bit
- Stop bit: 1 or 2 bits
- Parity bit: Even, odd, no parity

Note: Make sure the UART is halted (when RXEN/UART_CTL*x* register = 0) before changing transfer data format settings.

∗ **RXEN**: UART Enable Bit in the UART Ch.*x* Control (UART_CTL*x*) Register (D0/0x4104/0x4124)

Data length

The data length is selected by CHLN (D4/UART_MOD*x* register). Setting CHLN to 0 (default) sets the data length to 7 bits. Setting CHLN to 1 sets the data length to 8 bits.

∗ **CHLN**: Character Length Select Bit in the UART Ch.*x* Mode (UART_MOD*x*) Register (D4/0x4103/0x4123)

Stop bit

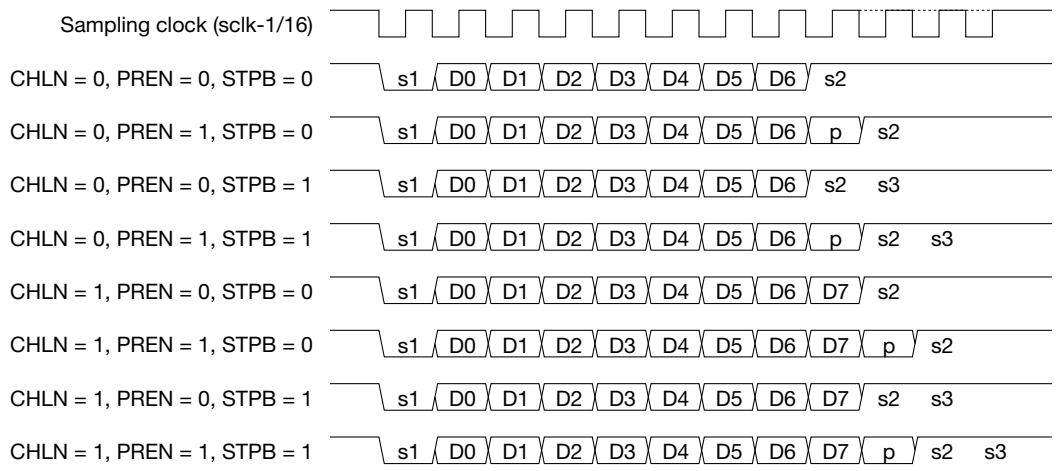
The stop bit length is selected by STPB (D1/UART_MOD*x* register). Setting STPB to 0 (default) sets the stop bit length to 1 bit. Setting STPB to 1 sets the stop bit length to 2 bits.

∗ **STPB**: Stop Bit Select Bit in the UART Ch.*x* Mode (UART_MOD*x*) Register (D1/0x4103/0x4123)

Parity bit

Whether the parity function is enabled or disabled is selected by PREN (D3/UART_MOD*x* register). Setting PREN to 0 (default) disables the parity function. In this case, no parity bit is added to the transfer data and the data is not checked for parity when received. Setting PREN to 1 enables the parity function. In this case, a parity bit is added to the transfer data and the data is checked for parity when received.
When the parity function is enabled, the parity mode is selected by PMD (D2/UART_MOD*x* register). Setting PMD to 0 (default) adds a parity bit and checks for even parity. Setting PMD to 1 adds a parity bit and checks for odd parity.

∗ **PREN**: Parity Enable Bit in the UART Ch.*x* Mode (UART_MOD*x*) Register (D3/0x4103/0x4123)
∗ **PMD**: Parity Mode Select Bit in the UART Ch.*x* Mode (UART_MOD*x*) Register (D2/0x4103/0x4123)



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

Figure 18.4.1: Transfer data format

18.5 Data Transfer Control

Make the following settings before starting data transfers.

(1) Select input clock. (See Section 18.3.)
    To use the internal clock, program the 8-bit timer to output the transfer clock. See Section 12.
(2) Set the transfer data format. (See Section 18.4.)
(3) To use the IrDA interface, set IrDA mode. (See Section 18.8.)
(4) Set interrupt conditions to use UART interrupts. (See Section 18.7.)

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

∗ **RXEN**: UART Enable Bit in the UART Ch.*x* Control (UART_CTL*x*) Register (D0/0x4104/0x4124)

Permitting data transfers

Set the RXEN bit (D0/UART_CTL*x* register) to 1 to permit data transfers. This switches transfer circuits to enable transfers.

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

Data transfer control

To start data transmission, program the transmission data to the UART_TXD*x* register (0x4101/0x4121).

∗ **UART_TXD*x***: UART Ch.*x* Transmit Data Register (0x4101/0x4121)

The data is written to the transmit data buffer, and the transmission circuit starts sending data.
The buffer data is sent to the transmit shift register, and the start bit is output from the SOUT pin. The data in the shift register is then output from the LSB. The transfer data bit is shifted in sync with the sampling clock rising edge and output in sequence via the SOUT pin. Following output of MSB, the parity bit (if parity is enabled) and stop bit are output.

The transmission circuit includes the TDBE (D0/UART_ST*x* register) and TRBS (D2/UART_ST*x* register) status flags.

∗ **TDBE**: Transmit Data Buffer Empty Flag in the UART Ch.*x* Status (UART_ST*x*) Register (D0/0x4100/0x4120)
∗ **TRBS**: Transmit Busy Flag in the UART Ch.*x* Status (UART_ST*x*) Register (D2/0x4100/0x4120)

The TDBE flag indicates the transmit data buffer status. This flag switches to 0 when the application program programs data to the transmit data buffer and reverts to 1 when the buffer data is sent to the transmit shift register. Interrupts can be generated when this flag is 1 (see Section 18.7). Subsequent data is sent after confirming that the transmit data buffer is empty either by using this interrupt or by inspecting the TDBE flag. The transmission buffer size is 1 byte, but a shift register is provided separately to allow data to be written while the previous data is being sent. Always confirm that the transmit data buffer is empty before writing transmission data. Writing data while the TDBE flag is 0 will overprogram earlier transmission data inside the transmit data buffer.

The TRBS flag indicates the shift register status. This flag switches to 1 when transmission data is loaded from the transmit data buffer to the shift register and reverts to 0 once the data is sent. Read this flag to check whether the transmission circuit is operating or at standby.

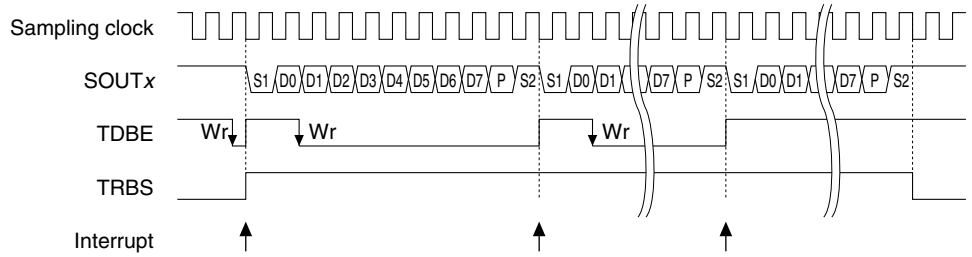


Figure 18.5.1: Data transmission timing chart

Data reception control

The receiving circuit is launched by setting the RXEN bit to 1, enabling data to be received from an external serial device.

When the external serial device sends the start bit, the receiving circuit detects its Low level and starts sampling the following data bits. The data bits are sampled at the sampling clock rising edge, and the lead bit is loaded into the receive shift register as LSB. Once the MSB has been received into the shift register, the received data is loaded into the receive data buffer. If parity checking is enabled, the receiving circuit checks parity at the same time by checking the parity bit received immediately after the MSB.

The receive data buffer, a 2-byte FIFO, receives data until full.

Received data in the buffer can be read from the UART_RXD*x* register (0x4102/0x4122). The oldest data is read out first, clearing the register.

∗ **UART_RXD*x***: UART Ch.*x* Receive Data Register (0x4102/0x4122)

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

∗ **RDRY**: Receive Data Ready Flag in the UART Ch.*x* Status (UART_ST*x*) Register (D1/0x4100/0x4120)
∗ **RD2B**: Second Byte Receive Flag in the UART Ch.*x* Status (UART_ST*x*) Register (D3/0x4100/0x4120)

The RDRY flag indicates that the receive data buffer still contains data. The RD2B flag indicates that the receive data buffer is full.

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

(2) RDRY = 1, RD2B = 0
One data item 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.

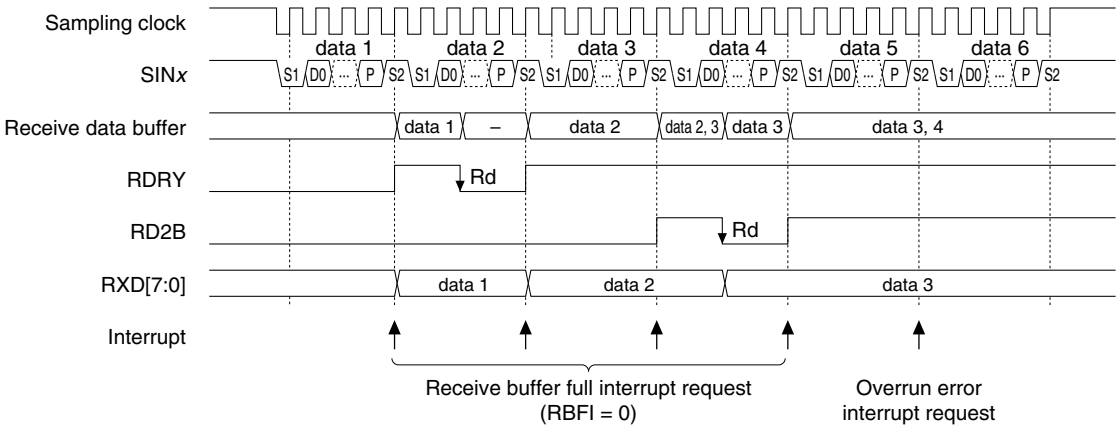
(3) RDRY = 1, RD2B = 1
Two data items have been received. Read the receive data buffer contents twice. The receive data buffer outputs the oldest data first. This resets the RD2B flag. The buffer then reverts to the state in (2) above. The second read outputs the most recent received data, after which 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. With default settings, a receive buffer full interrupt occurs when the receive data buffer receives one item of data (status (2) above). This can be changed by setting the RBFI bit (D1/UART_CTL*x* register) to 1 so that an interrupt occurs when the receive data buffer receives two items of data.

∗ **RBFI**: Receive Buffer Full Interrupt Condition Setup Bit in the UART Ch.*x* Control (UART_CTL*x*) Register (D1/0x4104/0x4124)

Three error flags are also provided in addition to the flags previously mentioned. See Section 18.6 for detailed information on flags and receive errors.



S1: Start bit, S2: Stop bit, P: Parity bit, Rd: Data bits from RXD[7:0]

Figure 18.5.2: Data receiving timing chart

Blocking data transfers

After a data transfer is completed (both transmission and reception), data transfers are blocked by writing 0 to the RXEN bit. Confirm that the TDBE flag is 1 and the TRBS and RDRY flags are both 0 before blocking data transfer.

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

18.6 Receive Errors

Three different receive errors may be detected while receiving data.
Since receive errors are interrupt factors, they can be processed by generating interrupts. For more information on UART interrupt control, refer to Section 18.7.

Parity error

If PREN (D3/UART_MOD*x* register) has been set to 1 (parity enabled), data received is checked for parity.
Data received in the shift register is checked for parity when sent to the receive data buffer. The matching is checked against the PMD (D2/UART_MOD*x* register) setting (odd or even parity). If the result is a non-match, a parity error is issued, and the parity error flag PER (D5/UART_ST*x* register) is set to 1.
Even if this error occurs, the data received is sent to the receive data buffer, and the receiving operation continues. However, the received data cannot be guaranteed if a parity error occurs.
The PER flag (D5/UART_ST*x* register) is reset to 0 by writing as 1.

- **PREN**: Parity Enable Bit in the UART Ch.*x* Mode (UART_MOD*x*) Register (D3/0x4103/0x4123)
- **PMD**: Parity Mode Select Bit in the UART Ch.*x* Mode (UART_MOD*x*) Register (D2/0x4103/0x4123)
- **PER**: Parity Error Flag in the UART Ch.*x* Status (UART_ST*x*) Register (D5/0x4100/0x4120)

Framing error

A framing error occurs if the stop bit is received as 0 and the UART determines sync offset. If the stop bit is set to two bits, only the first bit is checked.
The framing error flag FER (D6/UART_ST*x* register) is set to 1 if this error occurs. The received data is still transferred to the receive data buffer if this error occurs and the receiving operation continues, but the data cannot be guaranteed, even if no framing error occurs for subsequent data receiving.
The FER flag (D6/UART_ST*x* register) is reset to 0 by writing as 1.

- **FER**: Framing Error Flag in the UART Ch.*x* Status (UART_ST*x*) Register (D6/0x4100/0x4120)

Overrun error

Even if the receive data buffer is full (two data items already received), a third item of data can be received in the shift register. However, if the receive data buffer is not emptied (by reading out data received) by the time this data has been received, the third data received in the shift register will not be sent to the buffer and an overrun error will occur.
If an overrun error occurs, the overrun error flag OER (D4/UART_ST*x* register) is set to 1.
The receiving operation continues even if this error occurs.
The OER flag (D4/UART_ST*x* register) is reset to 0 by writing as 1.

- **OER**: Overrun Error Flag in the UART Ch.*x* Status (UART_ST*x*) Register (D4/0x4100/0x4120)

18.7 UART Interrupts

The UART includes a function for generating the following three different interrupt types.
- Transmit buffer empty interrupt
- Receive buffer full interrupt
- Receive error interrupt

The UART outputs one interrupt signal shared by the three above interrupt factor types to the interrupt controller (ITC). Inspect the status flag or error flag to determine the interrupt factor occurring.

Transmit buffer empty interrupt

To use this interrupt, set TIEN (D4/UART_CTL*x* register) to 1. If TIEN is set to 0 (default), interrupt requests for this factor will not be sent to the ITC.

∗ **TIEN**: Transmit Buffer Empty Interrupt Enable Bit in the UART Ch.*x* Control (UART_CTL*x*) Register (D4/0x4104/0x4124)

When transmission data written to the transmit data buffer is transferred to the shift register, the UART sets the TDBE bit (D0/UART_ST*x* register) to 1, indicating that the transmit data buffer is empty. If transmit buffer empty interrupts are permitted (TIEN = 1), an interrupt request pulse is sent simultaneously to the ITC.

∗ **TDBE**: Transmit Data Buffer Empty Flag in the UART Ch.*x* Status (UART_ST*x*) Register (D0/0x4100/0x4120)

An interrupt occurs if other interrupt conditions are met.
You can inspect the TDBE flag in the UART interrupt handler routine to determine whether the UART interrupt is attributable to a transmit buffer empty. If TDBE is 0, the next transmission data can be written to the transmit data buffer by the interrupt handler routine.

Receive buffer full interrupt

To use this interrupt, set RIEN (D5/UART_CTL*x* register) to 1. If RIEN is set to 0 (default), interrupt requests for this factor will not be sent to the ITC.

∗ **RIEN**: Receive Buffer Full Interrupt Enable Bit in the UART Ch.*x* Control (UART_CTL*x*) Register (D5/0x4104/0x4124)

If the specified volume of received data is loaded into the receive data buffer when a receive buffer full interrupt is permitted (RIEN = 1), the UART outputs an interrupt request pulse to the ITC. If RBFI (D1/UART_CTL*x* register) is 0, an interrupt request pulse is output as soon as one item of received data is loaded into the receive data buffer (RDRY flag (D1/UART_ST*x* register) is set to 1). If RBFI (D1/UART_CTL*x* register) is 1, an interrupt request pulse is output as soon as two items of received data are loaded into the receive data buffer (RD2B flag (D3/UART_ST*x* register) is set to 1).

∗ **RBFI**: Receive Buffer Full Interrupt Condition Ch.*x* Setup Bit in the UART Control (UART_CTL*x*) Register (D1/0x4104/0x4124)
∗ **RDRY**: Receive Data Ready Flag in the UART Ch.*x* Status (UART_ST*x*) Register (D1/0x4100/0x4120)
∗ **RD2B**: Second Byte Receive Flag in the UART Ch.*x* Status (UART_ST*x*) Register (D3/0x4100/0x4120)

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

Receive error interrupt

To use this interrupt, set REIEN (D6/UART_CTL register) to 1. If REIEN is set to 0 (default), interrupt requests will not be sent to the ITC for this factor.

* **REIEN**: Receive Error Interrupt Enable Bit in the UART Control (UART_CTL) Register (D6/0x4104)

The UART sets the error flags shown below to 1 if a parity error, framing error, or overrun error is detected when receiving data. If receive error interrupts are permitted (REIEN = 1), an interrupt request pulse is output at the same time to the ITC.

* **PER**: Parity Error Flag in the UART Ch.*x* Status (UART_ST*x*) Register (D5/0x4100/0x4120)
* **FER**: Framing Error Flag in the UART Ch.*x* Status (UART_ST*x*) Register (D6/0x4100/0x4120)
* **OER**: Overrun Error Flag in the UART Ch.*x* Status (UART_ST*x*) Register (D4/0x4100/0x4120)

If other interrupt conditions are satisfied, an interrupt occurs.
Inspect the error flags above as part of the UART interrupt handler routine to determine whether the UART interrupt was caused by a receive error. If any of the error flags has the value 1, the interrupt handler routine will proceed with error recovery.

Interrupt vectors

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

Table 18.7.1: UART interrupt vector

Channel	Vector number	Vector address
Ch.0	16 (0x10)	TTBR + 0x40
Ch.1	17 (0x11)	TTBR + 0x44

Other interrupt settings

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

18.8 IrDA Interface

This UART module incorporates an RZI modulation/demodulation circuit enabling implementation of IrDA 1.0-compatible infrared communication simply by adding basic external circuits.

The transmission data output from the UART transmit shift register is input to the modulation circuit and output from the SOUT pin after the Low pulse has been modulated to a 3/16 sclk cycle.

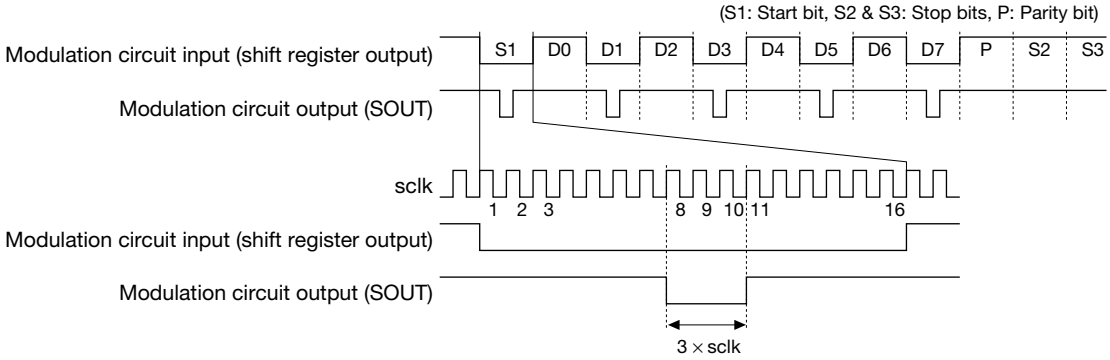


Figure 18.8.1: Transmission signal waveform

The received IrDA signal is input to the demodulation circuit and the Low pulse width is converted to 16 sclk cycles before entry to the receive shift register. The demodulation circuit uses the pulse detection clock selected from the prescaler output clock separately from the transfer cock to detect Low pulses input (when minimum pulse width = 1.41 µs/115,200 bps).

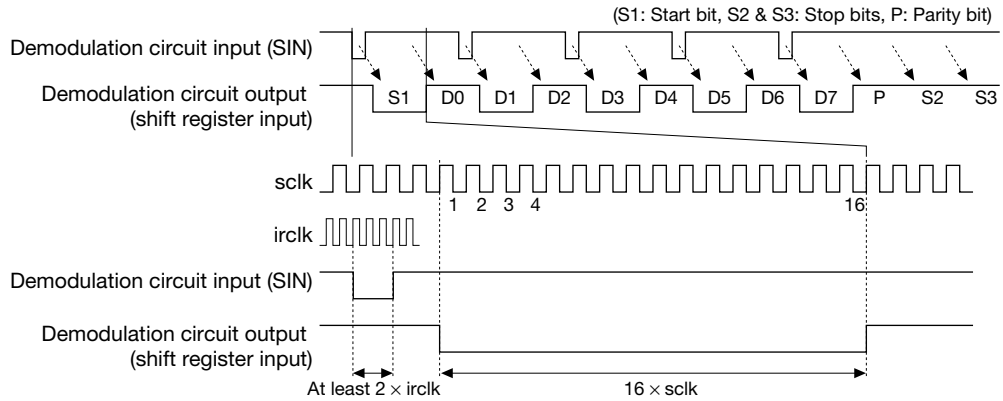


Figure 18.8.2: Receive signal waveform

IrDA enable

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

∗ **IRMD**: IrDA Mode Select Bit in the UART Ch.*x* Expansion (UART_EXP*x*) Register (D0/0x4105/0x4125)

Note: This must be set before setting other UART conditions.

IrDA receive detection clock selection

The input pulse detection clock is selected from among the prescaler output clock PCLK-1/1 to PCLK-1/128 using IRCLK[2:0] (D[6:4]/UART_EXP*x* register).

* **IRCLK[2:0]**: IrDA Receive Detection Clock Select Bits in the UART Ch. Expansion (UART_EXP*x*) Register (D[6:4]/0x4105/0x4125)

Table 18.8.1: IrDA receive detection clock selection

IRCLK[2:0]	Prescaler output clock
0x7	PCLK-1/128
0x6	PCLK-1/64
0x5	PCLK-1/32
0x4	PCLK-1/16
0x3	PCLK-1/8
0x2	PCLK-1/4
0x1	PCLK-1/2
0x0	PCLK-1/1

(Default: 0x0)

This clock must be selected as a clock faster than the 8-bit timer or transfer clock sclk input via the SCLK*x* pin. The demodulation circuit treats Low pulses with a width of at least 2 IrDA receive detection clock cycles as valid and converts them to 16 sclk cycle width Low pulses. Select the prescaler output clock to enable detection of input pulses with a minimum width of 1.41 µs.

Serial data transfer control

Data transfer control in IrDA mode is identical to that for normal interfaces. For detailed information on data format settings and data transfer and interrupt control methods, refer to the previous discussions.

18.9 Control Register Details

Table 18.9.1: UART register list

Address	Register name		Function
0x4100	UART_ST0	UART Ch.0 Status Register	Transfer, buffer, error status display
0x4101	UART_TXD0	UART Ch.0 Transmit Data Register	Transmission data
0x4102	UART_RXD0	UART Ch.0 Receive Data Register	Received data
0x4103	UART_MOD0	UART Ch.0 Mode Register	Transfer data format setting
0x4104	UART_CTL0	UART Ch.0 Control Register	Data transfer control
0x4105	UART_EXP0	UART Ch.0 Expansion Register	IrDA mode setting
0x4120	UART_ST1	UART Ch.1 Status Register	Transfer, buffer, error status display
0x4121	UART_TXD1	UART Ch.1 Transmit Data Register	Transmission data
0x4122	UART_RXD1	UART Ch.1 Receive Data Register	Received data
0x4123	UART_MOD1	UART Ch.1 Mode Register	Transfer data format setting
0x4124	UART_CTL1	UART Ch.1 Control Register	Data transfer control
0x4125	UART_EXP1	UART Ch.1 Expansion Register	IrDA mode setting

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

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

0x4100/0x4120: 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	

The "*x*" in register names indicates the channel number (0 or 1).

0x4100: UART Ch.0 Status Register (UART_ST0)
0x4120: UART Ch.1 Status Register (UART_ST1)

D7 Reserved

D6 FER: Framing Error Flag

Indicates whether a framing error has occurred.

1 (R): Error occurred
0 (R): No error (default)
1 (W): Reset to 0
0 (W): Disabled

FER is set to 1 when a framing error occurs. Framing errors occur when data is received with the stop bit set to 0.
FER is reset by writing 1.

D5 PER: Parity Error Flag

Indicates whether a parity error has occurred.

1 (R): Error occurred
0 (R): No error (default)
1 (W): Reset to 0
0 (W): Disabled

PER is set to 1 when a parity error occurs. Parity checking is enabled only when PREN (D3/ UART_MOD*x* register) is set to 1 and is performed when received data is transferred from the shift register to the receive data buffer.
PER is reset by writing 1.

D4 OER: Overrun Error Flag

Indicates whether an overrun error has occurred.

1 (R): Error occurred
0 (R): No error (default)
1 (W): Reset to 0
0 (W): Disabled

OER is set to 1 when an overrun error occurs. Overrun errors occur when data is received in the shift register when the receive data buffer is already full and additional data is sent. The receive data buffer is not overwritten if this error occurs. The shift register is overwritten as soon as the error occurs.
OER is reset by writing 1.

D3 RD2B: Second Byte Received Flag

Indicates that the receive data buffer contains two items of received data.

1 (R): Second byte can be read
0 (R): Second byte not received (default)

RD2B is set to 1 when the second byte of data is loaded into the receive data buffer and is reset to 0 when the first data is read from the receive data buffer.

D2 TRBS: Transmit Busy Flag

Indicates the transmit shift register status.

1 (R): Operating

0 (R): Standby (default)

TRBS is set to 1 when transmission data is loaded from the transmit data buffer into the shift register and is reset to 0 when the data transfer is complete. Inspect TRBS to determine whether the transmit circuit is operating or at standby.

D1 RDRY: Receive Data Ready Flag

Indicates that the receive data buffer contains valid received data.

1 (R): Data can be read

0 (R): Buffer empty (default)

RDRY is set to 1 when received data is loaded into the receive data buffer and is reset to 0 when all data has been read from the receive data buffer.

D0 TDBE: Transmit Data Buffer Empty Flag

Indicates the state of the transmit data buffer.

1 (R): Buffer empty (default)

0 (R): Data exists

TDBE is reset to 0 when transmit data is written to the transmit data buffer and is set to 1 when the data is transferred to the shift register.

0x4101/0x4121: UART Ch.*x* Transmit Data Registers (UART_TXD*x*)

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

The "*x*" in register names indicates the channel number (0 or 1).

0x4101: UART Ch.0 Transmit Data Register (UART_TXD0)
0x4121: UART Ch.1 Transmit Data Register (UART_TXD1)

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

Program transmit data to be set in the transmit data buffer. (Default: 0x0)
The UART begins transmitting when data is written to this register. Data written to TXD[7:0] is retained until sent to the transmit data buffer.
Transmitting data from within the transmit data buffer generates a transmit buffer empty interrupt factor.
TXD7 (MSB) is invalid in 7-bit mode.
Serial converted data is output from the SOUT pin, with the LSB first bits set to 1 as High level and bits set to 0 as Low level.
This register can also be read from.

0x4102/0x4122: UART Ch.*x* Receive Data Registers (UART_RXD*x*)

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

The "*x*" in register names indicates the channel number (0 or 1).

0x4102: UART Ch.0 Receive Data Register (UART_RXD0)
0x4122: UART Ch.1 Receive Data Register (UART_RXD1)

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

Data in the receive data buffer is read out in sequence, starting with the oldest. Received data is placed in the receive data buffer. The receive data buffer is a 2-byte FIFO that allows proper data receipt until it fills, even if data is not read out. If the buffer is full and the shift register also contains received data, an overrun error will occur, unless the data is read out before receipt of the subsequent data starts.

The receive circuit includes two receive buffer status flags: RDRY (D1/UART_ST*x* register) and RD2B (D3/UART_ST*x* register). The RDRY flag indicates the presence of valid received data in the receive data buffer, while RD2B flag indicates the presence of two items of received data in the receive data buffer.

A receive buffer full interrupt occurs when the received data in the receive data buffer reaches the number specified by RBFI (D1/UART_CTL*x* register).

0 is loaded into RXD7 in 7-bit mode.

Serial data input via the SIN pin is converted to parallel, with the initial bit as LSB, the High level bit as 1, and the Low level bit as 0. This data is then loaded into the receive data buffer.

This register is read-only. (Default: 0x0)

0x4103/0x4123: UART Ch.*x* Mode Registers (UART_MOD*x*)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
UART Ch.*x*	**0x4103**	D7–5	**–**	reserved	–				–	–	0 when being read.
Mode Register	**0x4123**	D4	**CHLN**	Character length	1	8 bits	0	7 bits	0	R/W	
(UART_MOD*x*)	(8 bits)	D3	**PREN**	Parity enable	1	With parity	0	No parity	0	R/W	
		D2	**PMD**	Parity mode select	1	Odd	0	Even	0	R/W	
		D1	**STPB**	Stop bit select	1	2 bits	0	1 bit	0	R/W	
		D0	**SSCK**	Input clock select	1	External	0	Internal	0	R/W	

The "*x*" in register names indicates the channel number (0 or 1).

0x4103: UART Ch.0 Mode Register (UART_MOD0)
0x4123: UART Ch.1 Mode Register (UART_MOD1)

D[7:5] Reserved

D4 CHLN: Character Length Select Bit

Selects the serial transfer data length.
1 (R/W): 8 bits
0 (R/W): 7 bits (default)

D3 PREN: Parity Enable Bit

Enables the parity function.
1 (R/W): With parity
0 (R/W): No parity (default)

PREN is used to select receive data parity checking and to determine whether a parity bit is added to transmitted data. Setting PREN to 1 parity-checks the received data. A parity bit is automatically added to the transmitted data. If PREN is set to 0, no parity bit is checked or added.

D2 PMD: Parity Mode Select Bit

Selects the parity mode.
1 (R/W): Odd parity
0 (R/W): Even parity (default)

Writing 1 to PMD selects odd parity; writing 0 to it selects even parity. Parity checking and parity bit addition are enabled only when PREN (D3) is set to 1. The PMD setting is disabled if PREN (D3) is 0.

D1 STPB: Stop Bit Select Bit

Selects the stop bit length.
1 (R/W): 2 bits
0 (R/W): 1 bit (default)

Writing 1 to STPB selects two stop bits; writing 0 to it selects one bit. The start bit is fixed at one bit.

D0 SSCK: Input Clock Select Bit

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

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

0x4104/0x4124: UART Ch.*x* Control Registers (UART_CTL*x*)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
UART Ch.*x* Control Register (UART_CTL*x*)	0x4104 0x4124 (8 bits)	D7	–	reserved	–				–	–	0 when being read.
		D6	REIEN	Receive error int. enable	1	Enable	0	Disable	0	R/W	
		D5	RIEN	Receive buffer full int. enable	1	Enable	0	Disable	0	R/W	
		D4	TIEN	Transmit buffer empty int. enable	1	Enable	0	Disable	0	R/W	
		D3–2	–	reserved	–				–	–	0 when being read.
		D1	RBFI	Receive buffer full int. condition	1	2 bytes	0	1 byte	0	R/W	
		D0	RXEN	UART enable	1	Enable	0	Disable	0	R/W	

The "*x*" in register names indicates the channel number (0 or 1).

0x4104: UART Ch.0 Control Register (UART_CTL0)
0x4124: UART Ch.1 Control Register (UART_CTL1)

D7 Reserved

D6 REIEN: Receive Error Interrupt Enable Bit

Permits interrupt requests to the ITC when a receive error occurs.
1 (R/W): Permitted
0 (R/W): Prohibited (default)

Set this bit to 1 to process receive errors using interrupts.

D5 RIEN: Receive Buffer Full Interrupt Enable Bit

Permits interrupt requests to the ITC caused when the received data quantity in the receive data buffer reaches the quantity specified in RBFI (D1).
1 (R/W): Permitted
0 (R/W): Prohibited (default)

Set this bit to 1 to read receive data using interrupts.

D4 TIEN: Transmit Buffer Empty Interrupt Enable Bit

Permits interrupt requests to the ITC caused when transmission data in the transmit data buffer is sent to the shift register (i.e. when data transmission begins).
1 (R/W): Permitted
0 (R/W): Prohibited (default)

Set this bit to 1 to program data to the transmit data buffer using interrupts.

D[3:2] Reserved

D1 RBFI: Receive Buffer Full Interrupt Condition Setup Bit

Sets the quantity of data in the receive buffer to generate a receive buffer full interrupt.
1 (R/W): 2 bytes
0 (R/W): 1 byte (default)

If receive buffer full interrupts are permitted (RIEN = 1), the UART outputs an interrupt request pulse to the ITC when the quantity of received data specified by RBFI is loaded into the receive data buffer. If the RBFI bit is 0, an interrupt request pulse is output as soon as one item of received data is loaded into the receive data buffer (when the RDRY flag (D1/UART_ST*x* register) is set to 1). If RBFI is 1, an interrupt request pulse is output as soon as two items of received data are loaded into the receive data buffer (when the RD2B flag (D3/UART_ST*x* register) is set to 1).

D0 RXEN: UART Enable Bit

Permits data transfer by the UART.
1 (R/W): Permitted
0 (R/W): Prohibited (default)

Set RXEN to 1 before starting UART transfers. Setting RXEN to 0 will stop data transfers. Set the transfer conditions while RXEN is 0.
Preventing transfers by writing 0 to RXEN also clears transmit data buffer.

0x4105/0x4125: UART Ch.*x* Expansion Registers (UART_EXP*x*)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
UART Ch.*x* Expansion Register (UART_EXP*x*)	**0x4105** **0x4125** (8 bits)	D7	**–**	reserved	–	–	–	0 when being read.
		D6–4	**IRCLK[2:0]**	IrDA receive detection clock select	IRCLK[2:0] / Clock: 0x7 PCLK-1/128; 0x6 PCLK-1/64; 0x5 PCLK-1/32; 0x4 PCLK-1/16; 0x3 PCLK-1/8; 0x2 PCLK-1/4; 0x1 PCLK-1/2; 0x0 PCLK-1/1	0x0	R/W	
		D3–1	**–**	reserved	–	–	–	0 when being read.
		D0	**IRMD**	IrDA mode select	1 On / 0 Off	0	R/W	

The "*x*" in register names indicates the channel number (0 or 1).

0x4105: UART Ch.0 Expansion Register (UART_EXP0)
0x4125: UART Ch.1 Expansion Register (UART_EXP1)

D7 Reserved

D[6:4] IRCLK[2:0]: IrDA Receive Detection Clock Select Bits

Select the prescaler output clock used as the IrDA input pulse detection clock.

Table 18.9.2: IrDA receive detection clock selection

IRCLK[2:0]	Prescaler output clock
0x7	PCLK-1/128
0x6	PCLK-1/64
0x5	PCLK-1/32
0x4	PCLK-1/16
0x3	PCLK-1/8
0x2	PCLK-1/4
0x1	PCLK-1/2
0x0	PCLK-1/1

(Default: 0x0)

This clock must be selected as a clock faster than the 8-bit timer or transfer clock sclk input via the SCLK pin.

The demodulation circuit treats Low pulses with a width of at least 2 IrDA receive detection clock cycles as valid. Select the appropriate prescaler output clock to enable detection of input pulses with a minimum width of 1.41 µs.

D[3:1] Reserved

D0 IRMD: IrDA Mode Select Bit

Switches the IrDA interface function on and off.

1 (R/W): On

0 (R/W): Off (default)

Set this to 1 to use the IrDA interface. When this bit is set to 0, this module functions as a normal UART, with no IrDA functions.

18.10 Precautions

- The following UART bits should be set with transfers blocked (RXEN = 0).
  - All UART_MOD*x* register (0x4103/0x4123) bits (SSCK, STPB, PMD, PREN, CHLN)
  - RBFI bit in the UART_CTL*x* register (0x4104/0x4124)
  - All UART_EXP*x* register (0x4105/0x4125) bits (IRMD, IRCLK[2:0])

  ∗ **RXEN**: UART Enable Bit in the UART Ch.*x* Control (UART_CTL*x*) Register (D0/0x4104/0x4124)
- Do not set RXEN to 0 while the UART is transmitting or receiving data.
- The UART transfer rate is capped at 460,800 bps (115,200 bps in IrDA mode). Do not set faster transfer rates.
- Preventing transfer by setting RXEN to 0 clears (initializes) the transmit data buffer. Before writing 0 to RXEN, confirm the absence of data in the buffer awaiting transmission.
- The IrDA receive detection clock must be selected as a clock faster than the 8-bit timer or transfer clock sclk input via the SCLK pin.
- The IrDA interface demodulation circuit treats Low pulses with a width of at least 2 IrDA receive detection clock cycles as valid. Select the appropriate prescaler output clock to enable detection of input pulses with a minimum width of 1.41 µs as an IrDA receive detection clock.

19 SPI

19.1 SPI Configuration

The S1C17702 incorporates a synchronized serial interface module (SPI). This SPI module supports both Master and Slave modes and is used for 8-bit data transfers. Four different data transfer timing patterns (clock phase and polarity) can be selected.
The SPI module includes a transmit data buffer and receive data buffer separate from the shift register, and is capable of generating two different interrupt types (transmit buffer empty and receive buffer full). This allows easy processing of continuous serial data transfer using interrupts.
Figure 19.1.1 illustrates the SPI module configuration.

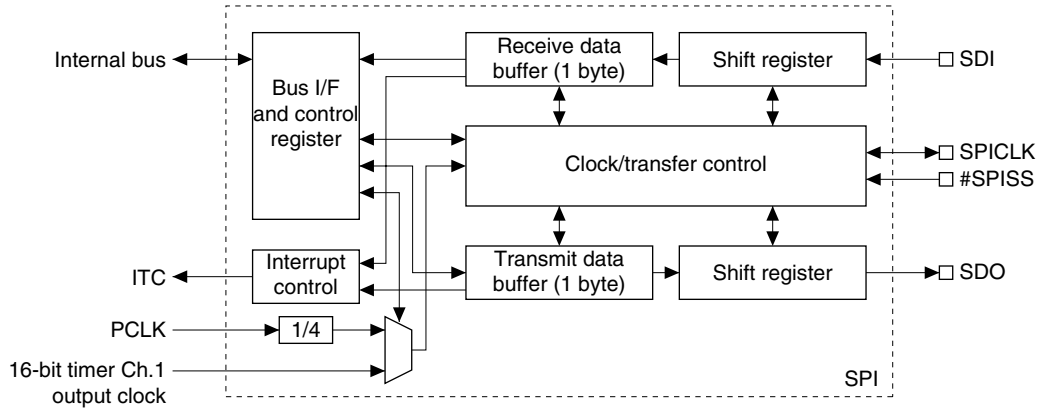


Figure 19.1.1: SPI module configuration

19.2 SPI Input/Output Pins

Table 19.2.1 lists the SPI pins.

Table 19.2.1: SPI pin list

Pin name	I/O	Qty	Function
SDI (P20)	I	1	SPI data input pin Inputs serial data from SPI bus.
SDO (P21)	O	1	SPI data output pin Outputs serial data to SPI bus.
SPICLK (P22)	I/O	1	SPI external clock input/output pin Outputs SPI clock when SPI is in Master mode. Inputs external clock when SPI is used in Slave mode.
#SPISS (P17)	I	1	SPI slave selection signal (active Low) input pin SPI (Slave mode) is selected as slave device by Low input to this pin.

The SPI input/output pins (SDI, SDO, SPICLK, #SPISS) are shared with general purpose input/output port pins (P20, P21, P22, P17) and are initially set as general purpose input/output port pins. The function must be switched using the P2_PMUX and P1_PMUX register settings to use general purpose input/output port pins as SPI input/output pins. Switch the pins to SPI mode by setting the following control bits to 1.

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

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

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

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

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

19.3 SPI Clock

The Master mode SPI uses the 16-bit timer Ch.1 output clock or a PCLK-1/4 clock to generate the SPI clock. This clock is output from the SPICLK pin to the slave device while also driving the shift register.
Use MCLK (D9/SPI_CTL register) to select whether the 16-bit timer Ch.1 output clock or PCLK-1/4 clock is used. Setting MCLK to 1 selects the 16-bit timer Ch.1 output clock; setting to 0 selects the PCLK-1/4 clock.

* **MCLK**: SPI Clock Source Select Bit in the SPI Control (SPI_CTL) Register (D9/0x4326)

Using the 16-bit timer Ch.1 output clock enables programmable transfer rates. For more information on 16-bit timer control, see "11 16-bit Timer (T16)."

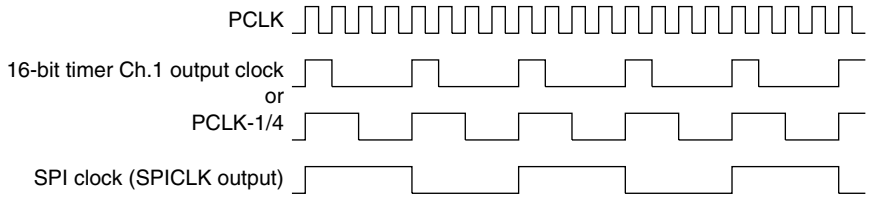


Figure 19.3.1: Master mode SPI clock

In Slave mode, the SPI clock is input via the SPICLK pin.

Note: The duty ratio of the clock input via the SPICLK pin must be 50%.

19.4 Data Transfer Condition Settings

The SPI module can be set to Master or Slave modes. The SPI clock polarity and phase can also be set via the SPI_CTL register. The data length is fixed at 8 bits.

Note: Make sure the SPI module is halted (when SPEN/SPI_CTL register = 0) before Master/Slave mode selection and clock condition settings.

∗ **SPEN**: SPI Enable Bit in the SPI Control (SPI_CTL) Register (D0/0x4326)

Master/Slave mode selection

MSSL (D1/SPI_CTL register) is used to set the SPI module to Master mode or Slave mode. Setting MSSL to 1 sets Master mode; setting it to 0 (default) sets Slave mode. In Master mode, data is transferred using the internal clock. In Slave mode, data is transferred by inputting the master device clock.

∗ **MSSL**: Master/Slave Mode Select Bit in the SPI Control (SPI_CTL) Register (D1/0x4326)

SPI clock polarity and phase settings

The SPI clock polarity is selected by CPOL (D2/SPI_CTL register). Setting CPOL to 1 treats the SPI clock as active Low; setting it to 0 (default) treats it as active High.

∗ **CPOL**: Clock Polarity Select Bit in the SPI Control (SPI_CTL) Register (D2/0x4326)

The SPI clock phase is selected by CPHA (D3/SPI_CTL register).

∗ **CPHA**: Clock Phase Select Bit in the SPI Control (SPI_CTL) Register (D3/0x4326)

As shown below, these control bits set transfer timing.

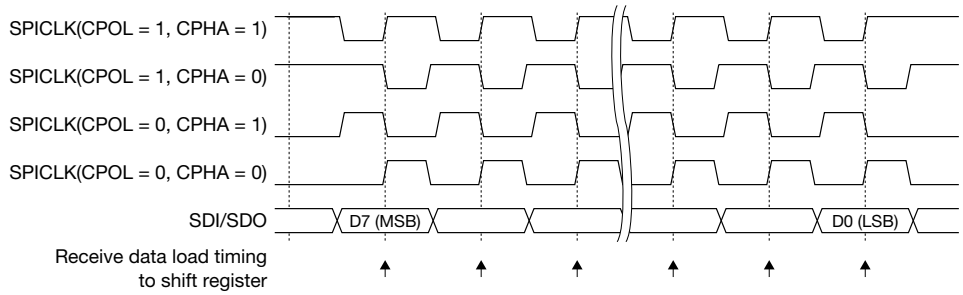


Figure 19.4.1: Clock and data transfer timing

MSB initial/LSB initial settings

Use MLSB (D8/SPI_CTL register) to select whether the data MSB or LSB is input or output first.
MSB initial is set when MLSB is 0 (the default value); LSB initial is set when MLSB is 1.

∗ **MLSB**: LSB/MSB First Mode Select Bit in the SPI Control (SPI_CTL) Register (D8/0x4326)

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.

