MF564-08

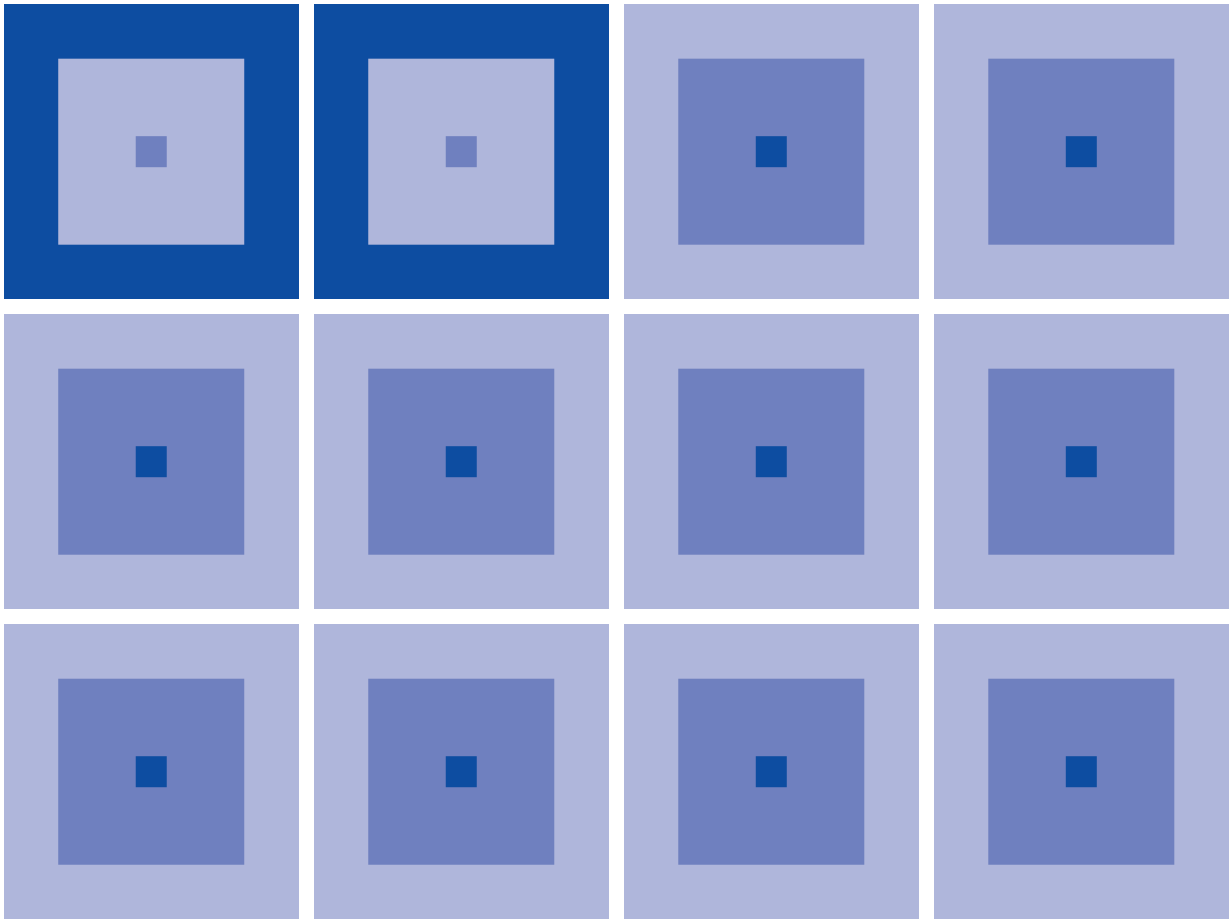
EPSON

CMOS 4-BIT SINGLE CHIP MICROCOMPUTER

S1C621A0

Technical Manual

S1C621A0 Technical Hardware/S1C621A0 Technical Software



SEIKO EPSON CORPORATION

ENERGY SAVING EPSON

PREFACE

This manual is individualy described about the hardware and the software of the S1C621A0.

I. S1C621A0 Technical Hardware

This part explains the function of the S1C621A0, the circuit configurations, and details the controlling method.

II. S1C621A0 Technical Software

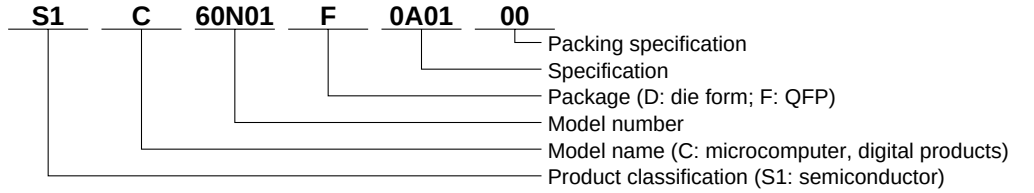
This part explains the programming method of the S1C621A0.

The information of the product number change

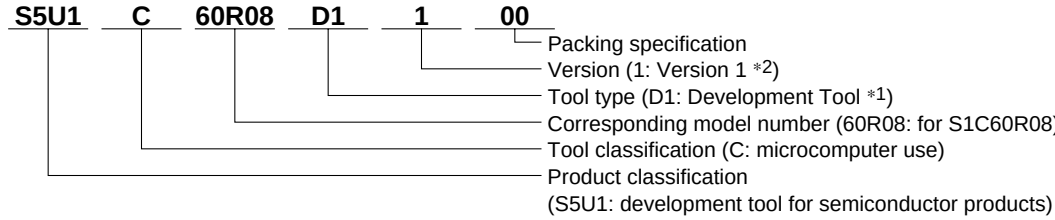
Starting April 1, 2001, the product number will be changed as listed below. To order from April 1, 2001 please use the new product number. For further information, please contact Epson sales representative.

Configuration of product number

Devices



Development tools



*1: For details about tool types, see the tables below. (In some manuals, tool types are represented by one digit.)
*2: Actual versions are not written in the manuals.