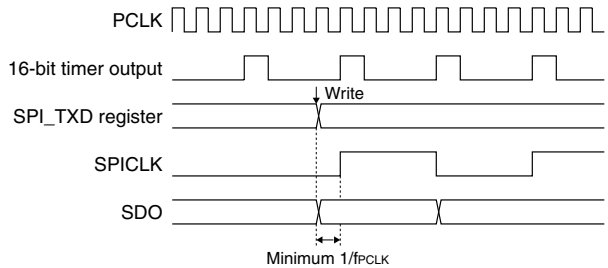


Figure 19.4.2: SDO and SPICLK Change Timings when CPHA = 0

The half SPICLK cycle will be secured from change of data to change of the clock for the second and following transmit data bits and the second and following bytes during continuous transfer.

19.5 Data Transfer Control

Make the following settings before starting data transfers.
(1) Set the 16-bit timer Ch.1 to output the SPI clock. (See Section 11.)
(2) Select Master mode or Slave mode. (See Section 19.4.)
(3) Set clock conditions. (See Section 19.4.)
(4) Set the interrupt conditions to use SPI interrupts. (See Section 19.6.)

Note: Make sure the SPI is halted (when SPEN/SPI_CTL register = 0) before changing the above settings.

∗ **SPEN**: SPI Enable Bit in the SPI Control (SPI_CTL) Register (D0/0x4326)

Permitting data transfers

Set the SPEN bit (D0/SPI_CTL register) to 1 to permit SPI operations. This enables SPI transfers and permits clock input/output.

Note: Do not set SPEN to 0 when the SPI module is transferring data.

Data transfer control

To start data transmission, program the transmission data to the SPI_TXD register (0x4322).

∗ **SPI_TXD**: SPI Transmit Data Register (0x4322)

The data is written to the transmit data buffer, and the SPI module begins sending data. The buffer data is sent to the transmit shift register. In Master mode, the module starts clock output from the SPICLK pin. In Slave mode, the module awaits clock input from the SPICLK pin. The data in the shift register is shifted in sequence at the clock rising or falling edge, as determined by CPHA (D3/SPI_CTL register) and CPOL (D2/SPI_CTL register) (see Figure 19.4.1) and sent from the SDO pin with MSB leading.

∗ **CPHA**: Clock Phase Select Bit in the SPI Control (SPI_CTL) Register (D3/0x4326)
∗ **CPOL**: Clock Polarity Select Bit in the SPI Control (SPI_CTL) Register (D2/0x4326)

The SPI module includes the SPTBE (D0/SPI_ST register) and SPBSY (D2/SPI_ST register) status flags for transfer control.

∗ **SPTBE**: Transmit Data Buffer Empty Flag in the SPI Status (SPI_ST) Register (D0/0x4320)
∗ **SPBSY**: Transfer Busy Flag in the SPI Status (SPI_ST) Register (D2/0x4320)

The SPTBE flag indicates the transmit data buffer status. This flag switches to 0 when the application program programs data to the SPI_TXD register (transmit data buffer) and reverts to 1 when the buffer data is sent to the transmit shift register. Interrupts can be generated when this flag is 1 (see Section 19.6). Subsequent data is sent after confirming that the transmit data buffer is empty either by using this interrupt or by inspecting the SPTBE flag. The transmission buffer size is 1 byte, but a shift register is provided separately to allow data to be written while the previous data is being sent. Always confirm that the transmit data buffer is empty before writing transmission data. Writing data while the SPTBE flag is 0 will overprogram earlier transmission data inside the transmit data buffer.

In Master mode, the SPBSY flag indicates the shift register status. This flag switches to 1 when transmission data is loaded from the transmit data buffer to the shift register and reverts to 0 once the data is sent. Read this flag to check whether the SPI module is operating or at standby.
The Slave mode SPBSY flag indicates the SPI slave selection signal (#SPISS pin) status. The flag has the value 1 when the SPI module is selected in Slave mode and the value 0 when the module is not selected.

Data receipt control

In Master mode, dummy data is written to the SPI_TXD register (0x4322). Writing to the SPI_TXD register creates the trigger for receipt as well as transmission start. Writing actual transmission data enables simultaneous transfers.
This starts the SPI clock output from SPICLK.

In Slave mode, the module waits until the clock is input from SPICLK. Slave mode involves only data receipt.

There is no need to program to the SPI_TXD register if no transmission is required. The receiving operation is started by clock input from the master device. If data is transferred simultaneously, the transmission data is written to the SPI_TXD register before the clock is input.

The data is contained in sequence in the shift register at the rising or falling edge for the clock determined by CPHA (D3/SPI_CTL register) and CPOL (D2/SPI_CTL register). (See Figure 19.4.1.)
The received data is loaded into the receive data buffer once the 8 bits of data are received in the shift register.

Received data in the buffer can be read from the SPI_RXD register (0x4324).

* **SPI_RXD**: SPI Receive Data Register (0x4324)

The SPI module includes an SPRBF flag (D1/SPI_ST register) for receipt control.

* **SPRBF**: Receive Data Buffer Full Flag in the SPI Status (SPI_ST) Register (D1/0x4320)

The SPRBF flag indicates the receive data buffer status. This flag is set to 1 when the data received in the shift register is loaded into the receive data buffer, indicating that the receive data can be read out. It reverts to 0 when the buffer data is read out from the SPI_RXD register. An interrupt can be generated as soon as the flag is set to 1 (see Section 19.6). The received data should be read out either by using this interrupt or by inspecting the SPRBF flag to confirm that the receive data buffer contains valid receive data. The receive data buffer is 1 byte in size, but a shift register is also provided, enabling received data to be retained in the buffer even while the subsequent data is being received. Note that the receive data buffer should be read out before receiving the subsequent data is complete. If receiving the subsequent data is complete before the receive data buffer contents are read out, the newly received data will overprogram the previous received data in the buffer.

In Master mode, the SPBSY flag indicating the shift register state can be used in the same way while transferring data.

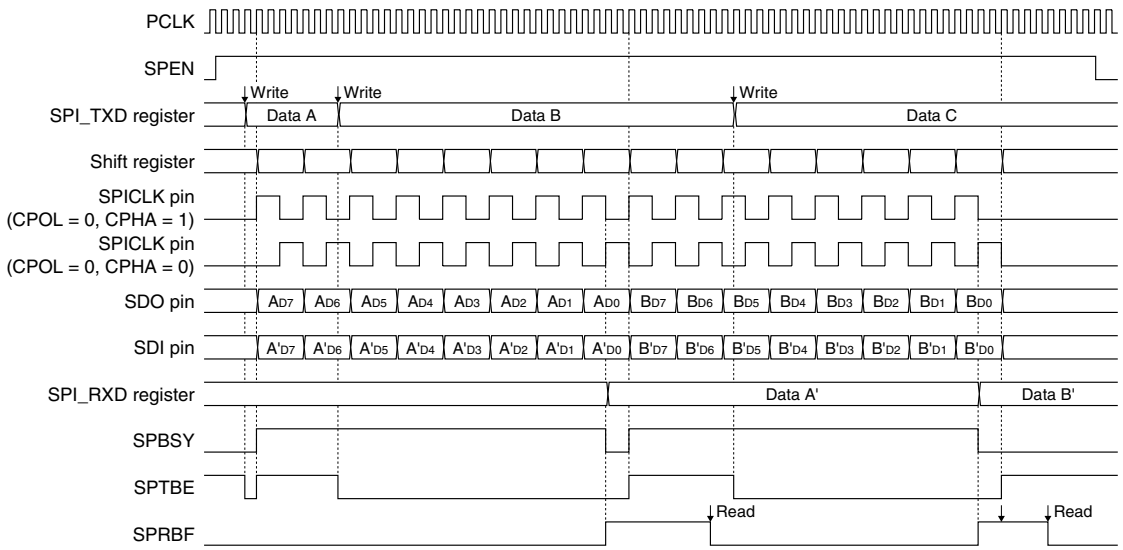


Figure 19.5.1: Data transmission/receiving timing chart (MSB first)

Blocking data transfers

After a data transfer is completed (both transmission and reception), data transfers are blocked by writing 0 to the SPEN bit. Confirm that the SPTBE flag is 1 and the SPBSY flag is 0 before blocking data transfer.
The data being transferred cannot be guaranteed if SPEN is set to 0 while data is being sent or received.

19.6 SPI Interrupts

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

The SPI module outputs one interrupt signal shared by the three above interrupt factor types to the interrupt controller (ITC). Inspect the status flag to determine the interrupt factor occurring.

Transmit buffer empty interrupt

To use this interrupt, set SPTIE (D4/SPI_CTL register) to 1. If SPTIE is set to 0 (default), interrupt requests for this factor will not be sent to the ITC.

∗ **SPTIE**: Transmit Data Buffer Empty Interrupt Enable Bit in the SPI Control (SPI_CTL) Register (D4/0x4326)

When transmission data written to the transmit data buffer is transferred to the shift register, the SPI module sets the SPTBE bit (D0/SPI_ST register) to 1, indicating that the transmit data buffer is empty. If transmit buffer empty interrupts are permitted (SPTIE = 1), an interrupt request pulse is sent simultaneously to the ITC.

∗ **SPTBE**: Transmit Data Buffer Empty Flag in the SPI Status (SPI_ST) Register (D0/0x4320)

An interrupt occurs if other interrupt conditions are met.
You can inspect the SPTBE flag in the SPI interrupt handler routine to determine whether the SPI interrupt is attributable to a transmit buffer empty. If SPTBE is 1, the next transmission data can be written to the transmit data buffer by the interrupt handler routine.

Receive buffer full interrupt

To use this interrupt, set SPRIE (D5/SPI_CTL register) to 1. If SPRIE is set to 0 (default), interrupt requests for this factor will not be sent to the ITC.

∗ **SPRIE**: Receive Data Buffer Full Interrupt Enable Bit in the SPI Control (SPI_CTL) Register (D5/0x4326)

When data received in the shift register is loaded into the receive data buffer, the SPI module sets the SPRBF bit (D1/SPI_ST register) to 1, indicating that the receive data buffer contains readable received data. If receive buffer full interrupts are permitted (SPRIE = 1), an interrupt request pulse is output to the ITC at the same time.

∗ **SPRBF**: Receive Data Buffer Full Flag in the SPI Status (SPI_ST) Register (D1/0x4320)

An interrupt occurs if other interrupt conditions are met.
You can inspect the SPRBF flag in the SPI interrupt 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.

Interrupt vectors

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

Vector number: 18 (0x12)
Vector address: TTBR + 0x48

Other interrupt settings

The SPI interrupt priority can be set for the ITC between level 0 (default) and level 7. The PSR (S1C17 core internal processor status register) IE (interrupt enable) bit must be set to 1 to generate actual interrupts.
For specific information on interrupt processing, refer to “6 Interrupt Controller (ITC).”

19.7 Control Register Details

Table 19.7.1: SPI register list

Address	Register name		Function
0x4320	SPI_ST	SPI Status Register	Transfer, buffer status display
0x4322	SPI_TXD	SPI Transmit Data Register	Transmission data
0x4324	SPI_RXD	SPI Receive Data Register	Received data
0x4326	SPI_CTL	SPI Control Register	SPI mode and data transfer permission setting

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

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

0x4320: SPI Status Register (SPI_ST)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
SPI Status Register (SPI_ST)	0x4320 (16 bits)	D15–3	–	reserved	–				–	–	0 when being read.
		D2	SPBSY	Transfer busy flag (master)	1	Busy	0	Idle	0	R	
				ss signal low flag (slave)	1	ss = L	0	ss = H			
		D1	SPRBF	Receive data buffer full flag	1	Full	0	Not full	0	R	
		D0	SPTBE	Transmit data buffer empty flag	1	Empty	0	Not empty	1	R	

D[15:3] Reserved

D2 SPBSY: Transfer Busy Flag (Master Mode)/ss Signal Low Flag (Slave Mode)

Master mode

Indicates the SPI transfer status.

1 (R): Operating

0 (R): Standby (default)

SPBSY is set to 1 when the SPI starts data transfer in Master mode and is maintained at 1 while transfer is underway.

It is cleared to 0 once the transfer is complete.

Slave mode

Indicates the slave selection (#SPISS) signal status.

1 (R): Low level (this SPI is selected)

0 (R): High level (this SPI is not selected) (default)

SPBSY is set to 1 when the master device sets the #SPISS signal to active to select this SPI module (slave device). It is returned to 0 when the master device clears the SPI module selection by returning the #SPISS signal to inactive.

D1 SPRBF: Receive Data Buffer Full Flag

Indicates the receive data buffer status.

1 (R): Data full

0 (R): No data (default)

SPRBF is set to 1 when data received in the shift register is sent to the receive data buffer (when receiving is complete), indicating that the data can be read. It reverts to 0 once the buffer data is read from the SPI_RXD register (0x4324).

D0 SPTBE: Transmit Data Buffer Empty Flag

Indicates the state of the transmit data buffer.

1 (R): Empty (default)

0 (R): Data exists

SPTBE is set to 0 when transmit data is written to the SPI_TXD register (transmit data buffer, 0x4322), and is set to 1 when the data is transferred to the shift register (when transmission starts).

Transmission data is written to the SPI_TXD register when this bit is 1.

0x4322: SPI Transmit Data Register (SPI_TXD)

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

D[15:8] Reserved

D[7:0] SPTDB[7:0]: SPI Transmit Data Buffer Bits

Set the transmission data to be written to the transmit data buffer. (Default: 0x0)

In Master mode, transmission is started by writing data to this register. In Slave mode, the contents of this register are sent to the shift register and transmission begins when the clock is input from the master.

SPTBE (D0/SPI_ST register) is set to 1 (empty) as soon as data written to this register has been transferred to the shift register. A transmit buffer empty interrupt is generated at the same time. The subsequent transmit data can then be written, even while data is being transmitted.

Serial converted data is output from the SDO pin with MSB leading, with the bit set to 1 as High level and the bit set to 0 as Low level.

Note: Make sure that SPEN (D0/SPI_CTL register) is set to 1 before writing data to the SPI_TXD register to start data transmission/reception.

0x4324: SPI Receive Data Register (SPI_RXD)

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

D[15:8] Reserved

D[7:0] SPRDB[7:0]: SPI Receive Data Buffer Bits

Contain the received data. (Default: 0x0)

SPRBF (D1/SPI_ST register) is set to 1 (data full) as soon as data is received and the shift register data has been transferred to the receive data buffer. A receive buffer full interrupt is generated at the same time. Data can then be read until subsequent data is received. If receiving the subsequent data is complete before the register has been read out, the new received data overprograms the contents.

Serial data input from the SDI pin with MSB leading is converted to parallel, with the High level bit set to 1 and the Low level bit set to 0. The data is the loaded into this register.

This register is read-only.

0x4326: SPI Control Register (SPI_CTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SPI Control Register (SPI_CTL)	0x4326 (16 bits)	D15–10	–	reserved	–	–	–	0 when being read.
		D9	MCLK	SPI clock source select	1 T16 Ch.1 0 PCLK-1/4	0	R/W	
		D8	MLSB	LSB/MSB first mode select	1 LSB 0 MSB	0	R/W	
		D7–6	–	reserved	–	–	–	0 when being read.
		D5	SPRIE	Receive data buffer full int. enable	1 Enable 0 Disable	0	R/W	
		D4	SPTIE	Transmit data buffer empty int. enable	1 Enable 0 Disable	0	R/W	
		D3	CPHA	Clock phase select	1 Data out 0 Data in	0	R/W	These bits must be set before setting SPEN to 1.
		D2	CPOL	Clock polarity select	1 Active L 0 Active H	0	R/W	
		D1	MSSL	Master/slave mode select	1 Master 0 Slave	0	R/W	
		D0	SPEN	SPI enable	1 Enable 0 Disable	0	R/W	

D[15:10] Reserved

D9 MCLK: SPI Clock Source Select Bit

Selects the SPI clock source.
1 (R/W): 16-bit timer Ch.1
0 (R/W): PCLK-1/4 (default)

D8 MLSB: LSB/MSB First Mode Select Bit

Selects whether data is transferred with MSB first or LSB first.
1 (R/W): LSB first
0 (R/W): MSB first (default)

D[7:6] Reserved

D5 SPRIE: Receive Data Buffer Full Interrupt Enable Bit

Permits or prohibits receive data buffer full SPI interrupts.
1 (R/W): Permitted
0 (R/W): Prohibited (default)

Setting SPRIE to 1 permits the output of SPI interrupt requests to the ITC due to a receive data buffer full. These interrupt requests are generated when the data received in the shift register is transferred to the receive data buffer (when receipt is complete).
SPI interrupts are not generated by receive data buffer full if SPRIE is set to 0.

D4 SPTIE: Transmit Data Buffer Empty Interrupt Enable Bit

Permits or prohibits transmit data buffer empty SPI interrupts.
1 (R/W): Permitted
0 (R/W): Prohibited (default)

Setting SPTIE to 1 permits the output of SPI interrupt requests to the ITC due to a transmit data buffer empty. These interrupt requests are generated when the data written to the transmit data buffer is transferred to the shift register (when transmission starts).
SPI interrupts are not generated by transmit data buffer empty if SPTIE is set to 0.

D3 CPHA: SPI Clock Phase Select Bit

Selects the SPI clock phase. (Default: 0)
Sets the data transfer timing together with CPOL (D2). (See Figure 19.7.1.)

D2 CPOL: SPI Clock Polarity Select Bit

Selects the SPI clock polarity.
1 (R/W): Active Low
0 (R/W): Active High (default)

Sets the data transfer timing together with CPHA (D3). (See Figure 19.7.1.)

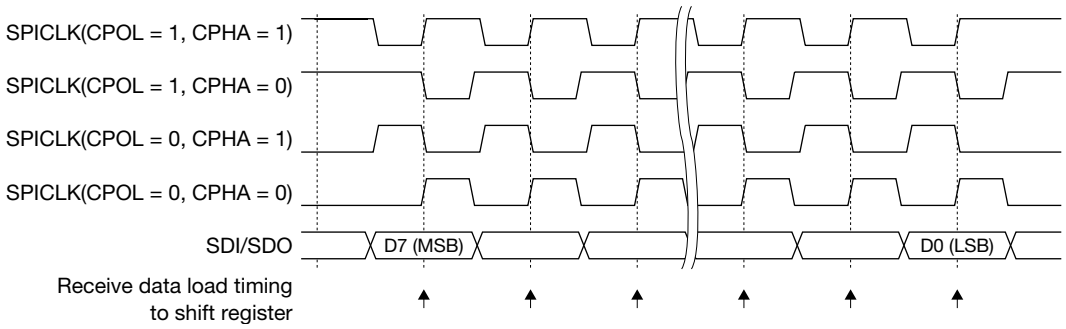


Figure 19.7.1: Clock and data transfer timing

D1 MSSL: Master/Slave Mode Select Bit

Sets the SPI module to Master or Slave mode.
1 (R/W): Master mode
0 (R/W): Slave mode (default)

Setting MSSL to 1 selects Master mode; setting it to 0 selects Slave mode. Master mode performs data transfer with the clock generated by the 16-bit timer Ch.1. In Slave mode, data is transferred by inputting the clock from the master device.

D0 SPEN: SPI Enable Bit

Permits or prohibits SPI module operation.
1 (R/W): Permitted
0 (R/W): Prohibited (default)

Setting SPEN to 1 starts the SPI module operation, enabling data transfer.
Setting SPEN to 0 stops the SPI module operation.

Note: The SPEN bit should be set to 0 before setting the CPHA, CPOL, and MSSL bits.

19.8 Precautions

- The duty ratio of the clock input via the SPICLK pin must be 50%.
- The SPI_CTL register (0x4326) must not be accessed while the SPBSY flag (D2/SPI_ST register) is 1 or the SPRBF flag (D1/SPI_ST register) is 1.

  * **SPBSY**: Transfer Busy Flag in the SPI Status (SPI_ST) Register (D2/0x4320)
  * **SPRBF**: Receive Data Buffer Full Flag in the SPI Status (SPI_ST) Register (D1/0x4320)

- When the SPI module is used in master mode with CPHA (D3/SPI_CTL register) set to 0, the clock may change a minimum of one system clock (PCLK) cycle time from change of the first transmit data bit.

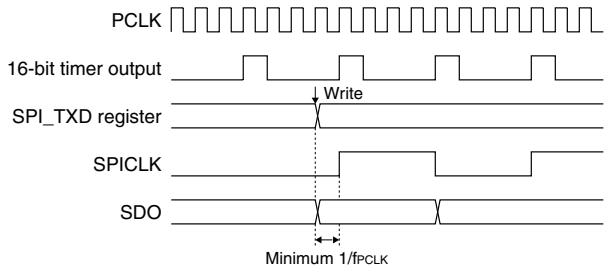


Figure 19.8.1: SDO and SPICLK Change Timings when CPHA = 0

- The half SPICLK 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.
- Make sure that SPEN (D0/SPI_CTL register) is set to 1 before writing data to the SPI_TXD register to start data transmission/reception.

  * **SPEN**: SPI Enable Bit in the SPI Control (SPI_CTL) Register (D0/0x4326)

- The SPEN bit should be set to 0 before setting the CPHA (D3/SPI_CTL register), CPOL (D2/SPI_CTL register), and MSSL (D1/SPI_CTL register) bits.

  * **CPHA**: Clock Phase Select Bit in the SPI Control (SPI_CTL) Register (D3/0x4326)
  * **CPOL**: Clock Polarity Select Bit in the SPI Control (SPI_CTL) Register (D2/0x4326)
  * **MSSL**: Master/Slave Mode Select Bit in the SPI Control (SPI_CTL) Register (D1/0x4326)

20 I²C

20.1 I²C Configuration

The S1C17702 incorporates an I²C bus interface module for high-speed synchronized serial communications. The I²C module operates as a master device (as single master only) using the clock fed from the 16-bit timer Ch.2. It supports standard (100 kbps) and fast (400 kbps) modes as well as 7-bit/10-bit slave address mode. It incorporates a noise filter function to help improve the reliability of data transfers.
This module is capable of generating two different types of interrupts (transmit buffer empty and receive buffer full interrupts) for easy and continuous processing of serial data transfers with interrupts.
Figure 20.1.1 shows the I²C module configuration.

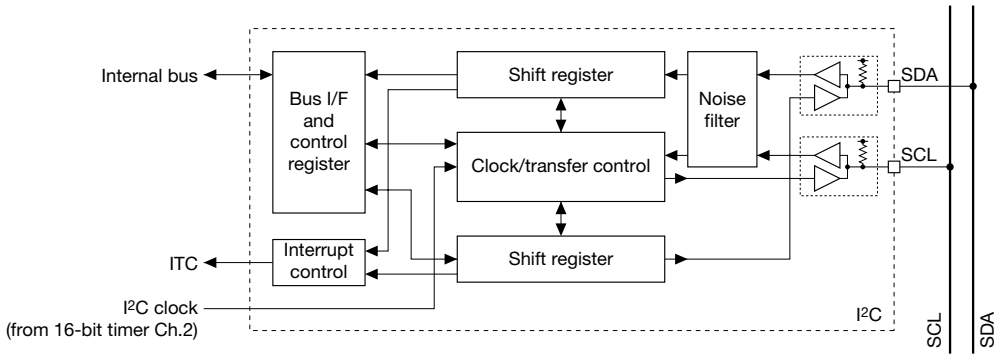


Figure 20.1.1: I²C module configuration

20.2 I²C Input/Output Pins

Table 20.2.1 lists the I²C pins.

Table 20.2.1: I²C pin list

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

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

P14 → SDA

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

P15 → SCL

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

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

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

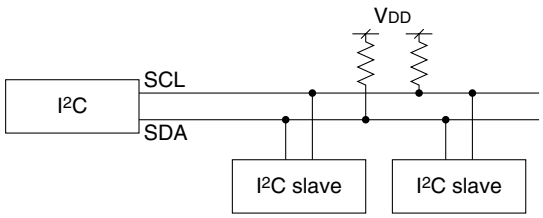


Figure 20.2.1: I²C connection example

20.3 I²C Clock

The I²C module uses the internal clock output by the 16-bit timer Ch.2 as the synchronizing clock. This clock is output from the SCL pin to the slave device while also driving the shift register. The clock should be programmed to output a signal matching the transfer rate from the 16-bit timer Ch.2. For more information on 16-bit timer control, refer to "11 16-bit Timer (T16)."

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

20.4 Settings Before Data Transfer

The I²C module includes an optional noise filter function that can be selected via the application program.

Noise filter function

The I²C module incorporates a function for filtering noise from the SDA and SCL pin input signals. This function is enabled by setting NSERM (D4/I2C_CTL register) to 1.
Note that using this function requires setting the I²C clock (16-bit timer Ch.2 output clock) frequency to 1/6 or less of PCLK.

* **NSERM**: Noise Remove On/Off Bit in the I²C Control (I2C_CTL) Register (D4/0x4342)

20.5 Data Transfer Control

Make the following settings before starting data transfers.
(1) Set the 16-bit timer Ch.2 to output the I²C clock. (See Section 11.)
(2) Select the option function. (See section 20.4.)
(3) Set the interrupt conditions to use I²C interrupts. (See Section 20.6.)

Note: Make sure the I²C module is halted (when I2CEN/I2C_EN register = 0) before changing the above settings.

∗ **I2CEN**: I²C Enable Bit in the I²C Enable (I2C_EN) Register (D0/0x4340)

Permitting data transfers

Set the I2CEN (D0/I2C_EN register) to 1 to permit I²C operations. This enables I²C transfers and permits clock input/output.

Note: Do not set I2CEN to 0 when the I²C module is transferring data.

Data transfer start

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

(1) Generate start condition

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

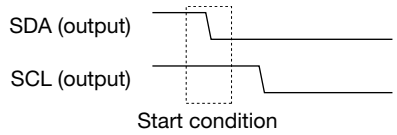


Figure 20.5.1: Start condition

The start condition is generated by setting STRT (D0/I2C_CTL register) to 1.

∗ **STRT**: Start Control Bit in the I²C Control (I2C_CTL) Register (D0/0x4342)

STRT is automatically reset to 0 once the start condition is generated. The I²C bus is busy from this point on.

(2) Slave address transmission

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

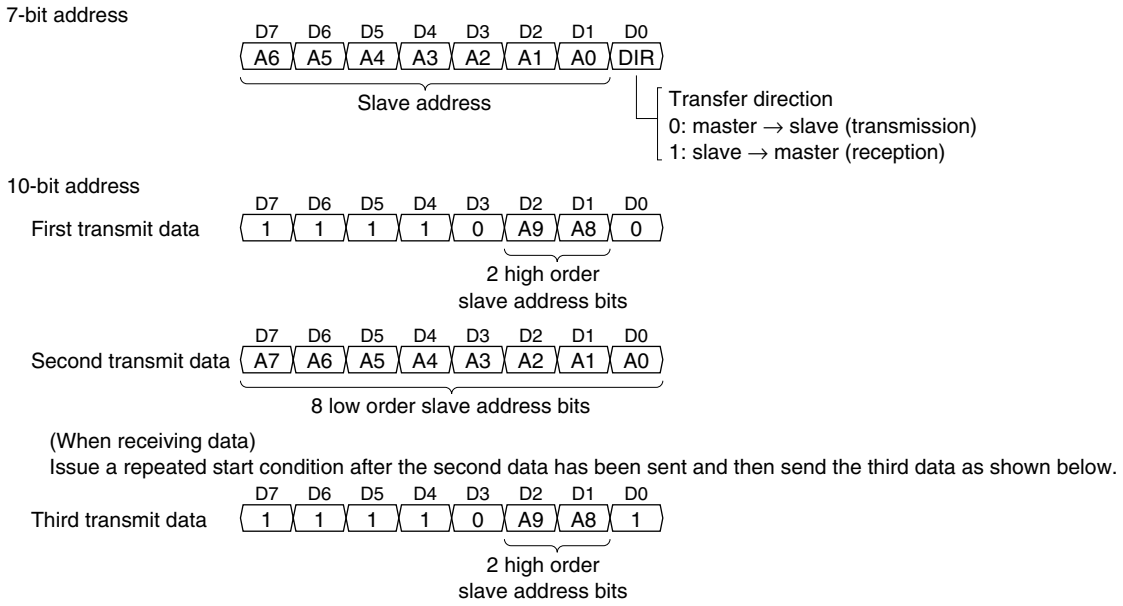


Figure 20.5.2: Slave address and transmission data specifying transfer direction

Transfer direction indicates the data transfer direction after the slave address. This is set to 0 when sending data from the master to the slave and to 1 when receiving data from the slave.
To send a slave address, set the transmission address to RTDT[7:0] (D[7:0]/I2C_DAT register). At the same time, set the TXE (D9/I2C_DAT register) transmitting the address to 1.

- **RTDT[7:0]**: Receive/Transmit Data Bits in the I²C Data (I2C_DAT) Register (D[7:0]/0x4344)
- **TXE**: Transmit Execution Bit in the I²C Data (I2C_DAT) Register (D9/0x4344)

After the slave address has been output, data can be sent and received as many times as required. Data must be sent or received according to the transfer direction set together with the slave address.

Data transmission control

The procedure for transmitting data is described below. Data transmission is performed by the same procedure as for slave address transmission.
To send byte data, set the transmission data to RTDT[7:0] (D[7:0]/I2C_DAT register). Set TXE (D9/I2C_DAT register) to 1 to transmit 1 byte.

When TXE is set to 1, the I²C module begins data transmission in sync with the clock. If the previous data is currently being transmitted, data transmission starts after this has been completed.
The I²C module first transfers the data written to the shift register, then starts outputting the clock from SCL. Resetting TXE to 0 at this point generates an interrupt, enabling the subsequent transmission data and TXE to be reset.
The data bits in the shift register are shifted in sequence at the clock falling edge and output via the SDA pin with the MSB leading.
The I²C module outputs 9 clocks with each data transmission. In the 9th clock cycle, an ACK or NAK is received from the slave device with the SDA signal as high impedance.
The slave device returns ACK(0) to the master if the data is received. If the data is not received, SDA is not pulled down, which the I²C module interprets to mean an NAK(1) (transmission failed).

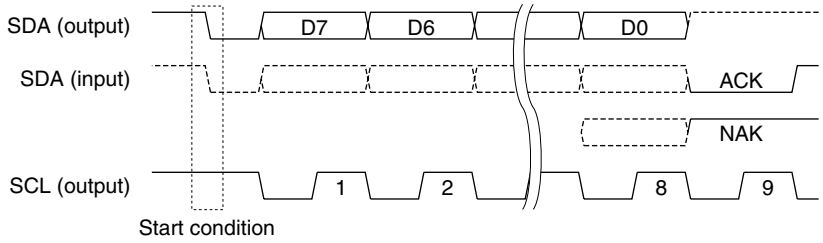


Figure 20.5.3: ACK and NAK

The I²C module includes two status bits, TBUSY (D8/I2C_CTL register) and RTACK (D8/I2C_DAT register), for transmission control.

- **TBUSY**: Transmit Busy Flag in the I²C Control (I2C_CTL) Register (D8/0x4342)
- **RTACK**: Receive/Transmit ACK Bit in the I²C Data (I2C_DAT) Register (D8/0x4344)

The TBUSY flag indicates the data transmission status. This flag becomes 1 when transmission starts (including slave address transmission) and reverts to 0 once data transmission ends.
Inspect the flag to check whether the I²C module is currently transmitting or at standby.
The RTACK bit indicates whether or not the slave device returned an ACK for the previous transmission. RTACK is 0 if an ACK was returned and 1 if ACK was not returned.

Data receipt control

The procedure for receiving data is described below. To receive data, the slave address must be sent with the transfer direction bit set to 1.
To receive data, set RXE (D10/I2C_DAT register) to 1 for receiving 1 byte.
TXE (D9/I2C_DAT register) is set to 1 when sending the slave address, but RXE can also be set to 1 at the same time. If both TXE and RXE are set to 1, TXE takes priority.

- **RXE**: Receive Execution Bit in the I²C Data (I2C_DAT) Register (D10/0x4344)

When the RXE bit is set to 1, allowing receiving to start, the I²C module starts outputting the clock from the SCL 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 receive data is loaded to RTDT[7:0] once the 8-bit data has been received in the shift register. The I²C module includes two status bits for receive control: RBRDY (D11/I2C_DAT register) and RBUSY (D9/ I2C_CTL register).

- **RBRDY**: Receive Buffer Ready Bit in the I²C Data (I2C_DAT) Register (D11/0x4344)
- **RBUSY**: Receive Busy Flag in the I²C Control (I2C_CTL) Register (D9/0x4342)

The RBRDY flag indicates the receive data status. This flag becomes 1 when the data received in the shift register is loaded to RTDT[7:0] and reverts to 0 when the receive data is read out from RTDT[7:0]. Interrupts can also be generated once the flag value becomes 1.
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 I²C module is currently receiving or in standby.
To wait for reception using polling, follow the procedures given below using the RBUSY flag.
Interrupts to the CPU are prohibited because polling accurately determines the two state transitions 3 and 4.

1. Prohibits CPU interrupts using di command.
2. Writes 1 to RXE to prepare for receiving.
3. Waits for RBUSY to become 1 (reception start).
4. Waits for RBUSY to become 0 (reception end).
5. Reads out RTDT (received data).
6. Returns to CPU interrupt permitted state using ei command.

The I²C module outputs 9 clocks with each data receipt. In the 9th clock cycle, an ACK or NAK is sent to the slave. The bit state sent can be set in RTACK (D8/I2C_DAT register). To send ACK, set RTACK to 0. To send NAK, set RTACK to 1.

Data transfer end (Stop condition generation)

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

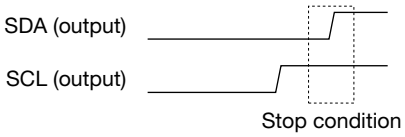


Figure 20.5.4: Stop condition

The stop condition is generated by setting STP (D1/I2C_CTL register) to 1.

∗ **STP**: Stop Control Bit in the I²C Control (I2C_CTL) Register (D1/0x4342)

When STP is set to 1, the I²C module switches the SDA line from Low to High and generates a stop condition while maintaining the I²C bus SCL line at High. The I²C bus subsequently switches to free state.
Before STP can be set to 1, confirm that TBUSY or RBUSY is reset to 0 from 1 (this indicates that the I²C 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 (Repeated start condition generation)

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

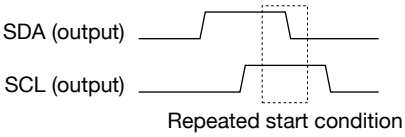


Figure 20.5.5: Repeated start condition

The repeated start condition is generated by setting STRT (D0/I2C_CTL register) to 1 when the I²C bus is busy.

∗ **STRT**: Start Control Bit in the I2C Control (I2C_CTL) Register (D0/0x4342)

STRT is automatically reset to 0 once the repeated start condition is generated. Slave address transmission is subsequently possible with the I²C bus remaining in the busy state.

Prohibiting data transfer

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

Timing chart

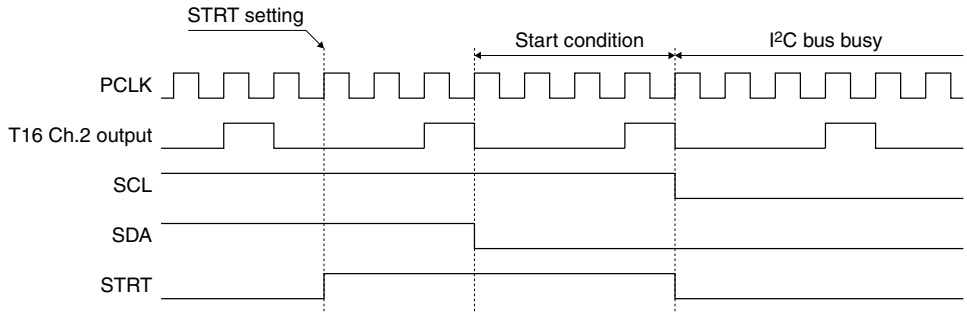


Figure 20.5.6: Start condition generation

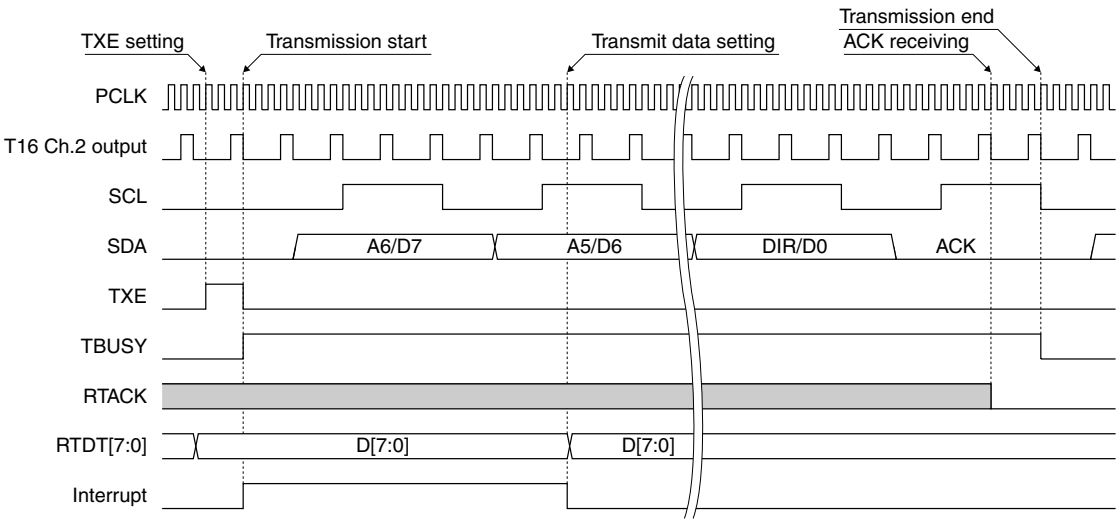


Figure 20.5.7: Slave address transmission/data transmission

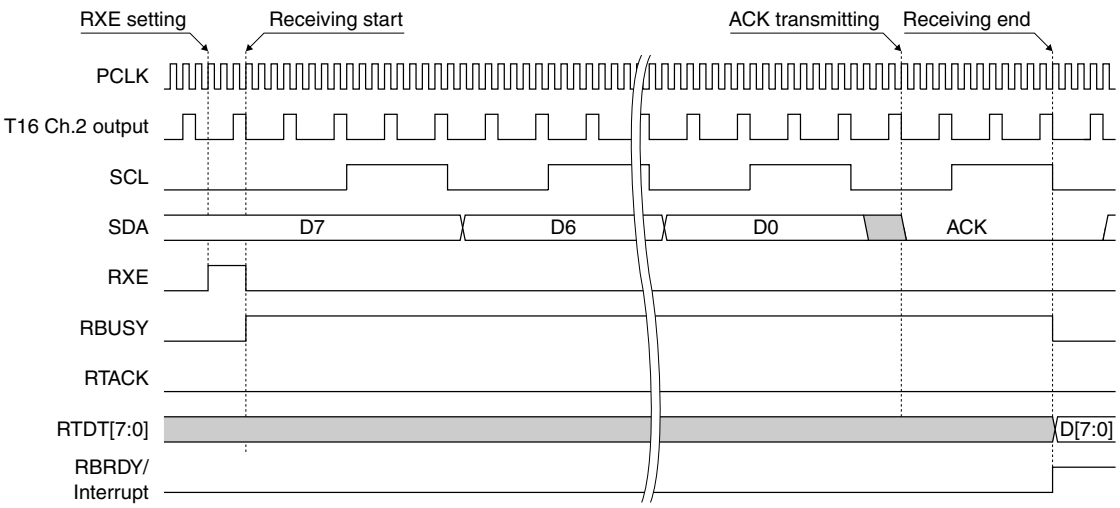


Figure 20.5.8: Data receiving

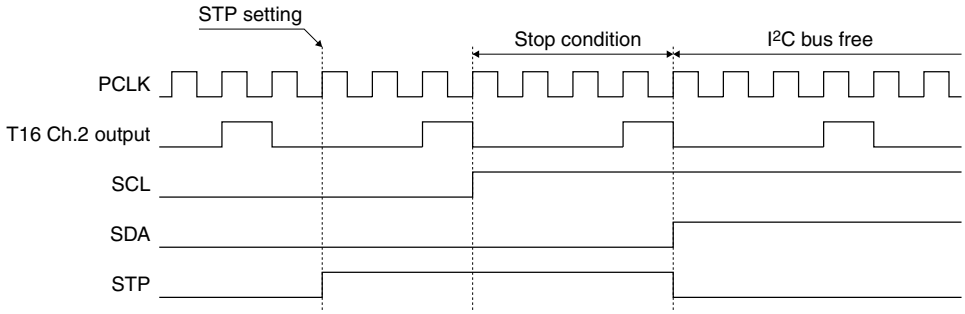


Figure 20.5.9: Stop condition generation

20.6 I²C Interrupts

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

The I²C module outputs one interrupt signal shared by the two above interrupt factor types to the interrupt controller (ITC).

Transmit buffer empty interrupt

To use this interrupt, set TINTE (D0/I2C_ICTL register) to 1. If TINTE is set to 0 (default), interrupt requests for this factor will not be sent to the ITC.

∗ **TINTE**: Transmit Interrupt Enable Bit in the I²C Interrupt Control (I2C_ICTL) Register (D0/0x4346)

If transmit buffer empty interrupts are permitted (TINTE = 1), an interrupt request pulse is output to the ITC as soon as the transmit data set in RTDT[7:0] (D[7:0]/I2C_DAT register) is transferred to the shift register.

∗ **RTDT[7:0]**: Receive/Transmit Data Bits in the I²C Data (I2C_DAT) Register (D[7:0]/0x4344)

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]. If TXE (D9/I2C_DAT register) is set to 0 at the same time, the I²C module only clears the cause of interrupt without sending the data written.

∗ **TXE**: Transmit Execution Bit in the I²C Data (I2C_DAT) Register (D9/0x4344)

Receive buffer full interrupt

To use this interrupt, set RINTE (D1/I2C_ICTL register) to 1. If RINTE is set to 0 (default), interrupt requests for this factor will not be sent to the ITC.

∗ **RINTE**: Receive Interrupt Enable Bit in the I²C Interrupt Control (I2C_ICTL) Register (D1/0x4346)

If receive buffer full interrupts are permitted (RINTE = 1), an interrupt request pulse is output to the ITC as soon as the data received in the shift register is loaded to RTDT[7:0].

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].

Note: After an I²C 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.

Interrupt vectors

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

Vector number: 19 (0x13)
Vector address: TTBR + 0x4c

Other interrupt settings

The ITC allows the priority of I²C interrupts to be set between level 0 (the default value) and level 7. To generate actual interrupts, the PSR (S1C17 core internal processor status register) IE (interrupt enable) bit must be set to 1.
For more information on interrupt processing, see “6 Interrupt Controller (ITC).”

20.7 Control Register Details

Table 20.7.1: I²C register list

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

The I²C module registers are described in detail below. These are 16-bit registers.

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

0x4340: I²C Enable Register (I2C_EN)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
I²C Enable Register (I2C_EN)	**0x4340** (16 bits)	D15–1	**–**	reserved	–				–	–	0 when being read.
		D0	**I2CEN**	I²C enable	1	Enable	0	Disable	0	R/W	

D[15:1] Reserved

D0 I2CEN: I²C Enable Bit

Permits or prohibits I²C module operation.
1 (R/W): Permitted
0 (R/W): Prohibited (default)

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

0x4342: I²C Control Register (I2C_CTL)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
I²C Control Register (I2C_CTL)	0x4342 (16 bits)	D15–10	–	reserved	–	–	–	0 when being read.
		D9	RBUSY	Receive busy flag	1 Busy 0 Idle	0	R	
		D8	TBUSY	Transmit busy flag	1 Busy 0 Idle	0	R	
		D7–5	–	reserved	–	–	–	0 when being read.
		D4	NSERM	Noise remove on/off	1 On 0 Off	0	R/W	
		D3–2	–	reserved	–	–	–	0 when being read.
		D1	STP	Stop control	1 Stop 0 Ignored	0	R/W	
		D0	STRT	Start control	1 Start 0 Ignored	0	R/W	

D[15:10] Reserved

D9 RBUSY: Receive Busy Flag

Indicates the I²C receipt status.

1 (R): Operating

0 (R): Standby (default)

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

D8 TBUSY: Transmit Busy Flag

Indicates the I²C transmission status.

1 (R): Operating

0 (R): Standby (default)

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

D[7:5] Reserved

D4 NSERM: Noise Remove On/Off Bit

Turns the noise filter function on or off.

1 (R/W): On

0 (R/W): Off (default)

The I²C module incorporates a function for filtering noise from the SDA and SCL pin input signals. This function is enabled by setting NSERM to 1.

Note that using this function requires setting the I²C clock (16-bit timer Ch.2 output clock) frequency to 1/6 or less of PCLK.

D[3:2] Reserved

D1 STP: Stop Control Bit

Generates the stop condition.

1 (R/W): Stop condition generated

0 (R/W): Disabled (default)

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

Before STP can be set to 1, confirm that TBUSY or RBUSY is reset to 0 from 1 (this indicates that the I²C 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 automatically reset to 0 if the stop condition is generated.

D0 **STRT: Start Control Bit**

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

With STRT set at 1, the I²C module generates the start condition by changing the SDA line to Low while maintaining the I²C bus SCL line at High.
The 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.

0x4344: I²C Data Register (I2C_DAT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
I²C Data Register (I2C_DAT)	0x4344 (16 bits)	D15–12	–	reserved	–	–	–	0 when being read.
		D11	RBRDY	Receive buffer ready	1 Ready 0 Empty	0	R	
		D10	RXE	Receive execution	1 Receive 0 Ignored	0	R/W	
		D9	TXE	Transmit execution	1 Transmit 0 Ignored	0	R/W	
		D8	RTACK	Receive/transmit ACK	1 Error 0 ACK	0	R/W	
		D7–0	RTDT[7:0]	Receive/transmit data RTDT7 = MSB RTDT0 = LSB	0x0 to 0xff	0x0	R/W	

D[15:12] Reserved

D11 RBRDY: Receive Buffer Ready Flag

Indicates the receive buffer status.

1 (R): Receive data exists

0 (R): No receive data (default)

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

Note: Use the RBUSY flag when awaiting reception using polling. The RBRDY flag cannot be used to await reception with polling. For more information on awaiting reception control procedures using polling, refer to "Data receipt control" in "20.5 Data Transfer Control."

D10 RXE: Receive Execution Bit

Receives 1 byte of data.

1 (R/W): Data receipt start

0 (R/W): Disabled (default)

Setting RXE to 1 and TXE (D9) to 0 starts receiving for 1 byte of data. RXE can be set to 1 for subsequent receipt, even if the slave address is being sent or data is being received. RXE is reset to 0 as soon as 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): Disabled (default)

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

D8 RTACK: Receive/Transmit ACK Bit

When transmitting data

Indicates the response bit status.

1 (R/W): Error (NAK)

0 (R/W): ACK (default)

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

When receiving data

Sets the response bit sent to the slave.

1 (R/W): Error (NAK)

0 (R/W): ACK (default)

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

To return an NAK, set RTACK to 1.

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

When sending data

Set the transmission data. (Default: 0x0)

Data transmission is started by setting TXE (D9) to 1. If a slave address or data is currently being transmitted, transmission begins once the previous transmission is completed. Serial converted data is output from the SDA pin with MSB leading and bits set to 0 as Low level.

A transmit buffer empty interrupt factor is generated as soon as the data written to this register is transferred to the shift register, after which the subsequent transmission data can be written.

When receiving data

Read the receive data. (Default: 0x0)

Data receipt is started by setting RXE (D10) to 1. If a slave address is currently being transmitted or data is currently being received, the new receipt starts once the previous data has been transferred. The RBRDY flag (D11) is set and a receive buffer full interrupt factor generated as soon as receipt is complete and the shift register data is transferred to this register. Data can then be read until the subsequent data has been received. If the subsequent data is received before this register is read out, the contents are overwritten by the most recent received data.

Serial data input from the SDA pin with MSB leading is converted to parallel, with the High level bit set to 1 and the Low level bit set to 0, then loaded to this register.

0x4346: I²C Interrupt Control Register (I2C_ICTL)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
I²C Interrupt Control Register (I2C_ICTL)	0x4346 (16 bits)	D15–2	–	reserved	–				–	–	0 when being read.
		D1	RINTE	Receive interrupt enable	1	Enable	0	Disable	0	R/W	
		D0	TINTE	Transmit interrupt enable	1	Enable	0	Disable	0	R/W	

D[15:2] Reserved

D1 RINTE: Receive Interrupt Enable Bit

Permits or prohibits receive buffer full I²C interrupts.
1 (R/W): Permitted
0 (R/W): Prohibited (default)

Setting RINTE to 1 permits the output of I²C interrupt requests to the ITC due to a receive data buffer full. These interrupt requests are generated when the data received in the shift register is transferred to RTDT[7:0] (D[7:0]/I2C_DAT register) (when receipt is complete).
I²C interrupts are not generated by receive data buffer full if RINTE is set to 0.

D0 TINTE: Transmit Interrupt Enable Bit

Permits or prohibits transmit buffer empty I²C interrupts.
1 (R/W): Permitted
0 (R/W): Prohibited (default)

Setting TINTE to 1 permits the output of I²C interrupt requests to the ITC due to a transmit buffer empty. These interrupt requests are generated when the data written to RTDT[7:0] (D[7:0]/I2C_DAT register) is transferred to the shift register.
I²C interrupts are not generated by transmit buffer empty if TINTE is set to 0.

21 Remote Controller (REMC)

21.1 REMC Configuration

The S1C17702 incorporates a remote controller (REMC) module for generating infrared remote control communication signals. The REMC module consists of a carrier generation circuit for generating a carrier signal using the prescaler output clock, an 8-bit down-counter for counting the transferred data length, a modulation circuit for generating transmission data of the specified carrier length, and an edge detection circuit for detecting input signal rising and falling edges.
The module is also capable of generating counter underflow interrupts indicating that the specified data length has been transmitted and input rising/falling edge detection interrupts for data receipt processing.
Figure 21.1.1 shows the REMC module configuration.

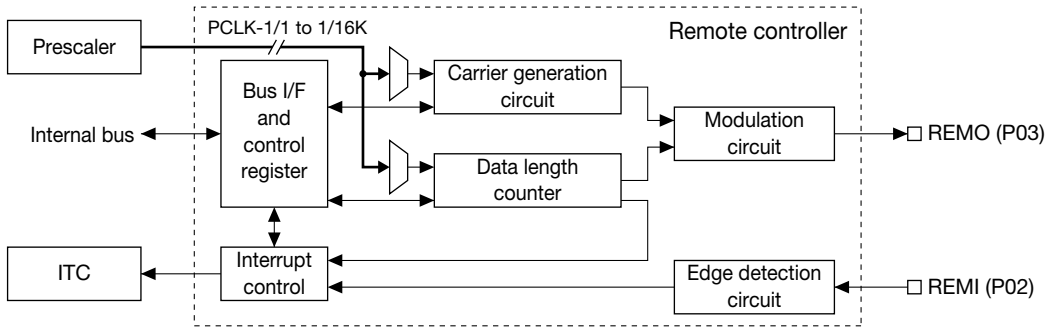


Figure 21.1.1: REMC module configuration

21.2 REMC Input/output Pin

Table 21.2.1 lists the REMC input/output pins.

Table 21.2.1: REMC input/output pin list

Pin name	I/O	Qty	Function
REMI (P02)	I	1	Remote control transmit data input pin Inputs receive data.
REMO (P03)	O	1	Remote control transmit data output pin Outputs modulated remote control transmit data.

The REMC module input/output pins (REMI, REMO) are shared with general purpose input/output port pins (P02, P03) and are initially set as general purpose input/output port pins. The function must be switched using the P0_PMUX register setting to use general purpose input/output port pins as REMC input/output pins. Switch the pins to REMC input/output by setting the following control bits to 1.

P02 → REMI

* **P02MUX**: P02 Port Function Select Bit in the P0 Port Function Select (P0_PMUX) Register (D2/0x52a0)

P03 → REMO

* **P03MUX**: P03 Port Function Select Bit in the P0 Port Function Select (P0_PMUX) Register (D3/0x52a0)

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

21.3 Carrier Generation

The REMC module incorporates a carrier generation circuit that generates a carrier signal for transmission in accordance with the clock set by the software and carrier H and L section lengths.

The prescaler output clock is used for the carrier signal generation clock. The prescaler generates 15 different clocks, dividing the PCLK clock from 1/1 to 1/16K. One is selected by CGCLK[3:0] (D[15:12]/REMC_PSC register).

* **CGCLK[3:0]**: Carrier Generator Clock Select Bits in the REMC Prescaler Clock Select (REMC_PSC) Register (D[15:12]/0x5340)

Table 21.3.1: Carrier generation clock selection

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

(Default: 0x0)

For more information on prescaler control, refer to "9 Prescaler (PSC)."

Note: The prescaler must run before the REMC module.

The carrier H and L section lengths are set by REMCH[5:0] (D[5:0]/REMC_CARH register) and REMCL[5:0] (D[13:8]/REMC_CARL register), respectively. These registers set a value corresponding to the number of clock cycles selected above + 1.

* **REMCH[5:0]**: H Carrier Length Setup Bits in the REMC H Carrier Length Setup (REMC_CARH) Register (D[5:0]/0x5342)
* **REMCL[5:0]**: L Carrier Length Setup Bits in the REMC L Carrier Length Setup (REMC_CARL) Register (D[13:8]/0x5342)

The carrier H and L section lengths can be calculated as follows:

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

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

REMCH: Carrier H section length register data value
REMCL: Carrier L section length register data value
clk_in: Prescaler output clock frequency

The carrier signal is generated from these settings as shown in Figure 21.3.1.

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

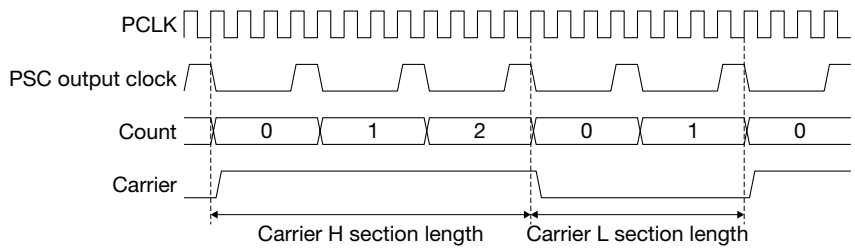


Figure 21.3.1: Carrier signal generation

21.4 Data Length Counter Clock Settings

The data length counter is an 8-bit counter for setting data lengths when transmitting data.
When a value corresponding to the data pulse width is written during data transmission, the data length counter begins counting down from that value, generating an underflow interrupt factor and halting when the counter reaches 0.
The subsequent transmit data is set using this interrupt.

This counter is also used for data receiving, enabling measurement of the receive data length. Interrupts can be generated at the input signal rising or falling edges when receiving data. The data pulse length can be obtained from the difference between data pulses by setting the data length counter to 0xff using the interrupt when the input changes and by reading out the count value when a subsequent interrupt occurs due to input changes.

This data length counter count clock also uses a prescaler output clock and can select one of 15 different types. The prescaler output clock is selected by the control bit LCCLK[3:0] (D[11:8]/REMC_CFG register) provided separately to the carrier generation clock.

* **LCCLK[3:0]**: Length Counter Clock Select Bits in the REMC Prescaler Clock Select (REMC_CFG) Register (D[11:8]/0x5340)

Table 21.4.1: Data length counter clock selection

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

(Default: 0x0)

The data length counter can count up to 256. The count clock should be selected to ensure that the data length fits within this range.

21.5 Data Transfer Control

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

Note: Make sure the REMC module is halted (when REMEN/REMC_CFG register = 0) before changing the above settings.

∗ **REMEN**: REMC Enable Bit in the REMC Configuration (REMC_CFG) Register (D0/0x5340)

Data transfer control

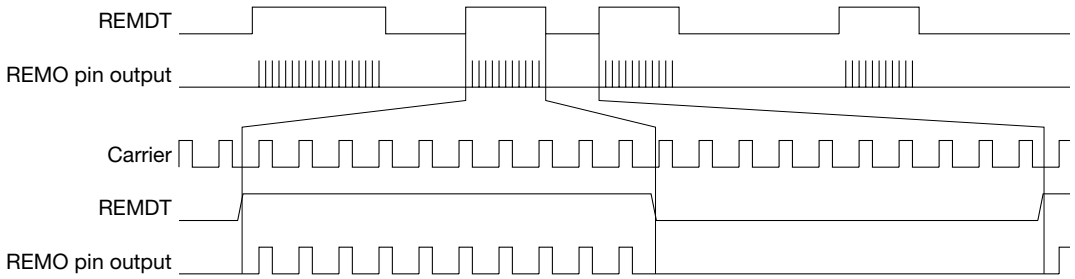


Figure 21.5.1: Data transmission

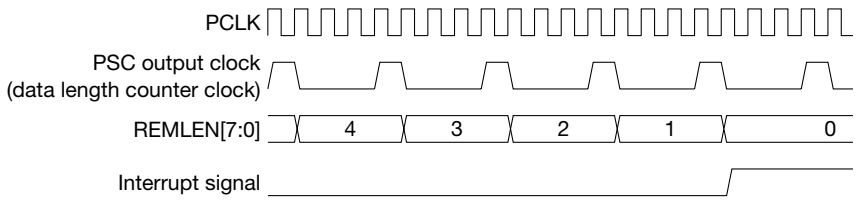


Figure 21.5.2: Underflow interrupt generation timing

(1) Data transmit mode setting

Set REMC to transmit mode by writing 0 to REMMD (D1/REMC_CFG register).

∗ **REMMD**: REMC Mode Select Bit in the REMC Configuration (REMC_CFG) Register (D1/0x5340)

(2) Permit data transmission

Permit REMC operation by setting REMEN (D0/REMC_CFG register) to 1. This initiates REMC transmission.

Set REMDT (D0/REMC_ST register) to 0 and REMLEN[7:0] (D[7:0]/REMC_LCNT register) to 0x0 before setting REMEN to 1 to prevent unnecessary data transmission.

(3) Transmission data settings

Set the data to be transmitted (High or Low) to REMDT (D0/REMC_ST register).

∗ **REMDT**: Transmit/Receive Data Bit in the REMC Status (REMC_ST) Register (D0/0x5344)

Setting REMDT to 1 outputs High; setting it to 0 outputs Low from the REMO pin after being modulated by the carrier signal.

(4) Data pulse length setting

Set the value corresponding to the data pulse length (High or Low section) at the start of transmission to REMLEN[7:0] (D[15:8]/REMC_LCNT register) to set to the data length counter.

∗ **REMLEN[7:0]**: Transmit/Receive Data Length Count Bits in the REMC Length Counter (REMC_LCNT) Register (D[15:8]/0x5344)

Given below are the values to which the data length counter is set:

Setting = Data pulse length (seconds) x prescaler output clock frequency (Hz)
The data length counter begins counting down from the value written using the prescaler output clock selected. An underflow interrupt factor occurs when the data length counter value reaches 0. If interrupts are permitted, an REMC interrupt request is output to the interrupt controller (ITC). The data length counter stops counting when it reaches 0.

(5) Interrupt processing

To transmit the subsequent data, set the subsequent data (step 3) and set the data pulse length (step 4) as part of the interrupt handler routine generated by the data length counter underflow.

(6) Data transmission end

To end data transmission, set REMEN to 0 after the final data transmission is complete (after underflow interrupt has occurred).

Data receipt control

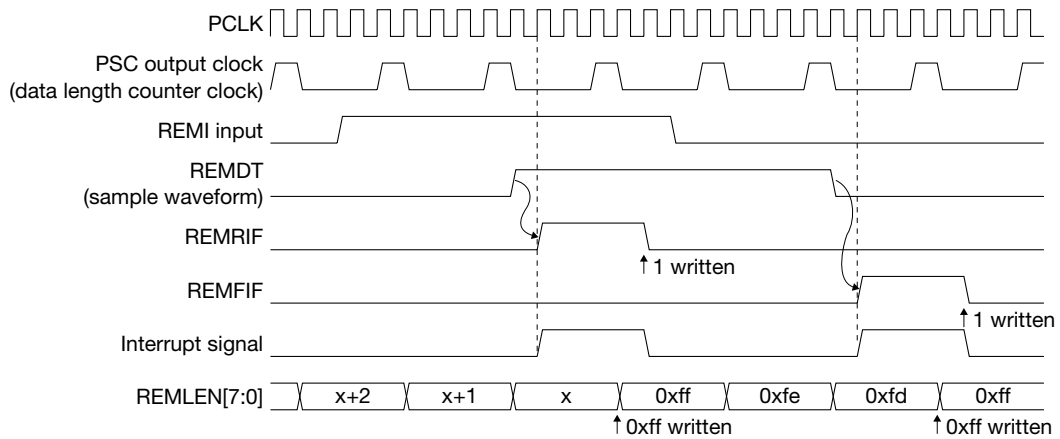


Figure 21.5.3: Data receipt

(1) Data receipt mode setting

Set REMC to receipt mode by writing 1 to REMMD (D1/REMC_CFG register).

(2) Permit data receipt

Permit REMC operation by setting REMEN (D0/REMC_CFG register) to 1. This initiates REMC transmission (input edge detection operation).

REMC detects input changes (signal rising or falling edges) by sampling the input signal from the REMI pin using the prescaler output clock selected for carrier generation. If a signal edge is detected, a rising or falling edge interrupt factor is generated. An REMC interrupt request is output to the ITC if interrupts are permitted. Rising edge and falling edge interrupts can be individually permitted or blocked.

Note that if the signal level after the input has changed is not detected for at least two continuous sampling clock cycles, the interrupt factor is interpreted as noise, and no rising or falling edge interrupt is generated.

(3) Interrupt processing

When a rising edge or falling edge interrupt occurs, 0xff is written to REMLEN[7:0] (D[15:8]/REMC_LCNT register) as part of the interrupt handler routine and set as the value of the data length counter.

The data length counter begins counting down using the selected prescaler output clock from the value written.

The data received can be read out from REMDT (D0/REMC_LCNT register).

The subsequent trailing or rising edge interrupt is generated once the data pulse ends, at which point the data length counter is read out. The data length can be calculated from the difference between 0xff and the value read. To receive the subsequent data, set the data length counter to 0xff once again, then wait for the subsequent interrupt.

If the data length counter becomes 0 after being set to 0xff without the occurrence of an edge interrupt, either data receiving is complete or a receive error has occurred. Data length counter underflow interrupts are generated even when receiving data and should be used for end/error processing.

(4) Data receipt end

To end data receipt, program 0 to REMEN after the final data has been received.

21.6 REMC Interrupts

The REMC module includes functions to generate the following three different interrupt types.
- Underflow interrupt
- Rising edge interrupt
- Falling edge interrupt

The REMC module outputs one interrupt signal shared by the three interrupt factors above to the interrupt controller (ITC). To identify the interrupt factor that occurred, inspect the interrupt flag within the REMC module.

Underflow interrupt

Generated when the data length counter has counted down to 0, this interrupt request sets the interrupt flag REMUIF (D0/REMC_IFLG register) inside the REMC to 1.
When data is being transmitted, the underflow interrupt indicates that the specified data length has been transmitted. When receiving data, the underflow interrupt indicates that data has been received or that a receive error has occurred.

∗ **REMUIF**: Underflow Interrupt Flag in the REMC Interrupt Flag (REMC_INT) Register (D0/0x5346)

To use this interrupt, set REMUIE (D0/REMC_IMSK register) to 1. If REMUIE is set to 0 (default), REMUIF will not be set to 1, and the interrupt request attributable to this factor will not be sent to the ITC.

∗ **REMUIE**: Underflow Interrupt Enable Bit in the REMC Interrupt Mask (REMC_IMSK) Register (D0/0x5346)

When REMUIF is set to 1, REMC outputs an interrupt request signal to the ITC. This interrupt request sets the REMC interrupt flag to 1 within the ITC, and generates an interrupt if the ITC and S1C17 core interrupt conditions are met.
REMUIF should be inspected as part of the REMC interrupt handler routine to determine whether the REMC interrupt is attributable to data length counter underflow.
The interrupt factor should be cleared as part of the interrupt handler routine by resetting both the ITC REMC interrupt flag and REMC module REMUIF (i.e., setting both to 1).

Rising edge interrupt

Generated when the REMI pin input signal changes from Low to High, this interrupt request sets the interrupt flag REMRIF (D1/REMC_IFLG register) to 1 within the REMC.
When data is being received, the data length counter can be operated between this interrupt and a falling edge interrupt to calculate the received data pulse width from that count value.

∗ **REMRIF**: Rising Edge Interrupt Flag in the REMC Interrupt Flag (REMC_INT) Register (D1/0x5346)

To use this interrupt, set REMRIE (D1/REMC_IMSK register) to 1. If REMRIE is set to 0 (default), REMRIF is not set to 1 and the interrupt request for this factor is not sent to the ITC.

∗ **REMRIE**: Rising Edge Interrupt Enable Bit in the REMC Interrupt Mask (REMC_INT) Register (D1/0x5346)

When REMRIF is set to 1, REMC outputs an interrupt request to the ITC. This interrupt request signal sets the REMC interrupt flag to 1 within the ITC, generating an interrupt if the ITC and S1C17 core interrupt conditions are met.
REMRIF should be inspected as part of the REMC interrupt handler routine to determine whether the REMC interrupt is attributable to input signal rising edge.
The interrupt factor should be cleared as part of the interrupt handler routine by resetting both the ITC REMC interrupt flag and REMC module REMRIF (i.e., setting both to 1).

Falling edge interrupt

Generated when the REMI pin input signal changes from High to Low, this interrupt request sets the interrupt flag REMRIF (D2/REMC_INT register) to 1 within the REMC.
When data is being received, the data length counter can be operated between this interrupt and a falling edge interrupt to calculate the received data pulse width from that count value.

* **REMFIF**: Falling Edge Interrupt Flag in the REMC Interrupt Flag (REMC_INT) Register (D2/0x5346)

To use this interrupt, set REMFIE (D2/REMC_IMSK register) to 1. If REMFIE is set to 0 (default), REMFIF is not set to 1 and the interrupt request for this factor is not sent to the ITC.

* **REMFIE**: Falling Edge Interrupt Enable Bit in the REMC Interrupt Mask (REMC_INT) Register (D2/0x5346)

When REMFIF is set to 1, REMC outputs an interrupt request to the ITC. This interrupt request signal sets the REMC interrupt flag to 1 within the ITC, generating an interrupt if the ITC and S1C17 core interrupt conditions are met.
REMFIF should be inspected as part of the REMC interrupt handler routine to determine whether the REMC interrupt is attributable to input signal falling edge.
The interrupt factor should be cleared as part of the interrupt handler routine by resetting both the ITC REMC interrupt flag and REMC module REMFIF (i.e., setting both to 1).

Interrupt vectors

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

Vector number: 17 (0x11)
Vector address: TTBR + 0x44

Other interrupt settings

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

21.7 Control Register Details

Table 21.7.1: REMC register list

Address	Register name		Function
0x5340	REMC_CFG	REMC Configuration Register	Clock and transfer control
0x5342	REMC_CAR	REMC Carrier Length Setup Register	Carrier H/L section length setting
0x5344	REMC_LCNT	REMC Length Counter Register	Transfer bit and transfer data length setting
0x5346	REMC_INT	REMC Interrupt Mask Register	Interrupt control

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

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

0x5340: REMC Configuration Register (REMC_CFG)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
REMC Configuration Register (REMC_CFG)	**0x5340** (16 bits)	D15–12	**CGCLK[3:0]**	Carrier generator clock select (Prescaler output clock)	CGCLK[3:0] / LCCLK[3:0]: 0xf reserved; 0xe PCLK-1/16384; 0xd PCLK-1/8192; 0xc PCLK-1/4096; 0xb PCLK-1/2048; 0xa PCLK-1/1024; 0x9 PCLK-1/512; 0x8 PCLK-1/256; 0x7 PCLK-1/128; 0x6 PCLK-1/64; 0x5 PCLK-1/32; 0x4 PCLK-1/16; 0x3 PCLK-1/8; 0x2 PCLK-1/4; 0x1 PCLK-1/2; 0x0 PCLK-1/1	0x0	R/W	
		D11–8	**LCCLK[3:0]**	Length counter clock select (Prescaler output clock)		0x0	R/W	
		D7–2	–	reserved	–	–	–	0 when being read.
		D1	**REMMD**	REMC mode select	1 Receive / 0 Transmit	0	R/W	
		D0	**REMEN**	REMC enable	1 Enable / 0 Disable	0	R/W	

D[15:12] CGCLK[3:0]: Carrier Generator Clock Select Bits

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

Table 21.7.2: Carrier generation clock selection

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

(Default: 0x0)

D[11:8] LCCLK[3:0]: Length Counter Clock Select Bits

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

Table 21.7.3: Carrier generation clock selection

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

(Default: 0x0)

Note: The clock should be set only while the REMC module is stopped (REMEN(D0) = 0).

D[7:2] Reserved

D1 REMMD: REMC Mode Select Bit

Selects the transfer direction.

1 (R/W): Receive

0 (R/W): Transmit (default)

D0 REMEN: REMC Enable Bit

Permits or prohibit data transfer by the REMC module
1 (R/W): Permitted
0 (R/W): Prohibited (default)

Setting REMEN to 1 begins transmission or receiving in accordance with REMMD (D1) settings.
Setting REMEN to 0 halts REMC module operations.

0x5342: REMC Carrier Length Setup Register (REMC_CAR)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
REMC Carrier Length Setup Register (REMC_CAR)	0x5342 (16 bits)	D15–14	–	reserved	–	–	–	0 when being read.
		D13–8	REMCL[5:0]	L carrier length setup	0x0 to 0x3f	0x0	R/W	
		D7–6	–	reserved	–	–	–	0 when being read.
		D5–0	REMCH[5:0]	H carrier length setup	0x0 to 0x3f	0x0	R/W	

D[15:14] Reserved

D[13:8] REMCL[5:0]: L Carrier Length Setup Bits

Set the carrier signal L section length. (Default: 0x0)
Specify a value corresponding to the number of carrier generation clock cycles selected by CGCLK[3:0] (D[15:12]/REMC_CFG register) + 1.
Calculate carrier L section length as follows:

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

REMCL: REMCL[5:0] settings
clk_in: Prescaler output clock frequency

The H section length is specified by REMCH[5:0] (D[5:0]).
The carrier signal is generated from these settings as shown in Figure 21.7.1.

D[7:6] Reserved

D[5:0] REMCH[5:0]: H Carrier Length Setup Bits

Set the carrier signal H section length. (Default: 0x0)
Specify a value corresponding to the number of carrier generation clock cycles selected by CGCLK[3:0] (D[15:12]/REMC_CFG register) + 1.
Calculate carrier H section length as follows:

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

REMCH: REMCH[5:0] settings
clk_in: Prescaler output clock frequency

The L section length is specified by REMCL[5:0] (D[13:8]).
The carrier signal is generated from these settings as shown in Figure 21.7.1.

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

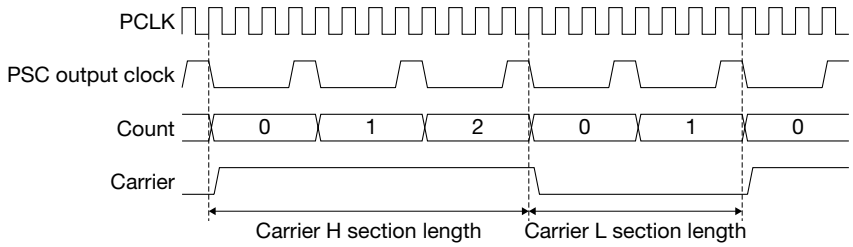


Figure 21.7.1: Carrier signal generation

0x5344: REMC Length Counter Register (REMC_LCNT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
REMC Length Counter Register (REMC_LCNT)	**0x5344** (16 bits)	D15–8	**REMLEN[7:0]**	Transmit/receive data length count (down counter)	0x0 to 0xff	0x0	R/W	
		D7–1	**–**	reserved	–	–	–	0 when being read.
		D0	**REMDT**	Transmit/receive data	1 1 (H) 0 0 (L)	0	R/W	

D[15:8] REMLEN[7:0]: Transmit/Receive Data Length Count Bits

Sets the data length counter value and begins counting. (Default: 0x0)
The counter stops when it reaches 0 and generates an underflow interrupt factor.

For data transmission

Sets the transmit data length for data transmission.
When a value corresponding to the data pulse width is written, the data length counter begins counting down from that value, generating an underflow interrupt and halting when the counter reaches 0.
The subsequent transmit data is set using this interrupt.

For data receiving

Interrupts can be generated at the input signal rising or falling edges when receiving data. The data pulse length can be obtained from the difference by setting the data length counter to 0xff using the interrupt when the input changes and reading out the count value when the next interrupt occurs due to an input change.

D[7:1] Reserved

D0 REMDT: Transmit/Receive Data Bit

Sets the transmit data for data transmission. Receive data can be read when receiving data.
1 (R/W): 1 (H)
0 (R/W): 0 (L) (default)

If REMEN (D0/REMC_CFG register) is set to 1, the REMDT setting is modulated by the carrier signal for data transmission and output from the REMO pin. For data receiving, this bit is set to the value corresponding to the signal level of the data pulse input.

0x5346: REMC Interrupt Control Register (REMC_INT)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
REMC Interrupt Control Register (REMC_INT)	0x5346 (16 bits)	D15–11	–	reserved	–				–	–	0 when being read.
		D10	REMFIF	Falling edge interrupt flag	1	Cause of interrupt occurred	0	Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D9	REMRIF	Rising edge interrupt flag					0	R/W	
		D8	REMUIF	Underflow interrupt flag					0	R/W	
		D7–3	–	reserved	–				–	–	0 when being read.
		D2	REMFIE	Falling edge interrupt enable	1	Enable	0	Disable	0	R/W	
		D1	REMRIE	Rising edge interrupt enable	1	Enable	0	Disable	0	R/W	
		D0	REMUIE	Underflow interrupt enable	1	Enable	0	Disable	0	R/W	

This register indicates the occurrence status of interrupt factors arising from data length counter underflow, input signal rising edge, or input signal falling edge. When an REMC interrupt occurs, the interrupt flag in this register should be inspected to identify the interrupt factor.
Setting the corresponding interrupt enable bit to 1 sets the interrupt flag to 1 when a data length counter underflow, input signal rising edge, or input signal falling edge occurs. The REMC outputs an interrupt request signal to the ITC at the same time, which sets the REMC interrupt flag to 1 within the ITC and generates an interrupt if the ITC and S1C17 core interrupt conditions are met.

Note: • **To prevent interrupt recurrences, the REMC module interrupt flag must be reset within the interrupt handler routine following an REMC interrupt.**

• **To prevent generating unnecessary interrupts, reset the interrupt flag before permitting interrupts by the interrupt enable bit.**

D[15:11] Reserved

D10 REMFIF: Falling Edge Interrupt Flag

Interrupt flag indicating the falling edge interrupt occurrence status.

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

Setting REMFIE (D2/REMC_IMSK register) to 1 sets SIF1 to 1 at the input signal falling edge.

D9 REMRIF: Rising Edge Interrupt Flag

Interrupt flag indicating the rising edge interrupt factor occurrence status.

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

Setting REMRIF (D1/REMC_IMSK register) to 1 sets REMRIE to 1 at the input signal falling edge.

D8 REMUIF: Underflow Interrupt Flag

Interrupt flag indicating the underflow interrupt factor occurrence status.

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

Setting REMUIE (D1/REMC_IMSK register) to 1 sets REMUIF to 1 when a data length counter underflow occurs.

D[7:3] Reserved

D2 REMFIE: Falling Edge Interrupt Enable Bit

Permits or blocks input signal falling edge interrupts.

1 (R/W): Interrupt permitted
0 (R/W): Interrupt prohibited (default)

D1 REMRIE: Rising Edge Interrupt Enable Bit

Permits or blocks input signal rising edge interrupts.

1 (R/W): Interrupt permitted

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

D0 REMUIE: Underflow Interrupt Enable Bit

Permits or blocks data length counter underflow interrupts.

1 (R/W): Interrupt permitted

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

21.8 Precautions

- The prescaler must run before operating the REMC module.
- To prevent interrupt recurrences, the REMC module interrupt flag must be reset within the interrupt handler routine following an REMC interrupt.
- To prevent unwanted interrupts, reset the interrupt flag before permitting interrupts with the interrupt enable bit.

22 LCD Driver (LCD)

22.1 LCD Driver Configuration

The S1C17702 incorporates a pixel-matrix LCD driver capable of driving LCD panels of up to 2,304 pixels (72 segments × 32 common) in size. Figure 22.1.1 illustrates the LCD driver and driver power supply configuration.

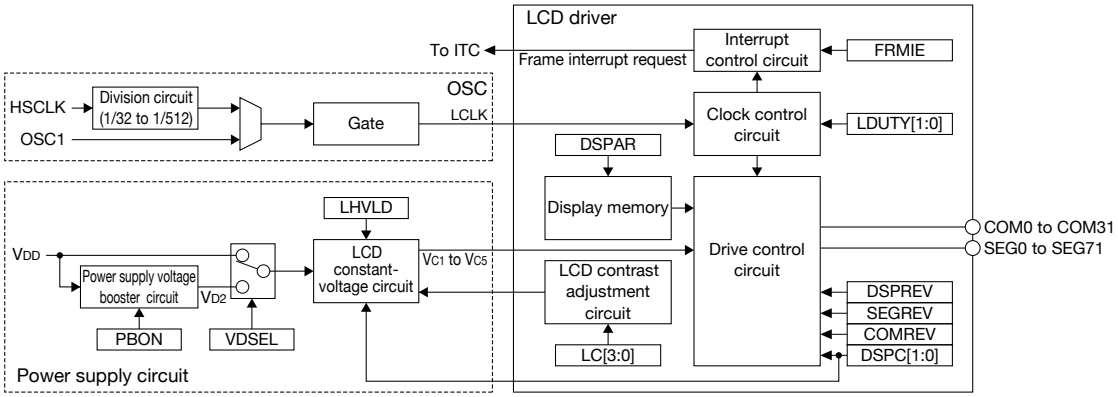


Figure 22.1.1: LCD driver and driver power supply configuration

22.2 LCD Power Supply

LCD driver voltages V_{C1} to V_{C5} are generated using the internal chip LCD voltage regulator and power supply voltage booster circuit. No external power supply is needed. For more information on the LCD power supply, see "4 Power Supply."

22.3 LCD Clock

Figure 22.3.1 illustrates the LCD clock feed system.

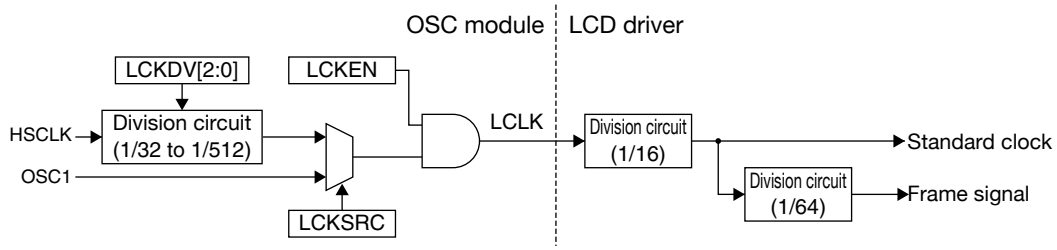


Figure 22.3.1: LCD clock system

22.3.1 LCD Operating Clock

The LCD driver operation clock (LCLK) is generated by the LCD clock generator within the OSC module. For more information on the OSC module, see "7 Oscillator Circuit (OSC)."

22.3.2 Frame Signal

The frame signal is generated by dividing LCLK into 1/1024. The frame frequency is as shown below. The time for 1 frame shown in Figures 22.4.1 and 22.4.2 is the frame frequency.

When OSC1 (32.768 kHz typ.) is selected as the LCD clock source

Frame frequency = 64 Hz (typ.)

When HSCLK is selected as the LCD clock source

$$\text{Frame frequency} = \frac{f_{HSCLK}}{512} \times \text{LCKDV [Hz]}$$

f_{HSCLK}: HSCLK (IOSC or OSC3) clock frequency [Hz]
LCKDV: HSCLK division ratio 1/32 to 1/512

22.4 Driver Duty Switching

Drive duty can be switched between 1/32 and 1/16 using LDUTY[1:0] (D[1:0]/LCD_CCTL register). Table 22.4.1 shows the correspondence between LDUTY[1:0] settings, drive duty, and maximum display pixel size.

* **LDUTY[1:0]**: LCD Duty Select Bits in the LCD Clock Control (LCD_CCTL) Register (D[1:0]/0x50a2)

Table 22.4.1: Drive duty settings

LDUTY[1:0]	Duty	Valid common pin	Valid segment pin	Max display pixel size
0x3	Reserved	–	–	–
0x2	1/32	COM0 to COM31	SEG0 to SEG71	2,304 pixels
0x1	1/16	COM0 to COM15	SEG0 to SEG87	1,408 pixels
0x0	Reserved	–	–	–

(Default: 0x2)

Pins COM16 to COM31 and SEG87 to SEG72 are set to common output pins when 1/32 duty is selected and to segment output pins when 1/16 duty is selected.
The drive bias is 1/5 (five potentials V_{C1}, V_{C2}, V_{C3}, V_{C4}, V_{C5}) for both 1/32 and 1/16 duty. The drive waveforms are as shown in Figure 22.4.1 and 22.4.2, respectively.

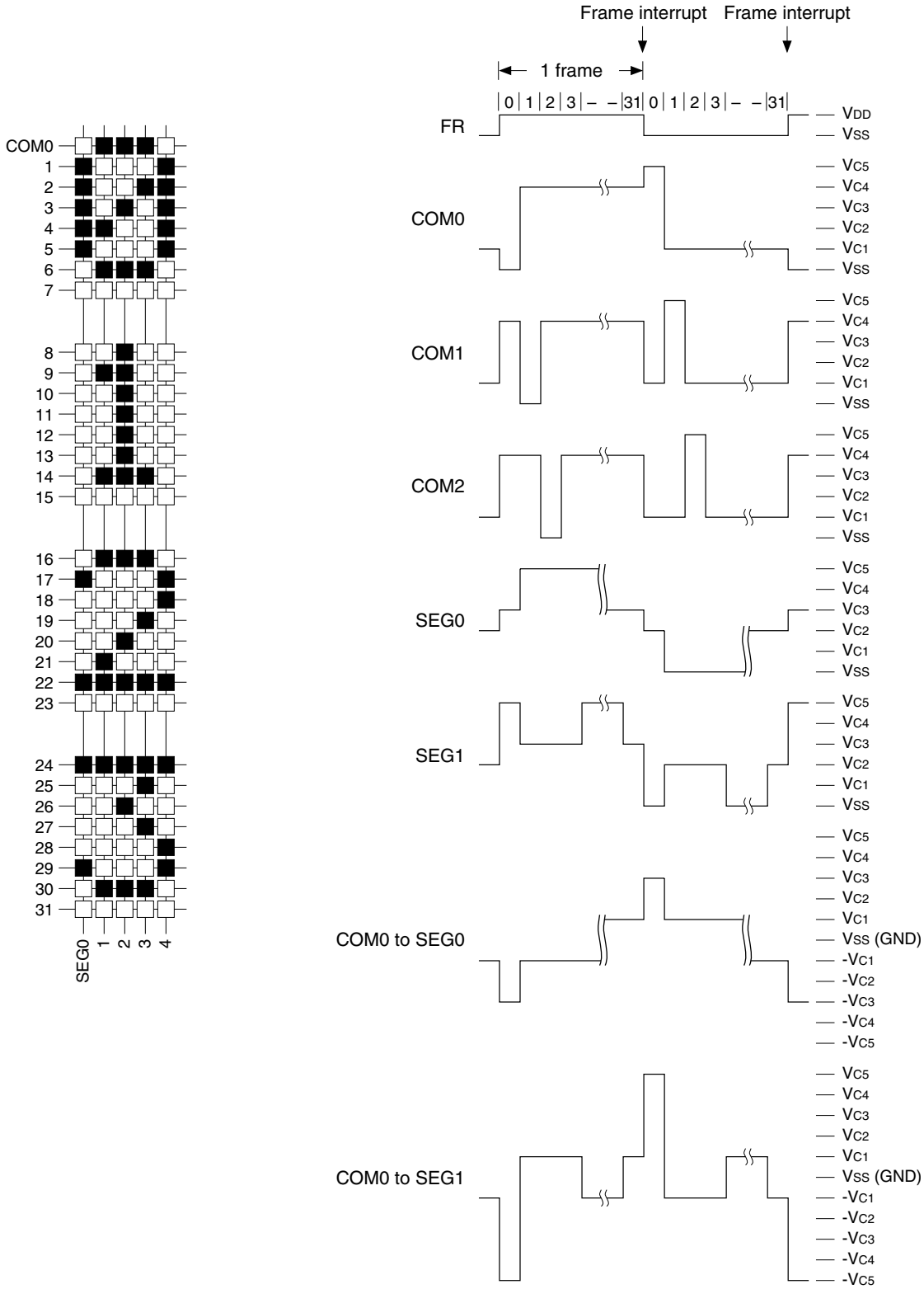


Figure 22.4.1: 1/32 duty drive waveform

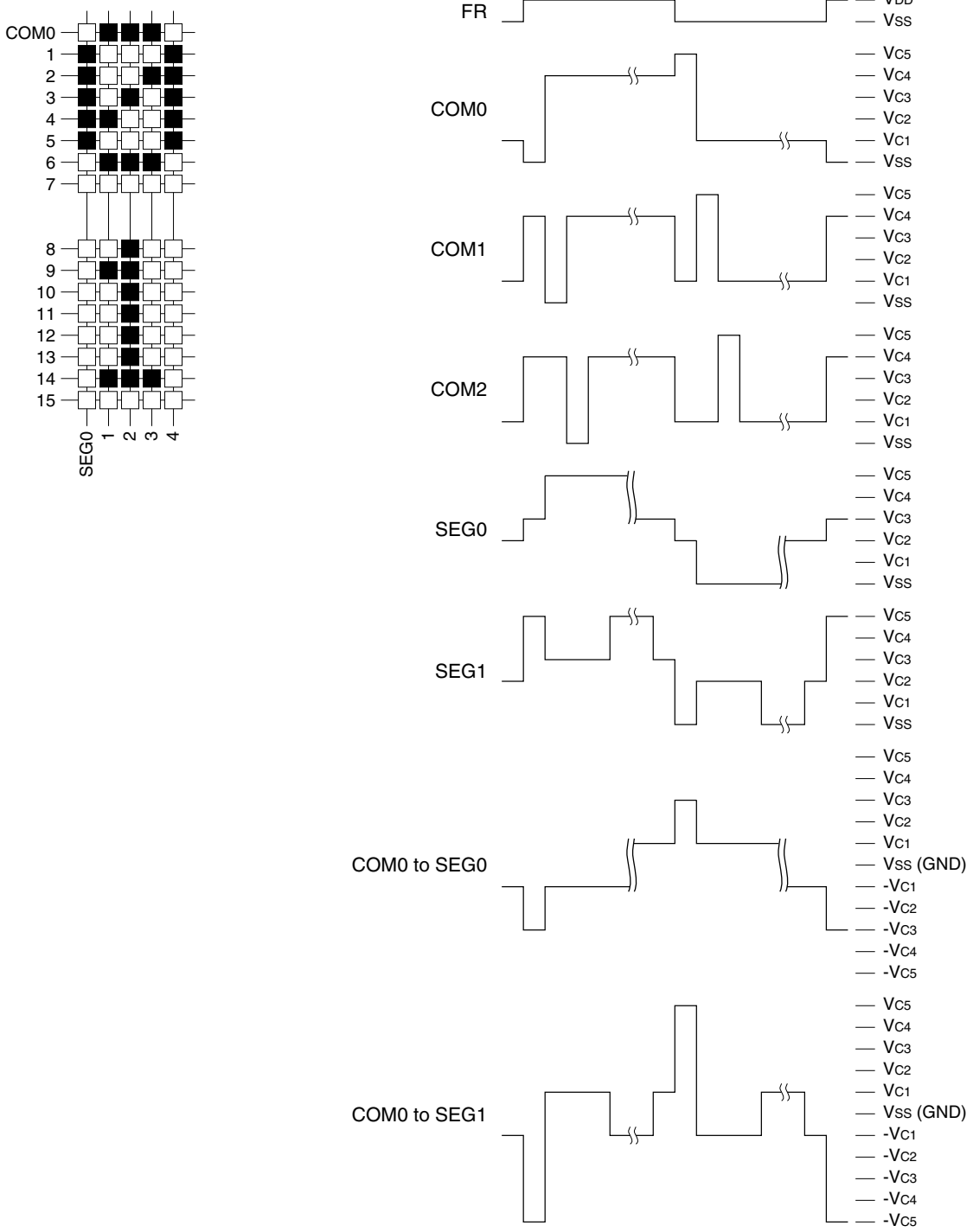


Figure 22.4.2: 1/16 duty drive waveform

22.5 Display Memory

The S1C17702 incorporates 576 bytes of display memory. The display memory is assigned to addresses 0x80000 to 0x8055f. The correspondence between memory bits and common/segment pins varies depending on the items selected, as follows.