Comparison table between new and previous number

S1C60 Family processors

Previous No.	New No.
E0C6001	S1C60N01
E0C6002	S1C60N02
E0C6003	S1C60N03
E0C6004	S1C60N04
E0C6005	S1C60N05
E0C6006	S1C60N06
E0C6007	S1C60N07
E0C6008	S1C60N08
E0C6009	S1C60N09
E0C6011	S1C60N11
E0C6013	S1C60N13
E0C6014	S1C60140
E0C60R08	S1C60R08

S1C62 Family processors

Previous No.	New No.
E0C621A	S1C621A0
E0C6215	S1C62150
E0C621C	S1C621C0
E0C6S27	S1C6S2N7
E0C6S37	S1C6S3N7
E0C623A	S1C6N3A0
E0C623E	S1C6N3E0
E0C6S32	S1C6S3N2
E0C6233	S1C62N33
E0C6235	S1C62N35
E0C623B	S1C6N3B0
E0C6244	S1C62440
E0C624A	S1C624A0
E0C6S46	S1C6S460

Previous No.	New No.
E0C6247	S1C62470
E0C6248	S1C62480
E0C6S48	S1C6S480
E0C624C	S1C624C0
E0C6251	S1C62N51
E0C6256	S1C62560
E0C6292	S1C62920
E0C6262	S1C62N62
E0C6266	S1C62660
E0C6274	S1C62740
E0C6281	S1C62N81
E0C6282	S1C62N82
E0C62M2	S1C62M20
E0C62T3	S1C62T30

Comparison table between new and previous number of development tools

Development tools for the S1C60/62 Family

Previous No.	New No.
ASM62	S5U1C62000A
DEV6001	S5U1C60N01D
DEV6002	S5U1C60N02D
DEV6003	S5U1C60N03D
DEV6004	S5U1C60N04D
DEV6005	S5U1C60N05D
DEV6006	S5U1C60N06D
DEV6007	S5U1C60N07D
DEV6008	S5U1C60N08D
DEV6009	S5U1C60N09D
DEV6011	S5U1C60N11D
DEV60R08	S5U1C60R08D
DEV621A	S5U1C621A0D
DEV621C	S5U1C621C0D
DEV623B	S5U1C623B0D
DEV6244	S5U1C62440D
DEV624A	S5U1C624A0D
DEV624C	S5U1C624C0D
DEV6248	S5U1C62480D
DEV6247	S5U1C62470D

Previous No.	New No.
DEV6262	S5U1C62620D
DEV6266	S5U1C62660D
DEV6274	S5U1C62740D
DEV6292	S5U1C62920D
DEV62M2	S5U1C62M20D
DEV6233	S5U1C62N33D
DEV6235	S5U1C62N35D
DEV6251	S5U1C62N51D
DEV6256	S5U1C62560D
DEV6281	S5U1C62N81D
DEV6282	S5U1C62N82D
DEV6S27	S5U1C6S2N7D
DEV6S32	S5U1C6S3N2D
DEV6S37	S5U1C6S3N7D
EVA6008	S5U1C60N08E
EVA6011	S5U1C60N11E
EVA621AR	S5U1C621A0E2
EVA621C	S5U1C621C0E
EVA6237	S5U1C62N37E
EVA623A	S5U1C623A0E

Previous No.	New No.
EVA623B	S5U1C623B0E
EVA623E	S5U1C623E0E
EVA6247	S5U1C62470E
EVA6248	S5U1C62480E
EVA6251R	S5U1C62N51E1
EVA6256	S5U1C62N56E
EVA6262	S5U1C62620E
EVA6266	S5U1C62660E
EVA6274	S5U1C62740E
EVA6281	S5U1C62N81E
EVA6282	S5U1C62N82E
EVA62M1	S5U1C62M10E
EVA62T3	S5U1C62T30E
EVA6S27	S5U1C6S2N7E
EVA6S32R	S5U1C6S3N2E2
ICE62R	S5U1C62000H
KIT6003	S5U1C60N03K
KIT6004	S5U1C60N04K
KIT6007	S5U1C60N07K

I. S1C621A0 Technical Hardware

CONTENTS

CHAPTER 1 DESCRIPTION

The S1C621A0 is a single-chip microcomputer which uses an S1C6200A CMOS 4-bit CPU as the core. It contains a 4,096 (words) × 12 (bits) ROM, 208 (words) × 4 (bits) RAM, LCD driver circuit, remote-control carrier output circuit, time base counter, analog comparator, and watchdog timer.

The S1C621A0 offers a superb solution to infrared remote controller and other applications requiring low power consumption.

1.1 Features

- ROM size: 4,096 words × 12 bits
- RAM size: 208 words × 4 bits
- Clock: 32.768 kHz or 455 kHz (selected by software)
- Instruction execution time:
  - 32 kHz operation: 153, 214 or 366 µsec (depending on instructions)
  - 455 kHz operation: 11, 15 or 26 µsec (depending on instructions)
- Instruction set: 100 instructions
- Input ports: 8 ports (with or without pull-up resistor)
- Output ports: 9 ports (clock output and buzzer output possible by using mask option)
- I/O ports: 4 ports
- Infrared remote-control output: 1 output
- LCD drivers: 32 segment × 3 common outputs
  32 segment × 4 common outputs
  (Duty 1/3 or 1/4 selected by using mask option)

- Crystal oscillation circuit: 32.768 kHz
- Ceramic oscillation circuit: 455 kHz (Alternative selections of either not using this circuit or using CR oscillation circuit instead is possible by using mask option)
- Interrupts:
  - External: 2 input interrupts
  - Internal: 3 timer interrupts (32 Hz, 8 Hz or 2 Hz)
    1 remote control output control interrupt
- SVD built-in: Detection voltage 2.3 V
- Power supply: 3 V (2.2 V to 3.5 V)
- Current consumption (Typ.):
  - 32 kHz operation: 2 µA in halt mode
    9 µA in full run mode
  - 455 kHz operation: 130 µA
- Supply form: QFP5-80pin, QFP14-80pin or chip

1.2 Block Diagram

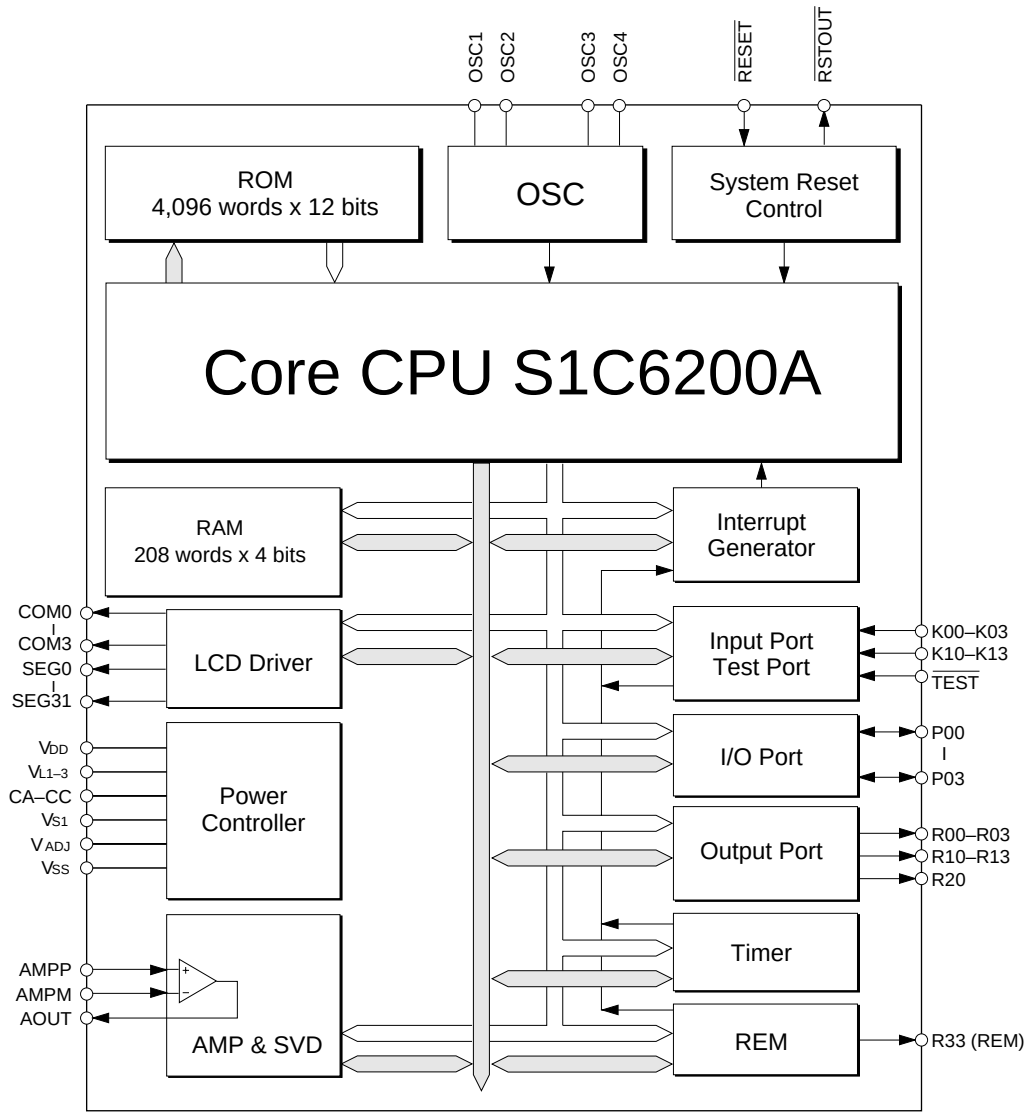


Fig. 1.2.1
S1C621A0 Block Diagram

1.3 Pin Configuration

QFP5-80pin
(plastic)

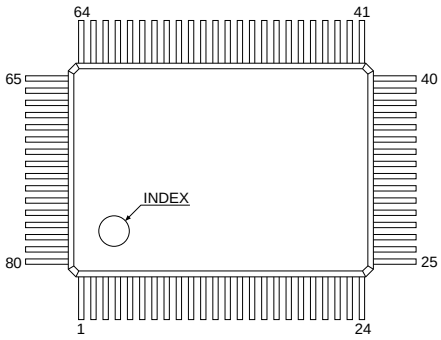


Fig. 1.3.1
S1C621A0 Pin Configuration
(QFP5-80pin)

Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name
1	SEG16	21	COM0	41	OSC4	61	K11
2	SEG15	22	CC	42	V_{S1}	62	K12
3	SEG14	23	CB	43	R00	63	K13
4	SEG13	24	CA	44	R01	64	$\overline{TEST}$
5	SEG12	25	V_{L3}	45	R02	65	N.C.
6	SEG11	26	V_{L2}	46	R03	66	SEG31
7	SEG10	27	$\overline{RESET}$	47	R10	67	SEG30
8	SEG9	28	V_{ADJ}	48	R11	68	SEG29
9	SEG8	29	V_{L1}	49	R12	69	SEG28
10	SEG7	30	R33(REM)	50	R13	70	SEG27
11	SEG6	31	$\overline{RSTOUT}$	51	R20	71	SEG26
12	SEG5	32	AMPP	52	P00	72	SEG25
13	SEG4	33	AMPM	53	P01	73	SEG24
14	SEG3	34	AOUT	54	P02	74	SEG23
15	SEG2	35	V_{DD}	55	P03	75	SEG22
16	SEG1	36	N.C.	56	K00	76	SEG21
17	SEG0	37	OSC1	57	K01	77	SEG20
18	COM3	38	OSC2	58	K02	78	SEG19
19	COM2	39	V_{SS}	59	K03	79	SEG18
20	COM1	40	OSC3	60	K10	80	SEG17

N.C. = No Connection

QFP14-80pin
(plastic)

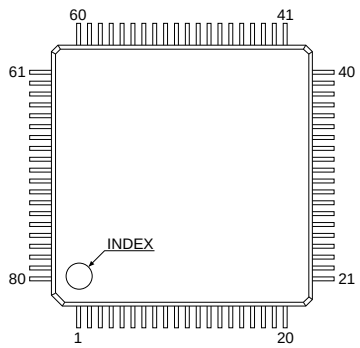


Fig. 1.3.2
S1C621A0 Pin Configuration
(QFP14-80pin)

Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name
1	CB	21	R01	41	$\overline{TEST}$	61	SEG14
2	CA	22	R02	42	N.C.	62	SEG13
3	V_{L3}	23	R03	43	SEG31	63	SEG12
4	V_{L2}	24	R10	44	SEG30	64	SEG11
5	$\overline{RESET}$	25	R11	45	SEG29	65	SEG10
6	V_{ADJ}	26	R12	46	SEG28	66	SEG9
7	V_{L1}	27	R13	47	SEG27	67	SEG8
8	R33(REM)	28	R20	48	SEG26	68	SEG7
9	$\overline{RSTOUT}$	29	P00	49	SEG25	69	SEG6
10	AMPP	30	P01	50	SEG24	70	SEG5
11	AMPM	31	P02	51	SEG23	71	SEG4
12	AOUT	32	P03	52	SEG22	72	SEG3
13	V_{DD}	33	K00	53	SEG21	73	SEG2
14	OSC1	34	K01	54	SEG20	74	SEG1
15	OSC2	35	K02	55	SEG19	75	SEG0
16	V_{SS}	36	K03	56	SEG18	76	COM3
17	OSC3	37	K10	57	SEG17	77	COM2
18	OSC4	38	K11	58	SEG16	78	COM1
19	V_{S1}	39	K12	59	N.C.	79	COM0
20	R00	40	K13	60	SEG15	80	CC