(1) Drive duty (1/32 or 1/16 duty)
(2) SEG pin assignment (normal or inverted)
(3) COM pin assignment (normal or inverted)

Figures 22.5.1 and 22.5.2 show the correspondence between display memory and common/segment pins for each drive duty.

Writing 1 to a display memory bit corresponding to pixels on the LCD panel switches on that pixel, while writing 0 turns off the pixel. Since the display memory is RAM allowing reading and writing, bits can be controlled individually using logic operation instructions (read modify program instructions).

Bits not assigned to the display area within the 576 byte display memory can be used as general RAM for reads and programs.

Address initial 16 bits	Bit	Address last 8 bits: 0x00 ··· 0x0f	0x10 ··· 0x1f	0x20 ··· 0x2f	0x30 ··· 0x3f	0x40 ··· 0x47	0x48 ··· 0x5f	0x60 ··· 0xff	COMREV = 1	COMREV = 0
0x800**	D0	Display area					Unused area (general memory)	Unavailable area	COM0	COM31
	D1								COM1	COM30
	D2								COM2	COM29
	D3								COM3	COM28
	D4								COM4	COM27
	D5								COM5	COM26
	D6								COM6	COM25
	D7								COM7	COM24
0x801**	D0								COM8	COM23
	D1								COM9	COM22
	D2								COM10	COM21
	D3								COM11	COM20
	D4								COM12	COM19
	D5								COM13	COM18
	D6								COM14	COM17
	D7								COM15	COM16
0x802**	D0								COM16	COM15
	D1								COM17	COM14
	D2								COM18	COM13
	D3								COM19	COM12
	D4								COM20	COM11
	D5								COM21	COM10
	D6								COM22	COM9
	D7								COM23	COM8
0x803**	D0								COM24	COM7
	D1								COM25	COM6
	D2								COM26	COM5
	D3								COM27	COM4
	D4								COM28	COM3
	D5								COM29	COM2
	D6								COM30	COM1
	D7								COM31	COM0
0x804**	D0									
	D1									
	D2									
	D3									
	D4									
	D5									
	D6									
	D7									
0x805**	D0									
	D1									
	D2									
	D3									
	D4									
	D5									
	D6									
	D7									
SEGREV = 1		SEG0 ··· SEG15	SEG16 ··· SEG31	SEG32 ··· SEG47	SEG48 ··· SEG63	SEG64 ··· SEG71				
SEGREV = 0		SEG71 ··· SEG56	SEG55 ··· SEG40	SEG39 ··· SEG24	SEG23 ··· SEG8	SEG7 ··· SEG0				

Figure 22.5.1: Display memory map (with 1/32 duty selected)

Address initial 16 bits	Bit	Address last 8 bits 0x00 ··· 0x0f	0x10 ··· 0x1f	0x20 ··· 0x2f	0x30 ··· 0x3f	0x40 ··· 0x4f	0x50 ··· 0x57	0x58 ··· 0x5f	0x60 ··· 0xff	COMREV = 1	COMREV = 0
0x800**	D0	Display area 0 (DSPAR = 0)								COM0	COM15
	D1									COM1	COM14
	D2									COM2	COM13
	D3									COM3	COM12
	D4									COM4	COM11
	D5									COM5	COM10
	D6									COM6	COM9
	D7									COM7	COM8
0x801**	D0									COM8	COM7
	D1									COM9	COM6
	D2									COM10	COM5
	D3									COM11	COM4
	D4									COM12	COM3
	D5									COM13	COM2
	D6									COM14	COM1
	D7									COM15	COM0
0x802**	D0	Display area 1 (DSPAR = 1)						Unused area (general memory)	Unavailable area	COM0	COM15
	D1									COM1	COM14
	D2									COM2	COM13
	D3									COM3	COM12
	D4									COM4	COM11
	D5									COM5	COM10
	D6									COM6	COM9
	D7									COM7	COM8
0x803**	D0									COM8	COM7
	D1									COM9	COM6
	D2									COM10	COM5
	D3									COM11	COM4
	D4									COM12	COM3
	D5									COM13	COM2
	D6									COM14	COM1
	D7									COM15	COM0
0x804**	D0										
	D1										
	D2										
	D3										
	D4										
	D5										
	D6										
	D7										
0x805**	D0										
	D1										
	D2										
	D3										
	D4										
	D5										
	D6										
	D7										
SEGREV = 1		SEG0 ··· SEG15	SEG16 ··· SEG31	SEG32 ··· SEG47	SEG48 ··· SEG63	SEG64 ··· SEG79	SEG80 ··· SEG87				
SEGREV = 0		SEG87 ··· SEG72	SEG71 ··· SEG56	SEG55 ··· SEG40	SEG39 ··· SEG24	SEG23 ··· SEG8	SEG7 ··· SEG0				

Figure 22.5.2: Display memory map (with 1/16 duty selected)

Display area selection (with 1/16 duty selected)

When 1/16 duty is selected as the drive duty, two screen areas can be reserved within the display memory, and DSPAR (D5/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.

* **DSPAR**: Display Memory Area Control Bit in the LCD Display Control (LCD_DCTL) Register (D5/0x50a0)

SEG pin assignment

The display memory address assignment for the SEG pins can be inverted using SEGREV (D7/LCD_DCTL register). When SEGREV is set to 1 (the default value), 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. (Refer to Figures 22.5.1 and 22.5.2)

* **SEGREV**: Segment Output Assignment Control Bit in the LCD Display Control (LCD_DCTL) Register (D7/0x50a0)

COM pin assignment

The display memory bit assignment for the COM pins can be inverted using COMREV (D6/LCD_DCTL register). When COMREV is set to 1 (the default value), 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. (Refer to Figures 22.5.1 and 22.5.2)

* **COMREV**: Common Output Assignment Control Bit in the LCD Display Control (LCD_DCTL) Register (D6/0x50a0)

22.6 Display Control

22.6.1 Display On/Off

The LCD display state is controlled using DSPC[1:0] (D[1:0]/LCD_DCTL register).

* **DSPC[1:0]**: LCD Display Control Bits in the LCD Display Control (LCD_DCTL) Register (D[1:0]/0x50a0)

Table 22.6.1.1: 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 feed must be underway at the time. (See section 22.3.)

Note: The LCD power supply may not generate the drive voltage normally if DSPC[1:0] is set to a value other than 0x0 before the clock is supplied to the LCD driver circuit. Make sure that DSPC[1:0] is set to 0x0 before starting the clock supply and alter DSPC[1:0] after the conditions listed below are all met.

1. **The LCD clock source oscillator is operating and the output clock is stabilized (oscillation start time and oscillation stabilization time have already been elapsed).**
2. **The LCD clock settings have been finished.**
3. **The LCD clock supply is enabled and the clock is fed into the LCD driver circuit.**

If display off is selected, the drive voltage feed from the LCD voltage regulator stops, and pins V_{C1} to V_{C5} 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. Common 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 initial resetting.

Note: Executing the slp instruction during displaying may cause deterioration of the LCD panel, as it does not reset DSPC[1:0] to 0x0 (Display off). Therefore, be sure to reset DSPC[1:0] to 0x0 (Display off) via software before executing the slp instruction.

22.6.2 LCD Contrast Adjustment

The LCD contrast can be adjusted to one of 16 gray levels using LC[3:0] (D[3:0]/LCD_CADJ register). Contrast is adjusted by controlling the voltages V_{C1} to V_{C5} output by the internal LCD voltage circuit.

* **LC[3:0]**: LCD Contrast Adjustment Bits in the LCD Contrast Adjust (LCD_CADJ) Register (D[3:0]/0x50a1)

Table 22.6.2.1: LCD contrast adjustment

LC[3:0]	Contrast
0xf	High (dark)
0xe	↑
:	:
0x1	↓
0x0	Low (light)

(Default: 0x0)

LC[3:0] is set to 0x0 after initial resetting. Initialization via software is required to achieve the required contrast.

22.6.3 Inverted Display

The LCD display can be inverted (black/white inversion) using merely control bit manipulation, without changing the display memory. Setting DSPREV (D4/LCD_DCTL register) to 0 inverts the display; setting to 1 returns the display to normal status.

* **DSPREV**: Reverse Display Control Bit in the LCD Display Control (LCD_DCTL) Register (D4/0x50a0)

Note that the display will not be inverted if All off is selected using DSPC[1:0] (D[1:0]/LCD_DCTL register). The display will be inverted if All on is selected.

22.6.4 Gradation Display Control

The LCD driver includes a function for generating interrupts for individual frames. These interrupts can be used to produce graduated shades of gray by turning pixels on or off.
Since the actual gray scales available will vary depending on LCD panel characteristics, the graduated shades of gray should be controlled by adjusting frame frequencies and frame intervals for turning pixels on and off.
For more information on frame interrupts, refer to section 22.7.

22.7 LCD Interrupt

The LCD module includes a function for generating interrupts due to frame signals.

Frame interrupt

This interrupt request generated for each frame sets the interrupt flag FRMIF (D0/LCD_IFLG register) to 1 within the LCD module.
See Figures 22.4.1 and 22.4.2 for more information on interrupt timing.

∗ **FRMIF**: Frame Signal Interrupt Flag in the LCD Interrupt Flag (LCD_IFLG) Register (D0/0x50a6)

To use this interrupt, set FRMIE (D0/LCD_IMSK register) to 1. When FRMIE is set to 0 (the default value), interrupt requests for this interrupt cause are not sent to the interrupt controller (ITC).

∗ **FRMIE**: Frame Signal Interrupt Enable Bit in the LCD Interrupt Mask (LCD_IMSK) Register (D0/0x50a5)

If FRMIF is set to 1 while FRMIE is set to 1 (interrupt permitted), the LCD module outputs an interrupt request to the ITC. An interrupt is generated if interrupt conditions are satisfied for the ITC and S1C17 core.
Within the interrupt handler routine, the interrupt cause should be cleared by resetting (writing 1 to) the LCD module FRMIF, rather than using the ITC LCD interrupt flag.

Note:
- **To prevent interrupt recurrences, the LCD module interrupt flag FRMIF must be reset within the interrupt handler routine following an LCD interrupt.**
- **To prevent unwanted interrupts, FRMIF should be reset before permitting LCD interrupts with FRMIE.**

Interrupt vector

The LCD interrupt vector number and vector address are as shown below:

Vector number: 10(0x0a)
Vector address: TTBR + 0x28

Other interrupt settings

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

22.8 Control Register Details

Table 22.8.1: LCD register list

Address	Register name		Function
0x50a0	LCD_DCTL	LCD Display Control Register	LCD display control
0x50a1	LCD_CADJ	LCD Contrast Adjust Register	Contrast control
0x50a2	LCD_CCTL	LCD Clock Control Register	LCD clock duty selection
0x50a3	LCD_VREG	LCD Voltage Regulator Control Register	LCD driver voltage regulator control
0x50a4	LCD_PWR	LCD Power Voltage Booster Control Register	LCD power supply voltage booster circuit control
0x50a5	LCD_IMSK	LCD Interrupt Mask Register	Interrupt mask setting
0x50a6	LCD_IFLG	LCD Interrupt Flag Register	Interrupt occurrence status display/reset

The individual LCD module registers are described below. These are all 8-bit registers.

Note: When writing data to the registers, always program 0 to bits indicated as "Reserved." Do not program 1.

0x50a0: 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]	Display			0x0	R/W	
					0x3	All off					
					0x2	All on					
					0x1	Normal display					
					0x0	Display off					

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 (the default value), 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. (Refer to Figures 22.5.1 and 22.5.2)

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 (the default value), 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. (Refer to Figures 22.5.1 and 22.5.2)

D5 DSPAR: Display Memory Area Control Bit

Selects the display area when driving using 1/16 duty.
1 (R/W): Display area 1
0 (R/W): Display area 0 (default)

Selects which of the two display areas reserved in the display area is displayed when driving using 1/16 duty. Setting DSPAR to 0 selects display area 0; setting to 1 selects display area 1. Refer to Figure 22.5.2 for more information on 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 display memory.

D[3:2] Reserved

D[1:0] DSPC[1:0]: LCD Display Control Bits

Control the LCD display.

Table 22.8.2: 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 feed must be underway at the time. (See section 22.3.)

Note: The LCD power supply may not generate the drive voltage normally if DSPC[1:0] is set to a value other than 0x0 before the clock is supplied to the LCD driver circuit. Make sure that DSPC[1:0] is set to 0x0 before starting the clock supply and alter DSPC[1:0] after the conditions listed below are all met.

1. **The LCD clock source oscillator is operating and the output clock is stabilized (oscillation start time and oscillation stabilization time have already been elapsed).**
2. **The LCD clock settings have been finished.**
3. **The LCD clock supply is enabled and the clock is fed into the LCD driver circuit.**

If display off is selected, the drive voltage feed from the LCD voltage regulator stops, and pins V_{C1} to V_{C5} are all set to V_{SS} level.

Since All on and All off directly control the driving waveform output by the LCD driver, the display memory data is not altered. Common pins will be 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 display memory.

DSPC[1:0] is reset to 0x0 (Display off) after initial resetting.

Note: Executing the slp instruction during displaying may cause deterioration of the LCD panel, as it does not reset DSPC[1:0] to 0x0 (Display off). Therefore, be sure to reset DSPC[1:0] to 0x0 (Display off) via software before executing the slp instruction.

0x50a1: LCD Contrast Adjust Register (LCD_CADJ)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
LCD Contrast Adjust 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	0x0	R/W	
					0xf	Dark			
					:	:			
					0x0	Light			

D[7:4] Reserved

D[3:0] LC[3:0]: LCD Contrast Adjustment Bits

Adjust LCD contrast by controlling voltages V_{C1} to V_{C5} output by the internal LCD voltage circuit.

Table 22.8.3: LCD contrast adjustment

LC[3:0]	Contrast
0xf	High (dark)
0xe	↑
:	:
0x1	↓
0x0	Low (light)

(Default: 0x0)

LC[3:0] is set to 0x0 after initial resetting. Initialization via software is required to achieve the required contrast.

0x50a2: 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)	D7–2	**–**	reserved	–		–	–	0 when being read.
		D1–0	**LDUTY[1:0]**	LCD duty select	LDUTY[1:0]	Duty	0x2	R/W	
					0x3	reserved			
					0x2	1/32			
					0x1	1/16			
					0x0	reserved			

D[7:2] Reserved

D[1:0] LDUTY[1:0]: LCD Duty Select Bits

Select the drive duty.

Table 22.8.4: Drive duty settings

LDUTY[1:0]	Duty	Valid common pin	Valid segment pin	Max display pixel size
0x3	Reserved	–	–	–
0x2	1/32	COM0 to COM31	SEG0 to SEG71	2,304 pixels
0x1	1/16	COM0 to COM15	SEG0 to SEG87	1,408 pixels
0x0	Reserved	–	–	–

(Default: 0x2)

0x50a3: 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	
		D3–0	**–**	reserved	–	–	–	0 when being read.

For more information on these control bits, see “0x50a3: LCD Voltage Regulator Control Register (LCD_VREG)” in section 4.5.

0x50a4: LCD Power Voltage Booster Control Register (LCD_PWR)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
LCD Power Voltage Booster Control Register (LCD_PWR)	**0x50a4** (8 bits)	D7–2	**–**	reserved	–				–	–	0 when being read.
		D1	**VDSEL**	Regulator power source select	1	V_{D2}	0	V_{DD}	0	R/W	
		D0	**PBON**	Power voltage booster control	1	On	0	Off	0	R/W	

For more information on these control bits, see "0x50a4: LCD Power Voltage Booster Control Register (LCD_PWR)" in section 4.5.

0x50a5: 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

Permits or prevents frame interrupts.

1 (R/W): Permit interrupt

0 (R/W): Prevent interrupt (default)

Setting FRMIE to 1 permits LCD interrupt requests to the ITC. Setting to 0 prevents interrupts.

0x50a6: 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

Interrupt flag indicating the frame interrupt cause occurrence status.
1 (R): Interrupt cause present
0 (R): No interrupt cause (default)
1 (W): Reset flag
0 (W): Disable

FRMIF is the LCD module interrupt flag and is set to the frame signal rising edge 1. If FRMIE (D0/LCD_IMSK register) is set to 1 here, an LCD interrupt request signal is output to the ITC. An interrupt is generated if interrupt conditions are satisfied for the ITC and S1C17 core.
FRMIF is reset by writing 1.

Note:
- **To prevent interrupt recurrences, the LCD module interrupt flag FRMIF must be reset within the interrupt handler routine following an LCD interrupt.**
- **To prevent unwanted interrupts, FRMIF should be reset before permitting LCD interrupts with FRMIE.**

22.9 Precautions

- To prevent interrupt recurrences, the LCD module interrupt flag FRMIF (D0/LCD_IFLG register) must be reset within the interrupt handler routine following an LCD interrupt.
- To prevent unwanted interrupts, FRMIF (D0/LCD_IFLG register) should be reset before permitting LCD interrupts with FRMIE (D0/LCD_IMSK register).
- See "4.6 Precautions" for LCD power supply precautions.
- The LCD power supply may not generate the drive voltage normally if DSPC[1:0] (D[1:0]/LCD_DCTL register) is set to a value other than 0x0 before the clock is supplied to the LCD driver circuit. Make sure that DSPC[1:0] is set to 0x0 before starting the clock supply and alter DSPC[1:0] after the conditions listed below are all met.
  1. The LCD clock source oscillator is operating and the output clock is stabilized (oscillation start time and oscillation stabilization time have already been elapsed).
  2. The LCD clock settings have been finished.
  3. The LCD clock supply is enabled and the clock is fed into the LCD driver circuit.
- Executing the slp instruction during displaying may cause deterioration of the LCD panel, as it does not reset DSPC[1:0] to 0x0 (Display off). Therefore, be sure to reset DSPC[1:0] to 0x0 (Display off) via software before executing the slp instruction.

23 Power Supply Voltage Detection Circuit (SVD)

23.1 SVD Module Configuration

The S1C17702 incorporates an SVD (supply voltage detection) circuit to detect power supply voltage drops. Software can be used to turn the SVD circuit on/off, set the comparison voltage, and read out the detection results. Interrupts can also be generated when a voltage drop is detected. Figure 23.1.1 illustrates the SVD circuit configuration.

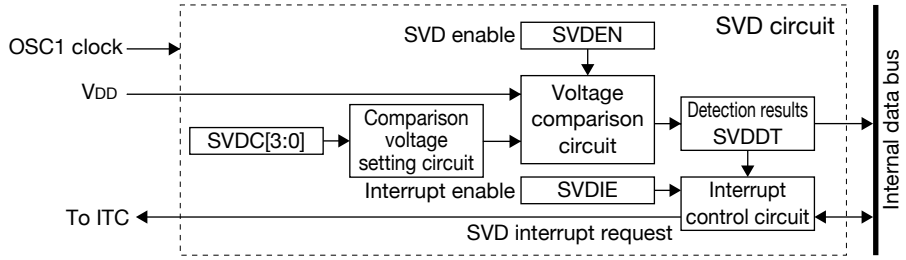


Figure 23.1.1: SVD circuit configuration

23.2 Comparison Voltage Setting

The SVD circuit compares the power supply voltage (V_{DD}) against the comparison voltage set by the software and outputs results indicating whether the power supply voltage exceeds this comparison voltage. The comparison voltage can be selected from among the 15 listed in Table 23.2.1 with the SVDC[3:0] (D[3:0]/SVD_CMP register).

∗ **SVDC[3:0]**: SVD Compare Voltage Select Bits in the SVD Compare Voltage (SVD_CMP) Register (D[3:0]/0x5101)

Table 23.2.1: 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)

23.3 SVD Circuit Control

Power supply voltage detection using the SVD circuit is initiated by writing 1 to SVDEN (D0/SVD_EN register) and is stopped by writing 0.

* **SVDEN**: SVD Enable Bit in the SVD Enable (SVD_EN) Register (D0/0x5100)

The results can be read out from the SVDDT (D0/SVD_RSLT register).

* **SVDDT**: SVD Detection Result Bit in the SVD Detection Result (SVD_RSLT) Register (D0/0x5102)

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 permitted and SVDEN is set to 1, an interrupt is generated 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 heavy load protection. 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 read and checked in the interrupt handler routine.

Note:
- **Up to 500 µs may be required to produce stable detection results after SVD circuit operations start. When reading detection results without using interrupts, allow for this stabilization time before reading SVDDT (D0/SVD_RSLT register) after writing 1 to SVDEN (D0/SVD_EN register).**
- **Operating the SVD circuit increases current consumption. If power supply voltage detection is not required, stop SVD operations by setting SVDEN to 0.**

23.4 SVD Interrupt

The SVD module includes a function for generating interrupts when power supply voltage drops are detected.

Power supply voltage detection interrupt

This interrupt request is generated when the power supply voltage (V_{DD}) detected value drops below the comparison voltage while SVD is operating (SVDEN (D0/SVD_EN register) = 1). It sets the interrupt flag SVDIF (D0/SVD_IFLG register) to 1 within the SVD module. Once set, SVDIF is not reset even if the power supply voltage subsequently returns to a value exceeding the comparison voltage.

∗ **SVDIF**: SVD Interrupt Flag in the SVD Interrupt Flag (SVD_IFLG) Register (D0/0x5104)

To use this interrupt, set SVDIE (D0/SVD_IMSK register) to 1. When SVDIE is set to 0 (the default value), interrupt request for this cause will not be sent to the interrupt controller (ITC).

∗ **SVDIE**: SVD Interrupt Enable Bit in the SVD Interrupt Mask (SVD_IMSK) Register (D0/0x5103)

If SVDIF is set to 1 while SVDIE is set to 1 (interrupt permitted), the SVD module outputs an interrupt request to the ITC. An interrupt is generated if interrupt conditions are satisfied for the ITC and S1C17 core.

Note:
- **To prevent interrupt recurrences, the SVD module interrupt flag SVDIF must be reset within the interrupt handler routine following an SVD interrupt.**
- **To prevent unwanted interrupts, reset SVDIF before permitting SVD interrupts with SVDIE.**

Interrupt vector

The SVD interrupt vector number and vector address are as shown below:

Vector number: 9(0x09)
Vector address: TTBR + 0x24

Other interrupt settings

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

23.5 Control Register Details

Table 23.5.1: SVD register list

Address	Register name		Function
0x5100	SVD_EN	SVD Enable Register	SVD operation permission
0x5101	SVD_CMP	SVD Compare Voltage Register	Comparison voltage setting
0x5102	SVD_RSLT	SVD Detection Result Register	Voltage detection result
0x5103	SVD_IMSK	SVD Interrupt Mask Register	Interrupt mask setting
0x5104	SVD_IFLG	SVD Interrupt Flag Register	Interrupt occurrence status display/reset

The individual SVD module registers are described below. These are all 8-bit registers.

Note: When writing data to the registers, always program 0 to bits indicated as "Reserved." Do not program 1.

0x5100: 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

Permits or prevents SVD circuit operations.
1 (R/W): Permit
0 (R/W): Prevent (default)

Setting SVDEN to 1 initiates power supply voltage detection; setting to 0 stops detection.

Note: • Up to 500 µs may be required to obtain stable detection results after the SVD circuit begins operating. When reading detection results without using interrupts, allow this stabilization time to elapse before reading SVDDT (D0/SVD_RSLT register) after writing 1 to SVDEN.

• Operating the SVD circuit increases current consumption. If power supply voltage detection is not required, stop SVD operations by setting SVDEN to 0.

0x5101: SVD Compare Voltage Register (SVD_CMP)

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
SVD Compare Voltage Register (SVD_CMP)	**0x5101** (8 bits)	D7–4	**–**	reserved	–		–	–	0 when being read.
		D3–0	**SVDC[3:0]**	SVD compare voltage	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 Compare Voltage Select Bits

Selects one of 15 comparison voltages for detecting voltage drops.

Table 23.5.2: 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.

0x5102: SVD Detection Result Register (SVD_RSLT)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
SVD Detection	0x5102	D7–1	–	reserved	–				–	–	0 when being read.
Result Register (SVD_RSLT)	(8 bits)	D0	SVDDT	SVD detection result	1	Low	0	Normal	×	R	

D[7:1] Reserved

D0 SVDDT: SVD Detection Result Bit

Reads out 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] (D[3:0]/SVD_CMP register) while SVDEN (D0/SVD_EN register) = 1. The current power supply voltage status can be checked by reading SVDDT.

0x5103: 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

Permits or prevents interrupts when a power supply voltage drop is detected.

1 (R/W): Permit interrupt

0 (R/W): Prevent interrupt (default)

Setting SVDIE to 1 permits SVD interrupt requests to the ITC; setting to 0 prevents interrupts.

0x5104: 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

Interrupt flag indicating the power supply voltage drop detection interrupt cause occurrence status.
1 (R): Interrupt cause present
0 (R): No interrupt cause (default)
1 (W): Reset flag
0 (W): Disable

SVDIF, the SVD module interrupt flag, is set to 1 when a power supply voltage drop is detected. If SVDIE (D0/SVD_IMSK register) is set to 1 here, an SVD interrupt request signal is output to the ITC. An interrupt is generated if interrupt conditions are satisfied for the ITC and S1C17 core.
SVDIF is reset by writing a 1 to it.

Note:
- **To prevent interrupt recurrences, the SVD module interrupt flag SVDIF must be reset within the interrupt handler routine following an SVD interrupt.**
- **To prevent unwanted interrupts, reset SVDIF before permitting SVD interrupts with SVDIE (D0/SVD_IMSK register).**

23.6 Precautions

- Up to 500 µs may be required to obtain stable detection results after the SVD circuit begins operating. When reading detection results without using interrupts, allow this stabilization time to elapse before reading SVDDT (D0/SVD_RSLT register) after writing 1 to SVDEN (D0/SVD_EN register).
- Operating the SVD circuit increases current consumption. If power supply voltage detection is not required, stop SVD operations by setting SVDEN (D0/SVD_EN register) to 0.
- To prevent interrupt recurrences, the SVD module interrupt flag SVDIF (D0/SVD_IFLG register) must be reset within the interrupt handler routine following an SVD interrupt.
- To prevent unwanted interrupts, reset SVDIF (D0/SVD_IFLG register) before permitting SVD interrupts with SVDIE (D0/SVD_IMSK register).
- The OSC1 clock feed must be underway for SVD operations. If the OSC1 oscillation is halted, initiate oscillation. Allow the oscillation start time and stabilization time to elapse before operating the SVD circuit.

24 On-chip Debugger (DBG)

24.1 Resource Requirements and Debugging Tool

Debugging work area

Debugging requires a 64-byte debugging work area. In the S1C17702, RAM addresses 0x0007c0 to 0x0007ff are assigned as the debugging work area. When using the debugging function, avoid using this area for any other user applications.

The start address for this debugging work area can be read from the DBRAM register (0xffff90).

Debugging tool

Debugging involves connecting an ICD (In-Circuit Debugger) such as S5U1C17001H (ICD Mini) to the S1C17702 debug pin and inputting the debug instruction from the PC debugger.

The following tools are required:

- S1C17 Family In-Circuit Debugger (e.g., S5U1C17001H)
- S1C17 Family C compiler package (S5U1C17001C)

Debug pins

The following debug pins are used to connect an ICD (e.g., S5U1C17001H).

Table 24.1.1: Debug pin list

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

Shared with general purpose input/output port pins (P31, P32, P33), the on-chip debugger input/output pins (DCLK, DST2, DSIO) are initially set for use as debugger pins. If the debugging function is not used, these pins can be switched via the P3_PMUX register to enable use as general purpose input/output port pins. Set the control bits shown below to 1 to switch the pins to general purpose input/output port use.

DCLK → P31

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

DST2 → P32

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

DSIO → P33

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

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

24.2 Debug Break Operation Status

The S1C17 core switches to debug mode when the `brk` instruction is executed or a debug interrupt is generated by a break signal (Low) input to the DSIO pin. This state persists until the `retd` instruction is executed.
During this time, hardware interrupts and NMIs are disabled.
The default setting halts peripheral circuit operations. This setting can be modified even when debugging is underway.

Peripheral circuits that operate using the prescaler output clock

- 8-bit timer
- 16-bit timer
- PWM & capture timer
- Remote controller
- P port
- UART
- SPI
- I^2C

With the default settings, the prescaler will stop in debug mode, also stopping the peripheral circuits above that use the prescaler output clock. The prescaler includes PRUND (D1/PSC_CTL register) to specify prescaler operations during debug mode. When PRUND is set to 1, the prescaler operates even in debug mode, allowing the peripheral circuits above to operate as well. When PRUND is 0 (default), the prescaler and the peripheral circuits above will stop when the S1C17 core switches to debug mode.

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

Peripheral circuits that operate using the OSC1 clock

- Clock timer
- Watchdog timer
- Stopwatch timer
- 8-bit OSC1 timer

The MISC register includes O1DBG (D0/MISC_OSC1 register) to specify the operation of the above OSC1 peripheral circuits during debug mode. When O1DBG is set to 1, the OSC1 peripheral circuits operate even in debug mode. When O1DBG is 0 (default), the OSC1 peripheral circuits will stop when the S1C17 core switches to debug mode.

∗ **O1DBG**: OSC1 Peripheral Control (in Debug Mode) Bit in the OSC1 Peripheral Control (MISC_OSC1) Register (D0/0x5324)

24.3 Additional Debugging Function

The S1C17702 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 S1C17702 also allows designation of address 0x0 (beginning address of internal RAM) as the branching destination when debug mode is activated. The branching destination address is selected using DBADR (D8/MISC_IRAMSZ register). When the DBADR is set to "0" (default), the branching destination is set to 0xfffc00. When it is set to "1," the branching destination is set to 0x0.

- **DBADR**: Debug Base Address Select Bit in the IRAM Size Select (MISC_IRAMSZ) Register (D8/0x5326)

Adding instruction breaks

The S1C17 core supports two instruction breaks (hardware PC breaks). The S1C17702 increased this number to five, adding the control bits and registers given below.

- **IBE2**: Instruction Break #2 Enable Bit in the Debug Control (DCR) Register (D5/0xffffa0)
- **IBE3**: Instruction Break #3 Enable Bit in the Debug Control (DCR) Register (D6/0xffffa0)
- **IBE4**: Instruction Break #4 Enable Bit in the Debug Control (DCR) Register (D7/0xffffa0)
- **IBAR2[23:0]**: Instruction Break Address #2 Bits in the Instruction Break Address (IBAR2) Register 2 (D[23:0]/0xffffb8)
- **IBAR3[23:0]**: Instruction Break Address #3 Bits in the Instruction Break Address (IBAR3) Register 3 (D[23:0]/0xffffbc)
- **IBAR4[23:0]**: Instruction Break Address #4 Bits in the Instruction Break Address (IBAR4) Register 4 (D[23:0]/0xffffd0)

Note that the debugger included in the S5U1C17001C (Ver. 1.2.1) or newer is required to use five hardware PC breaks.

24.4 Control Register Details

Table 24.4.1: Debug register list

Address	Register name		Function
0x5322	MISC_OSC1	OSC1 Peripheral Control Register	OSC1 operation peripheral function setting for debugging
0x5326	MISC_IRAMSZ	IRAM Size Select Register	IRAM size selection
0xffff90	DBRAM	Debug RAM Base Register	Debug RAM base address display
0xffffa0	DCR	Debug Control Register	Debug control
0xffffb8	IBAR2	Instruction Break Address Register 2	Instruction break address #2 setting
0xffffbc	IBAR3	Instruction Break Address Register 3	Instruction break address #3 setting
0xffffd0	IBAR4	Instruction Break Address Register 4	Instruction break address #4 setting

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

Note:
- **When data is written to the registers, the "Reserved" bits must always be written as 0 and not 1.**
- **For debug registers not described here, refer to the *S1C17 Core Manual*.**

0x5322: OSC1 Peripheral Control Register (MISC_OSC1)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
OSC1 Peripheral Control Register (MISC_OSC1)	0x5322 (16 bits)	D15–1	–	reserved	–				–	–	0 when being read.
		D0	O1DBG	OSC1 peripheral control in debug mode	1	Run	0	Stop	0	R/W	

D[7:1] Reserved

D0 O1DBG: OSC1 Peripheral Control in Debug Mode Bit

Sets OSC1 peripheral circuit operation in debug mode.

1 (R/W): Operate

0 (R/W): Stop (default)

OSC1 peripheral circuit refers to the following peripheral circuits that operate using the OSC1 clock.

- Clock timer
- Watchdog timer
- Stopwatch timer
- 8-bit OSC1 timer

0x5326: IRAM Size Select Register (MISC_IRAMSZ)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
IRAM Size Select Register (MISC_IRAMSZ)	0x5326 (16 bits)	D15–9	–	reserved	–	–	–	0 when being read.
		D8	DBADR	Debug base address select	1 0x0 0 0xfffc00	0	R/W	
		D7–2	–	reserved	–	–	–	0 when being read.
		D1–0	IRAMSZ[1:0]	IRAM size select	IRAMSZ[1:0] / Read cycle 0x3 reserved 0x2 reserved 0x1 reserved 0x0 reserved	0x0	R/W	

D[15:9] Reserved

D8 DBADR: Debug Base Address Select Bit

Selects the address to branch to in the event of a debug interrupt.
1(R/W): 0x0
0(R/W): 0xfffc00 (default)

D[7:2] Reserved

D[1:0] IRAMSZ[1:0]: IRAM Size Select Bits

Selects the size of the internal RAM to be used.

Table 24.4.2 Selecting the size of internal RAM

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

(Default: 0x0)

Note: • The IRAM Size Select Register is write-protected. The write-protection must be overridden by writing 0x96 to the MISC Protect Register (0x5324). Note that MISC Protect Register (0x5324) should normally be set to a value other than 0x96, except when writing to the IRAM Size Select Register. Unnecessary programs may result in system malfunctions.
• Do not alter IRAMSZ[1:0] from the default value.

0xffff90: Debug RAM Base Register (DBRAM)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Debug RAM Base Register (DBRAM)	**0xffff90** (32 bits)	D31–24	**–**	Unused (fixed at 0)	0x0	0x0	R	
		D23–0	**DBRAM[23:0]**	Debug RAM base address	0x2fc0	0x2fc0	R	

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

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

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

0xffffa0: Debug Control Register (DCR)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
Debug Control Register (DCR)	0xffffa0 (8 bits)	D7	IBE4	Instruction break #4 enable	1	Enable	0	Disable	0	R/W	
		D6	IBE3	Instruction break #3 enable	1	Enable	0	Disable	0	R/W	
		D5	IBE2	Instruction break #2 enable	1	Enable	0	Disable	0	R/W	
		D4	DR	Debug request flag	1	Occurred	0	Not occurred	0	R/W	Reset by writing 1.
		D3	IBE1	Instruction break #1 enable	1	Enable	0	Disable	0	R/W	
		D2	IBE0	Instruction break #0 enable	1	Enable	0	Disable	0	R/W	
		D1	SE	Single step enable	1	Enable	0	Disable	0	R/W	
		D0	DM	Debug mode	1	Debug mode	0	User mode	0	R	

D7 IBE4: Instruction Break #4 Enable Bit

Permits or prohibits instruction break #4.
1(R/W): Permit
0(R/W): Prohibit (default)

If this bit is set to 1, the instruction fetch address and the value set in the Instruction Break Address Register 4 (0xffffd0) are compared. If they match, an instruction break is generated. If this bit is set to 0, no comparison is performed.

D6 IBE3: Instruction Break #3 Enable Bit

Permits or prohibits instruction break #3.
1(R/W): Permit
0(R/W): Prohibit (default)

If this bit is set to 1, the instruction fetch address and the value set in the Instruction Break Address Register 3 (0xffffbc) are compared. If they match, an instruction break is generated. If this bit is set to 0, no comparison is performed.

D5 IBE2: Instruction Break #2 Enable Bit

Permits or prohibits instruction break #2.
1(R/W): Permit
0(R/W): Prohibit (default)

If this bit is set to 1, the instruction fetch address and the value set in the Instruction Break Address Register 2 (0xffffb8) are compared. If they match, an instruction break is generated. If this bit is set to 0, no comparison is performed.

D4 DR: Debug Request Flag

Indicates the presence or absence of an external debug request.
1(R): Request generated
0(R): None (default)
1(W): Resets flag
0(W): Invalid

This flag is cleared (reset to 0) when 1 is written. It must be cleared before the debug processing routine is terminated by the retd instruction.

D3 IBE1: Instruction Break #1 Enable Bit

Permits or prohibits instruction break #1.
1(R/W): Permit
0(R/W): Prohibit (default)

If this bit is set to 1, the instruction fetch address and the value set in the Instruction Break Address Register 1 (0xffffb4) are compared. If they match, an instruction break is generated. If this bit is set to 0, no comparison is performed.

D2 **IBE0: Instruction Break #0 Enable Bit**

Permits or prohibits instruction break #0.
1(R/W): Permit
0(R/W): Prohibit (default)

If this bit is set to 1, the instruction fetch address and the value set in the Instruction Break Address Register 0 (0xffffb0) are compared. If they match, an instruction break is generated. If this bit is set to 0, no comparison is performed.

D1 **SE: Single Step Enable Bit**

Permits or prohibits single-step operations.
1(R/W): Permit
0(R/W): Prohibit (default)

D0 **DM: Debug Mode Bit**

Indicates the processor operating mode (debug mode or user mode).
1(R): Debug mode
0(R): User mode (default)

0xffffb8: Instruction Break Address Register 2 (IBAR2)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Instruction Break Address Register 2 (IBAR2)	**0xffffb8** (32 bits)	D31–24	**–**	reserved	–	–	–	0 when being read.
		D23–0	**IBAR2[23:0]**	Instruction break address #2 IBAR223 = MSB IBAR20 = LSB	0x0 to 0xffffff	0x0	R/W	

D[31:24] Reserved

D[23:0] IBAR2[23:0]: Instruction Break Address #2 Bits

Sets instruction break address #2. (default: 0x000000)

0xffffbc: Instruction Break Address Register 3 (IBAR3)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Instruction Break Address Register 3 (IBAR3)	**0xffffbc** (32 bits)	D31–24	**–**	reserved	–	–	–	0 when being read.
		D23–0	**IBAR3[23:0]**	Instruction break address #3 IBAR323 = MSB IBAR30 = LSB	0x0 to 0xffffff	0x0	R/W	

D[31:24] Reserved

D[23:0] IBAR3[23:0]: Instruction Break Address #3 Bits

Sets instruction break address #3. (default: 0x000000)

0xffffd0: Instruction Break Address Register 4 (IBAR4)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Instruction Break Address Register 4 (IBAR4)	**0xffffd0** (32 bits)	D31–24	**–**	reserved	–	–	–	0 when being read.
		D23–0	**IBAR4[23:0]**	Instruction break address #4 IBAR423 = MSB IBAR40 = LSB	0x0 to 0xffffff	0x0	R/W	

D[31:24] Reserved

D[23:0] IBAR4[23:0]: Instruction Break Address #4 Bits

Sets instruction break address #4. (default: 0x000000)

25 Multiplier/Divider

25.1 Overview

The S1C17702 incorporates a coprocessor that provides signed/unsigned 16 × 16 bit multiplication functions, 16 ÷ 16 bit division functions, and signed 16 × 16 bit + 32-bit product-sum processor (MAC = Multiply and Accumulator) functions enabling overflow detection.
Use of these functions is discussed below.

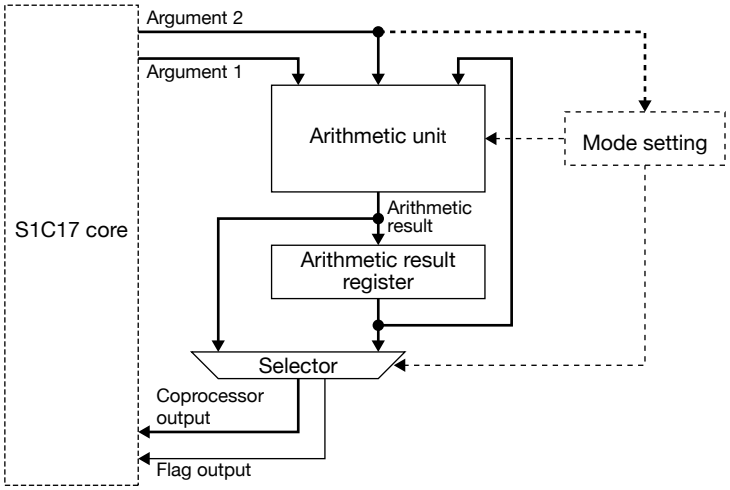


Figure 25.1.1: Multiplier/divider block diagram

Table 25.1.1: Arithmetic cycles

Operation	Cycles
Multiplication	1 cycle
Product-sum calculation	1 cycle
Division	17 to 20 cycles

25.2 Operating Mode and Output Mode

The multiplier/divider operates in accordance with the operating mode specified by the application program. The multiplier/divider supports six different operations, as shown in Table 25.2.1.
The results of multiplication, division, and product-sum operations are 32-bit data. This means the S1C17 core cannot read out results in a single access cycle. The output mode is provided to specify whether the first 16 bits or last 16 bits of the multiplier/divider arithmetic results are read out.
Specify the operating and output modes by writing 7-bit data to the multiplier/divider internal mode setting register. Use the "`ld.cw`" instruction for writing.

`ld.cw %rd,%rs` %rs[6:0] is written to the mode setting register. (%rd: not used)
`ld.cw %rd,`*`imm7`* *imm7*[6:0] is written to the mode setting register. (%rd: not used)

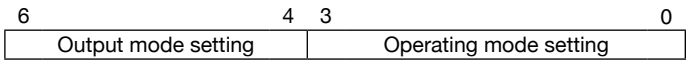


Figure25.2.1: Mode setting registers

Table 25.2.1: Mode setting

Setting (D[6:4])	Output mode	Setting (D[3:0])	Operating mode
0x0	Last 16-bit output mode Last 16 bits of the arithmetic results are read out as coprocessor output.	0x0	Initialization mode 0 Clears the arithmetic results register to 0x0.
0x1	First 16-bit output mode First 16 bits of the arithmetic results are read out as coprocessor output.	0x1	Initialization mode 1 Loads the 16-bit arithmetic augend into the last 16 bits of the arithmetic results register.
0x2 to 0x7	Reserved	0x2	Initialization mode 2 Loads the 32-bit arithmetic augend into the arithmetic results register.
		0x3	Arithmetic results reading mode Outputs the arithmetic results register data without performing calculations.
		0x4	Unsigned multiplication mode Performs unsigned multiplication.
		0x5	Signed multiplication mode Performs signed multiplication.
		0x6	Reserved
		0x7	Signed product-sum operation mode Performs signed product-sum operation.
		0x8	Unsigned division mode Performs unsigned division.
		0x9	Signed division mode Performs signed division.
		0xa to 0xf	Reserved

25.3 Multiplication

The multiplication function executes "A (32 bits) = B (16 bits) × C (16 bits)."
To perform multiplication, set the operating mode to 0x4 (unsigned multiplication) or 0x5 (signed multiplication). Next, transfer the 16-bit multiplicand (B) and 16-bit multiplier (C) to the multiplier/divider using the "`ld.ca`" instruction. Half of the arithmetic result (16 bits, A [15:0] or A[31:16], depending on output mode) is returned to the CPU register, together with the flag status.
The remaining half of the arithmetic result is read out by setting the multiplier/divider to arithmetic result reading mode.