N.C. = No Connection

1.4 Pin Description

Table 1.4.1 Pin Description

Pin Name	Pin No.		In / Out	Function
	QFP5-80	QFP14-80		
V_{DD}	35	13	I	Power supply (+)
V_{SS}	39	16	I	Power supply (-)
V_{S1}	42	19	–	Oscillation circuit power supply (about V_{DD} -2.0 V)
V_{L1}	29	7	–	LCD power supply (V_{DD} -V_L)
V_{L2}	26	4	–	LCD booster supply (V_{DD} -2V_L)
V_{L3}	25	3	–	LCD booster supply (V_{DD} -3V_L)
CA–CC	22–24	1, 2, 80	–	LCD booster capacitor connection pin
V_{ADJ}	28	6	I	V_L input adjustment pin
OSC1	37	14	I	Crystal oscillation input pin
OSC2	38	15	O	Crystal oscillation output pin (include drain capacitor)
OSC3	40	17	I	Ceramic or CR oscillation input pin (selected by using mask option)
OSC4	41	18	O	Ceramic or CR oscillation output pin (selected by using mask option)
COM0–3	18–21	76–79	O	LCD common outputs (duty 1/3 or 1/4 selected possible by using mask option)
SEG0–31	1–17 66–80	43–58 60–75	O	LCD segment outputs (DC output possible by using mask option)
AMPP	32	10	I	Analog comparator non-inverted input pin
AMPM	33	11	I	Analog comparator inverted input pin
AOUT	34	12	O	Analog comparator output pin
$\overline{TEST}$	64	41	I	Test input pin
$\overline{RESET}$	27	5	I	Initial reset input pin
K00–03	56–59	33–36	I	Input pins
K10–13	60–63	37–39	I	Input pins
P00–03	52–55	29–32	I/O	I/O pins
R00–03	43–46	20–23	O	Output pins (R12 : DC, FOUT or $\overline{BZ}$ selected by using mask option) (R13 : DC or BZ selected by using mask option)
R10–13	47–50	24–27		
R20	51	28		
R33 (REM)	30	8	O	Remote control carrier output pin
$\overline{RSTOUT}$	31	9	O	Initial reset output pin

CHAPTER 2 CPU, ROM AND RAM

2.1 CPU

The S1C621A0 uses an S1C6200A 4-bit CPU as the core and therefore has nearly the same register composition and instruction set as other family processors using the S1C6200A core CPU. (For details of the S1C6200A, see the "S1C6200/6200A Core CPU Manual".)

Due to features such as the ROM/RAM size and two clock operating modes, the S1C621A0 differs from the S1C6200A in the following points:

(1) The index registers IX and IY each consist of 8 bits as the RAM can be addressed on a 8-bit basis. (The register has no 4-bit page part.) Therefore, the instructions relating to XP and YP ("PUSH XP", "PUSH YP", "LD XP,r", "LD YP,r", "LD r,XP" and "LD r,YP") cannot be used with the S1C621A0.

(2) The S1C621A0 does not support the sleep mode. The SLP instruction is therefore unavailable with the S1C621A0.

(3) Since the ROM had a size of 4,096 words, no bank bit is required. The S1C621A0 thus uses no PCB or NBP registers.

2.2 ROM

The S1C621A0 has a mask ROM for storage of programs. Its size is 4,096 words × 12 bits.

The program area consists of 16 pages, each having 256 steps. After initial reset, the program starting address is page 1, step 0. The interrupt vector addresses range from steps 01H to 0FH of page 1.

2.3 RAM

This RAM stores various data. It is also used as a stack area from which subroutines are called or into which registers are saved. As the size of the RAM is 208 words × 4 bits, the stack area must be set at addresses up to CFH. (For the RAM map, see section 3.1 "Memory Map".)

In programming the RAM, it is necessary that the data area be separated from the stack area. (A stack area of 3 words is needed for subroutine call, interrupt handling, etc.)

The RAM area at addresses from 00H to 0FH accommodates memory registers which are addressed by the register pointer RP.

CHAPTER 3 PERIPHERAL CIRCUITS

All peripheral circuits (timer, I/O, etc.) of the S1C621A0 are configured in memory-mapped I/O form, and can be controlled by using memory operation instructions. This section describes the operation of each peripheral circuit in detail.

3.1 Memory Map

The data memory (RAM) of the S1C621A0 has an address space of 256 words, of which the high-order 48 words are allocated to the peripheral circuits.

Figure 3.1.1 shows the S1C621A0 memory map. Tables 3.1.1(a) to 3.1.1(d) gives details of the I/O memory map (addresses from 0F0H to 0FFH).

Address Page	Low / High	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	M0	M1	M2	M3	M4	M5	M6	M7	M8	M9	MA	MB	MC	MD	ME	MF
	1	RAM (208 words x 4 bits) R/W															
	2																
	3																
	4																
	5																
	6																
	7																
	8																
	9																
	A																
	B																
	C																
	D	Display memory (32 words x 4 bits) W															
	E																
	F	I/O memory [See Tables 3.1.1(a)–3.1.1(d)]															

Fig. 3.1.1 Memory Map

Note *Memory is not mounted in unused area within the memory map and in memory area not indicated in this chapter. For this reason, normal operation cannot be assured for programs that have been prepared with access to these areas.*

Table 3.1.1(a) I/O memory map (0F0H–0F3H)

Address	Register								Comment
	D3	D2	D1	D0	Name	Init	1	0	
0F0H	REMSO	IREM	IK1	IK0	REMSO	0	On	Off	Forced REM output (on/off)
	R/W	R			IREM	x	Yes	No	Interrupt factor flag (REM) Clear to 0 after read
					IK1	0	Yes	No	Interrupt factor flag (K10–K13) Clear to 0 after read
					IK0	0	Yes	No	Interrupt factor flag (K00–K03) Clear to 0 after read
0F1H	WDRST	TI2	TI8	TI32	WDRST	Clear	Clear	–	Watchdog timer reset
	W	R			TI2	0	Yes	No	Interrupt factor flag (Timer 2 Hz) Clear to 0 after read
					TI8	0	Yes	No	Interrupt factor flag (Timer 8 Hz) Clear to 0 after read
					TI32	0	Yes	No	Interrupt factor flag (Timer 32 Hz) Clear to 0 after read
0F2H	REMC	EIREM	EIK1	EIK0	REMC	1	On	Off	REM carrier generation on/off
	R/W				EIREM	0	Enable	Mask	Interrupt mask register (REM)
					EIK1	0	Enable	Mask	Interrupt mask register (K10–K13)
					EIK0	0	Enable	Mask	Interrupt mask register (K00–K03)
0F3H	TMRUN	ETI2	ETI8	ETI32	TMRUN	0	Run	Clear & Stop	Timer Run/Clear & Stop
	R/W				ETI2	0	Enable	Mask	Interrupt mask register (Timer 2 Hz)
					ETI8	0	Enable	Mask	Interrupt mask register (Timer 8 Hz)
					ETI32	0	Enable	Mask	Interrupt mask register (Timer 32 Hz)