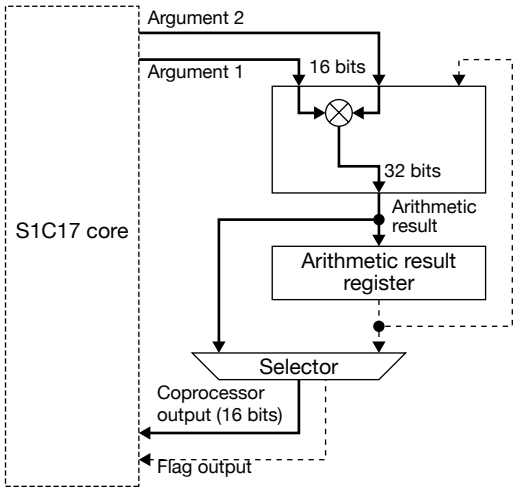


Figure 25.3.1: Multiplier mode data paths

Table 25.3.1: Multiplier mode operations

Mode setting	Instruction	Operation	Flag	Remarks
0x04 or 0x05	`ld.ca %rd,%rs`	res[31:0] ← %rd × %rs %rd ← res[15:0]	psr (CVZN) ← 0b0000	The arithmetic result register retains arithmetic results until the results are overwritten by another operation.
	(`ext` *`imm9`*) `ld.ca %rd,`*`imm7`*	res[31:0] ← %rd × *imm7/16* %rd ← res[15:0]		
0x14 or 0x15	`ld.ca %rd,%rs`	res[31:0] ← %rd × %rs %rd ← res[31:16]		
	(`ext` *`imm9`*) `ld.ca %rd,`*`imm7`*	res[31:0] ← %rd × *imm7/16* %rd ← res[31:16]		

res: Arithmetic result register

Examples:

```
ld.cw %r0,0x4   ; Mode setting (unsigned multiplication mode & last 16 bit output mode)
ld.ca %r0,%r1   ; Executes "res = %r0 × %r1" and loads the last 16 bits of the result to %r0 register.
ld.cw %r0,0x13  ; Mode setting (arithmetic result reading mode & first 16 bit output mode)
ld.ca %r1,%r0   ; Loads the first 16 bits of the result to %r1 register.
```

25.4 Division

The division function executes "A (16 bits) = B (16 bits) ÷ C (16 bits), D (16 bits) = Remainder."
To perform a division, set the operating mode to 0x8 (unsigned division) or 0x9 (signed division). Next, transfer the 16-bit dividend (B) and 16-bit divisor (C) to the multiplier/divider using the "ld.ca" instruction. The quotient will be placed in the lower 16 bits of the arithmetic result register, while the remainder is placed in the upper 16 bits. When the calculation is completed, the 16 bits corresponding to the quotient or remainder as specified in the output mode and the flag status are returned to the CPU register. The other 16 bits of the arithmetic result can be read out by setting the multiplier/divider to arithmetic result reading mode.

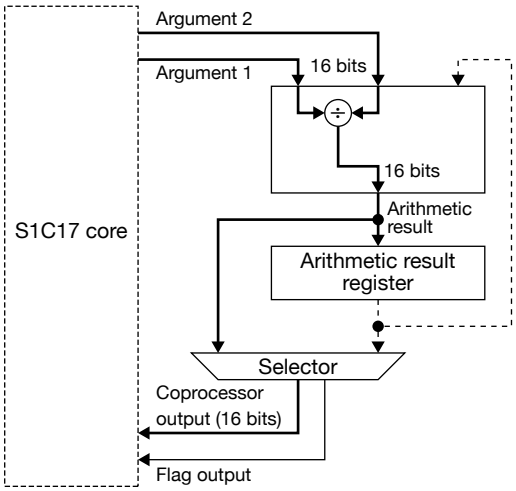


Figure 25.4.1 Division mode data path

Table 25.4.1 Division mode operations

Mode setting	Instruction	Operation	Flag	Remarks
0x08 or 0x09	`ld.ca %rd,%rs`	res[31:0] ← %rd ÷ %rs %rd ← res[15:0] (quotient)	psr (CVZN) ← 0b0000	The arithmetic result register retains the calculated result until it is overwritten by the result of another arithmetic operation.
	(`ext` *imm9*) `ld.ca %rd,`*imm7*	res[31:0] ← %rd ÷ *imm7/16* %rd ← res[15:0] (quotient)		
0x18 or 0x19	`ld.ca %rd,%rs`	res[31:0] ← %rd ÷ %rs %rd ← res[31:16] (remainder)		
	(`ext` *imm9*) `ld.ca %rd,`*imm7*	res[31:0] ← %rd ÷ *imm7/16* %rd ← res[31:16] (remainder)		

res: Arithmetic result register

Example:

```
ld.cw %r0,0x8   ; Mode setting (unsigned division mode & lower 16-bit output mode)
ld.ca %r0,%r1   ; Executes "res = %r0 + %r1" and loads the lower 16 bits (quotient) of the result to the
                  %r0 register.
ld.cw %r0,0x13  ; Mode setting (arithmetic result reading mode & upper 16-bit output mode)
ld.ca %r1,%r0   ; Loads the upper 16 bits (remainder) of the result to the %r1 register.
```

25.5 Product-sum Operation

The product-sum operation function executes "A (32 bits) = B (16 bits) × C (16 bits) + A (32 bits)."
The initial value (A) must be set to the arithmetic result register before performing product-sum operations.
To clear the arithmetic result register (A = 0), set the operating mode to 0x0. There is no need to send 0x0 to the multiplier/divider using separate instructions.
To load 16-bit or 32-bit values to the arithmetic result register, set the operating mode to 0x1 (16 bits) or 0x2 (32 bits). Next, transfer the initial value to the multiplier/divider using the "`ld.cf`" instruction.

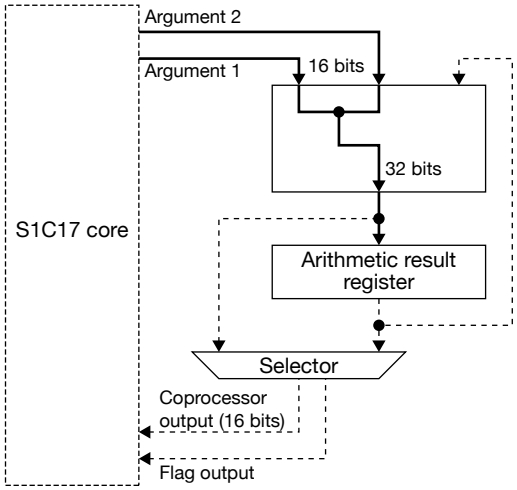


Figure 25.5.1: Initialization mode data paths

Table 25.5.1: Arithmetic result register initialization

Mode setting	Instruction	Operation	Remarks
0x0	–	res[31:0] ← 0x0	Initializes using operating mode settings only (no data transfer).
0x1	`ld.cf %rd,%rs`	res[31:16] ← 0x0 res[15:0] ← %rs	
	(`ext` *imm9*) `ld.cf %rd,`*imm7*	res[31:16] ← 0x0 res[15:0] ← *imm7/16*	
0x2	`ld.cf %rd,%rs`	res[31:16] ← %rd res[15:0] ← %rs	
	(`ext` *imm9*) `ld.cf %rd,`*imm7*	res[31:16] ← %rd res[15:0] ← *imm7/16*	

res: Arithmetic result register

To perform product-sum operations, set the operating mode to 0x7 (signed product-sum). Next, transfer the 16-bit multiplicand (B) and 16-bit multiplier (C) to the multiplier/divider using the "`ld.ca`" instruction. Half of the arithmetic result (16 bits, A [15:0] or A[31:16], depending on output mode) is returned to the CPU register together with the flag status. The remaining half of the arithmetic result is read out by setting the multiplier/divider to arithmetic result reading mode.
The PSR overflow flag (V) is set to 1 by the arithmetic results. Other flags are cleared to 0.
Transfer the required number of multiplicands and multipliers to continue product-sum operations without switching to arithmetic result reading mode. In this case, there is no need to set to product-sum operation mode each time data is sent.

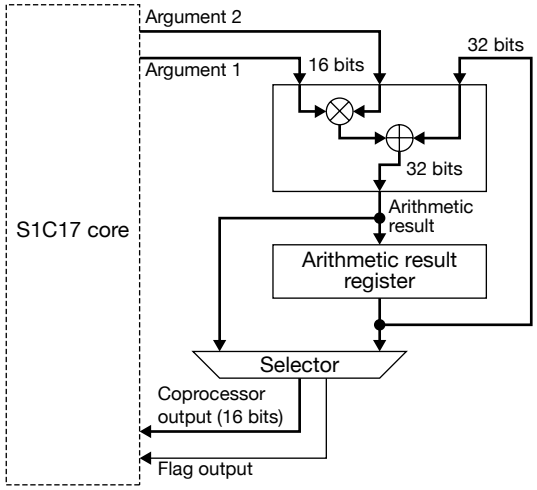


Figure 25.5.2: Product-sum operation mode data paths

Table 25.5.2: Product-sum operation mode operations

Mode setting	Instruction	Operation	Flag	Remarks
0x07	`ld.ca %rd,%rs`	res[31:0] ← %rd × %rs + res[31:0] %rd ← res[15:0]	If overflow occurs psr (CVZN) ← 0b0100 Other cases psr (CVZN) ← 0b0000	The arithmetic result register retains arithmetic results until the results are overwritten by another operation.
	(`ext` *imm9*) `ld.ca %rd,`*imm7*	res[31:0] ← %rd × *imm7/16* + res[31:0] %rd ← res[15:0]		
0x17	`ld.ca %rd,%rs`	res[31:0] ← %rd × %rs + res[31:0] %rd ← res[31:16]		
	(`ext` *imm9*) `ld.ca %rd,`*imm7*	res[31:0] ← %rd × *imm7/16* + res[31:0] %rd ← res[31:16]		

res: Arithmetic result register

Examples:

```
ld.cw %r0,0x7  ;  Mode setting (signed product-sum operation mode & last 16 bit output mode)
ld.ca %r0,%r1  ;  Executes "res = %r0 × %r1 + res" and loads the last 16 bits of the result to %r0 register.
ld.cw %r0,0x13 ;  Mode setting (arithmetic result reading mode & first 16 bit output mode)
ld.ca %r1,%r0  ;  Loads first 16 bits of the result to %r1 register.
```

Overflow flag (V) setting conditions

If the multiplication result sign, arithmetic result register sign, and arithmetic result sign satisfy the following conditions in product-sum operations, an overflow occurs, and the overflow flag (V) is set to 1.

Table 25.5.3: Overflow flag (V) setting conditions

Mode setting	Multiplication result sign	Arithmetic result register sign	Arithmetic result sign
0x07	0 (Positive)	0 (Positive)	1 (Negative)
0x07	1 (Negative)	1 (Negative)	0 (Positive)

An overflow occurs if positive values are summed giving a negative result in product-sum operations or if negative values are summed giving a positive result. The result is retained in the coprocessor until the overflow flag (V) is cleared.

Overflow flag (V) clear conditions

The overflow flag (V) set is cleared if the "`ld.ca`" instruction is executed for product-sum operation without causing an overflow or if the "`ld.ca`" or "`ld.cf`" instruction is executed in other than arithmetic result reading mode.

25.6 Arithmetic Results Reading

Since the "ld.ca" instruction cannot load 32-bit arithmetic results to the CPU register, multiplication and product-sum operation return half of the arithmetic result (16 bits, A[15:0] or A[31:16], depending on output mode) together with the flag status to the CPU register. The remaining half of the arithmetic result is read by setting the multiplier to arithmetic result reading mode. The arithmetic result register retains arithmetic results until the results are overwritten by another operation.

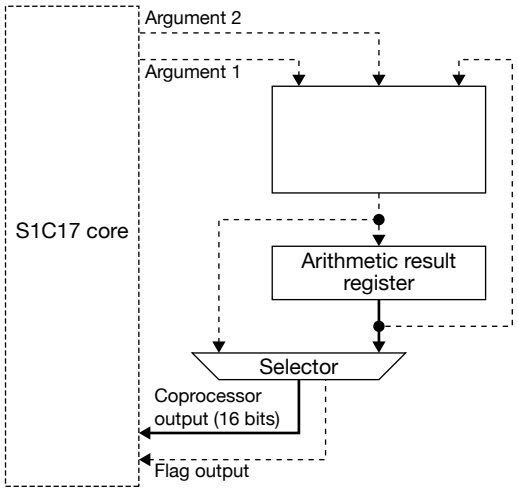


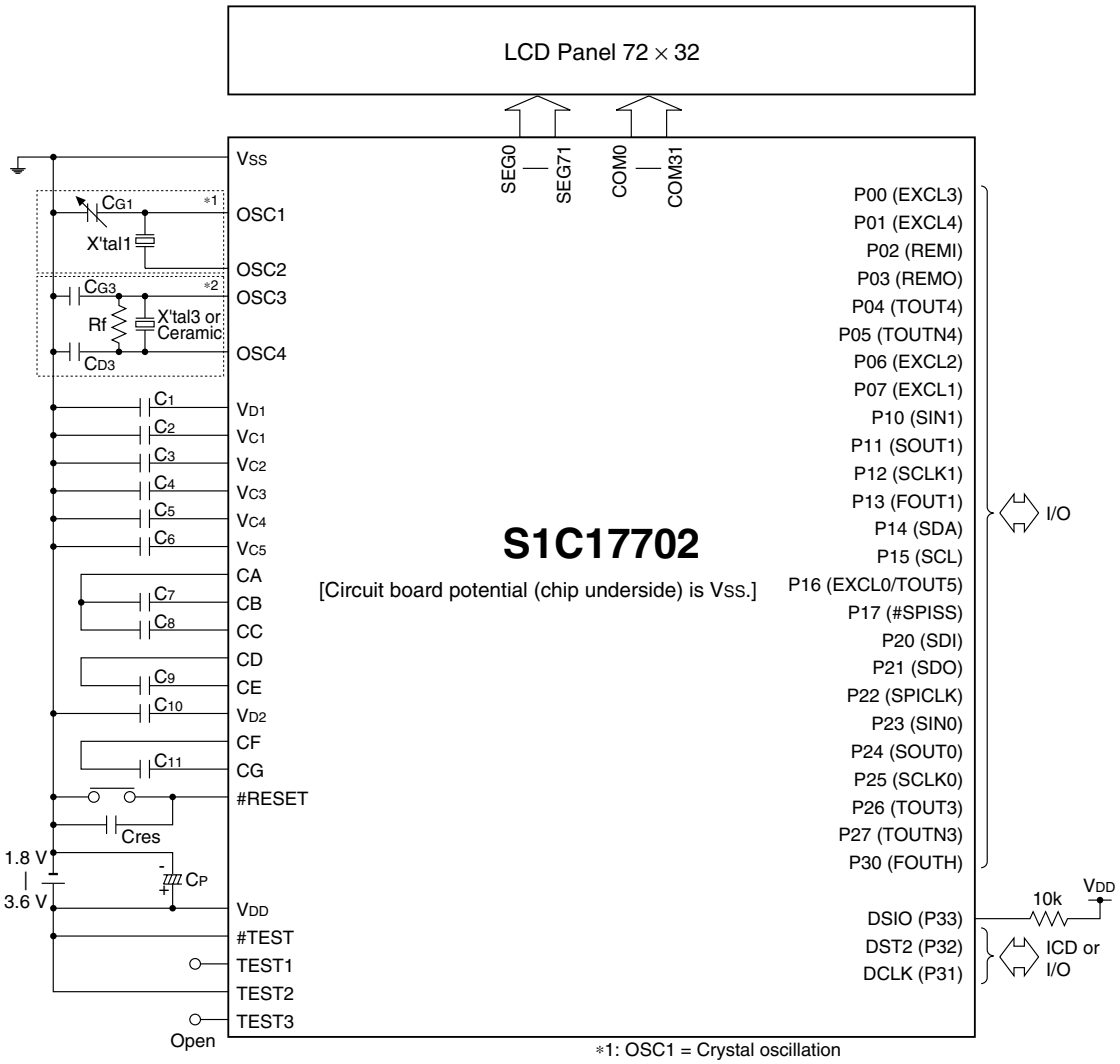
Figure 25.6.1: Arithmetic result reading mode data paths

Table 25.6.1: Arithmetic result reading mode operations

Mode setting	Instruction	Operation	Flag	Remarks
0x03	ld.ca %rd,%rs	%rd ← res[15:0]	psr (CVZN) ← 0b0000	This operating mode does not affect the arithmetic result register.
	ld.ca %rd,*imm7*	%rd ← res[15:0]		
0x13	ld.ca %rd,%rs	%rd ← res[31:16]		
	ld.ca %rd,*imm7*	%rd ← res[31:16]		

res: Arithmetic result register

26 Basic External Connection Diagram



*1: OSC1 = Crystal oscillation
*2: OSC3 = Crystal or ceramic oscillation

Recommended values for external components

Symbol	Name	Recommended value
X'tal1	Crystal oscillator	32.768 kHz
C_{G1}	Trimmer capacitor	0 to 25 pF
X'tal3	Crystal oscillator	0.2 to 8 MHz
Ceramic	Ceramic oscillator	0.2 to 8 MHz
Rf	Drain resistor	1 MΩ
C_{G3}	Gate capacitor	15 pF (Crystal oscillation)
		30 pF (Ceramic oscillation)
C_{D3}	Drain capacitor	15 pF (Crystal oscillation)
		30 pF (Ceramic oscillation)
Cres	#RESET pin capacitor	0.47 µF

Symbol	Name	Recommended value
C_1	Capacitor between V_{SS} and V_{D1}	0.1 µF
C_2	Capacitor between V_{SS} and V_{C1}	0.1 µF
C_3	Capacitor between V_{SS} and V_{C2}	0.1 µF
C_4	Capacitor between V_{SS} and V_{C3}	0.1 µF
C_5	Capacitor between V_{SS} and V_{C4}	0.1 µF
C_6	Capacitor between V_{SS} and V_{C5}	0.1 µF
C_7 to C_9	Booster capacitor	0.1 µF
C_{10}	Capacitor between V_{SS} and V_{D2}	0.1 µF
C_{11}	Booster capacitor	0.1 µF
C_P	Inter-power supply capacitor	3.3 µF

27 Electrical Characteristics

27.1 Absolute Maximum Ratings

(V_{SS} = 0V)

Item	Code	Condition	Rating	Units
Power supply voltage	V_{DD}		-0.3 to 4.0	V
LCD power supply voltage	V_{C5}		-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 for all pins	-20	mA
Low-level output current	I_{OL}	1 pin	5	mA
		Total for all pins	20	mA
Permissible losses*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 s (leads)	–

*1: For plastic package

27.2 Recommended Operating Conditions

(Ta = -25 to 70°C)

Item	Code	Condition	Min.	Typ.	Max.	Units
Operating power supply voltage	V_{DD}	Normal operating mode	1.8		3.6	V
		Flash memory programming mode	2.7		3.6	V
Operating frequency	f_{IOSC}			2.5		MHz
	f_{OSC3}	Crystal/ceramic oscillation	0.2		8.2	MHz
	f_{OSC1}	Crystal oscillation		32.768	100	kHz
Capacitor between V_{SS} and V_{D1}*1	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 V_{SS} and V_{C4}*1	C_5			0.1		µF
Capacitor between V_{SS} and V_{C5}*1	C_6			0.1		µF
Capacitor between CA and CB*1	C_7			0.1		µF
Capacitor between CA and CC*1	C_8			0.1		µF
Capacitor between CD and CE*1	C_9			0.1		µF
Capacitor between V_{SS} and V_{D2}*1	C_{10}			0.1		µF
Capacitor between CF and CG*1	C_{11}			0.1		µF

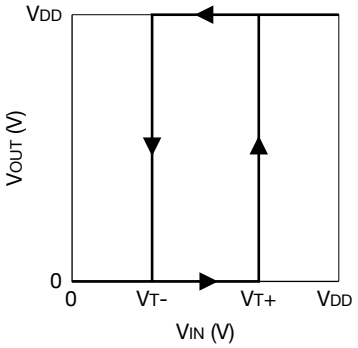
*1: No capacitors are required if no LCD drive is used. V_{C1} to V_{C5} and CA to CG should be left open.

27.3 DC Characteristics

Unless otherwise specified: V_{DD} = 1.8 to 3.6 V, V_{SS} = 0 V, Ta = -25 to 70°C

Item	Code	Condition	Rating	Units	Max.	Units
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 voltage (1)	V_{T1+}	#RESET	0.5V_{DD}		0.9V_{DD}	V
Low level Schmitt input voltage (1)	V_{T1-}	#RESET	0.1V_{DD}		0.5V_{DD}	V
High level Schmitt input voltage (2) *1	V_{T2+}	P*xx*	0.5V_{DD}		0.9V_{DD}	V
Low level Schmitt input voltage (2) *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
Input leakage current	I_{LI}	P*xx*, #RESET	-1		1	µA
Output leakage current	I_{LO}	P*xx*	-1		1	µA
Input pull-up resistance	R_{IN}	P*xx*, #RESET	100		500	kΩ
Input pin capacitance	C_{IN}	P*xx*, V_{IN} = 0 V, f = 1 MHz, Ta = 25°C			15	pF
Segment, common output current	I_{SEGH}	SEG*xx*, COM*xx*, V_{SEGH} = V_{C5} - 0.1 V			-5	µA
	I_{SEGL}	SEG*xx*, COM*xx*, V_{SEGL} = 0.1 V	5			µA

*1: When Schmitt input is enabled



27.4 Analog Circuit Characteristics

LCD driver

Since Typ values for the LCD driver will vary depending on panel load (e.g., panel size, drive duty, display pixel illumination number, display patterns), they should be evaluated by connecting the actual panel to be used. See "27.8 Characteristics Graph" for more information on load characteristics.

Unless otherwise stated, V_{DD} = 1.8 V to 3.6 V, V_{SS} = 0 V, Ta = 25°C, C1 to C11 = 0.1 µF, output is checkerboard pattern, no panel load.

Item	Symbol	Condition		Min.	Typ.	Max.	Units
LCD drive voltage	V_{C1}	1 MΩ load resistor connected between Vss and V_{C1}		0.18V_{C5}		0.22V_{C5}	V
	V_{C2}	1 MΩ load resistor connected between Vss and V_{C2}		0.39V_{C5}		0.43V_{C5}	V
	V_{C3}	1 MΩ load resistor connected between Vss and V_{C3}		0.59V_{C5}		0.63V_{C5}	V
	V_{C4}	1 MΩ load resistor connected between Vss and V_{C4}		0.79V_{C5}		0.83V_{C5}	V
	V_{C5}	1 MΩ load resistor connected between Vss and V_{C5}	LC[3:0] = 0x0	Typ. × 0.94	4.20	Typ. × 1.06	V
			LC[3:0] = 0x1		4.30		V
			LC[3:0] = 0x2		4.40		V
			LC[3:0] = 0x3		4.50		V
			LC[3:0] = 0x4		4.60		V
			LC[3:0] = 0x5		4.70		V
			LC[3:0] = 0x6		4.80		V
			LC[3:0] = 0x7		4.90		V
			LC[3:0] = 0x8		5.00		V
			LC[3:0] = 0x9		5.10		V
			LC[3:0] = 0xa		5.20		V
			LC[3:0] = 0xb		5.30		V
			LC[3:0] = 0xc		5.40		V
			LC[3:0] = 0xd		5.50		V
			LC[3:0] = 0xe		5.60		V
			LC[3:0] = 0xf		5.70		V

SVD circuit

Unless otherwise stated, V_{DD} = 1.8 V to 3.6 V, V_{SS} = 0 V, Ta = 25°C.

Item	Symbol	Condition	Min.	Typ.	Max.	Units
SVD voltage	V_{SVD}	SVDC[3:0] = 0x0	–	–	–	V
		SVDC[3:0] = 0x1	Typ. × 0.91	1.8	Typ. × 1.09	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 response time	t_{SVD}				500	µs

Flash memory

Unless otherwise stated, V_{DD} = 1.8 V to 3.6 V, V_{SS} = 0 V, Ta = 25°C.

Item	Symbol	Condition	Min.	Typ.	Max.	Units
Erase time*1	t_{SE}	4 Kbyte erase			25	ms
Program time*1	t_{BP}	16 byte writing			20	µs
Overwriting cycles*2	C_{FEP}	Data retention guaranteed 10 years	1000			Cycles

*1: Including data transfer and verification but excluding erase/program start control times.
*2: Treating erasing + writing or writing only as a single cycle.

27.5 Current Consumption

Unless otherwise stated, V_{DD} = 1.8 V to 3.6 V, V_{SS} = 0 V, Ta = 25°C, C_1 to C_{11} = 0.1 µF, no LCD panel load, PCKEN[1:0] = 0x3 (ON), VD1MD = 0x0, FLCYC[2:0] = 0x4 (1cycle), CCKGR[1:0] = 0x1 (gear ratio 1/1)

Item	Symbol	Condition	Min.	Typ.	Max.	Units
SLEEP current consumption	I_{SLP}	OSC1 = OFF, IOSC = OFF, OSC3 = OFF		1.2	3.0	µA
HALT current consumption	I_{HALT1}	OSC1 = 32 kHz, IOSC = OFF, OSC3 = OFF, PCKEN[1:0] = 0x0 (OFF)		2.7	5.0	µA
	I_{HALT2}	OSC1 = 32 kHz, IOSC = OFF, OSC3 = 8 MHz (ceramic)		580	800	µA
	I_{HALT3}	OSC1 = 32 kHz, IOSC = ON, OSC3 = OFF		220	300	µA
Current consumption when operating *1	I_{EXE1}	OSC1 = 32 kHz, IOSC = OFF, OSC3 = OFF, CPU = OSC1		16	25	µA
		OSC1 = 32 kHz, IOSC = OFF, OSC3 = OFF, CCKGR[1:0] = 0x2 (gear ratio 1/4), CPU = OSC1		9	14	µA
	I_{EXE2}	OSC1 = 32 kHz, IOSC = OFF, OSC3 = 1 MHz (ceramic), CPU = OSC3		450	650	µA
		OSC1 = 32 kHz, IOSC = OFF, OSC3 = 8 MHz (ceramic), CPU = OSC3		3300	4700	µA
		OSC1 = 32 kHz, IOSC = OFF, OSC3 = 8 MHz (ceramic), CCKGR[1:0] = 0x2 (gear ratio 1/4), CPU = OSC3		1600	2400	µA
	I_{EXE3}	OSC1 = 32 kHz, IOSC = ON, OSC3 = OFF, VD1MD = 1, CPU = IOSC		1100	1600	µA
	I_{EXE11}	OSC1 = 32 kHz, IOSC = OFF, OSC3 = OFF, VD1MD = 1, CPU = OSC1		35	50	µA
	I_{EXE21}	OSC1 = 32 kHz, IOSC = OFF, OSC3 = 1 MHz (ceramic), VD1MD = 1, CPU = OSC3		780	1100	µA
		OSC1 = 32 kHz, IOSC = OFF, OSC3 = 8 MHz (ceramic), VD1MD = 1, CPU = OSC3		5600	8000	µA
	I_{EXE31}	OSC1 = 32 kHz, IOSC = ON, OSC3 = OFF, VD1MD = 1, CPU = IOSC		1900	2700	µA
Current consumption when operating in heavy load protection mode *1	I_{EXE1H}	OSC1 = 32 kHz, IOSC = OFF, OSC3 = OFF, CPU = OSC1, HVLD = 1		22	33	µA
LCD circuit current *2	I_{LCD1}	DSPC[1:0] = 0x3 (all on), LC[3:0] = 0xf, f_{OSC1} = 32.768 kHz, V_{DD} = 2.5 to 3.6 V		7	14	µA
LCD circuit current for boosting power supply voltage *3	I_{LCD2}	DSPC[1:0] = 0x3 (all on), LC[3:0] = 0xf, f_{OSC1} = 32.768 kHz, PBON = 1, V_{DD} = 1.8 to 2.5 V		20	40	µA
LCD circuit current in heavy load protection mode *4	I_{LCD1H}	DSPC[1:0] = 0x3 (all on), LC[3:0] = 0xf, f_{OSC1} = 32.768 kHz, V_{DD} = 2.5 to 3.6 V, LHVLD = 1		10	20	µA
LCD circuit current for boosting power supply voltage in heavy load protection mode *5	I_{LCD2H}	DSPC[1:0] = 0x3 (all on), LC[3:0] = 0xf, f_{OSC1} = 32.768 kHz, PBON = 1, V_{DD} = 2.5 to 3.6 V, LHVLD = 1		38	75	µA
SVD circuit current *6	I_{SVD}	V_{DD} = 3.6 V		5	10	µA
Flash EEPROM erasure current *7	I_{FERS}	8 MHz CPU operations, VD1MD = 1		7	14	mA
Flash EEPROM programming current *8	I_{FPRG}	8 MHz CPU operations, VD1MD = 1		7	14	mA

*1: Value for continuous operation while fetching “ALU commands 60.5%, branching commands 17%, memory reading 12%, memory writing 10.5%” program from Flash memory.

*2: Added to HALT/operation current consumption for LCD circuit operation.
Current consumption may be higher, depending on display patterns and panel loads.

*3: Added to operation current consumption in heavy load protection mode for LCD circuit operation.
Current consumption may be higher, depending on display patterns and panel loads.

*4: Added to HALT/operation current consumption for power supply voltage booster circuit + LCD circuit operation.
Current consumption may be higher, depending on display patterns and panel loads.

*5: Added to operation current consumption in heavy load protection mode for power supply voltage booster circuit + LCD circuit operation.
Current consumption may be higher, depending on display patterns and panel loads.

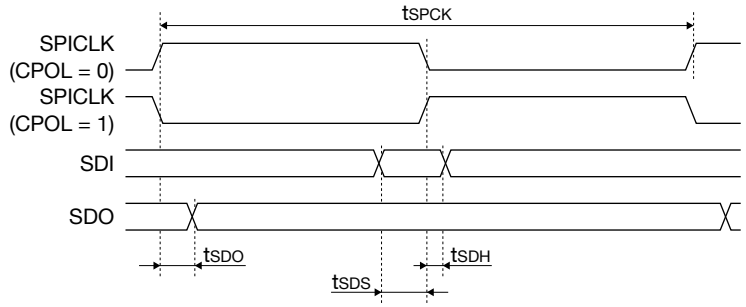
*6: Added to operation current consumption/operation current consumption in heavy load protection mode for SVD circuit operation.

*7: Added to operation current consumption for self programming erasing.

*8: Added to operation current consumption for self programming operation.

27.6 AC Characteristics

27.6.1 SPI AC Characteristics



Master mode

Unless otherwise specified: V_{DD} = 1.8 to 3.6 V, V_{SS} = 0 V, Ta = -25 to 70°C

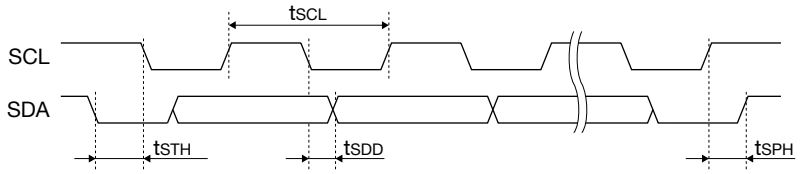
Item	Code	Min.	Typ.	Max.	Units
SPICLK cycle time	t_{SPCK}	500			ns
SDI setup time	t_{SDS}	70			ns
SDI hold time	t_{SDH}	10			ns
SDO output delay time	t_{SDO}			20	ns

Slave mode

Unless otherwise specified: V_{DD} = 1.8 to 3.6 V, V_{SS} = 0 V, Ta = -25 to 70°C

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

27.6.2 I²C AC Characteristics

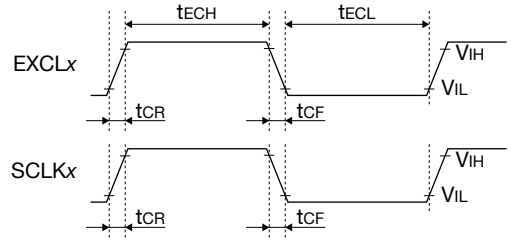


Unless otherwise specified: V_{DD} = 1.8 to 3.6 V, V_{SS} = 0 V, Ta = -25 to 70°C

Item	Code	Min.	Typ.	Max.	Units
SCL cycle time	t_{SCL}	2500			ns
Start condition hold time	t_{STH}	1/f_{SYS}			ns
Data output delay time	t_{SDD}	1/f_{SYS}			ns
Stop condition hold time	t_{SPH}	1/f_{SYS}			ns

∗ f_{SYS}: System operation clock frequency

27.6.3 External Clock Input AC Characteristics

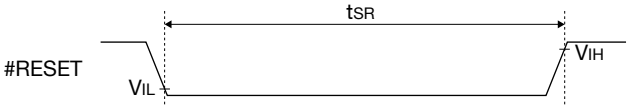


Unless otherwise specified: V_{DD} = 1.8 to 3.6 V, V_{SS} = 0 V, V_{IH} = 0.8V_{DD}, V_{IL} = 0.2V_{DD}, Ta = -25 to 70°C

Item	Code	Min.	Typ.	Max.	Units
EXCL*x* input High pulse width	t_{ECH}	1/f_{SYS}			s
EXCL*x* input Low pulse width	t_{ECL}	1/f_{SYS}			s
UART transfer rate	R_U			460,800	bps
UART transfer rate (IrDA mode)	R_{UIrDA}			115,200	bps
Input rising edge time	t_{CR}			80	ns
Input drop-off time	t_{CF}			80	ns
OSC3 clock cycle time	t_{OSC3}	125			ns
OSC3 clock input duty	t_{OSC3D}	46		54	%

∗f_{SYS}: System operation clock frequency

27.6.4 System AC Characteristics



Unless otherwise specified: V_{DD} = 1.8 to 3.6 V, V_{SS} = 0 V, V_{IH} = 0.8V_{DD}, V_{IL} = 0.2V_{DD}, Ta = -25 to 70°C

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

27.7 Oscillation Characteristics

Oscillation characteristics depend on various parameters, including circuit board patterns and the components used. Use the values given for the following characteristics as reference values. In particular, if ceramic or crystal oscillators are used for OSC3, parameters such as capacitance and resistance should have the values recommended by the oscillator manufacturer. Oscillation start time is an important factor, since it establishes the wait time when using the OSC3 clock.

OSC1 crystal oscillator

Unless otherwise specified: V_{DD} = 1.8 to 3.6 V, V_{SS} = 0 V, Ta = 25°C,
crystal oscillator = FC-255X (R_1 = 30 kΩ to 70 kΩ Typ., C_L = 12.5 pF)*1,
C_{G1} = 25 pF external, C_{D1} = internal.

Item	Symbol	Condition	Min.	Typ.	Max.	Units
Oscillation start time	t_{sta}				3	s
External gate capacitance	C_{G1}	Including board capacitance	0		25	pF
Internal drain capacitance	C_{D1}	For chip		10		pF
Frequency IC deviation	∂f/∂IC	V_{DD} = Constant	-10		10	ppm
Frequency power supply voltage deviation	∂f/∂V				1	ppm/V
Frequency adjustment range	∂f/∂C_G	V_{DD} = Constant, C_G = 0 to 25 pF	25			ppm

*1: FC-255X: Epson Toyocom

OSC3 crystal oscillator

Unless otherwise specified: V_{DD} = 1.8 to 3.6 V, V_{SS} = 0 V, Ta = 25°C,
crystal oscillator = CA-301*1, R_f = 1 MΩ, C_{G3} = C_{D3} = 15 pF

Item	Symbol	Condition	Min.	Typ.	Max.	Units
Oscillation start time*2	t_{sta}				10	ms

*1: CA-301: Seiko Epson
*2: The crystal oscillator oscillation start time varies with the crystal oscillator used and CG3 and CD3.

OSC3 ceramic oscillator

Unless otherwise specified: V_{DD} = 1.8 to 3.6 V, V_{SS} = 0 V, Ta = 25°C,
ceramic oscillator = KBR-4.0MSB/KBR-8.0MSB*1, R_f = 1 MΩ,
C_{G3} = C_{D3} = 30 pF

Item	Symbol	Condition	Min.	Typ.	Max.	Units
Oscillation start time*2	t_{sta}				1	ms

*1: KBR-4.0MSB/KBR-8.0MSB: Kyocera
*2: Ceramic oscillator oscillation start time varies depending on the ceramic oscillator used and CG3 and CD3.

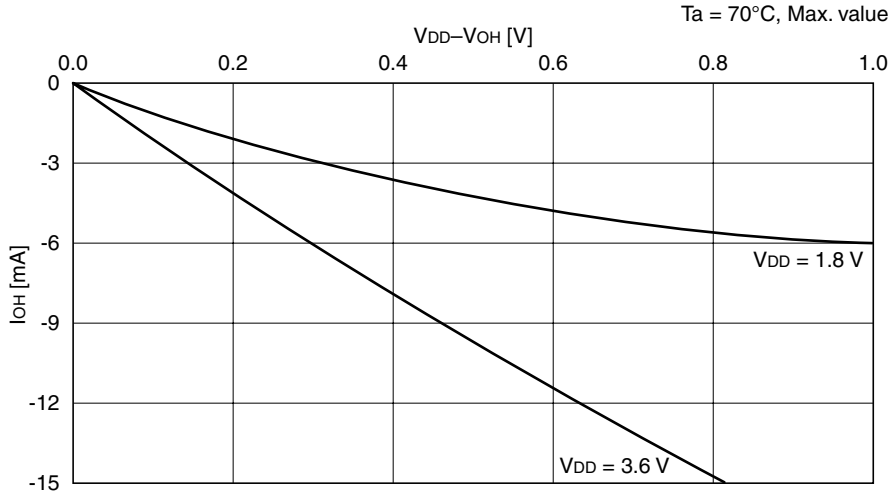
IOSC CR oscillator

Unless otherwise specified: V_{DD} = 1.8 to 3.6 V, V_{SS} = 0 V, Ta = 25°C

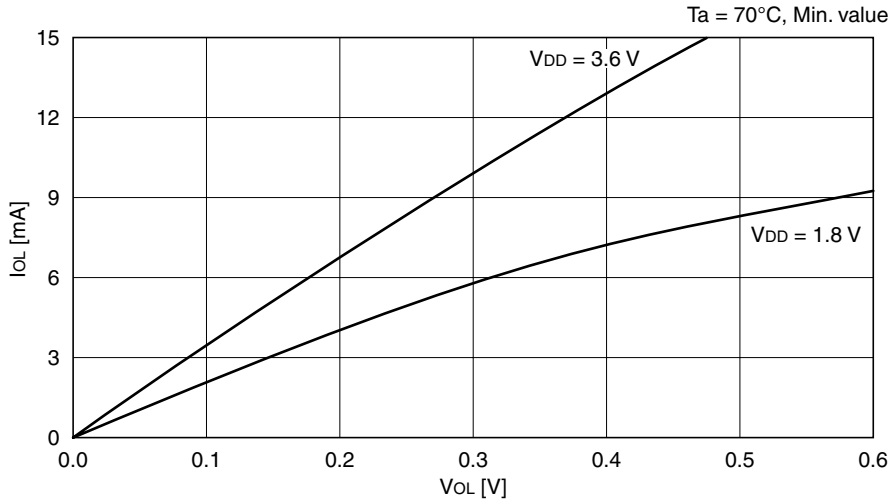
Item	Symbol	Condition	Min.	Typ.	Max.	Units
Oscillation start time	t_{sta}				5	µs
Oscillation frequency	f_{IOSC}	V_{D1} = 1.8 V	2.16	2.70	3.24	MHz

27.8 Characteristics Graphs (Reference Values)

High-level output current characteristics

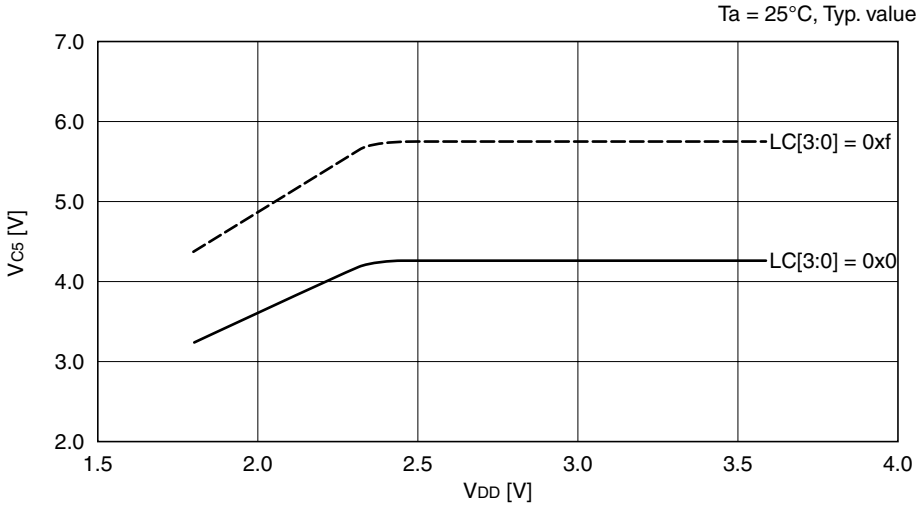


Low-level output current characteristics



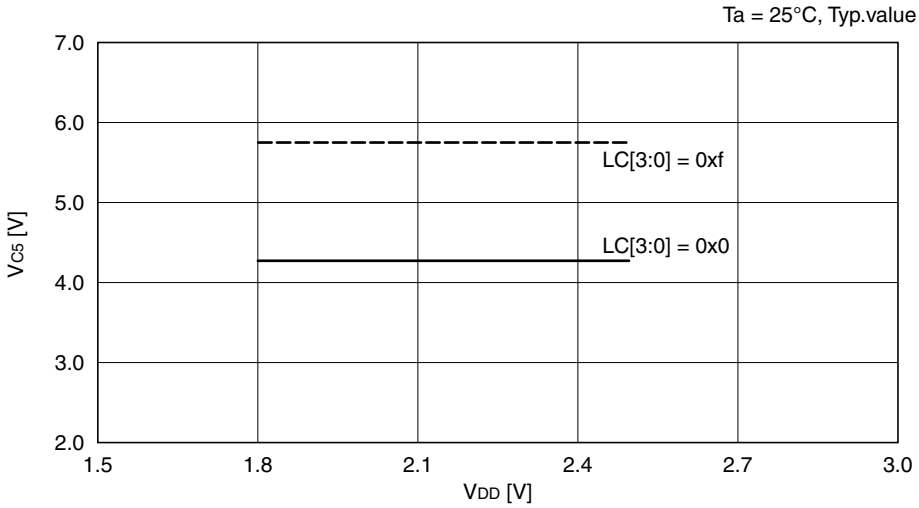
LCD drive voltage power supply voltage characteristics (Not using power supply voltage booster circuit)

With 1 MΩ load resistance connected between V_{SS} and V_{C5} (no panel load)