Note: Init = value at initial reset, x = undefined, – = not set on the circuit

Table 3.1.1(b) I/O memory map (0F4H–0F7H)

Address	Register								Comment
	D3	D2	D1	D0	Name	Init	1	0	
0F4H	TM03	TM02	TM01	TM00	TM03	0			Timer data (low-order) 16 Hz
	R				TM02	0			Timer data (low-order) 32 Hz
					TM01	0			Timer data (low-order) 64 Hz
					TM00	0			Timer data (low-order) 128 Hz
0F5H	TM13	TM12	TM11	TM10	TM13	0			Timer data (High-order) 1 Hz
	R				TM12	0			Timer data (High-order) 2 Hz
					TM11	0			Timer data (High-order) 4 Hz
					TM10	0			Timer data (High-order) 8 Hz
0F6H	SVDDT	SVDON	CLKCHG	OSCC	SVDDT	–	Normal	Low voltage	SVD data (1 when SVDON = 0)
	R	R/W			SVDON	0	On	Off	SVD circuit on/off
					CLKCHG	0	OSC1	OSC3	CPU CLK change OSC1/OSC3
					OSCC	1	On	Off	OSC3 oscillation on/off
0F7H	RCDIV	RCDUTY	RT1	RT0	RCDIV	x			REM carrier cycle set
	R/W				RCDUTY	x			REM carrier duty set
					RT1	x			REM τ-cycle set
					RT0	x			REM τ-cycle set

Note: Init = value at initial reset, x = undefined, – = not set on the circuit

Table 3.1.1(c) I/O memory map (0F8H–0F9H, 0FAH–0FBH)

Address	Register								Comment
	D3	D2	D1	D0	Name	Init	1	0	
0F8H	RIC3	RIC2	RIC1	RIC0	RIC3	x			REM interrupt counter set (τ clock) (all :1 after count completed)
	W				RIC2	x			
					RIC1	x			
					RIC0	x			
0F9H	ROUT1	ROUT0	MF91	MF90	ROUT1	0			REM output-on time set (0 τ – 3 τ)
	R/W				ROUT0	0			REM output-on time set (0 τ – 3 τ)
					MF91	x			General-purpose register (bit)
					MF90	x			General-purpose register (bit)
0FAH	K03	K02	K01	K00	K03	–	High	Low	Input port (fall : interrupt factor)
	R				K02	–	High	Low	Input port (fall : interrupt factor)
					K01	–	High	Low	Input port (fall : interrupt factor)
					K00	–	High	Low	Input port (fall : interrupt factor)
0FBH	K13	K12	K11	K10	K13	–	High	Low	Input port (fall : interrupt factor)
	R				K12	–	High	Low	Input port (fall : interrupt factor)
					K11	–	High	Low	Input port (fall : interrupt factor)
					K10	–	High	Low	Input port (fall : interrupt factor)

Note: Init = value at initial reset, x = undefined, – = not set on the circuit

Table 3.1.1(d) I/O memory map (0FCH–0FFH)

Address	Register								Comment
	D3	D2	D1	D0	Name	Init	1	0	
0FCH	R03	R02	R01	R00	R03	1	High	Low	Output port
	R/W				R02	1	High	Low	Output port
					R01	1	High	Low	Output port
					R00	1	High	Low	Output port
0FDH	R13	R12	R11	R10	R13	1	High	Low	Output port
	R/W				R12	1	High	Low	Output port
					R11	1	High	Low	Output port
					R10	1	High	Low	Output port
0FEH	P03	P02	P01	P00	P03	1	High	Low	I/O port (used as input port after initial reset)
	R/W				P02	1	High	Low	I/O port (used as input port after initial reset)
					P01	1	High	Low	I/O port (used as input port after initial reset)
					P00	1	High	Low	I/O port (used as input port after initial reset)
0FFH	OPDT	OPON	IOC	R20	OPDT	–	High	Low	Analog comparator data (0 or 1)
	R	R/W			OPON	0	On	Off	Analog comparator on/off
					IOC	0	Out	In	I/O port control out/in
					R20	0	High	Low	Output port

Note: Init = value at initial reset, x = undefined, – = not set on the circuit

3.2 Power Supply

The S1C621A0 is given 3 V by an single external power supply and generates voltages needed to operate its LCD driver and oscillation circuit.

A voltage reduction can be detected by the on-chip supply voltage detection circuit. (See section 3.10 "AMP and SVD".)

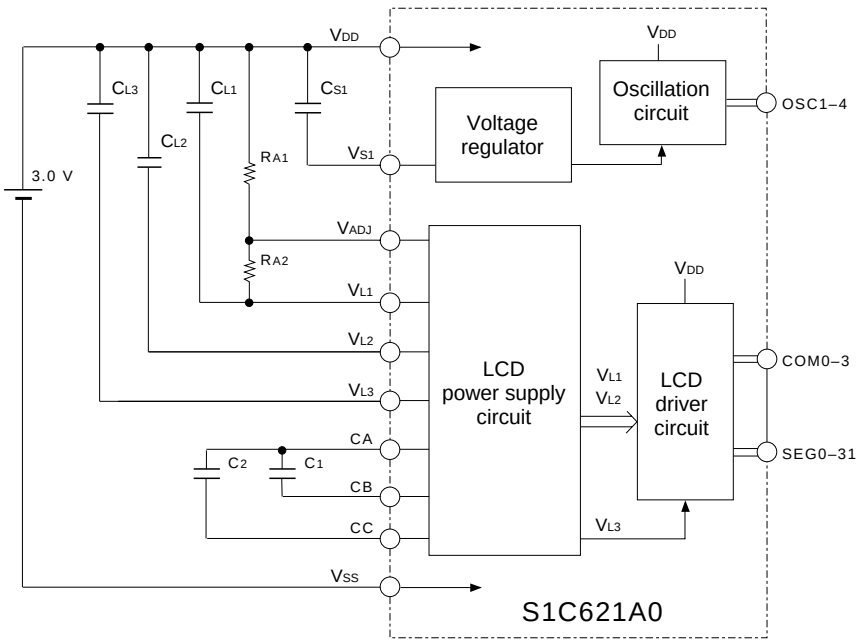


Fig. 3.2.1
Power Supply Block Diagram

Oscillation circuit power supply (V_{S1})

The S1C621A0 uses its on-chip voltage regulator to generate voltage level V_{S1} for stabilization of the oscillation circuit.

LCD power supply

The on-chip voltage regulator and booster are used to generate voltage levels (V_{DD}, V_{L1}, V_{L2} and V_{L3}) needed to drive the LCDs.