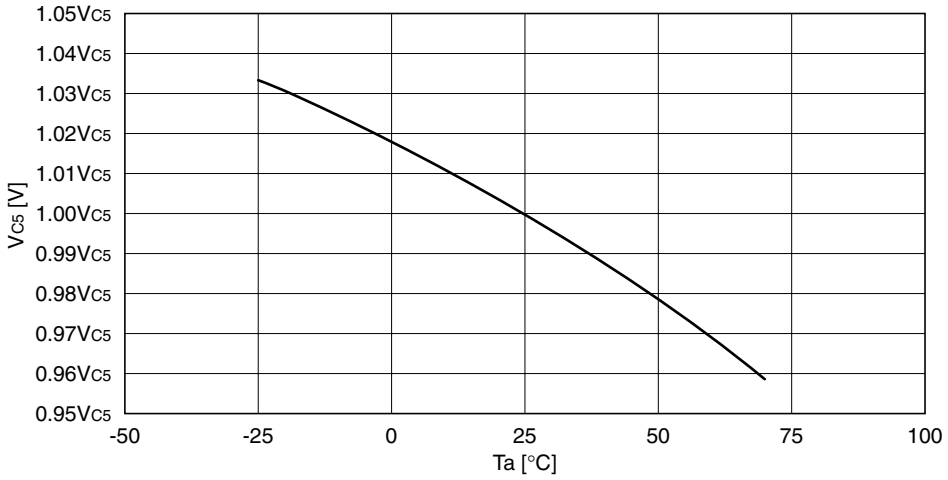
LCD drive voltage power supply voltage characteristics (Using power supply voltage booster circuit)

With 1 MΩ load resistance connected between V_{SS} and V_{C5} (no panel load)



LCD drive voltage temperature characteristics

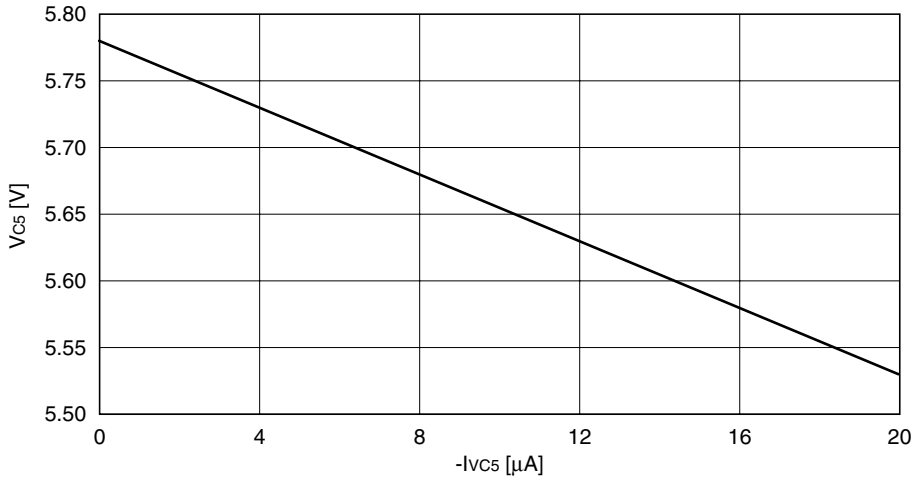
Typ. value



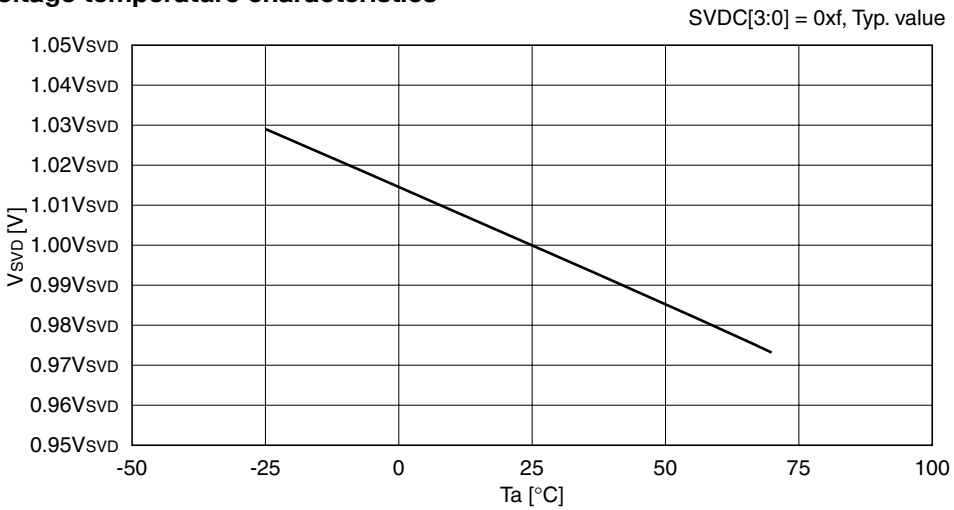
LCD drive voltage load characteristics

With load connected to V_{C5} only

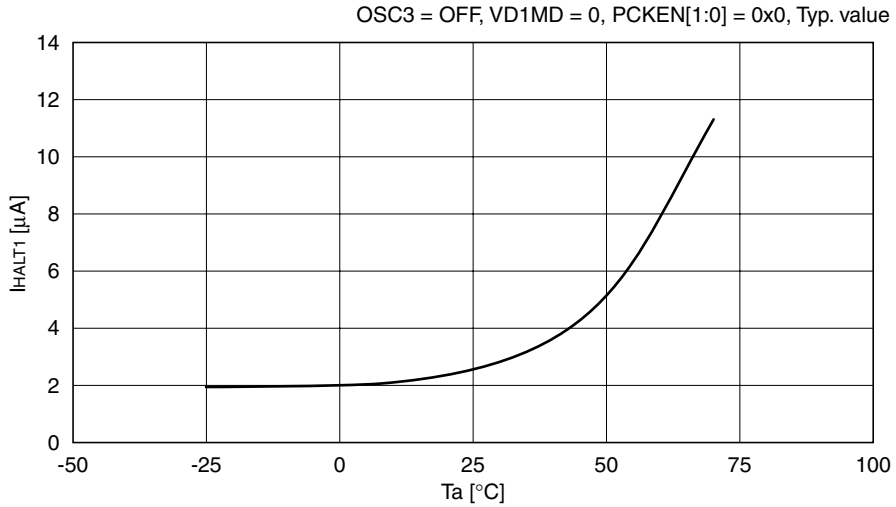
LC[3:0] = 0xf, Ta = 25°C, Typ. value



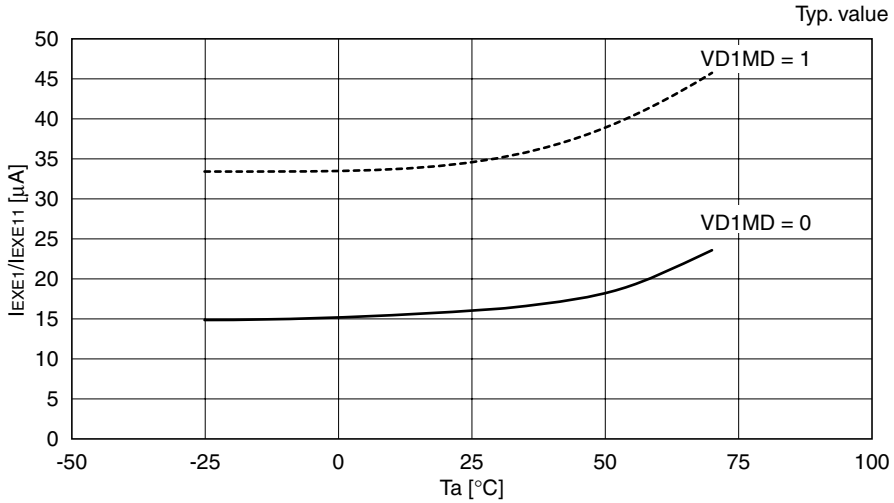
SVD voltage temperature characteristics



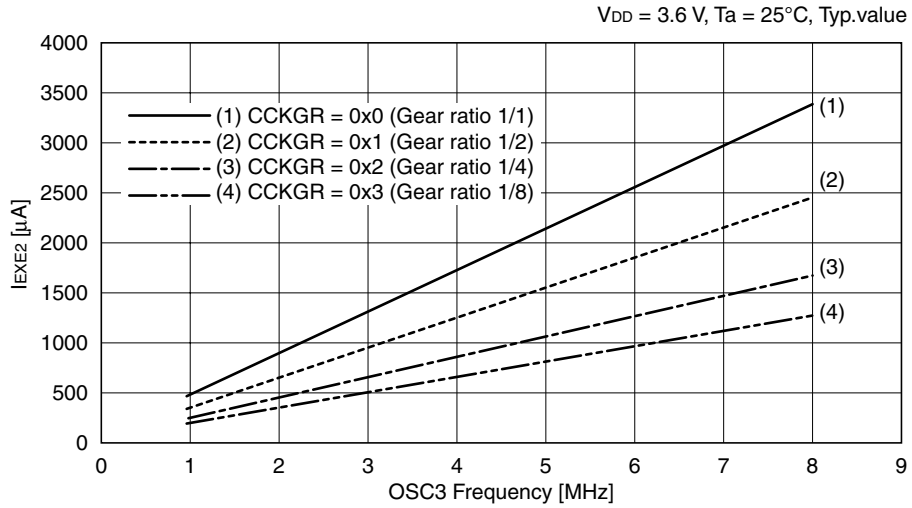
HALT current consumption temperature characteristics (OSC1 operation) (Crystal oscillation, fOSC1 = 32.768 kHz)



Execution current consumption temperature characteristics (OSC1 operation) (Crystal oscillation, f_{OSC1} = 32.768 kHz)



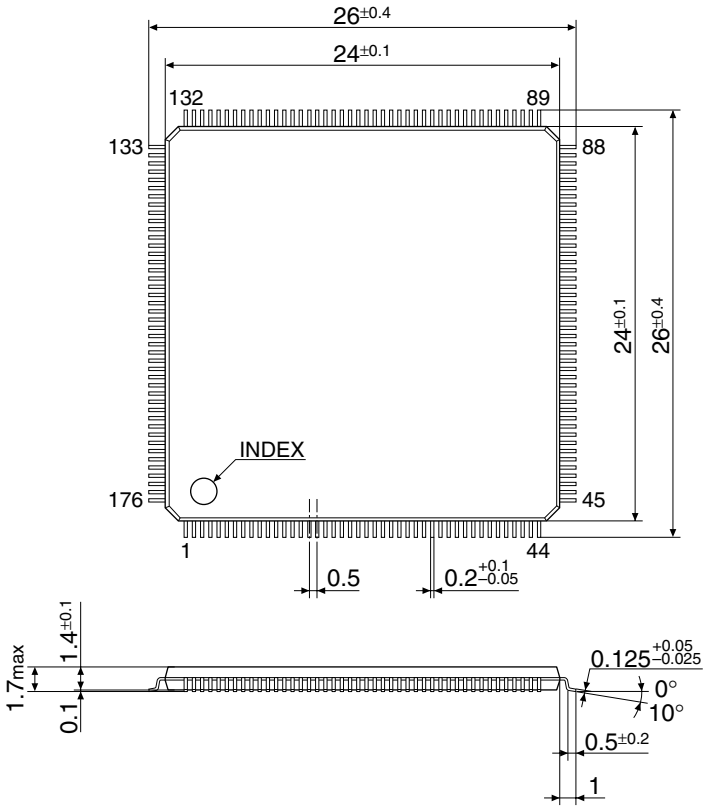
Execution current consumption frequency characteristics (OSC3 operation) (Crystal oscillation/ceramic oscillation)



28 Package

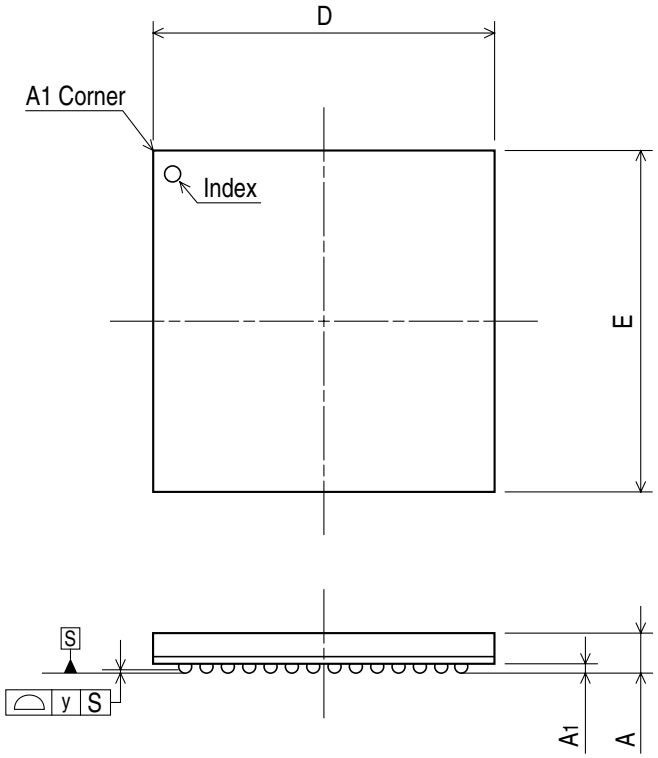
QFP21-176 pin package

(Units: mm)

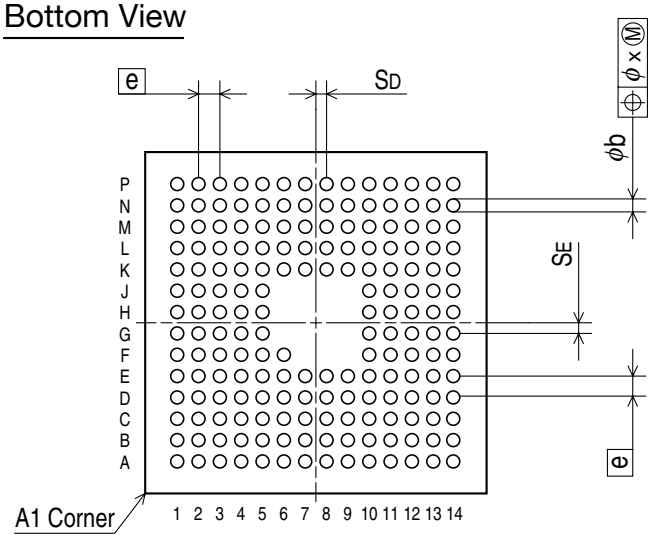


VFBGA8H-181 package

Top View



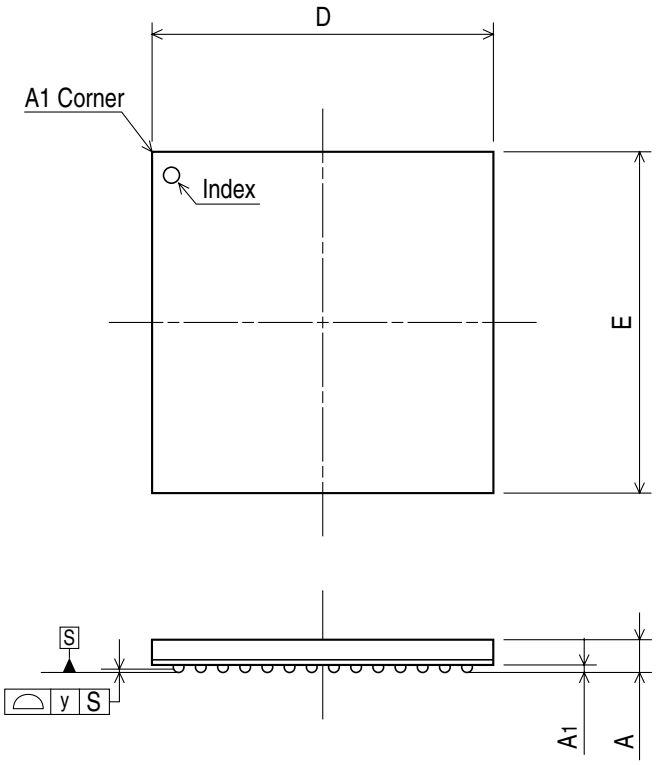
Bottom View



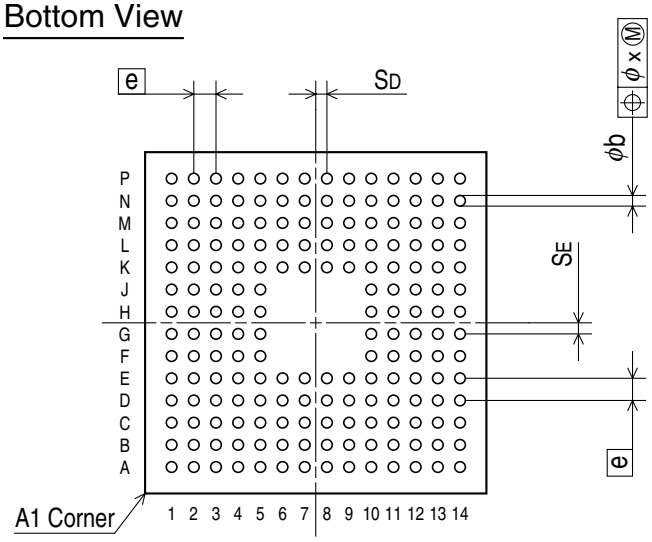
Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
D	–	8	–
E	–	8	–
A	–	–	1.0
A_1	–	0.23	–
e	–	0.5	–
b	0.26	–	0.36
x	–	–	0.08
y	–	–	0.1
S_D	–	0.25	–
S_E	–	0.25	–

VFBGA10H-180 package

Top View



Bottom View



Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
D	–	10	–
E	–	10	–
A	–	–	1.0
A_1	–	0.22	–
e	–	0.65	–
b	0.27	–	0.37
x	–	–	0.08
y	–	–	0.1
S_D	–	0.325	–
S_E	–	0.325	–

Appendix A I/O Register List

Internal Peripheral Circuit Area 1 (0x4000 onward)

Peripheral circuit	Address	Register name		Function
Prescaler (8-bit device)	0x4020	PSC_CTL	Prescaler Control Register	Prescaler start/stop control
	0x4021–0x403f	–	–	Reserved
UART (with IrDA) Ch.0 (8-bit device)	0x4100	UART_ST0	UART Ch.0 Status Register	Transfer, buffer, error status display
	0x4101	UART_TXD0	UART Ch.0 Transmit Data Register	Transmission data
	0x4102	UART_RXD0	UART Ch.0 Receive Data Register	Receiving data
	0x4103	UART_MOD0	UART Ch.0 Mode Register	Transfer data format setting
	0x4104	UART_CTL0	UART Ch.0 Control Register	Data transfer control
	0x4105	UART_EXP0	UART Ch.0 Expansion Register	IrDA mode setting
	0x4106–0x411f	–	–	Reserved
UART (with IrDA) Ch.1 (8-bit device)	0x4120	UART_ST1	UART Ch.1 Status Register	Transfer, buffer, error status display
	0x4121	UART_TXD1	UART Ch.1 Transmit Data Register	Transmission data
	0x4122	UART_RXD1	UART Ch.1 Receive Data Register	Receiving data
	0x4123	UART_MOD1	UART Ch.1 Mode Register	Transfer data format setting
	0x4124	UART_CTL1	UART Ch.1 Control Register	Data transfer control
	0x4125	UART_EXP1	UART Ch.1 Expansion Register	IrDA mode setting
	0x4126–0x413f	–	–	Reserved
8-bit timer (with F mode) Ch.0 (16-bit device)	0x4200	T8F_CLK0	8-bit Timer Ch.0 Input Clock Select Register	Prescaler output clock selection
	0x4202	T8F_TR0	8-bit Timer Ch.0 Reload Data Register	Reload data setting
	0x4204	T8F_TC0	8-bit Timer Ch.0 Counter Data Register	Counter data
	0x4206	T8F_CTL0	8-bit Timer Ch.0 Control Register	Timer mode setting and timer RUN/STOP
	0x4208	T8F_INT0	8-bit Timer Ch.0 Interrupt Control Register	Interrupt control
	0x420a–0x421f	–	–	Reserved
16-bit timer Ch.0 (16-bit device)	0x4220	T16_CLK0	16-bit Timer Ch.0 Input Clock Select Register	Prescaler output clock selection
	0x4222	T16_TR0	16-bit Timer Ch.0 Reload Data Register	Reload data setting
	0x4224	T16_TC0	16-bit Timer Ch.0 Counter Data Register	Counter data
	0x4226	T16_CTL0	16-bit Timer Ch.0 Control Register	Timer mode setting and timer RUN/STOP
	0x4228	T16_INT0	16-bit Timer Ch.0 Interrupt Control Register	Interrupt control
	0x422a–0x423f	–	–	Reserved
16-bit timer Ch.1 (16-bit device)	0x4240	T16_CLK1	16-bit Timer Ch.1 Input Clock Select Register	Prescaler output clock selection
	0x4242	T16_TR1	16-bit Timer Ch.1 Reload Data Register	Reload data setting
	0x4244	T16_TC1	16-bit Timer Ch.1 Counter Data Register	Counter data
	0x4246	T16_CTL1	16-bit Timer Ch.1 Control Register	Timer mode setting and timer RUN/STOP
	0x4248	T16_INT1	16-bit Timer Ch.1 Interrupt Control Register	Interrupt control
	0x424a–0x425f	–	–	Reserved
16-bit timer Ch.2 (16-bit device)	0x4260	T16_CLK2	16-bit Timer Ch.2 Input Clock Select Register	Prescaler output clock selection
	0x4262	T16_TR2	16-bit Timer Ch.2 Reload Data Register	Reload data setting
	0x4264	T16_TC2	16-bit Timer Ch.2 Counter Data Register	Counter data
	0x4266	T16_CTL2	16-bit Timer Ch.2 Control Register	Timer mode setting and timer RUN/STOP
	0x4268	T16_INT2	16-bit Timer Ch.2 Interrupt Control Register	Interrupt control
	0x426a–0x427f	–	–	Reserved
8-bit timer (with F mode) Ch.1 (16-bit device)	0x4280	T8F_CLK1	8-bit Timer Ch.1 Input Clock Select Register	Prescaler output clock selection
	0x4282	T8F_TR1	8-bit Timer Ch.1 Reload Data Register	Reload data setting
	0x4284	T8F_TC1	8-bit Timer Ch.1 Counter Data Register	Counter data
	0x4286	T8F_CTL1	8-bit Timer Ch.1 Control Register	Timer mode setting and timer RUN/STOP
	0x4288	T8F_INT1	8-bit Timer Ch.1 Interrupt Control Register	Interrupt control
	0x428a–0x429f	–	–	Reserved
Interrupt controller (16-bit device)	0x4300–0x4304	–	–	Reserved
	0x4306	ITC_LV0	Interrupt Level Setup Register 0	P0/P1 interrupt level setting
	0x4308	ITC_LV1	Interrupt Level Setup Register 1	SWT/CT interrupt level setting
	0x430a	ITC_LV2	Interrupt Level Setup Register 2	T8OSC1/SVD interrupt level setting
	0x430c	ITC_LV3	Interrupt Level Setup Register 3	LCD/T16E Ch.0 interrupt level setting
	0x430e	ITC_LV4	Interrupt Level Setup Register 4	T8F/T16 Ch.0 interrupt level setting
	0x4310	ITC_LV5	Interrupt Level Setup Register 5	T16 Ch.1/Ch.2 interrupt level setting
	0x4312	ITC_LV6	Interrupt Level Setup Register 6	UART Ch.0/Ch.1 interrupt level setting
	0x4314	ITC_LV7	Interrupt Level Setup Register 7	SPI/I^2C interrupt level setting
	0x4316	ITC_LV8	Interrupt Level Setup Register 8	REMC/T16E Ch.1 interrupt level setting
	0x4318–0x431f	–	–	Reserved

Peripheral circuit	Address	Register name		Function
SPI (16-bit device)	0x4320	SPI_ST	SPI Status Register	Transfer and buffer status display
	0x4322	SPI_TXD	SPI Transmit Data Register	Transmission data
	0x4324	SPI_RXD	SPI Receive Data Register	Receiving data
	0x4326	SPI_CTL	SPI Control Register	SPI mode and data transfer permission setting
	0x4328–0x433f	–	–	Reserved
I²C (16-bit device)	0x4340	I2C_EN	I²C Enable Register	I²C module enable
	0x4342	I2C_CTL	I²C Control Register	I²C control and transfer status display
	0x4344	I2C_DAT	I²C Data Register	Transfer data
	0x4346	I2C_ICTL	I²C Interrupt Control Register	I²C interrupt control
	0x4348–0x435f	–	–	Reserved

Internal Peripheral Circuit Area 2 (0x5000 onward)

Peripheral circuit	Address	Register name		Function
Clock timer (8-bit device)	0x5000	CT_CTL	Clock Timer Control Register	Timer reset and RUN/STOP control
	0x5001	CT_CNT	Clock Timer Counter Register	Counter data
	0x5002	CT_IMSK	Clock Timer Interrupt Mask Register	Interrupt mask setting
	0x5003	CT_IFLG	Clock Timer Interrupt Flag Register	Interrupt occurrence status display/reset
	0x5004–0x501f	–	–	Reserved
Stopwatch timer (8-bit device)	0x5020	SWT_CTL	Stopwatch Timer Control Register	Timer reset and RUN/STOP control
	0x5021	SWT_BCNT	Stopwatch Timer BCD Counter Register	BCD counter data
	0x5022	SWT_IMSK	Stopwatch Timer Interrupt Mask Register	Interrupt mask setting
	0x5023	SWT_IFLG	Stopwatch Timer Interrupt Flag Register	Interrupt occurrence status display/reset
	0x5024–0x503f	–	–	Reserved
Watchdog timer (8-bit device)	0x5040	WDT_CTL	Watchdog Timer Control Register	Timer reset and RUN/STOP control
	0x5041	WDT_ST	Watchdog Timer Status Register	Timer mode setting and NMI status display
	0x5042–0x505f	–	–	Reserved
Oscillator circuit (8-bit device)	0x5060	OSC_SRC	Clock Source Select Register	Clock source selection
	0x5061	OSC_CTL	Oscillation Control Register	Oscillation control
	0x5062	OSC_NFEN	Noise Filter Enable Register	Noise filter ON/OFF
	0x5063	OSC_LCLK	LCD Clock Setup Register	Reserved
	0x5064	OSC_FOUT	FOUT Control Register	Clock external output control
	0x5065	OSC_T8OSC1	T8OSC1 Clock Control Register	8-bit OSC1 timer clock setting
	0x5066–0x507f	–	–	Reserved
Clock generator (8-bit device)	0x5080	CLG_PCLK	PCLK Control Register	PCLK feed control
	0x5081	CLG_CCLK	CCLK Control Register	CCLK division ratio setting
	0x5082–0x509f	–	–	Reserved
LCD driver (8-bit device)	0x50a0	LCD_DCTL	LCD Display Control Register	LCD display control
	0x50a1	LCD_CADJ	LCD Contrast Adjust Register	Contrast control
	0x50a2	LCD_CCTL	LCD Clock Control Register	LCD clock duty selection
	0x50a3	LCD_VREG	LCD Voltage Regulator Control Register	LCD driver voltage regulator control
	0x50a4	LCD_PWR	LCD Power Voltage Booster Control Register	LCD power supply voltage booster circuit control
	0x50a5	LCD_IMSK	LCD Interrupt Mask Register	Interrupt mask setting
	0x50a6	LCD_IFLG	LCD Interrupt Flag Register	Interrupt occurrence status display/reset
	0x50a7–0x50bf	–	–	Reserved
8-bit OSC1 timer (8-bit device)	0x50c0	T8OSC1_CTL	8-bit OSC1 Timer Control Register	Timer mode setting and timer RUN/STOP
	0x50c1	T8OSC1_CNT	8-bit OSC1 Timer Counter Data Register	Counter data
	0x50c2	T8OSC1_CMP	8-bit OSC1 Timer Compare Data Register	Compare data setting
	0x50c3	T8OSC1_IMSK	8-bit OSC1 Timer Interrupt Mask Register	Interrupt mask data
	0x50c4	T8OSC1_IFLG	8-bit OSC1 Timer Interrupt Flag Register	Interrupt occurrence status display/reset
	0x50c5	T8OSC1_DUTY	8-bit OSC1 Timer PWM Duty Data Register	PWM output data setting
	0x50c6–0x50df	–	–	Reserved
SVD circuit (8-bit device)	0x5100	SVD_EN	SVD Enable Register	SVD operation permitted/prevented
	0x5101	SVD_CMP	SVD Compare Voltage Register	Comparison voltage setting
	0x5102	SVD_RSLT	SVD Detection Result Register	Voltage detection results
	0x5103	SVD_IMSK	SVD Interrupt Mask Register	Interrupt mask setting
	0x5104	SVD_IFLG	SVD Interrupt Flag Register	Interrupt occurrence status display/reset
	0x5105–0x511f	–	–	Reserved
Power supply circuit (8-bit device)	0x5120	VD1_CTL	V_{D1} Control Register	V_{D1} voltage and load protection control
	0x5121–0x513f	–	–	Reserved

Peripheral circuit	Address	Register name		Function
P port & port MUX (8-bit device)	0x5200	P0_IN	P0 Port Input Data Register	P0 port input data
	0x5201	P0_OUT	P0 Port Output Data Register	P0 port output data
	0x5202	P0_OEN	P0 Port Output Enable	P0 port output enable
	0x5203	P0_PU	P0 Port Pull-up Control Register	P0 port pull-up control
	0x5204	P0_SM	P0 Port Schmitt Trigger Control Register	P0 port Schmitt trigger control
	0x5205	P0_IMSK	P0 Port Interrupt Mask Register	P0 port interrupt mask setting
	0x5206	P0_EDGE	P0 Port Interrupt Edge Select Register	P0 port interrupt edge selection
	0x5207	P0_IFLG	P0 Port Interrupt Flag Register	P0 port interrupt occurrence status display/reset
	0x5208	P0_CHAT	P0 Port Chattering Filter Control Register	P0 port chattering filter control
	0x5209	P0_KRST	P0 Port Key-Entry Reset Configuration Register	P0 port key entry reset setting
	0x520a	P0_IEN	P0 Port Input Enable Register	P0 port input enable
	0x520b–0x520f	–	–	Reserved
	0x5210	P1_IN	P1 Port Input Data Register	P1 port input data
	0x5211	P1_OUT	P1 Port Output Data Register	P1 port output data
	0x5212	P1_OEN	P1 Port Output Enable	P1 port output enable
	0x5213	P1_PU	P1 Port Pull-up Control Register	P1 port pull-up control
	0x5214	P1_SM	P1 Port Schmitt Trigger Control Register	P1 port Schmitt trigger control
	0x5215	P1_IMSK	P1 Port Interrupt Mask Register	P1 port interrupt mask setting
	0x5216	P1_EDGE	P1 Port Interrupt Edge Select Register	P1 port interrupt edge selection
	0x5217	P1_IFLG	P1 Port Interrupt Flag Register	P1 port interrupt occurrence status display/reset
	0x5218	P1_CHAT	P1 Port Chattering Filter Control Register	P1 port chattering filter control
	0x5219	–	–	Reserved
	0x521a	P1_IEN	P1 Port Input Enable Register	P1 port input enable
	0x521b–0x521f	–	–	Reserved
	0x5220	P2_IN	P2 Port Input Data Register	P2 port input data
	0x5221	P2_OUT	P2 Port Output Data Register	P2 port output data
	0x5222	P2_OEN	P2 Port Output Enable	P2 port output enable
	0x5223	P2_PU	P2 Port Pull-up Control Register	P2 port pull-up control
	0x5224	P2_SM	P2 Port Schmitt Trigger Control Register	P2 port Schmitt trigger control
	0x5225–0x5229	–	–	Reserved
	0x522a	P2_IEN	P2 Port Input Enable Register	P2 port input enable
	0x522b–0x522f	–	–	Reserved
	0x5230	P3_IN	P3 Port Input Data Register	P3 port input data
	0x5231	P3_OUT	P3 Port Output Data Register	P3 port output data
	0x5232	P3_OEN	P3 Port Output Enable	P3 port output enable
	0x5233	P3_PU	P3 Port Pull-up Control Register	P3 port pull-up control
	0x5234	P3_SM	P3 Port Schmitt Trigger Control Register	P3 port Schmitt trigger control
	0x5235–0x5239	–	–	Reserved
	0x523a	P3_IEN	P3 Port Input Enable Register	P3 port input enable
	0x523b–0x527f	–	–	Reserved
	0x52a0	P0_PMUX	P0 Port Function Select Register	P0 port function selection
	0x52a1	P1_PMUX	P1 Port Function Select Register	P1 port function selection
	0x52a2	P2_PMUX	P2 Port Function Select Register	P2 port function selection
	0x52a3	P3_PMUX	P3 Port Function Select Register	P3 port function selection
	0x52a4–0x52bf	–	–	Reserved
PWM & capture timer Ch.0 (16-bit device)	0x5300	T16E_CA0	PWM Timer Ch.0 Compare Data A Register	Compare data A setting
	0x5302	T16E_CB0	PWM Timer Ch.0 Compare Data B Register	Compare data B setting
	0x5304	T16E_TC0	PWM Timer Ch.0 Counter Data Register	Counter data
	0x5306	T16E_CTL0	PWM Timer Ch.0 Control Register	Timer mode setting and timer RUN/STOP
	0x5308	T16E_CLK0	PWM Timer Ch.0 Input Clock Select Register	Prescaler output clock selection
	0x530a	T16E_IMSK0	PWM Timer Ch.0 Interrupt Mask Register	Interrupt mask setting
	0x530c	T16E_IFLG0	PWM Timer Ch.0 Interrupt Flag Register	Interrupt occurrence status display/reset
	0x530e–0x531f	–	–	Reserved
MISC register (16-bit device)	0x5320	MISC_FL	FLASHC/SRAMC Control Register	FLASHC/SRAMC access condition setting
	0x5322	MISC_OSC1	OSC1 Peripheral Control Register	OSC1 operation peripheral function setting for debugging
	0x5324	MISC_PROT	MISC Protect Register	MISC register program protection
	0x5326	MISC_IRAMSZ	IRAM Size Select Register	IRAM size selection
	0x5328	MISC_TTBRL	Vector Table Address Low Register	Vector table address setting
	0x532a	MISC_TTBRH	Vector Table Address High Register	
	0x532c	MISC_PSR	PSR Register	S1C17 core PSR readout
	0x532e–0x533f	–	–	Reserved
Remote controller (16-bit device)	0x5340	REMC_CFG	REMC Configuration Register	Clock transfer control
	0x5342	REMC_CAR	REMC Carrier Length Setup Register	Carrier H/L section length setting
	0x5344	REMC_LCNT	REMC Length Counter Register	Transfer bit and transfer data length setting
	0x5346	REMC_INT	REMC Interrupt Control Register	Interrupt control
	0x5348–0x535f	–	–	Reserved

Peripheral circuit	Address	Register name		Function
PWM & capture timer Ch.1 (16-bit device)	0x5360	T16E_CA1	PWM Timer Ch.1 Compare Data A Register	Compare data A setting
	0x5362	T16E_CB1	PWM Timer Ch.1 Compare Data B Register	Compare data B setting
	0x5364	T16E_TC1	PWM Timer Ch.1 Counter Data Register	Counter data
	0x5366	T16E_CTL1	PWM Timer Ch.1 Control Register	Timer mode setting and timer RUN/STOP
	0x5368	T16E_CLK1	PWM Timer Ch.1 Input Clock Select Register	Prescaler output clock selection
	0x536a	T16E_IMSK1	PWM Timer Ch.1 Interrupt Mask Register	Interrupt mask setting
	0x536c	T16E_IFLG1	PWM Timer Ch.1 Interrupt Flag Register	Interrupt occurrence status display/reset
	0x536e–0x537f	–	–	Reserved

Core I/O Reserved Area 2 (0xffff84 onward)

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

Note: Addresses marked as “Reserved” or unused peripheral circuit areas not marked in the table must not be accessed by application programs.

0x4020 Prescaler

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

0x4100–0x4124 UART (with IrDA)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
UART Ch.0 Status Register (UART_ST0)	**0x4100** (8 bits)	D7	**–**	reserved	–	–	–	0 when being read.
		D6	**FER**	Framing error flag	1 Error 0 Normal	0	R/W	Reset by writing 1.
		D5	**PER**	Parity error flag	1 Error 0 Normal	0	R/W	
		D4	**OER**	Overrun error flag	1 Error 0 Normal	0	R/W	
		D3	**RD2B**	Second byte receive flag	1 Ready 0 Empty	0	R	
		D2	**TRBS**	Transmit busy flag	1 Busy 0 Idle	0	R	Shift register status
		D1	**RDRY**	Receive data ready flag	1 Ready 0 Empty	0	R	
		D0	**TDBE**	Transmit data buffer empty flag	1 Empty 0 Not empty	1	R	
UART Ch.0 Transmit Data Register (UART_TXD0)	**0x4101** (8 bits)	D7–0	**TXD[7:0]**	Transmit data TXD7(6) = MSB TXD0 = LSB	0x0 to 0xff (0x7f)	0x0	R/W	
UART Ch.0 Receive Data Register (UART_RXD0)	**0x4102** (8 bits)	D7–0	**RXD[7:0]**	Receive data in the receive data buffer RXD7(6) = MSB RXD0 = LSB	0x0 to 0xff (0x7f)	0x0	R	Older data in the buffer is read out first.
UART Ch.0 Mode Register (UART_MOD0)	**0x4103** (8 bits)	D7–5	**–**	reserved	–	–	–	0 when being read.
		D4	**CHLN**	Character length	1 8 bits 0 7 bits	0	R/W	
		D3	**PREN**	Parity enable	1 With parity 0 No parity	0	R/W	
		D2	**PMD**	Parity mode select	1 Odd 0 Even	0	R/W	
		D1	**STPB**	Stop bit select	1 2 bits 0 1 bit	0	R/W	
		D0	**SSCK**	Input clock select	1 External 0 Internal	0	R/W	
UART Ch.0 Control Register (UART_CTL0)	**0x4104** (8 bits)	D7	**–**	reserved	–	–	–	0 when being read.
		D6	**REIEN**	Receive error int. enable	1 Enable 0 Disable	0	R/W	
		D5	**RIEN**	Receive buffer full int. enable	1 Enable 0 Disable	0	R/W	
		D4	**TIEN**	Transmit buffer empty int. enable	1 Enable 0 Disable	0	R/W	
		D3–2	**–**	reserved	–	–	–	0 when being read.
		D1	**RBFI**	Receive buffer full int. condition	1 2 bytes 0 1 byte	0	R/W	
		D0	**RXEN**	UART enable	1 Enable 0 Disable	0	R/W	
UART Ch.0 Expansion Register (UART_EXP0)	**0x4105** (8 bits)	D7	**–**	reserved	–	–	–	0 when being read.
		D6–4	**IRCLK[2:0]**	IrDA receive detection clock select	IRCLK[2:0] Clock 0x7 PCLK•1/128 0x6 PCLK•1/64 0x5 PCLK•1/32 0x4 PCLK•1/16 0x3 PCLK•1/8 0x2 PCLK•1/4 0x1 PCLK•1/2 0x0 PCLK•1/1	0x0	R/W	
		D3–1	**–**	reserved	–	–	–	0 when being read.
		D0	**IRMD**	IrDA mode select	1 On 0 Off	0	R/W	
UART Ch.1 Status Register (UART_ST1)	**0x4120** (8 bits)	D7	**–**	reserved	–	–	–	0 when being read.
		D6	**FER**	Framing error flag	1 Error 0 Normal	0	R/W	Reset by writing 1.
		D5	**PER**	Parity error flag	1 Error 0 Normal	0	R/W	
		D4	**OER**	Overrun error flag	1 Error 0 Normal	0	R/W	
		D3	**RD2B**	Second byte receive flag	1 Ready 0 Empty	0	R	
		D2	**TRBS**	Transmit busy flag	1 Busy 0 Idle	0	R	Shift register status
		D1	**RDRY**	Receive data ready flag	1 Ready 0 Empty	0	R	
		D0	**TDBE**	Transmit data buffer empty flag	1 Empty 0 Not empty	1	R	
UART Ch.1 Transmit Data Register (UART_TXD1)	**0x4121** (8 bits)	D7–0	**TXD[7:0]**	Transmit data TXD7(6) = MSB TXD0 = LSB	0x0 to 0xff (0x7f)	0x0	R/W	
UART Ch.1 Receive Data Register (UART_RXD1)	**0x4122** (8 bits)	D7–0	**RXD[7:0]**	Receive data in the receive data buffer RXD7(6) = MSB RXD0 = LSB	0x0 to 0xff (0x7f)	0x0	R	Older data in the buffer is read out first.
UART Ch.1 Mode Register (UART_MOD1)	**0x4123** (8 bits)	D7–5	**–**	reserved	–	–	–	0 when being read.
		D4	**CHLN**	Character length	1 8 bits 0 7 bits	0	R/W	
		D3	**PREN**	Parity enable	1 With parity 0 No parity	0	R/W	
		D2	**PMD**	Parity mode select	1 Odd 0 Even	0	R/W	
		D1	**STPB**	Stop bit select	1 2 bits 0 1 bit	0	R/W	
		D0	**SSCK**	Input clock select	1 External 0 Internal	0	R/W	
UART Ch.1 Control Register (UART_CTL1)	**0x4124** (8 bits)	D7	**–**	reserved	–	–	–	0 when being read.
		D6	**REIEN**	Receive error int. enable	1 Enable 0 Disable	0	R/W	
		D5	**RIEN**	Receive buffer full int. enable	1 Enable 0 Disable	0	R/W	
		D4	**TIEN**	Transmit buffer empty int. enable	1 Enable 0 Disable	0	R/W	
		D3–2	**–**	reserved	–	–	–	0 when being read.
		D1	**RBFI**	Receive buffer full int. condition	1 2 bytes 0 1 byte	0	R/W	
		D0	**RXEN**	UART enable	1 Enable 0 Disable	0	R/W	

0x4125 UART (with IrDA)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
UART Ch.1 Expansion Register (UART_EXP1)	**0x4125** (8 bits)	D7	**–**	reserved	–	–	–	0 when being read.
		D6–4	**IRCLK[2:0]**	IrDA receive detection clock select	IRCLK[2:0] Clock 0x7 PCLK-1/128 0x6 PCLK-1/64 0x5 PCLK-1/32 0x4 PCLK-1/16 0x3 PCLK-1/8 0x2 PCLK-1/4 0x1 PCLK-1/2 0x0 PCLK-1/1	0x0	R/W	
		D3–1	**–**	reserved	–	–	–	0 when being read.
		D0	**IRMD**	IrDA mode select	1 On 0 Off	0	R/W	

0x4200–0x4208 8-bit Timer (with Fine Mode) Ch.0

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
8-bit Timer Ch.0 Input Clock Select Register (T8F_CLK0)	**0x4200** (16 bits)	D15–4	**–**	reserved	–	–	–	0 when being read.
		D3–0	**DF[3:0]**	8-bit timer input clock select (Prescaler output clock)	DF[3:0] / Clock 0xf reserved 0xe PCLK-1/16384 0xd PCLK-1/8192 0xc PCLK-1/4096 0xb PCLK-1/2048 0xa PCLK-1/1024 0x9 PCLK-1/512 0x8 PCLK-1/256 0x7 PCLK-1/128 0x6 PCLK-1/64 0x5 PCLK-1/32 0x4 PCLK-1/16 0x3 PCLK-1/8 0x2 PCLK-1/4 0x1 PCLK-1/2 0x0 PCLK-1/1	0x0	R/W	
8-bit Timer Ch.0 Reload Data Register (T8F_TR0)	**0x4202** (16 bits)	D15–8	**–**	reserved	–	–	–	0 when being read.
		D7–0	**TR[7:0]**	8-bit timer reload data TR7 = MSB TR0 = LSB	0x0 to 0xff	0x0	R/W	
8-bit Timer Ch.0 Counter Data Register (T8F_TC0)	**0x4204** (16 bits)	D15–8	**–**	reserved	–	–	–	0 when being read.
		D7–0	**TC[7:0]**	8-bit timer counter data TC7 = MSB TC0 = LSB	0x0 to 0xff	0xff	R	
8-bit Timer Ch.0 Control Register (T8F_CTL0)	**0x4206** (16 bits)	D15–12	**–**	reserved	–	–	–	0 when being read.
		D11–8	**TFMD[3:0]**	Fine mode setup	0x0 to 0xf	0x0	R/W	Set a number of times to insert delay into a 16-underflow period.
		D7–5	**–**	reserved	–	–	–	0 when being read.
		D4	**TRMD**	Count mode select	1 One shot / 0 Repeat	0	R/W	
		D3–2	**–**	reserved	–	–	–	0 when being read.
		D1	**PRESER**	Timer reset	1 Reset / 0 Ignored	0	W	
		D0	**PRUN**	Timer run/stop control	1 Run / 0 Stop	0	R/W	
8-bit Timer Ch.0 Interrupt Control Register (T8F_INT0)	**0x4208** (16 bits)	D15–9	**–**	reserved	–	–	–	0 when being read.
		D8	**T8IE**	8-bit timer interrupt enable	1 Enable / 0 Disable	0	R/W	
		D7–1	**–**	reserved	–	–	–	0 when being read.
		D0	**T8IF**	8-bit timer interrupt flag	1 Cause of interrupt occurred / 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.