V_{L2} is obtained by boosting V_{L1} two times and V_{L3} is obtained by boosting V_{L1} three times. V_{L1} can be adjusted by feeding it back to the V_{ADJ} pin through R_{A1} and R_{A2} as shown in Figure 3.2.2. V_L ($\approx V_{DD} - V_{L1}$) is defined by following equation:

$$V_L \approx 1 \times (R_{A1} + R_{A2}) / R_{A1}$$

Example:

V_L	R_{A1}	R_{A2}
≈ 1 V	∞	0 Ω
≈ 1.5 V	2 MΩ	1 MΩ

An LCD driving voltage suited to each LCD panel can be obtained by adjusting V_L at the V_{ADJ} pin.

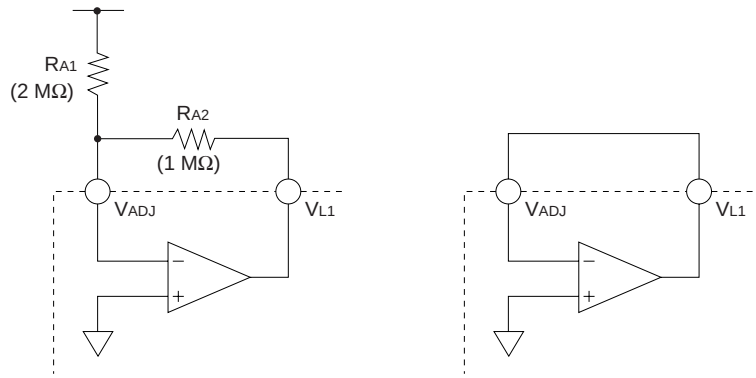


Fig. 3.2.2
V_L Adjust Circuit

3.3 Oscillation Circuit

The S1C621A0 contains an OSC1 (32.768 kHz) crystal oscillation circuit and an OSC3 (455 kHz) ceramic oscillation circuit*. Either the OSC1 or OSC3 can be selected as the CPU operation clock by software. Figures 3.3.1 and 3.3.2 show block diagrams of the oscillation circuit.

* By using the mask option it is possible to connect a CR oscillation circuit in place of the ceramic oscillation circuit, or it is possible to forgo the use of the OSC3 circuit.

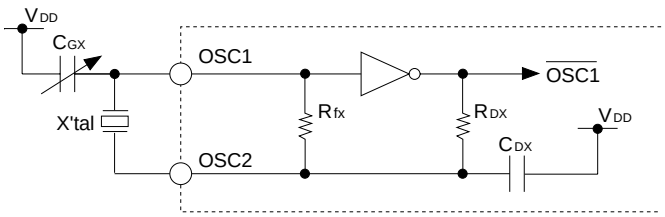
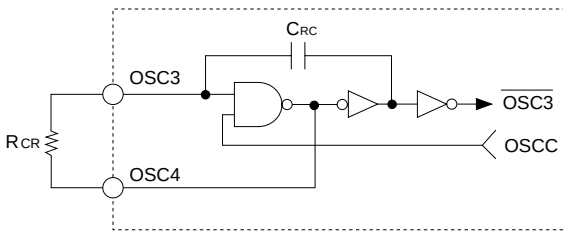


Fig. 3.3.1
OSC1 Oscillation Circuit
(Crystal)



a) OSC3 – CR oscillation circuit

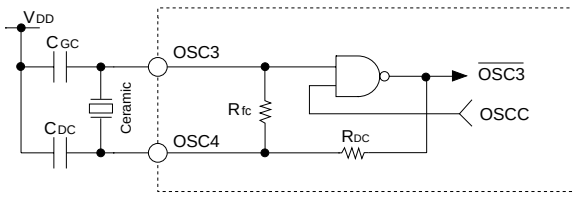
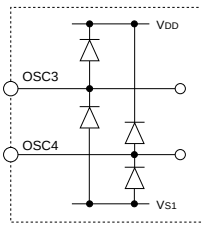


Fig. 3.3.2
OSC3 Oscillation
Circuit

b) OSC3 – Ceramic oscillation circuit



c) OSC3 – "Not Use" is selected by mask option

Oscillation circuit control by software

As shown in Table 3.3.1, the oscillation circuit can be controlled by software using D1 and D0 at I/O memory address 0F6H.

Table 3.3.1 I/O Memory Map

Address	Register								Comment
	D3	D2	D1	D0	Name	Init	1	0	
0F6H	SVDDT	SVDON	CLKCHG	OSCC	SVDDT	–	Normal	Low voltage	SVD data (1 when SVDON = 0)
	R	R/W			SVDON	0	On	Off	SVD circuit on/off
					CLKCHG	0	OSC1	OSC3	CPU CLK change OSC1/OSC3
					OSCC	1	On	Off	OSC3 oscillation on/off

Note: Init = value at initial reset, x = undefined, – = not set on the circuit

The following two bits relate to the oscillation circuit control by software.

• **OSCC:** OSC3 oscillation circuit drive control

This bit controls OSC3 oscillation ON/OFF condition. OSC3 oscillation is ON when OSCC = 1 and OFF when OSCC = 0.

At initial reset, the OSC3 circuit is turned ON. The circuit must be turned ON when it is necessary to obtain a remote control carrier signal output or a faster CPU speed. In all other cases, the circuit must stay OFF to save power.

• **CLKCHG:** CPU clock frequency change

This bit selects either a CPU operation clock, OSC3 clock or OSC1 clock. The OSC3 clock is selected with CLKCHG = 0 and the OSC1 clock with CLKCHG = 1. At initial reset, the bit is 0 (OSC3 selected).

Table 3.3.2 lists the relation between the CPU clock frequency and instruction execution time.

Table 3.3.2
CPU Clock Frequency and Instruction Execution Time

CPU clock frequency	Instruction execution time		
	5-clock instruction	7-clock instruction	12-clock instruction
32.768 kHz	152.6 µs	213.6 µs	366.2 µs
455 kHz	11.0 µs	15.4 µs	26.4 µs

Programming notes

(1) The OSC3 clock must not be selected unless its oscillation is stable.

(2) It takes at least 5 ms for the OSC3 clock (if generated by the 455 kHz ceramic oscillation circuit) to become stable after it is switched from OFF to ON. When changing the CPU clock from OSC3 to OSC1, wait at least 5 ms after OSC3 oscillation turns ON.

(3) When changing the CPU clock from OSC3 to OSC1 (32 kHz) at power-on, wait at least 3 seconds after the initial reset is cleared.

(4) The time until OSC3 or OSC1 oscillation becomes stable depends on the characteristics, operating conditions and other factors of the externally connected oscillator. Before using an oscillator, check its characteristics and set a proper waiting time with sufficient margin.

3.4 Input Ports (K00–K03, K10–K13)

The S1C621A0 has two 4-bit general-purpose input ports (K00–K03 and K10–K13). As shown in Figure 3.4.1, each input port pin is provided with a pull-up and a feedback pull-up so that the port is suitable for push switch or key matrix switch input. As the pull-up can be removed by using mask option, the input port can also be used for slide switch input or interface with another LSI.

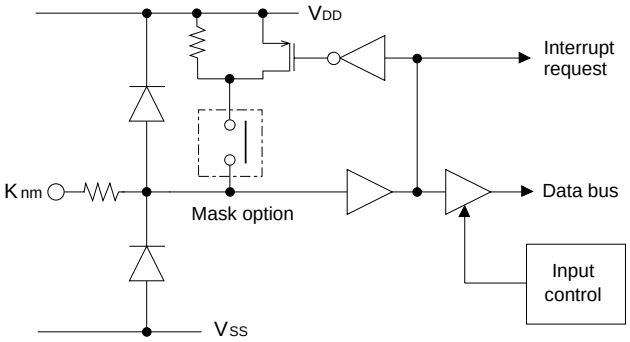


Fig. 3.4.1
Input Port Configuration

Input port data

Input port data can be read on a 4-bit basis (K00–K03, K10–K13) addressed 0FAH and 0FBH. Table 3.4.1 shows data assignments to the input port bits.

Table 3.4.1 I/O Memory Map

Address	Register								Comment
	D3	D2	D1	D0	Name	Init	1	0	
0FAH	K03	K02	K01	K00	K03	–	High	Low	Input port (fall : interrupt factor)
	R				K02	–	High	Low	Input port (fall : interrupt factor)
					K01	–	High	Low	Input port (fall : interrupt factor)
					K00	–	High	Low	Input port (fall : interrupt factor)
0FBH	K13	K12	K11	K10	K13	–	High	Low	Input port (fall : interrupt factor)
	R				K12	–	High	Low	Input port (fall : interrupt factor)
					K11	–	High	Low	Input port (fall : interrupt factor)
					K10	–	High	Low	Input port (fall : interrupt factor)

Note: Init = value at initial reset, – = not set on the circuit

Input interrupt

An interrupt request to the CPU can be invoked by the falling edge of an input port pin. For details, see section 3.11 "Interrupts and Halt".

Programming note

If the Knm pin is used as an input with an on-chip pull-up resistor, the time constant of the input line capacitive load and on-chip pull-up resistor causes the logic level to change with a certain delay from the start of an input signal change.

For example, consider a key matrix combined with some N-channel open drain outputs where turning a switch ON causes the key scan program to start. Input data is read in such a way that all the N-channel open drain outputs, except one, are turned off to key-scan the input signal now at low level as a result of switch-on, and the switch between that output and the others is checked to determine has turned ON. Even with switch-off, however, reading data without a delay results in an input error (the input is at low level). To prevent this, data must be programmed to assure that it is read with a delay, or that all N-channel open drain outputs are turned OFF, followed by reading the input repeatedly to confirm that it returns to high level and turns one output ON, then read input data.
Please consider delay of inputs on the program.

3.5 Output Ports (R00–R03, R10–R13, R20)

The S1C621A0 has two 4-bit and one 1-bit general-purpose output ports. As shown in Figure 3.5.1, the output may take either complementary or N-channel open drain form (selected by using mask option). Even with an open drain output, however, the Rnm pin cannot be pulled up to above VDD level.

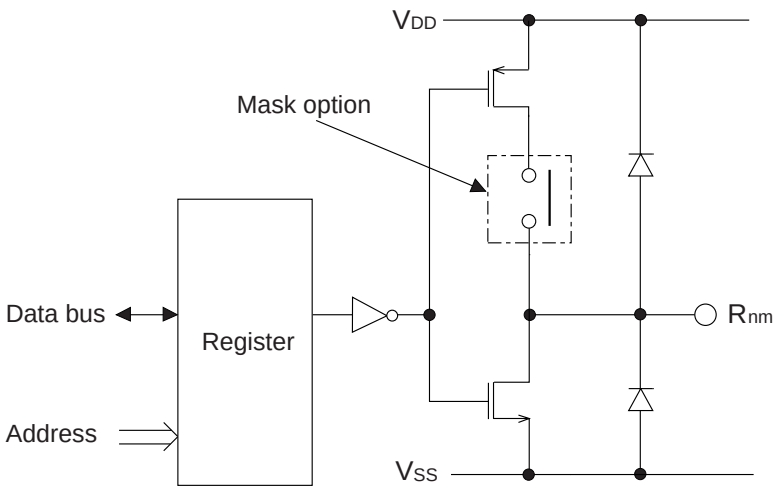


Fig. 3.5.1
Output Port Configuration

Output port data

The output port is a read/write register as shown in Figure 3.5.1. The data in the register is output by the output port pin. Table 3.5.1 shows data assignment to the output port bits on the memory map.

At initial reset, R00–R03 and R10–R13 are set to 1 and R20 to 0, is shown in the Table 3.5.1.

Table 3.5.1 I/O Memory Map

Address	Register								Comment
	D3	D2	D1	D0	Name	Init	1	0	
0FCH	R03	R02	R01	R00	R03	1	High	Low	Output port
	R/W				R02	1	High	Low	Output port
					R01	1	High	Low	Output port
					R00	1	High	Low	Output port
0FDH	R13	R12	R11	R10	R13	1	High	Low	Output port
	R/W				R12	1	High	Low	Output port
					R11	1	High	Low	Output port
					R10	1	High	Low	Output port
0FFH	OPDT	OPON	IOC	R20	OPDT	–	High	Low	Analog comparator data (0 or 1)
	R	R/W			OPON	0	On	Off	Analog comparator on/off
					IOC	0	Out	In	I/O port control out/in
					R20	0	High	Low	Output port

Note: Init = value at initial reset, – = not set on the circuit

R12 and R13 (FOUT, BZ, $\overline{BZ}$)

Among the output ports, R12 and R13 each have mask options (normally, DC output) as listed in Table 3.5.2.

Table 3.5.2
R12 and R13 Mask Options

Pin	Mask options
R12	DC, FOUT, $\overline{BZ}$ ($f_{OSC1}/8$, $\overline{f_{OSC1}/16}$)
R13	DC, BZ ($f_{OSC1}/8$, $f_{OSC1}/16$)

FOUT (frequency output divided OSC1 clock) becomes valid when 0 is written into the register, and appears a high-level DC output when 1 is written into the register (see Figure 3.5.2 at R12 pin).

BZ (buzzer) and $\overline{BZ}$ become valid when 0 is written into the register, and appear a low-level DC output when 1 is written into the register (see Figure 3.5.2 at R-port pins).