0x4220–0x4244 16-bit Timer

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
16-bit Timer Ch.0 Input Clock Select Register (T16_CLK0)	**0x4220** (16 bits)	D15–4	**–**	reserved	–	–	–	0 when being read.
		D3–0	**DF[3:0]**	Timer input clock select (Prescaler output clock)	DF[3:0] / Clock 0xf reserved 0xe PCLK-1/16384 0xd PCLK-1/8192 0xc PCLK-1/4096 0xb PCLK-1/2048 0xa PCLK-1/1024 0x9 PCLK-1/512 0x8 PCLK-1/256 0x7 PCLK-1/128 0x6 PCLK-1/64 0x5 PCLK-1/32 0x4 PCLK-1/16 0x3 PCLK-1/8 0x2 PCLK-1/4 0x1 PCLK-1/2 0x0 PCLK-1/1	0x0	R/W	
16-bit Timer Ch.0 Reload Data Register (T16_TR0)	**0x4222** (16 bits)	D15–0	**TR[15:0]**	16-bit timer reload data TR15 = MSB TR0 = LSB	0x0 to 0xffff	0x0	R/W	
16-bit Timer Ch.0 Counter Data Register (T16_TC0)	**0x4224** (16 bits)	D15–0	**TC[15:0]**	16-bit timer counter data TC15 = MSB TC0 = LSB	0x0 to 0xffff	0xffff	R	
16-bit Timer Ch.0 Control Register (T16_CTL0)	**0x4226** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10	**CKACTV**	External clock active level select	1 High / 0 Low	1	R/W	
		D9–8	**CKSL[1:0]**	Input clock and pulse width measurement mode select	CKSL[1:0] / Mode 0x3 reserved 0x2 Pulse width 0x1 External clock 0x0 Internal clock	0x0	R/W	
		D7–5	**–**	reserved	–	–	–	0 when being read.
		D4	**TRMD**	Count mode select	1 One shot / 0 Repeat	0	R/W	
		D3–2	**–**	reserved	–	–	–	0 when being read.
		D1	**PRESER**	Timer reset	1 Reset / 0 Ignored	0	W	
		D0	**PRUN**	Timer run/stop control	1 Run / 0 Stop	0	R/W	
16-bit Timer Ch.0 Interrupt Control Register (T16_INT0)	**0x4228** (16 bits)	D15–9	**–**	reserved	–	–	–	0 when being read.
		D8	**T16IE**	16-bit timer interrupt enable	1 Enable / 0 Disable	0	R/W	
		D7–1	**–**	reserved	–	–	–	0 when being read.
		D0	**T16IF**	16-bit timer interrupt flag	1 Cause of interrupt occurred / 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
16-bit Timer Ch.1 Input Clock Select Register (T16_CLK1)	**0x4240** (16 bits)	D15–4	**–**	reserved	–	–	–	0 when being read.
		D3–0	**DF[3:0]**	Timer input clock select (Prescaler output clock)	DF[3:0] / Clock 0xf reserved 0xe PCLK-1/16384 0xd PCLK-1/8192 0xc PCLK-1/4096 0xb PCLK-1/2048 0xa PCLK-1/1024 0x9 PCLK-1/512 0x8 PCLK-1/256 0x7 PCLK-1/128 0x6 PCLK-1/64 0x5 PCLK-1/32 0x4 PCLK-1/16 0x3 PCLK-1/8 0x2 PCLK-1/4 0x1 PCLK-1/2 0x0 PCLK-1/1	0x0	R/W	
16-bit Timer Ch.1 Reload Data Register (T16_TR1)	**0x4242** (16 bits)	D15–0	**TR[15:0]**	16-bit timer reload data TR15 = MSB TR0 = LSB	0x0 to 0xffff	0x0	R/W	
16-bit Timer Ch.1 Counter Data Register (T16_TC1)	**0x4244** (16 bits)	D15–0	**TC[15:0]**	16-bit timer counter data TC15 = MSB TC0 = LSB	0x0 to 0xffff	0xffff	R	

0x4246–0x4268 16-bit Timer

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
16-bit Timer Ch.1 Control Register (T16_CTL1)	**0x4246** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10	**CKACTV**	External clock active level select	1 High 0 Low	1	R/W	
		D9–8	**CKSL[1:0]**	Input clock and pulse width measurement mode select	CKSL[1:0] Mode 0x3 reserved 0x2 Pulse width 0x1 External clock 0x0 Internal clock	0x0	R/W	
		D7–5	**–**	reserved	–	–	–	0 when being read.
		D4	**TRMD**	Count mode select	1 One shot 0 Repeat	0	R/W	
		D3–2	**–**	reserved	–	–	–	0 when being read.
		D1	**PRESER**	Timer reset	1 Reset 0 Ignored	0	W	
		D0	**PRUN**	Timer run/stop control	1 Run 0 Stop	0	R/W	
16-bit Timer Ch.1 Interrupt Control Register (T16_INT1)	**0x4248** (16 bits)	D15–9	**–**	reserved	–	–	–	0 when being read.
		D8	**T16IE**	16-bit timer interrupt enable	1 Enable 0 Disable	0	R/W	
		D7–1	**–**	reserved	–	–	–	0 when being read.
		D0	**T16IF**	16-bit timer interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
16-bit Timer Ch.2 Input Clock Select Register (T16_CLK2)	**0x4260** (16 bits)	D15–4	**–**	reserved	–	–	–	0 when being read.
		D3–0	**DF[3:0]**	Timer input clock select (Prescaler output clock)	DF[3:0] Clock 0xf reserved 0xe PCLK-1/16384 0xd PCLK-1/8192 0xc PCLK-1/4096 0xb PCLK-1/2048 0xa PCLK-1/1024 0x9 PCLK-1/512 0x8 PCLK-1/256 0x7 PCLK-1/128 0x6 PCLK-1/64 0x5 PCLK-1/32 0x4 PCLK-1/16 0x3 PCLK-1/8 0x2 PCLK-1/4 0x1 PCLK-1/2 0x0 PCLK-1/1	0x0	R/W	
16-bit Timer Ch.2 Reload Data Register (T16_TR2)	**0x4262** (16 bits)	D15–0	**TR[15:0]**	16-bit timer reload data TR15 = MSB TR0 = LSB	0x0 to 0xffff	0x0	R/W	
16-bit Timer Ch.2 Counter Data Register (T16_TC2)	**0x4264** (16 bits)	D15–0	**TC[15:0]**	16-bit timer counter data TC15 = MSB TC0 = LSB	0x0 to 0xffff	0xffff	R	
16-bit Timer Ch.2 Control Register (T16_CTL2)	**0x4266** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10	**CKACTV**	External clock active level select	1 High 0 Low	1	R/W	
		D9–8	**CKSL[1:0]**	Input clock and pulse width measurement mode select	CKSL[1:0] Mode 0x3 reserved 0x2 Pulse width 0x1 External clock 0x0 Internal clock	0x0	R/W	
		D7–5	**–**	reserved	–	–	–	0 when being read.
		D4	**TRMD**	Count mode select	1 One shot 0 Repeat	0	R/W	
		D3–2	**–**	reserved	–	–	–	0 when being read.
		D1	**PRESER**	Timer reset	1 Reset 0 Ignored	0	W	
		D0	**PRUN**	Timer run/stop control	1 Run 0 Stop	0	R/W	
16-bit Timer Ch.2 Interrupt Control Register (T16_INT2)	**0x4268** (16 bits)	D15–9	**–**	reserved	–	–	–	0 when being read.
		D8	**T16IE**	16-bit timer interrupt enable	1 Enable 0 Disable	0	R/W	
		D7–1	**–**	reserved	–	–	–	0 when being read.
		D0	**T16IF**	16-bit timer interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.

0x4280–0x4288 8-bit Timer (with Fine Mode) Ch.1

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
8-bit Timer Ch.1 Input Clock Select Register (T8F_CLK1)	**0x4280** (16 bits)	D15–4	**–**	reserved	–	–	–	0 when being read.
		D3–0	**DF[3:0]**	8-bit timer input clock select (Prescaler output clock)	DF[3:0] / Clock 0xf reserved 0xe PCLK-1/16384 0xd PCLK-1/8192 0xc PCLK-1/4096 0xb PCLK-1/2048 0xa PCLK-1/1024 0x9 PCLK-1/512 0x8 PCLK-1/256 0x7 PCLK-1/128 0x6 PCLK-1/64 0x5 PCLK-1/32 0x4 PCLK-1/16 0x3 PCLK-1/8 0x2 PCLK-1/4 0x1 PCLK-1/2 0x0 PCLK-1/1	0x0	R/W	
8-bit Timer Ch.1 Reload Data Register (T8F_TR1)	**0x4282** (16 bits)	D15–8	**–**	reserved	–	–	–	0 when being read.
		D7–0	**TR[7:0]**	8-bit timer reload data TR7 = MSB TR0 = LSB	0x0 to 0xff	0x0	R/W	
8-bit Timer Ch.1 Counter Data Register (T8F_TC1)	**0x4284** (16 bits)	D15–8	**–**	reserved	–	–	–	0 when being read.
		D7–0	**TC[7:0]**	8-bit timer counter data TC7 = MSB TC0 = LSB	0x0 to 0xff	0xff	R	
8-bit Timer Ch.1 Control Register (T8F_CTL1)	**0x4286** (16 bits)	D15–12	**–**	reserved	–	–	–	0 when being read.
		D11–8	**TFMD[3:0]**	Fine mode setup	0x0 to 0xf	0x0	R/W	Set a number of times to insert delay into a 16-underflow period.
		D7–5	**–**	reserved	–	–	–	0 when being read.
		D4	**TRMD**	Count mode select	1 One shot 0 Repeat	0	R/W	
		D3–2	**–**	reserved	–	–	–	0 when being read.
		D1	**PRESER**	Timer reset	1 Reset 0 Ignored	0	W	
		D0	**PRUN**	Timer run/stop control	1 Run 0 Stop	0	R/W	
8-bit Timer Ch.1 Interrupt Control Register (T8F_INT1)	**0x4288** (16 bits)	D15–9	**–**	reserved	–	–	–	0 when being read.
		D8	**T8IE**	8-bit timer interrupt enable	1 Enable 0 Disable	0	R/W	
		D7–1	**–**	reserved	–	–	–	0 when being read.
		D0	**T8IF**	8-bit timer interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.

0x4306–0x4316 Interrupt Controller

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Interrupt Level Setup Register 0 (ITC_LV0)	**0x4306** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**ILV1[2:0]**	P1 interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**ILV0[2:0]**	P0 interrupt level	0 to 7	0x0	R/W	
Interrupt Level Setup Register 1 (ITC_LV1)	**0x4308** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**ILV3[2:0]**	CT interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**ILV2[2:0]**	SWT interrupt level	0 to 7	0x0	R/W	
Interrupt Level Setup Register 2 (ITC_LV2)	**0x430a** (16 bits)	D15–11	**–**	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 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 interrupt level	0 to 7	0x0	R/W	
Interrupt Level Setup Register 5 (ITC_LV5)	**0x4310** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**ILV11[2:0]**	T16 Ch.2 interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**ILV10[2:0]**	T16 Ch.1 interrupt level	0 to 7	0x0	R/W	
Interrupt Level Setup Register 6 (ITC_LV6)	**0x4312** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**ILV13[2:0]**	UART Ch.1 interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**ILV12[2:0]**	UART Ch.0 interrupt level	0 to 7	0x0	R/W	
Interrupt Level Setup Register 7 (ITC_LV7)	**0x4314** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**ILV15[2:0]**	I^2C interrupt level	0 to 7	0x0	R/W	
		D7–3	**–**	reserved	–	–	–	0 when being read.
		D2–0	**ILV14[2:0]**	SPI interrupt level	0 to 7	0x0	R/W	
Interrupt Level Setup Register 8 (ITC_LV8)	**0x4316** (16 bits)	D15–11	**–**	reserved	–	–	–	0 when being read.
		D10–8	**ILV17[2:0]**	T16E 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	

0x4320–0x4326 SPI

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SPI Status Register (SPI_ST)	**0x4320** (16 bits)	D15–3	–	reserved	–	–	–	0 when being read.
		D2	**SPBSY**	Transfer busy flag (master)	1 Busy 0 Idle	0	R	
				ss signal low flag (slave)	1 ss = L 0 ss = H			
		D1	**SPRBF**	Receive data buffer full flag	1 Full 0 Not full	0	R	
		D0	**SPTBE**	Transmit data buffer empty flag	1 Empty 0 Not empty	1	R	
SPI Transmit Data Register (SPI_TXD)	**0x4322** (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	**SPTDB[7:0]**	SPI transmit data buffer SPTDB7 = MSB SPTDB0 = LSB	0x0 to 0xff	0x0	R/W	
SPI Receive Data Register (SPI_RXD)	**0x4324** (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	**SPRDB[7:0]**	SPI receive data buffer SPRDB7 = MSB SPRDB0 = LSB	0x0 to 0xff	0x0	R	
SPI Control Register (SPI_CTL)	**0x4326** (16 bits)	D15–10	–	reserved	–	–	–	0 when being read.
		D9	**MCLK**	SPI clock source select	1 T16 Ch.1 0 PCLK•1/4	0	R/W	
		D8	**MLSB**	LSB/MSB first mode select	1 LSB 0 MSB	0	R/W	
		D7–6	–	reserved	–	–	–	0 when being read.
		D5	**SPRIE**	Receive data buffer full int. enable	1 Enable 0 Disable	0	R/W	
		D4	**SPTIE**	Transmit data buffer empty int. enable	1 Enable 0 Disable	0	R/W	
		D3	**CPHA**	Clock phase select	1 Data out 0 Data in	0	R/W	These bits must be
		D2	**CPOL**	Clock polarity select	1 Active L 0 Active H	0	R/W	set before setting
		D1	**MSSL**	Master/slave mode select	1 Master 0 Slave	0	R/W	SPEN to 1.
		D0	**SPEN**	SPI enable	1 Enable 0 Disable	0	R/W	

0x4340–0x4346 I²C

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
I²C Enable Register (I2C_EN)	**0x4340** (16 bits)	D15–1	**–**	reserved	–	–	–	0 when being read.
		D0	**I2CEN**	I²C enable	1 Enable 0 Disable	0	R/W	
I²C Control Register (I2C_CTL)	**0x4342** (16 bits)	D15–10	**–**	reserved	–	–	–	0 when being read.
		D9	**RBUSY**	Receive busy flag	1 Busy 0 Idle	0	R	
		D8	**TBUSY**	Transmit busy flag	1 Busy 0 Idle	0	R	
		D7–5	**–**	reserved	–	–	–	0 when being read.
		D4	**NSERM**	Noise remove on/off	1 On 0 Off	0	R/W	
		D3–2	**–**	reserved	–	–	–	0 when being read.
		D1	**STP**	Stop control	1 Stop 0 Ignored	0	R/W	
		D0	**STRT**	Start control	1 Start 0 Ignored	0	R/W	
I²C Data Register (I2C_DAT)	**0x4344** (16 bits)	D15–12	**–**	reserved	–	–	–	0 when being read.
		D11	**RBRDY**	Receive buffer ready	1 Ready 0 Empty	0	R	
		D10	**RXE**	Receive execution	1 Receive 0 Ignored	0	R/W	
		D9	**TXE**	Transmit execution	1 Transmit 0 Ignored	0	R/W	
		D8	**RTACK**	Receive/transmit ACK	1 Error 0 ACK	0	R/W	
		D7–0	**RTDT[7:0]**	Receive/transmit data RTDT7 = MSB RTDT0 = LSB	0x0 to 0xff	0x0	R/W	
I²C Interrupt Control Register (I2C_ICTL)	**0x4346** (16 bits)	D15–2	**–**	reserved	–	–	–	0 when being read.
		D1	**RINTE**	Receive interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	**TINTE**	Transmit interrupt enable	1 Enable 0 Disable	0	R/W	

0x5000–0x5003 Clock Timer

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Clock Timer Control Register (CT_CTL)	**0x5000** (8 bits)	D7–5	**–**	reserved	–	–	–	0 when being read.
		D4	**CTRST**	Clock timer reset	1 Reset 0 Ignored	0	W	
		D3–1	**–**	reserved	–	–	–	
		D0	**CTRUN**	Clock timer run/stop control	1 Run 0 Stop	0	R/W	
Clock Timer Counter Register (CT_CNT)	**0x5001** (8 bits)	D7–0	**CTCNT[7:0]**	Clock timer counter value	0x0 to 0xff	0	R	
Clock Timer Interrupt Mask Register (CT_IMSK)	**0x5002** (8 bits)	D7–4	**–**	reserved	–	–	–	0 when being read.
		D3	**CTIE32**	32 Hz interrupt enable	1 Enable 0 Disable	0	R/W	
		D2	**CTIE8**	8 Hz interrupt enable	1 Enable 0 Disable	0	R/W	
		D1	**CTIE2**	2 Hz interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	**CTIE1**	1 Hz interrupt enable	1 Enable 0 Disable	0	R/W	
Clock Timer Interrupt Flag Register (CT_IFLG)	**0x5003** (8 bits)	D7–4	**–**	reserved	–	–	–	0 when being read.
		D3	**CTIF32**	32 Hz interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D2	**CTIF8**	8 Hz interrupt flag		0	R/W	
		D1	**CTIF2**	2 Hz interrupt flag		0	R/W	
		D0	**CTIF1**	1 Hz interrupt flag		0	R/W	

0x5020–0x5023 Stopwatch Timer

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Stopwatch Timer Control Register (SWT_CTL)	0x5020 (8 bits)	D7–5	–	reserved	–	–	–	0 when being read.
		D4	SWTRST	Stopwatch timer reset	1 Reset 0 Ignored	0	W	
		D3–1	–	reserved	–	–	–	
		D0	SWTRUN	Stopwatch timer run/stop control	1 Run 0 Stop	0	R/W	
Stopwatch Timer BCD Counter Register (SWT_BCNT)	0x5021 (8 bits)	D7–4	BCD10[3:0]	1/10 sec. BCD counter value	0 to 9	0	R	
		D3–0	BCD100[3:0]	1/100 sec. BCD counter value	0 to 9	0	R	
Stopwatch Timer Interrupt Mask Register (SWT_IMSK)	0x5022 (8 bits)	D7–3	–	reserved	–	–	–	0 when being read.
		D2	SIE1	1 Hz interrupt enable	1 Enable 0 Disable	0	R/W	
		D1	SIE10	10 Hz interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	SIE100	100 Hz interrupt enable	1 Enable 0 Disable	0	R/W	
Stopwatch Timer Interrupt Flag Register (SWT_IFLG)	0x5023 (8 bits)	D7–3	–	reserved	–	–	–	0 when being read.
		D2	SIF1	1 Hz interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D1	SIF10	10 Hz interrupt flag		0	R/W	
		D0	SIF100	100 Hz interrupt flag		0	R/W	

0x5040–0x5041 Watchdog Timer

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Watchdog Timer Control Register (WDT_CTL)	0x5040 (8 bits)	D7–5	–	reserved	–	–	–	0 when being read.
		D4	WDTRST	Watchdog timer reset	1 Reset / 0 Ignored	0	W	
		D3–0	WDTRUN[3:0]	Watchdog timer run/stop control	Other than 1010 Run / 1010 Stop	1010	R/W	
Watchdog Timer Status Register (WDT_ST)	0x5041 (8 bits)	D7–2	–	reserved	–	–	–	0 when being read.
		D1	WDTMD	NMI/Reset mode select	1 Reset / 0 NMI	0	R/W	
		D0	WDTST	NMI status	1 NMI occurred / 0 Not occurred	0	R	

0x5060–0x5065 Oscillator

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Clock Source Select Register (OSC_SRC)	0x5060 (8 bits)	D7–2	–	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	
LCD Clock Setup 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 HSCLK-1/512; 0x3 HSCLK-1/256; 0x2 HSCLK-1/128; 0x1 HSCLK-1/64; 0x0 HSCLK-1/32	0x0	R/W	
		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	
FOUT Control Register (OSC_FOUT)	0x5064 (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.
		D3–2	FOUTHD [1:0]	FOUTH clock division ratio select	FOUTHD[1:0] / Division ratio: 0x3 reserved; 0x2 HSCLK-1/4; 0x1 HSCLK-1/2; 0x0 HSCLK-1/1	0x0	R/W	
		D1	FOUTHE	FOUTH output enable	1 Enable 0 Disable	0	R/W	
		D0	FOUT1E	FOUT1 output enable	1 Enable 0 Disable	0	R/W	
T8OSC1 Clock Control Register (OSC_T8OSC1)	0x5065 (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.
		D3–1	T8O1CK[2:0]	T8OSC1 clock division ratio select	T8O1CK[2:0] / Division ratio: 0x7–0x6 reserved; 0x5 OSC1-1/32; 0x4 OSC1-1/16; 0x3 OSC1-1/8; 0x2 OSC1-1/4; 0x1 OSC1-1/2; 0x0 OSC1-1/1	0x0	R/W	
		D0	T8O1CE	T8OSC1 clock output enable	1 Enable 0 Disable	0	R/W	

0x5080–0x5081 Clock Generator

Register name	Address	Bit	Name	Function	Setting		Init.	R/W	Remarks
PCLK Control Register (CLG_PCLK)	**0x5080** (8 bits)	D7–2	–	reserved	–		–	–	0 when being read.
		D1–0	**PCKEN[1:0]**	PCLK enable	PCKEN[1:0]	PCLK supply	0x3	R/W	
					0x3	Enable			
					0x2	Not allowed			
					0x1	Not allowed			
					0x0	Disable			
CCLK Control Register (CLG_CCLK)	**0x5081** (8 bits)	D7–2	–	reserved	–		–	–	0 when being read.
		D1–0	**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			

0x50a0–0x50a6 LCD Driver

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] / Display: 0x3 All off; 0x2 All on; 0x1 Normal display; 0x0 Display off	0x0	R/W	
LCD Contrast Adjust 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	0x0	R/W	
LCD Clock Control Register (LCD_CCTL)	**0x50a2** (8 bits)	D7–2	**–**	reserved	–	–	–	0 when being read.
		D1–0	**LDUTY[1:0]**	LCD duty select	LDUTY[1:0] / Duty: 0x3 reserved; 0x2 1/32; 0x1 1/16; 0x0 reserved	0x2	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	
		D3–0	**–**	reserved	–	–	–	0 when being read.
LCD Power Voltage Booster Control Register (LCD_PWR)	**0x50a4** (8 bits)	D7–2	**–**	reserved	–	–	–	0 when being read.
		D1	**VDSEL**	Regulator power source select	1 V_{D2} / 0 V_{DD}	0	R/W	
		D0	**PBON**	Power voltage booster control	1 On / 0 Off	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.

0x50c0–0x50c5 8-bit OSC1 Timer

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
8-bit OSC1 Timer Control Register (T8OSC1_CTL)	**0x50c0** (8 bits)	D7–5	–	reserved	–	–	–	0 when being read.
		D4	**T8ORST**	Timer reset	1 Reset 0 Ignored	0	W	
		D3–2	–	reserved	–	–	–	
		D1	**T8ORMD**	Count mode select	1 One shot 0 Repeat	0	R/W	
		D0	**T8ORUN**	Timer run/stop control	1 Run 0 Stop	0	R/W	
8-bit OSC1 Timer Counter Data Register (T8OSC1_CNT)	**0x50c1** (8 bits)	D7–0	**T8OCNT[7:0]**	Timer counter data T8OCNT7 = MSB T8OCNT0 = LSB	0x0 to 0xff	0x0	R	
8-bit OSC1 Timer Compare Data Register (T8OSC1_CMP)	**0x50c2** (8 bits)	D7–0	**T8OCMP[7:0]**	Compare data T8OCMP7 = MSB T8OCMP0 = LSB	0x0 to 0xff	0x0	R/W	
8-bit OSC1 Timer Interrupt Mask Register (T8OSC1_IMSK)	**0x50c3** (8 bits)	D7–1	–	reserved	–	–	–	0 when being read.
		D0	**T8OIE**	8-bit OSC1 timer interrupt enable	1 Enable 0 Disable	0	R/W	
8-bit OSC1 Timer Interrupt Flag Register (T8OSC1_IFLG)	**0x50c4** (8 bits)	D7–1	–	reserved	–	–	–	0 when being read.
		D0	**T8OIF**	8-bit OSC1 timer interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
8-bit OSC1 Timer PWM Duty Data Register (T8OSC1_DUTY)	**0x50c5** (8 bits)	D7–0	**T8ODTY[7:0]**	PWM output duty data T8ODTY7 = MSB T8ODTY0 = LSB	0x0 to 0xff	0x0	R/W	

0x5100–0x5104 SVD Circuit

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SVD Enable Register (SVD_EN)	**0x5100** (8 bits)	D7–1	**–**	reserved	–	–	–	0 when being read.
		D0	**SVDEN**	SVD enable	1 Enable 0 Disable	0	R/W	
SVD Compare Voltage Register (SVD_CMP)	**0x5101** (8 bits)	D7–4	**–**	reserved	–	–	–	0 when being read.
		D3–0	**SVDC[3:0]**	SVD compare voltage	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 reseved	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	
		D4–1	–	reserved	–	–	–	0 when being read.
		D0	VD1MD	Flash erase/program mode	1 Flash (2.5 V) 0 Norm.(1.8 V)	0	R/W	

0x5200–0x5213 P Port & Port MUX

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
P0 Port Input Data Register (P0_IN)	**0x5200** (8 bits)	D7–0	**P0IN[7:0]**	P0[7:0] port input data	1: 1 (H) 0: 0 (L)	×	R	
P0 Port Output Data Register (P0_OUT)	**0x5201** (8 bits)	D7–0	**P0OUT[7:0]**	P0[7:0] port output data	1: 1 (H) 0: 0 (L)	0	R/W	
P0 Port Output Enable Register (P0_OEN)	**0x5202** (8 bits)	D7–0	**P0OEN[7:0]**	P0[7:0] port output enable	1: Enable 0: Disable	0	R/W	
P0 Port Pull-up Control Register (P0_PU)	**0x5203** (8 bits)	D7–0	**P0PU[7:0]**	P0[7:0] port pull-up enable	1: Enable 0: Disable	1 (0xff)	R/W	
P0 Port 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 (0xff)	R/W	
P0 Port Interrupt Mask Register (P0_IMSK)	**0x5205** (8 bits)	D7–0	**P0IE[7:0]**	P0[7:0] port interrupt enable	1: Enable 0: Disable	0	R/W	
P0 Port Interrupt Edge Select Register (P0_EDGE)	**0x5206** (8 bits)	D7–0	**P0EDGE[7:0]**	P0[7:0] port interrupt edge select	1: Falling edge 0: Rising edge	0	R/W	
P0 Port Interrupt Flag Register (P0_IFLG)	**0x5207** (8 bits)	D7–0	**P0IF[7:0]**	P0[7:0] port interrupt flag	1: Cause of interrupt occurred 0: Cause of interrupt not occurred	0	R/W	Reset by writing 1.
P0 Port Chattering Filter Control Register (P0_CHAT)	**0x5208** (8 bits)	D7	–	reserved	–	–	–	0 when being read.
		D6–4	**P0CF2[2:0]**	P0[7:4] chattering filter time	P0CF2[2:0] / Filter time 0x7: 16384/f_{PCLK} 0x6: 8192/f_{PCLK} 0x5: 4096/f_{PCLK} 0x4: 2048/f_{PCLK} 0x3: 1024/f_{PCLK} 0x2: 512/f_{PCLK} 0x1: 256/f_{PCLK} 0x0: None	0x0	R/W	
		D3	–	reserved	–	–	–	0 when being read.
		D2–0	**P0CF1[2:0]**	P0[3:0] chattering filter time	P0CF1[2:0] / Filter time 0x7: 16384/f_{PCLK} 0x6: 8192/f_{PCLK} 0x5: 4096/f_{PCLK} 0x4: 2048/f_{PCLK} 0x3: 1024/f_{PCLK} 0x2: 512/f_{PCLK} 0x1: 256/f_{PCLK} 0x0: None	0x0	R/W	
P0 Port Key-Entry Reset Configuration Register (P0_KRST)	**0x5209** (8 bits)	D7–2	–	reserved	–	–	–	0 when being read.
		D1–0	**P0KRST[1:0]**	P0 port key-entry reset configuration	P0KRST[1:0] / Configuration 0x3: P0[3:0] = 0 0x2: P0[2:0] = 0 0x1: P0[1:0] = 0 0x0: Disable	0x0	R/W	
P0 Port Input Enable Register (P0_IEN)	**0x520a** (8 bits)	D7–0	**P0IEN[7:0]**	P0[7:0] port input enable	1: Enable 0: Disable	0xff	R/W	
P1 Port Input Data Register (P1_IN)	**0x5210** (8 bits)	D7–0	**P1IN[7:0]**	P1[7:0] port input data	1: 1 (H) 0: 0 (L)	×	R	
P1 Port Output Data Register (P1_OUT)	**0x5211** (8 bits)	D7–0	**P1OUT[7:0]**	P1[7:0] port output data	1: 1 (H) 0: 0 (L)	0	R/W	
P1 Port Output Enable Register (P1_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	

0x5214–0x5233 P Port & Port MUX

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
P1 Port Schmitt Trigger Control Register (P1_SM)	**0x5214** (8 bits)	D7–0	**P1SM[7:0]**	P1[7:0] port Schmitt trigger input enable	1 Enable (Schmitt) 0 Disable (CMOS)	1 (0xff)	R/W	
P1 Port Interrupt Mask Register (P1_IMSK)	**0x5215** (8 bits)	D7–0	**P1IE[7:0]**	P1[7:0] port interrupt enable	1 Enable 0 Disable	0	R/W	
P1 Port Interrupt Edge Select Register (P1_EDGE)	**0x5216** (8 bits)	D7–0	**P1EDGE[7:0]**	P1[7:0] port interrupt edge select	1 Falling edge 0 Rising edge	0	R/W	
P1 Port Interrupt Flag Register (P1_IFLG)	**0x5217** (8 bits)	D7–0	**P1IF[7:0]**	P1[7:0] port interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
P1 Port Chattering Filter Control Register (P1_CHAT)	**0x5218** (8 bits)	D7	–	reserved	–	–	–	0 when being read.
		D6–4	**P1CF2[2:0]**	P1[7:4] chattering filter time	P1CF2[2:0] / Filter time: 0x7 16384/fPCLK; 0x6 8192/fPCLK; 0x5 4096/fPCLK; 0x4 2048/fPCLK; 0x3 1024/fPCLK; 0x2 512/fPCLK; 0x1 256/fPCLK; 0x0 None	0	R/W	
						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	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 Disable (CMOS)	1 (0xff)	R/W	
P2 Port Input Enable Register (P2_IEN)	**0x522a** (8 bits)	D7–0	**P2IEN[7:0]**	P2[7:0] port input enable	1 Enable 0 Disable	0xff	R/W	
P3 Port Input Data Register (P3_IN)	**0x5230** (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.
		D3–0	**P3IN[3:0]**	P3[3:0] port input data	1 1 (H) 0 0 (L)	×	R	
P3 Port Output Data Register (P3_OUT)	**0x5231** (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.
		D3–0	**P3OUT[3:0]**	P3[3:0] port output data	1 1 (H) 0 0 (L)	0	R/W	
P3 Port Output Enable Register (P3_OEN)	**0x5232** (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.
		D3–0	**P3OEN[3:0]**	P3[3:0] port output enable	1 Enable 0 Disable	0	R/W	
P3 Port Pull-up Control Register (P3_PU)	**0x5233** (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.
		D3–0	**P3PU[3:0]**	P3[3:0] port pull-up enable	1 Enable 0 Disable	1 (0xff)	R/W	

0x5234–0x52a3 P Port & Port MUX

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
P3 Port Schmitt Trigger Control Register (P3_SM)	0x5234 (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.
		D3–0	P3SM[3:0]	P3[3:0] port Schmitt trigger input enable	1 Enable (Schmitt) 0 Disable (CMOS)	1 (0xff)	R/W	
P3 Port Input Enable Register (P3_IEN)	0x523a (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.
		D3–0	P3IEN[3:0]	P3[3:0] port input enable	1 Enable 0 Disable	0xff	R/W	
P0 Port Function Select Register (P0_PMUX)	0x52a0 (8 bits)	D7–6	–	reserved	–	–	–	0 when being read.
		D5	P05MUX	P05 port function select	1 TOUTN4 0 P05	0	R/W	
		D4	P04MUX	P04 port function select	1 TOUT4 0 P04	0	R/W	
		D3	P03MUX	P03 port function select	1 REMO 0 P03	0	R/W	
		D2	P02MUX	P02 port function select	1 REMI 0 P02	0	R/W	
		D1–0	–	reserved	–	–	–	0 when being read.
P1 Port Function Select Register (P1_PMUX)	0x52a1 (8 bits)	D7	P17MUX	P17 port function select	1 #SPISS 0 P17	0	R/W	
		D6	P16MUX	P16 port function select	1 TOUT5 0 P16/EXCL0	0	R/W	
		D5	P15MUX	P15 port function select	1 SCL 0 P15	0	R/W	
		D4	P14MUX	P14 port function select	1 SDA 0 P14	0	R/W	
		D3	P13MUX	P13 port function select	1 FOUT1 0 P13	0	R/W	
		D2	P12MUX	P12 port function select	1 SCLK1 0 P12	0	R/W	
		D1	P11MUX	P11 port function select	1 SOUT1 0 P11	0	R/W	
		D0	P10MUX	P10 port function select	1 SIN1 0 P10	0	R/W	
P2 Port Function Select Register (P2_PMUX)	0x52a2 (8 bits)	D7	P27MUX	P27 port function select	1 TOUTN3 0 P27	0	R/W	
		D6	P26MUX	P26 port function select	1 TOUT3 0 P26	0	R/W	
		D5	P25MUX	P25 port function select	1 SCLK0 0 P25	0	R/W	
		D4	P24MUX	P24 port function select	1 SOUT0 0 P24	0	R/W	
		D3	P23MUX	P23 port function select	1 SIN0 0 P23	0	R/W	
		D2	P22MUX	P22 port function select	1 SPICLK 0 P22	0	R/W	
		D1	P21MUX	P21 port function select	1 SDO 0 P21	0	R/W	
		D0	P20MUX	P20 port function select	1 SDI 0 P20	0	R/W	
P3 Port Function Select Register (P3_PMUX)	0x52a3 (8 bits)	D7–4	–	reserved	–	–	–	0 when being read.
		D3	P33MUX	P33 port function select	1 P33 0 DSIO	0	R/W	
		D2	P32MUX	P32 port function select	1 P32 0 DST2	0	R/W	
		D1	P31MUX	P31 port function select	1 P31 0 DCLK	0	R/W	
		D0	P30MUX	P30 port function select	1 FOUTH 0 P30	0	R/W	

0x5300–0x530c PWM & Capture Timer Ch.0

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
PWM Timer Ch.0 Compare Data A Register (T16E_CA0)	0x5300 (16 bits)	D15–0	T16ECA[15:0]	Compare data A T16ECA15 = MSB T16ECA0 = LSB	0x0 to 0xffff	0x0	R/W	
PWM Timer Ch.0 Compare Data B Register (T16E_CB0)	0x5302 (16 bits)	D15–0	T16ECB[15:0]	Compare data B T16ECB15 = MSB T16ECB0 = LSB	0x0 to 0xffff	0x0	R/W	
PWM Timer Ch.0 Counter Data Register (T16E_TC0)	0x5304 (16 bits)	D15–0	T16ETC[15:0]	Counter data T16ETC15 = MSB T16ETC0 = LSB	0x0 to 0xffff	0x0	R/W	
PWM Timer 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	
PWM Timer Ch.0 Input Clock Select Register (T16E_CLK0)	0x5308 (16 bits)	D15–4	–	reserved	–	–	–	0 when being read.
		D3–0	T16EDF[3:0]	Timer input clock select (Prescaler output clock) T16EDF[3:0] / Clock: 0xf reserved 0xe PCLK-1/16384 0xd PCLK-1/8192 0xc PCLK-1/4096 0xb PCLK-1/2048 0xa PCLK-1/1024 0x9 PCLK-1/512 0x8 PCLK-1/256 0x7 PCLK-1/128 0x6 PCLK-1/64 0x5 PCLK-1/32 0x4 PCLK-1/16 0x3 PCLK-1/8 0x2 PCLK-1/4 0x1 PCLK-1/2 0x0 PCLK-1/1		0x0	R/W	
PWM Timer Ch.0 Interrupt Mask Register (T16E_IMSK0)	0x530a (16 bits)	D15–2	–	reserved	–	–	–	0 when being read.
		D1	CBIE	Compare B interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	CAIE	Compare A interrupt enable	1 Enable 0 Disable	0	R/W	
PWM Timer Ch.0 Interrupt Flag Register (T16E_IFLG0)	0x530c (16 bits)	D15–2	–	reserved	–	–	–	0 when being read.
		D1	CBIF	Compare B interrupt flag	1 Cause of interrupt occurred 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D0	CAIF	Compare A interrupt flag		0	R/W	

0x5320–0x532c MISC Registers

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
FLASHC/ SRAMC Control Register (MISC_FL)	0x5320 (16 bits)	D15–10	–	reserved	–	–	–	0 when being read.
		D9–8	SRCYC[1:0]	SRAMC access cycle	SRCYC[1:0] / Access cycle: 0x3 / 5 cycles; 0x2 / 4 cycles; 0x1 / 3 cycles; 0x0 / 2 cycles	0x3	R/W	
		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	OSC1 peripheral control in debug mode	1 Run / 0 Stop	0	R/W	
MISC Protect Register (MISC_PROT)	0x5324 (16 bits)	D15–0	PROT[15:0]	MISC register write protect	Writing 0x96 removes the write protection of the MISC registers (0x5326–0x532a). Writing another value set the write protection.	0x0	R/W	
IRAM Size Select Register (MISC_IRAMSZ)	0x5326 (16 bits)	D15–9	–	reserved	–	–	–	0 when being read.
		D8	DBADR	Debug base address select	1 0x0 / 0 0xfffc00	0	R/W	
		D7–2	–	reserved	–	–	–	0 when being read.
		D1–0	IRAMSZ[1:0]	IRAM size select	IRAMSZ[1:0] / Read cycle: 0x3 / reseved; 0x2 / reseved; 0x1 / reseved; 0x0 / reseved	0x0	R/W	
Vector Table Address Low Register (MISC_TTBRL)	0x5328 (16 bits)	D15–8	TTBR[15:8]	Vector table base address A[15:8]	0x0–0xff	0x80	R/W	
		D7–0	TTBR[7:0]	Vector table base address A[7:0] (fixed at 0)	0x0	0x0	R	
Vector Table Address High Register (MISC_TTBRH)	0x532a (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–0	TTBR[23:16]	Vector table base address A[23:16]	0x0–0xff	0x0	R/W	
PSR Register (MISC_PSR)	0x532c (16 bits)	D15–8	–	reserved	–	–	–	0 when being read.
		D7–5	PSRIL[2:0]	PSR interrupt level (IL) bits	0x0 to 0x7	0x0	R	
		D4	PSRIE	PSR interrupt enable (IE) bit	1 1 (enable) / 0 0 (disable)	0	R	
		D3	PSRC	PSR carry (C) flag	1 1 (set) / 0 0 (cleared)	0	R	
		D2	PSRV	PSR overflow (V) flag	1 1 (set) / 0 0 (cleared)	0	R	
		D1	PSRZ	PSR zero (Z) flag	1 1 (set) / 0 0 (cleared)	0	R	
		D0	PSRN	PSR negative (N) flag	1 1 (set) / 0 0 (cleared)	0	R	

0x5340–0x5346 Remote Controller

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
REMC Configuration Register (REMC_CFG)	**0x5340** (16 bits)	D15–12	**CGCLK[3:0]**	Carrier generator clock select (Prescaler output clock)	CGCLK[3:0] / LCCLK[3:0] = 0xf: reserved; 0xe: PCLK-1/16384; 0xd: PCLK-1/8192; 0xc: PCLK-1/4096; 0xb: PCLK-1/2048; 0xa: PCLK-1/1024; 0x9: PCLK-1/512; 0x8: PCLK-1/256; 0x7: PCLK-1/128; 0x6: PCLK-1/64; 0x5: PCLK-1/32; 0x4: PCLK-1/16; 0x3: PCLK-1/8; 0x2: PCLK-1/4; 0x1: PCLK-1/2; 0x0: PCLK-1/1	0x0	R/W	
		D11–8	**LCCLK[3:0]**	Length counter clock select (Prescaler output clock)		0x0	R/W	
		D7–2	–	reserved	–	–	–	0 when being read.
		D1	**REMMD**	REMC mode select	1 Receive / 0 Transmit	0	R/W	
		D0	**REMEN**	REMC enable	1 Enable / 0 Disable	0	R/W	
REMC Carrier Length Setup Register (REMC_CAR)	**0x5342** (16 bits)	D15–14	–	reserved	–	–	–	0 when being read.
		D13–8	**REMCL[5:0]**	L carrier length setup	0x0 to 0x3f	0x0	R/W	
		D7–6	–	reserved	–	–	–	0 when being read.
		D5–0	**REMCH[5:0]**	H carrier length setup	0x0 to 0x3f	0x0	R/W	
REMC Length Counter Register (REMC_LCNT)	**0x5344** (16 bits)	D15–8	**REMLEN[7:0]**	Transmit/receive data length count (down counter)	0x0 to 0xff	0x0	R/W	
		D7–1	–	reserved	–	–	–	0 when being read.
		D0	**REMDT**	Transmit/receive data	1 1 (H) / 0 0 (L)	0	R/W	
REMC Interrupt Control Register (REMC_INT)	**0x5346** (16 bits)	D15–11	–	reserved	–	–	–	0 when being read.
		D10	**REMFIF**	Falling edge interrupt flag	1 Cause of interrupt occurred / 0 Cause of interrupt not occurred	0	R/W	Reset by writing 1.
		D9	**REMRIF**	Rising edge interrupt flag		0	R/W	
		D8	**REMUIF**	Underflow interrupt flag		0	R/W	
		D7–3	–	reserved	–	–	–	0 when being read.
		D2	**REMFIE**	Falling edge interrupt enable	1 Enable / 0 Disable	0	R/W	
		D1	**REMRIE**	Rising edge interrupt enable	1 Enable / 0 Disable	0	R/W	
		D0	**REMUIE**	Underflow interrupt enable	1 Enable / 0 Disable	0	R/W	

0x5360–0x536c PWM & Capture Timer Ch.1

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
PWM Timer Ch.1 Compare Data A Register (T16E_CA1)	0x5360 (16 bits)	D15–0	T16ECA[15:0]	Compare data A T16ECA15 = MSB T16ECA0 = LSB	0x0 to 0xffff	0x0	R/W	
PWM Timer Ch.1 Compare Data B Register (T16E_CB1)	0x5362 (16 bits)	D15–0	T16ECB[15:0]	Compare data B T16ECB15 = MSB T16ECB0 = LSB	0x0 to 0xffff	0x0	R/W	
PWM Timer Ch.1 Counter Data Register (T16E_TC1)	0x5364 (16 bits)	D15–0	T16ETC[15:0]	Counter data T16ETC15 = MSB T16ETC0 = LSB	0x0 to 0xffff	0x0	R/W	
PWM Timer Ch.1 Control Register (T16E_CTL1)	0x5366 (16 bits)	D15–9	–	reserved	–	–	–	0 when being read.
		D8	INITOL	Initial output level	1 High 0 Low	0	R/W	
		D7	–	reserved	–	–	–	0 when being read.
		D6	SELFM	Fine mode select	1 Fine mode 0 Normal mode	0	R/W	
		D5	CBUFEN	Comparison buffer enable	1 Enable 0 Disable	0	R/W	
		D4	INVOUT	Inverse output	1 Invert 0 Normal	0	R/W	
		D3	CLKSEL	Input clock select	1 External 0 Internal	0	R/W	
		D2	OUTEN	Clock output enable	1 Enable 0 Disable	0	R/W	
		D1	T16ERST	Timer reset	1 Reset 0 Ignored	0	W	0 when being read.
		D0	T16ERUN	Timer run/stop control	1 Run 0 Stop	0	R/W	
PWM Timer Ch.1 Input Clock Select Register (T16E_CLK1)	0x5368 (16 bits)	D15–4	–	reserved	–	–	–	0 when being read.
		D3–0	T16EDF[3:0]	Timer input clock select (Prescaler output clock) T16EDF[3:0] Clock 0xf reserved 0xe PCLK-1/16384 0xd PCLK-1/8192 0xc PCLK-1/4096 0xb PCLK-1/2048 0xa PCLK-1/1024 0x9 PCLK-1/512 0x8 PCLK-1/256 0x7 PCLK-1/128 0x6 PCLK-1/64 0x5 PCLK-1/32 0x4 PCLK-1/16 0x3 PCLK-1/8 0x2 PCLK-1/4 0x1 PCLK-1/2 0x0 PCLK-1/1		0x0	R/W	
PWM Timer Ch.1 Interrupt Mask Register (T16E_IMSK1)	0x536a (16 bits)	D15–2	–	reserved	–	–	–	0 when being read.
		D1	CBIE	Compare B interrupt enable	1 Enable 0 Disable	0	R/W	
		D0	CAIE	Compare A interrupt enable	1 Enable 0 Disable	0	R/W	
PWM Timer Ch.1 Interrupt Flag Register (T16E_IFLG1)	0x536c (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	

0xffff84–0xffffd0 S1C17 Core I/O

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
Processor ID Register (IDIR)	**0xffff84** (8 bits)	D7–0	**IDIR[7:0]**	Processor ID 0x10: S1C17 Core	0x10	0x10	R	
Debug RAM Base Register (DBRAM)	**0xffff90** (32 bits)	D31–24	**–**	Unused (fixed at 0)	0x0	0x0	R	
		D23–0	**DBRAM[23:0]**	Debug RAM base address	0x2fc0	0x2fc0	R	
Debug Control Register (DCR)	**0xffffa0** (8 bits)	D7	**IBE4**	Instruction break #4 enable	1 Enable 0 Disable	0	R/W	
		D6	**IBE3**	Instruction break #3 enable	1 Enable 0 Disable	0	R/W	
		D5	**IBE2**	Instruction break #2 enable	1 Enable 0 Disable	0	R/W	
		D4	**DR**	Debug request flag	1 Occurred 0 Not occurred	0	R/W	Reset by writing 1.
		D3	**IBE1**	Instruction break #1 enable	1 Enable 0 Disable	0	R/W	
		D2	**IBE0**	Instruction break #0 enable	1 Enable 0 Disable	0	R/W	
		D1	**SE**	Single step enable	1 Enable 0 Disable	0	R/W	
		D0	**DM**	Debug mode	1 Debug mode 0 User mode	0	R	
Instruction Break Address Register 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 Flash Memory Programming

Flash memory programming consists of programming via a debugger using the flash programer function possessed by ICDs (in-circuit debuggers) such as the S5U1C17001H (ICD Mini) or self-programming via user programs.

B.1 Debugger Programming

The debuggers included in the S1C17 Family C compiler packages provide functions that allow an ICD (e.g., S5U1C17001H) to be used as a flash programer.

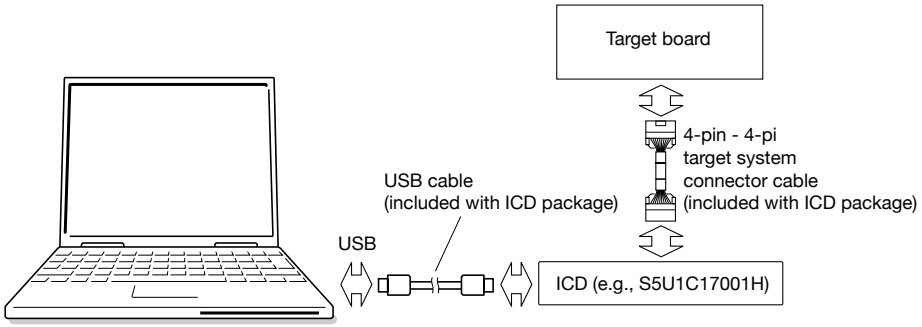


Figure B.1.1: Flash memory programming system using debugger

To program S1C17702 flash memory using this function, you must install a 4-pin connector on the target board to connect the ICD (e.g., S5U1C17001H).
Connect a 4-pin connector using the S1C17702 DCLK (P31), DST2 (P32), and DSIO (P33) pins as debugging pins. (Note that this means P31 to P33 general input/output ports cannot be used.)

For more information on flash memory programming using this system, refer to the manual for the S1C17 Family C compiler package (e.g., S5U1C17001H). For more information on the 4-pin connector pin layout, refer to the ICD (e.g., S5U1C17001H) manual.

B.2 Self-programming via User Programs

The S1C17702 includes self-programming functions for erasing and overwriting flash memory by user programs executed while operating on the target board. A separately provided self-programming package provides function routines as object files for self-programming. Self-programming functions are easily added by linking these objects to user application programs. For more information, see the self-programming package manual.

Appendix C 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.

C.1 Clock Control Power Saving

Figure C.1.1 illustrates the S1C17702 clock system.

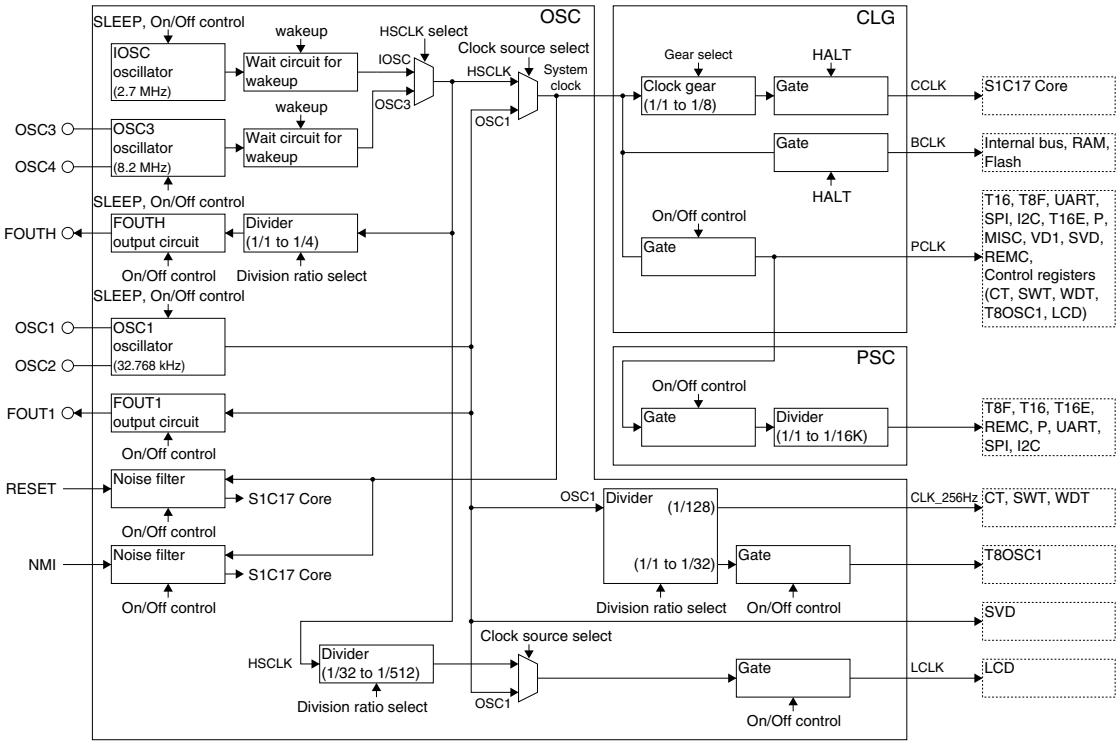


Figure C.1.1 Clock system

This section describes clock systems that can be controlled via software and power-saving control details. For more information on control registers and control methods, refer to the respective module sections.

System SLEEP (All clocks stopped)

- Execute `slp` instruction
  Execute the `slp` instruction when the entire system can be stopped. The CPU switches to SLEEP mode and the system clocks stop. This also stops all peripheral circuits using clocks. Starting up the CPU from SLEEP mode is therefore limited to startup using ports (described later).

System clocks

- Clock source selection (OSC module)
  Select between IOSC/OSC3 and OSC1 for the system clock source. Reduce current consumption by selecting the OSC1 clock when low-speed processing is possible.
- IOSC/OSC3 oscillator circuit stop (OSC module)
  Operate the oscillator circuit comprising the system clock source. Where possible, stop the other circuit. You can reduce current consumption by using OSC1 as the system clock and stopping the IOSC/OSC3 oscillator circuit.

CPU clock (CCLK)

- Execute the `halt` instruction
  Execute the `halt` instruction when program execution by the CPU is not required—for example, when only the display is required or for interrupt standby. The CPU switches to HALT mode and suspends operations, but the peripheral circuits maintain the status in place at the time of the `halt` instruction, enabling use of peripheral circuits for timers and interrupts. You can reduce power consumption even further by suspending unnecessary peripheral circuits before executing the halt instruction. The CPU is started from HALT mode using the port or interrupts from the peripheral circuit operating in HALT mode.
- Low-speed clock gear selection (CLG module)
  The CLG module can reduce CPU clock speeds to between 1/1 and 1/8 of the system clock via the clock gear settings. Reduce current consumption by operating the CPU at the minimum speed required for applications.

Peripheral clock (PCLK)

- PCLK stop (CLG module)
  Stop the PCLK clock feed from the CLG to peripheral circuits if none of the following peripheral circuits is required.

  Peripheral circuits operating with PCLK
  - Prescaler (PWM & capture timer, remote controller, P port)
  - UART
  - 8-bit time Ch.0 to Ch.1
  - 16-bit timer Ch.0 to Ch.2
  - SPI
  - I^2C
  - SVD circuit
  - Power supply control circuit
  - P port and port MUX (control register, chattering filter)
  - PWM & capture timer
  - MISC register
  - Remote controller

The peripheral modules listed below are operated by clocks other than PCLK. This means PCLK is not required.

- Clock timer
- Stopwatch timer
- Watchdog timer
- 8-bit OSC1 timer

The peripheral modules shown below require PCLK only when accessing the control register. PCLK is not required after the control register has been set and operations have begun.

- LCD driver

Table C.1.1 shows a list of methods for clock control and starting/stopping the CPU.

Table C.1.1: Clock control list

Current consumption	OSC1	OSC3	CPU (CCLK)	PCLK peripheral	OSC1 peripheral	CPU stop method	CPU startup method
↑ Low	Stop	Stop	Stop	Stop	Stop	Execute **slp** instruction	1
	Oscillation (system CLK)	Stop	Stop	Stop	Operation	Execute **halt** instruction	1, 2
	Oscillation (system CLK)	Stop	Stop	Operation	Operation	Execute **halt** instruction	1, 2, 3
	Oscillation	Stop	Operation(1/1)	Operation	Operation		
	Oscillation	Oscillation (system CLK)	Stop	Operation	Operation	Execute **halt** instruction	1, 2, 3
	Oscillation	Oscillation (system CLK)	Operation (Low gear)	Operation	Operation		
High ↓	Oscillation	Oscillation (system CLK)	Operation(1/1)	Operation	Operation		

HALT and SLEEP mode clearing methods (CPU startup method)

1. Startup by port
   Started up by input/output port interrupt and debug interrupt (ICD forced break).
2. Startup by OSC1 peripheral circuit
   Started up by clock timer, stopwatch timer, watchdog timer, 8-bit OSC1 timer, or LCD driver interrupts.
3. Startup by PCLK peripheral circuit
   Started up by PCLK peripheral circuit interrupt.

C.2 Reducing Power Consumption via Power Supply Control

The available power supply controls are listed below.

Internal voltage regulator

- Setting the internal operating voltage VD1 to 2.5 V increases current consumption.
  For normal operations, set VD1 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

- Turning on the power supply voltage booster circuit will increase current consumption.
  Turn off the power supply voltage booster circuit if the power supply voltage VDD is 2.5 V or more. Use VDD to drive the LCD voltage regulator. Use the power supply voltage booster circuit only if VDD is less 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 and the power supply voltage booster circuit.

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 D Mounting Precautions

This section describes various precautions for circuit board design and IC mounting.

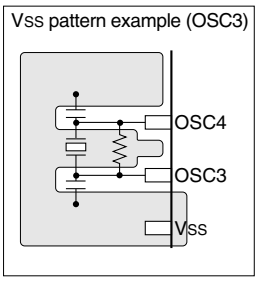
Oscillator circuit

- Oscillation characteristics depend on factors such as components used (oscillator, R_f, C_G, C_D) and circuit board patterns. In particular, with ceramic or crystal oscillators, select the appropriate external resistors (R_f,) and capacitors (C_G, C_D) only after fully evaluating components actually mounted on the circuit board.
- Oscillator clock disturbances caused by noise may cause malfunctions. To prevent such disturbances, consider the following points. The latest devices, in particular, are manufactured by microscopic processes, making them especially susceptible to noise.
  Areas in which noise countermeasures are especially important include the OSC2 pin and related circuit components and wiring. OSC1 pin handling is equally important. The noise precautions required for the OSC1 and OSC2 pins are described below.
  We also recommend applying similar noise countermeasures to high-speed oscillator circuits, such as the OSC3 and OSC4 pins and wiring.

(1) Components such as oscillators, resistors, and capacitors connected to the OSC1 (OSC3) and OSC2 (OSC4) pins should have the shortest connections possible.

(2) Wherever possible, avoid locating digital signal lines within 3 mm of the OSC1 (OSC3) and OSC2 (OSC4) pins or related circuit components and wiring. Rapidly-switching signals, in particular, should be kept at a distance from these components. Since the spacing between layers of multi-layer printed circuit boards is a mere 0.1 mm to 0.2 mm, the above precautions also apply when positioning digital signal lines on other layers.
Never place digital signal lines alongside such components or wiring, even if more than 3 mm distance or located on other layers. Avoid crossing wires.

(3) Use V_{SS} to shield OSC1 (OSC3) and OSC2 (OSC4) pins and related wiring (including wiring for adjacent circuit board layers).
Layers wired should be adequately shielded as shown to the right.
Fully ground adjacent layers, where possible. At minimum, shield the area at least 5 mm around the above pins and wiring.
Even after implementing these precautions, avoid configuring digital signal lines in parallel, as described in (2) above. Avoid crossing even on discrete layers, except for lines carrying signals with low switching frequencies.



(4) After implementing these precautions, check the output clock waveform by running the actual application program within the product.
Use an oscilloscope to check outputs from the FOUT1 and FOUTH pins.
You can check the quality of the OSC3 output waveform via the FOUTH output. Confirm that the frequency is as designed, is free of noise, and has minimal jitter.
You can check the quality of the OSC1 waveform via the FOUT1 output. In particular, enlarge the areas before and after the clock rising and falling edges and take special care to confirm that the regions approximately 100 ns to either side are free of clock or spiking noise.

Failure to observe precautions (1) to (3) adequately may lead to jitter in the OSC3 output and noise in the OSC1 output. Jitter in the OSC3 output will reduce operating frequencies, while noise in the OSC1 output will destabilize timers operated by the OSC1 clock as well as CPU core operations when the system clock switches to OSC1.

Reset circuit

- The reset signal input to the #RESET pin when power is turned on will vary, depending on various factors, such as power supply start-up time, components used, and circuit board patterns. Constants such as capacitance and resistance should be determined through thorough testing with real-world products. Account for resistance fluctuations when setting the #RESET pin pull-up resistance for constants settings.
- Components such as capacitors and resistors connected to the #RESET pin should have the shortest connections possible to prevent noise-induced resets.

Power supply circuit

Sudden power supply fluctuations due to noise will cause malfunctions. Consider the following issues.

(1) Connections from the power supply to the VDD and VSS pins should be implemented via the shortest, thickest patterns possible.

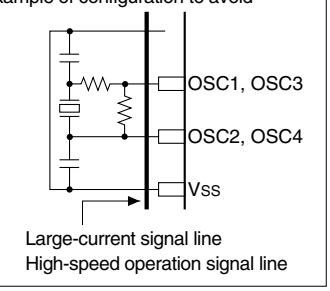
(2) If a bypass capacitor is connected between VDD and VSS, connections between the VDD and VSS pins should be as short as possible.

Bypass capacitor connection example

Signal line location

- To prevent electromagnetically-induced noise arising from mutual induction large-current signal lines should not be positioned close to circuits susceptible to noise, such as oscillators.
- Locating signal lines in parallel over significant distances or crossing signal lines operating at high speed will cause malfunctions due to noise generated by mutual interference.
  Specifically, avoid positioning crossing signal lines operating at high speed close to circuits susceptible to noise, such as oscillators.

Example of configuration to avoid



Noise-induced malfunctions

Check the following three points if you suspect the presence of noise-induced IC malfunctions.

(1) DSIO pin
Low-level noise to this pin will cause a switch to Debug mode. The switch to Debug mode can be confirmed by the clock output from DCLK and a High signal from the DST2 pin.
For the product version, we recommend connecting the DSIO pin directly to HV_{DD} or pulling up the DISO pin using a resistor not exceeding 10 kΩ.
The IC includes an internal pull-up resistor. The resistor has a relatively high impedance of 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) V_{DD} and V_{SS} power supply
The IC will malfunction the instant noise falling below the rated voltage is input.
Incorporate countermeasures on the circuit board, including close patterns for circuit board power supply circuits, noise-filtering decoupling capacitors, and surge/noise prevention components on the power supply line.

Perform the inspections described above using an oscilloscope capable of observing waveforms of at least 200 MHz. It may not be possible to observe high-speed noise events with a low-speed oscilloscope.

If you detect potential noise-induced malfunctions while observing the waveform with an oscilloscope, recheck with a low-impedance (less than 1 kΩ) resistor connecting the relevant pin to GND or to the power supply. Malfunctions at that pin are likely if changes are visible, such as the malfunction disappearing, becoming less frequent, or the phenomena changing.

The DSIO and #RESET input circuits described above detect input signal edges and are susceptible to malfunctions induced by spike noise. This makes these digital signal pins the most susceptible to noise.
To reduce potential noise, keep the following two points in mind when designing circuit boards:

(A) It is important to use low impedance resistors when driving the signals, as described above. Avoid connecting impedance exceeding 1 kΩ (ideally, 0 Ω) to the power supply or GND. The signal lines connected should be no longer than approximately 5 mm.

(B) Signals switching from 1 to 0 or 0 to 1 may generate noise if signal lines run parallel to other digital lines on the circuit board.
The highest risk of noise occurs in configurations in which a line is sandwiched between multiple signal lines that vary in synchrony. You can minimize noise effects by reducing the length of parallel sections (limit to a few cm) or by increasing the separation (to at least 2 mm).

Handling of light (for bare chip mounting)

The characteristics of semiconductor components can vary when exposed to light. ICs may malfunction or nonvolatile memory data may be corrupted if ICs are exposed to light.
Consider the following precautions for circuit boards and products in which this IC is mounted to prevent IC malfunctions attributable to light exposure.

(1) Design and mount the product so that the IC is shielded from light during use.

(2) Shield the IC from light during inspection processes.

(3) Shield the IC on the upper, underside, and side faces of the IC chip.

(4) Mount the IC chip within one week of opening the package. If the IC chip must be stored before mounting, take measures to ensure light shielding.

(5) Adequate evaluations are required to assess nonvolatile memory data retention characteristics before product delivery if the product is subjected to heat stress exceeding regular reflow conditions during mounting processes.

Miscellaneous

This product series is manufactured using 0.25 µm microscopic processes.
Although it is designed to ensure basic IC reliability meeting EIAJ and MIL standards, consider the following points when mounting the product.
All oscillator circuit input/output pins use direct connections to internal 0.25 µm transistors. In addition to physical damage during mounting, minor variations over time may result in electrical damage arising from disturbances in the form of voltages exceeding the absolute maximum rating (2.5 V). The following factors can give rise to these variations:

(1) electromagnetically-induced noise from industrial power supplies used in mounting reflow, reworking after mounting, and individual characteristic evaluation (testing) processes;

(2) electromagnetically-induced noise from a solder iron when soldering.

In particular, during soldering, take care to ensure that the soldering iron GND (tip potential) has the same potential as the IC GND.

Appendix E Initialization Routine

This section lists typical vector tables and initialization routines.

boot.s

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

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

.long BOOT              ; 0x00       0x00    reset                            ...(2)
.long unalign_handler   ; 0x01       0x04    unalign
.long nmi_handler       ; 0x02       0x08    NMI
.long int03_handler     ; 0x03       0x0c    -
.long p0_handler        ; 0x04       0x10    P0 port
.long p1_handler        ; 0x05       0x14    P1 port
.long swt_handler       ; 0x06       0x18    SWT
.long ct_handler        ; 0x07       0x1c    CT
.long t8osc1_handler    ; 0x08       0x20    T8OSC1
.long int09_handler     ; 0x09       0x24    SVD
.long lcd_handler       ; 0x0a       0x28    LCD
.long t16e_0_handler    ; 0x0b       0x2c    T16E ch0
.long t8f_handler       ; 0x0c       0x30    T8F
.long t16_0_handler     ; 0x0d       0x34    T16 ch0
.long t16_1_handler     ; 0x0e       0x38    T16 ch1
.long t16_2_handler     ; 0x0f       0x3c    T16 ch2
.long uart_0_handler    ; 0x10       0x40    UART ch0
.long uart_1_handler    ; 0x11       0x44    UART ch1
.long spi_handler       ; 0x12       0x48    SPI
.long i2c_handler       ; 0x13       0x4c    I2C
.long remc_handler      ; 0x14       0x50    REMC
.long t16e_1_handler    ; 0x15       0x54    T16E ch1
.long int16_handler     ; 0x16       0x58    -
.long int17_handler     ; 0x17       0x5c    -
.long int18_handler     ; 0x18       0x60    -
.long int19_handler     ; 0x19       0x64    -
.long int1a_handler     ; 0x1a       0x68    -
.long int1b_handler     ; 0x1b       0x6c    -
.long int1c_handler     ; 0x1c       0x70    -
.long int1d_handler     ; 0x1d       0x74    -
.long int1e_handler     ; 0x1e       0x78    -
.long int1f_handler     ; 0x1f       0x7c    -

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

.text                                                                         ...(3)
.align 1

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

        ; ----- Stack pointer --------------------
        Xld.a   %sp, 0x0f00                                                   ...(4)

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

        ; FLASHC
        Xld.a   %r0, 0x04       ; 1 cycle access, under 6 MHz system clock
        ld.b    [%r1], %r0      ; [0x5320] <= 0x04                            ...(5)

        ; SRAMC
        Xld.a   %r0, 0x00       ; 2 cycle access
        ext     0x01
        ld.b    [%r1], %r0      ; [0x5321] <= 0x01                            ...(6)
```

```
        ; ===== Main routine ========================================
         ...

; ========================================================================
;       Interrupt handler
; ========================================================================
; ----- Address unalign --------------------------
unalign_handler:
        ...

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

---

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

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

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

(4) Sets stack pointer.

(5) Sets the number of flash memory controller access cycles.
Can be set to 1-cycle access for C17702.
(See "3 Memory Map and Bus Control.")

(6) Sets the SRAM controller access conditions.
(See "3 Memory Map and Bus Control.")

Revision History

Code No.	Page	Contents
411581700	All	New establishment
411581702	1-1	Overview (Old) Based on an S1C17 CPU core, this product incorporates 128 KB of flash memory, ... a serial interfaces supporting sensors such as UART, SPI, and I2C to support high-bit rate ... 32 kHz and max. 8.2 MHz oscillator circuit, and internal 1.8 V voltage regulator. (New) Based on an S1C17 CPU core, this product incorporates a 128KB of flash memory, ... serial interfaces such as UART supporting high-bit rate and IrDA1.0, SPI, and I^2C to support various sensors, ... 32 kHz and max. 8.2 MHz oscillator circuits, and a 1.8 V voltage regulator.
	1-1, 1-2	Overview (Old) S5U1C17702H (New) S5U1C17001H
	1-2	Features: Power supply voltage detection (SVD) circuit (Old) • 16-value programmable (1.7 V to 3.2 V) (New) • 15-value programmable (1.8 V to 3.2 V)
	1-2, 27-1, 27-2, 27-6, 27-7	Features: Operating temperature (Old) • -20°C to 70°C (New) • -25°C to 70°C
	1-3	Features: Configuration as shipped (Old) No description (New) • VFBGA10H-180 package (10 mm × 10 mm, ball pitch: 0.65 mm)
	1-6	Pinout diagram: VFBGA8H-181 Modified Figure 1.3.1.2
	1-7	Pinout diagram: VFBGA10H-180 Added Figure 1.3.1.3
	1-11	Pin descriptions: ∗4: VSS ball numbers (Old) B14, ... F6, F... M8, M14 (New) B14, ... F6(note), F... M8, M14 (note) The F6 pin does not exist in the VFBGA10H-180 package.
	2-5	CPU: S1C17 core instruction list (Table 2.3.1) (Old) ipa.d (New) jpa.d
	3-1	S1C17702 memory map Modified Figure 3.1
	3-5	Internal RAM (Old) The S1C17702 enables the RAM size used to apply restrictions to 12 KB, 8 KB, 4 KB, or 2 KB. ... The RAM size is selected using IRAMSZ[1:0] (D[1:0]/MISC_IRAMSZ register). (New) Deleted
	3-5, 24-6	Internal RAM Modified the register table, Tables 3.3.1.1 and 24.4.2
		Internal RAM (Old) Note: The IRAM Size Select Register is write-protected. ... may result in system malfunctions. (New) Note: • The IRAM Size Select Register is write-protected. ... may result in system malfunctions. • Do not alter IRAMSZ[1:0] (D[1:0]/MISC_IRAMSZ register) from the default value.
	4-1	Power supply (Old) 4 Power Supply Voltage (New) 4 Power Supply
		Power supply: Power supply voltage (Old) Supply voltages within the respective ranges to LVDD and HVDD pins with the VSS pin as GND. (New) Supply voltages within the range to VDD pins with the VSS pin as GND.
		Power supply: Power supply voltage (Old) The VFBGA8H-181 package has three VDD pins and nineteen VSS pins. (New) The VFBGA8H-181 package has three VDD pins and 19 VSS pins. The VFBGA10H-180 package has three VDD pins and 18 VSS pins.
	4-5, 4-7	Power supply: Heavy load protection function (Old) Make this setting before driving heavy loads such as lamps and buzzers using the port output. (New) VD1 may become unstable in the operations shown below and in other conditions. ... Note: Always cancel heavy load protection mode ... while it repeating the processing.
	4-10	Power supply: Precautions (Old) No description (New) • Always cancel heavy load protection mode ... while it repeating the processing.
	7-1	OSC: OSC module configuration Modified Figure 7.1.1
	7-2	OSC: IOSC oscillator circuit (Old) The oscillator clock is operated by the S1C17 core and peripheral circuits following an initial reset. (New) The S1C17 core and peripheral circuits operate with this oscillator clock after an initial reset.

REVISION HISTORY

Code No.	Page	Contents
411581702	7-2	OSC: IOSC oscillator circuit (Old) The IOSC oscillator circuit stops oscillating even in SLEEP mode. (New) The IOSC oscillator circuit also stops oscillating in SLEEP mode.
		OSC: IOSC oscillator circuit (Old) Following initial resetting, IOSCEN is set to 1, and the IOSC oscillator circuit is on. (New) Following initial resetting, IOSCEN is set to 1 to turn the IOSC oscillator circuit on.
	7-3	OSC: OSC3 oscillator circuit (Old) The OSC3 oscillator circuit stops oscillating even in SLEEP mode. (New) The OSC3 oscillator circuit also stops oscillating in SLEEP mode.
		OSC: OSC3 oscillator circuit (Old) No description (New) OSC3 external clock input An external clock can be input to the OSC3 pin. ... see “27 Electrical Characteristics.”
	7-4	OSC: OSC1 oscillator circuit (Old) The OSC1 oscillator circuit stops oscillating even in SLEEP mode. (New) The OSC1 oscillator circuit also stops oscillating in SLEEP mode.
	7-5	OSC: High-speed clock (HSCLK) selection (Old) When switch the system clock between the HSCLK and OSC1 clocks, it takes switching time from one cycle of HSCLK (min.) to one cycle of OSC1 (max.) (New) Deleted
	7-5, 7-19	OSC: High-speed clock (HSCLK) selection (Old) Note: Both the IOSC and OSC3 oscillator circuits ... the HSCLKSEL value will remain unchanged. (New) Note: Both the IOSC and OSC3 oscillator circuits ... the HSCLKSEL value will remain unchanged. When selecting HSCLK, make sure PCKEN[1:0] (D[1:0]/CLG_PCLK register) is set to 0x3 (on) before writing to HSCLKSEL. ∗ PCKEN[1:0]: PCLK Enable Bits in the PCLK Control (CLG_PCLK) Register (D[1:0]/0x5080)
	7-5	OSC: System clock (OSC1 or HSCLK) selection (Old) To select HSCLK as the system clock, write 0 to SRCSRC with HSCLK running. (New) To select HSCLK as the system clock, write 0 to CLKSRC with HSCLK running. When switch the system clock between the HSCLK and OSC1 clocks, it takes switching time from one cycle of HSCLK (min.) to one cycle of OSC1 (max.).
	7-6	OSC: LCD clock control (Old) Clock feed control ... Setting LCKEN to 1 sends the clock generated as above to the LCD driver. If no LCD display is required, stop the clock feed to minimize current consumption. ∗ LCKEN: LCD Clock Enable Bit in the LCD Clock Setup (OSC_LCLK) Register (D0/0x5063) (New) Clock feed control ... Setting LCKEN to 1 sends the clock generated as above to the LCD driver. ∗ LCKEN: LCD Clock Enable Bit in the LCD Clock Setup (OSC_LCLK) Register (D0/0x5063) Note: Be sure to wait at least one LCLK clock cycle before setting LCKEN to 0 to stop the LCLK ... DSPC[1:0] is set to a value other than 0x0 after LCKEN is set to 1 to resume clock feed. ∗ DSPC[1:0]: LCD Display Control Bits in the ... (LCD_DCTL) Register (D[1:0]/0x50a0)
	7-8, 7-9	OSC: Clock external output (FOUTH, FOUT1) Modified Figures 7.8.1, 7.8.2, and 7.8.3
	7-8, 7-17	OSC: Clock external output (FOUTH, FOUT1) Modified Table 7.8.1
	7-8	OSC: FOUTH clock frequency selection (Old) Select the division ratio for the OSC3 clock using FOUTHD[1:0] (D[3:2]/OSC_FOUT register). (New) Select the division ratio for the OSC3 clock using FOUTHD[1:0] (D[3:2]/OSC_FOUT register).
	7-10	OSC: RESET and NMI input noise filters (Old) Note: • All noise filters should normally be enabled. (New) Note: • The RESET input noise filter should normally be enabled.
	7-19	OSC: Precautions (Old) No description (New) • Be sure to wait at least one LCLK clock ... 0x0 after LCKEN is set to 1 to resume clock feed.
	8-1	CLG: CLG module configuration Modified Figure 8.1.1
	8-3	CLG: Peripheral module clock control circuit Modified Figure 8.3.1
	8-5, 8-7	CLG: PCLK Control Register (CLG_PCLK) - (D[1:0]) PCKEN[1:0]: PCLK Enable Bits (Old) Peripheral modules operated using PCLK ... • 16-bit timer Ch.0 to 2 • Interrupt controller • SPI ... (New) Peripheral modules operated using PCLK ... • 16-bit timer Ch.0 to Ch.2 • SPI ...

Code No.	Page	Contents
411581702	10-8	P: Port input interrupt circuit configuration Modified Figure 10.7.1
	10-9	P: Interrupt flags (Old) Setting the corresponding P*x*IE[7:0] to 1 sets P*x*IF[7:0] to 1 at the specified edge (rising or falling edge) of the input signal. A P0 or P1 port interrupt request signal is also output to the ITC at the same time. (New) P*x*IF[7:0] is set to 1 at the specified edge (rising or falling edge) of the input signal. A P0 or P1 port interrupt request signal is also output to the ITC at the same time if the corresponding P*x*IE[7:0] is set to 1.
	10-18	P: P*x* Port Interrupt Flag Registers (P*x*_IFLG) - (D[7:0]) P*x*IF[7:0]: P*x*[7:0] Port Interrupt Flags (Old) Setting the corresponding P*x*IE[7:0] (P*x*_IMSK register) to 1 sets P*x*IF[7:0] to 1 at the specified edge (rising or falling edge) of the input signal. A P0 or P1 port interrupt request signal is also output to the ITC at the same time. (New) P*x*IF[7:0] is set to 1 at the specified edge (rising or falling edge) of the input signal. A P0 or P1 port interrupt request signal is also output to the ITC at the same time if the corresponding P*x*IE[7:0] (P*x*_IMSK register) is set to 1.
	13-1	T16E: PWM & capture timer configuration Modified Figure 13.1.1
	18-1	UART: UART Configuration Modified Figure 18.1.1
	18-6	UART: Data transfer control Modified Figure 18.5.1
	18-6	UART: Data reception control (Old) (2) RDRY = 1, RD2B = 0 ... Read the receive data buffer contents once to clear the data inside the buffer and reset the RDRY flag. ... (3) RDRY = 1, RD2B = 1 ... The receive data buffer outputs the oldest data first, clearing the buffer data read out and resetting the RD2B flag. ... Even when the receive buffer is full, ... and the new data will overprogram the shift register data. (New) (2) RDRY = 1, RD2B = 0 ... Read the receive data buffer contents once. 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, 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.
	18-7	UART: Data reception control Modified Figure 18.5.2
		UART: Blocking data transfers (Old) Setting the RXEN bit to 0 empties the transmission and receive data buffers, clearing any remaining data. (New) Setting the RXEN bit to 0 empties the transmission data buffer, clearing any remaining data.
	18-8	UART: Overrun error (Old) However, if the receive data buffer is not emptied ... A fourth item of data sent in this state will overprogram the third item of data in the shift register and generate an overrun error. (New) However, if the receive data buffer is not emptied ... sent to the buffer and an overrun error will occur.
	18-14	UART: UART Ch.*x* Status Registers (UART_ST*x*) (Old) 0x4100: UART Status Register (UART_ST) (New) 0x4100/0x4120: UART Ch.*x* Status Registers (UART_ST*x*) Modified the register table
		UART: UART Ch.*x* Status Registers (UART_ST*x*) - (D6) FER: Framing Error Flag - (D5) PER: Parity Error Flag - (D4) OER: Overrun Error Flag (Old) FER/PER/OER is reset by writing as 1 or by setting RXEN (D0/UART_CTL*x* register) to 0. (New) FER/PER/OER is reset by writing 1.
	18-19	UART: UART Ch.*x* Control Registers (UART_CTL*x*) - (D0) RXEN: UART Enable Bit (Old) Preventing transfers by writing 0 to RXEN also clears transfer data buffers. (New) Preventing transfers by writing 0 to RXEN also clears transmit data buffer.
	18-21	UART: Precautions (Old) • The following UART bits should be set with transfers blocked (RXEN = 0). ... - All UART_CTL*x* register (0x4104/0x4124) bits other than RXEN (RBFI, TIEN, RIEN, REIEN) ... (New) • The following UART bits should be set with transfers blocked (RXEN = 0). ... - RBFI bit in the UART_CTL*x* register (0x4104/0x4124) ...

REVISION HISTORY

Code No.	Page	Contents
411581702	18-21	UART: Precautions (Old) Preventing transfer by setting RXEN to 0 clears (initializes) transfer data buffers. Before writing 0 to RXEN, confirm the absence of data in the buffers awaiting transmission or reading. (New) Preventing transfer by setting RXEN to 0 clears (initializes) the transmit data buffer. Before writing 0 to RXEN, confirm the absence of data in the buffer awaiting transmission.
	19-3	SPI: SPI clock (Old) The Master mode SPI uses the internal clock output by the 16-bit timer Ch.1 as the SPI clock. ... Use the MCLK (D9/SPI_CTL register) to select to use the 16-bit timer Ch.1 output clock or PCLK-1/4 clock. (New) The Master mode SPI uses the 16-bit timer Ch.1 output clock or a PCLK-1/4 clock to generate the ... Use MCLK (D9/SPI_CTL register) to select whether the 16-bit timer Ch.1 output clock or PCLK-1/4 clock is used.
	19-3	SPI: SPI clock Modified Figure 19.3.1
		SPI: SPI clock (Old) Since the internal circuit operates in sync with the PCLK clock, the input clock is used to synchronize the differentiated PCLK clock. (New) Deleted
	19-3, 19-14	SPI: SPI clock (Old) Note: The frequency of the clock input via the SPICLK pin must be less than 1/3 of the PCLK and have a clock duty ratio of 50%. (New) Note: The duty ratio of the clock input via the SPICLK pin must be 50%. Deleted Figure 19.3.2
	19-4, 19-14	SPI: Data transfer condition settings (Old) No description (New) Note: When the SPI module is used in master mode with CPHA set to 0, ... (Added Figure 19.4.2) ... transmit data bits and the second and following bytes during continuous transfer.
	19-5	SPI: Data transmission timing chart Deleted Figure 19.5.1
		SPI: Data receipt control (Old) Writing to the SPI TXD register creates the trigger for receipt as well as transmission start. (New) Writing to the SPI_TXD register creates the trigger for receipt as well as transmission start.
	19-6	SPI: Data transmission/receiving timing chart Modified Figure 19.5.1
		SPI: Blocking data transfers (Old) After a data transfer is completed ... the SPRBF flag is 0 before blocking data transfer. Setting the SPEN bit 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 blocking data transfer. The data being transferred cannot be ... if SPEN is set to 0 while data is being sent or received.
	19-7	SPI: Transmit buffer empty interrupt (Old) If SPTBE is 0, the next transmission data can be written ... by the interrupt handler routine. (New) If SPTBE is 1, the next transmission data can be written ... by the interrupt handler routine.
	19-10, 19-14	SPI: SPI Transmit Data Register (SPI_TXD) (Old) No description (New) Note: Make sure that SPEN (D0/SPI_CTL register) is set ... to start data transmission/reception.
	20-1 to 20-18	I^2C (Old) NAK, NACK (New) NAK
	20-2	I^2C: I^2C connection example Added Figure 20.2.1
	20-3	I^2C: I^2C clock (upper limit of transfer rate) (Old) No description (New) When the I^2C module is used to ... 50 kbps in standard mode or 200 kbps in fast mode.
	20-5	I^2C: Slave address transmission (Old) ... In 10-bit mode, data is sent twice under software control. ... (New) ... In 10-bit mode, data is sent twice or three times under software control. ...
	20-6	I^2C: Slave address and transmission data specifying transfer direction Modified Figure 20.5.2
	20-7	I^2C: Data receipt 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.
		I^2C: Data receipt control (Old) This flag is 1 when receiving starts and reverts to 0 when the data is received. It also reverts to 0 for the Wait state. (New) This flag is 1 when receiving starts and reverts to 0 when the data is received.

Code No.	Page	Contents
411581702	20-8	I²C: Data transfer end (Stop condition generation) (Old) Stop condition generation can be reserved. To reserve the stop condition, check that I²C is ... The stop condition is generated as soon as 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 ... has finished and the time for the slave device to finish clock stretching has elapsed.
		I²C: Wait state for TXE, RXE, STRT, and STP settings (Old) Wait state for TXE, RXE, STRT, and STP settings The module will switch to Wait state with the SCL output fixed at Low if all of the TXE (D9/ ... by writing 1 to TXE or RXE to restart data transfer or by generating the stop condition with STP. (New) Deleted
		I²C: Prohibiting data transfer (Old) After data transfer is complete (both transmission and receipt), program 0 to the I2CEN bit ... Data being transferred cannot be guaranteed if I2CEN is set to 0 during the transfer. (New) After the stop condition has been generated, write 0 to I2CEN to disable data transfers. ... the SCL and SDA output levels and transfer data at that point cannot be guaranteed.
	20-9, 20-10	I²C: Timing chart Modified Figures 20.5.6 to 20.5.9
	20-11	I²C: Transmit buffer empty interrupt (Old) An interrupt occurs if other interrupt conditions are satisfied. (New) 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 ... cause of interrupt without sending the data written. ∗ TXE: Transmit Execution Bit in the I²C Data (I2C_DAT) Register (D9/0x4344)
		I²C: Receive buffer full interrupt (Old) An interrupt occurs if other interrupt conditions are met. (New) 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 ... from RTDT[7:0]. Note: After an I²C interrupt occurs, ... Note that it cannot be checked using a register.
	20-14	I²C: I²C Control Register (I2C_CTL) - (D1) STP: Stop Control Bit (Old) Note that the stop condition will be generated only if STP is 1 and TXE (D9/I2C_DAT register), ... STP is disabled if any of TXE, RXE, or STRT is 1. (New) Before STP can be set to 1, confirm that TBUSY or RBUSY is reset to 0 from 1 (this indicates ... has finished and the time for the slave device to finish clock stretching has elapsed.
	20-16	I²C: I²C Data Register (I2C_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.
	22-9, 22-14, 22-21	LCD: Display on/off (Old) No description (New) Note: The LCD power supply may not generate the drive voltage normally if DSPC[1:0] is set to ... 3. The LCD clock supply is enabled and the clock is fed into the LCD driver circuit.
		LCD: Display on/off (Old) DSPC[1:0] is reset to 0x0 (Display off) after initial resetting or when the slp instruction is executed. (New) DSPC[1:0] is reset to 0x0 (Display off) after initial resetting. Note: Executing the slp instruction ... via software before executing the slp instruction.
	23-2, 23-7, 27-3, AP-22	SVD: Comparison voltage setting (Old) Comparison voltage: 16 voltages, SVDC[3:0] = 0x0: 1.7 V (New) Comparison voltage: 15 voltages, SVDC[3:0] = 0x0: reserved
	27-7	External clock input AC characteristics Modified the characteristic table
	27-11, 27-12, 27-13	LCD drive voltage temperature characteristics, SVD voltage temperature characteristics, SVD voltage temperature characteristics, Execution current consumption temperature characteristics (OSC1 operation) Modified the graphs
	28-3	VFBGA10H-180 package Added the figure
	AP-28	I/O register list: IRAM Size Select Register (MISC_IRAMSZ) Modified the table
	AP-35	Power saving: Clock system Modified Table C.1.1
	AP-36	Power saving: Peripheral clock (PCLK) (Old) Peripheral circuits operating with PCLK ... • 16-bit timer Ch.0 to Ch.2 • Interrupt controller • SPI ... (New) Peripheral circuits operating with PCLK ... • 16-bit timer Ch.0 to Ch.2 • SPI ...

EPSON

International Sales Operations

AMERICA

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

EUROPE

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

ASIA

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

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

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

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

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

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

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

SEIKO EPSON CORP.
MICRODEVICES OPERATIONS DIVISION

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

Document Code: 411581702
First Issue October 2008 Ⓓ
Revised April 2011 in JAPAN Ⓛ