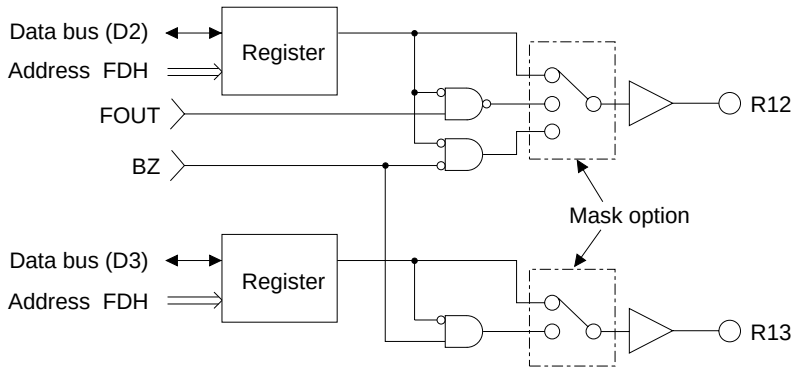


Fig. 3.5.2
Configuration of R12 and R13

Note *It is possible that hazard appear when the FOUT, BZ or $\overline{BZ}$ output turns ON or OFF.*

With FOUT selected, the pin provides the OSC1 oscillation clock (f_{OSC1}) or an OSC1 clock divided by 2 to 128 (selected by using mask option).

With BZ selected, the pin provides f_{OSC1} divided by 8 or 16 for driving a piezo-electric buzzer. A division ratio (8 or 16) can be selected by using the mask option i.e. when f_{OSC1} = 32 kHz, either 4 kHz or 2 kHz can be selected as f_{BZ}.

$\overline{BZ}$ is the complementary output of BZ as shown in Figure 3.5.3. If BZ alone cannot develop an effective voltage high enough to drive the buzzer, $\overline{BZ}$ is used with BZ to double the effective voltage.

The $\overline{BZ}$ output can be used only when the BZ output is selected at port R13.

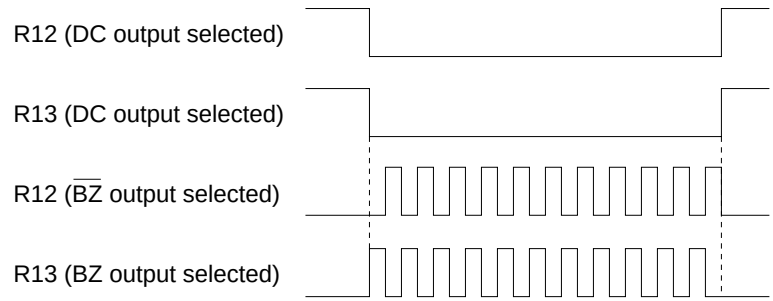


Fig. 3.5.3
R12 and R13 Output Waveforms

3.6 I/O Ports (P00–P03)

The S1C621A0 has a single 4-bit general-purpose I/O port. As shown in Figure 3.6.1, the Pnm pin works as an output port and it goes high or low levels when the value of the control register (IOC) is 1. The Pnm pin works as an input port and it goes to high-impedance state when the value of the control register is 0. (in this case, however, the Pnm pin is pulled up during and input operation under program control.)
At initial reset, the control register is set to 0 and the I/O port works as input.

Table 3.6.1 shows data assignment to the I/O port bits and the I/O control register on the memory map.

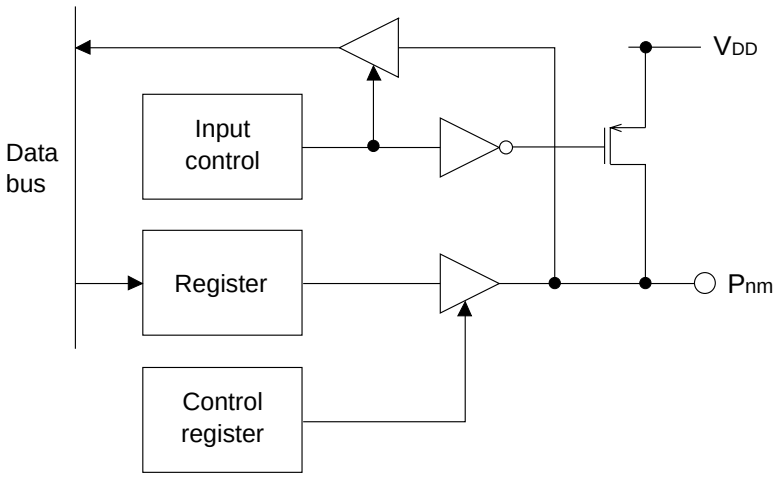


Fig. 3.6.1
I/O Port Configuration

Output data

The I/O port works as an output port when the control register is set to 1. The data at address 0FEH on the data memory is output to pins P00–P03.

Note *When the data at address 0FEH is read with the I/O port in the output mode, the data at the pin is directly loaded to the data bus. Note that, with the Pnm pin connected to a low-impedance load, the data stored in the data memory may differ from the data that is read.*

Table 3.6.1 I/O Memory Map

Address	Register								Comment
	D3	D2	D1	D0	Name	Init	1	0	
0FEH	P03	P02	P01	P00	P03	1	High	Low	I/O port (used as input port after initial reset)
	R/W				P02	1	High	Low	I/O port (used as input port after initial reset)
					P01	1	High	Low	I/O port (used as input port after initial reset)
					P00	1	High	Low	I/O port (used as input port after initial reset)
0FFH	OPDT	OPON	IOC	R20	OPDT	–	High	Low	Analog comparator data (0 or 1)
	R	R/W			OPON	0	On	Off	Analog comparator on/off
					IOC	0	Out	In	I/O port control out/in
					R20	0	High	Low	Output port

Note: Init = value at initial reset, – = not set on the circuit

Input data

Independently of the value of the I/O control register, the I/O port reads the value of the Pnm pin as data. Only during the read operation, the Pnm pin is pulled up (see Figure 3.6.1). This means that, in the input mode, the I/O port reads the data given to the Pnm pin and, in the output mode, reads the data loaded from the Pnm register to the Pnm pin.

I/O control (IOC) register

A register (named IOC) which controls the output state of the I/O port (Pnm) on a 4-bit basis is assigned to 0FFH, D1 on the memory map. With 1 set in this register, the I/O port (Pnm) works as output. With 0 set in the register, the I/O port is a high-impedance state and can accept external data, and pulled up only on reading operation. At initial reset, the IOC register is set to 0.

Notes on operation with pull-up resistors

With the Pnm pin used as input and the on-chip pull-up resistor connected to it, an input error will result if the time constant of the input line capacitive load and the on-chip pull-up resistor is greater than the read time. Although the pull-up time equals 1.5 times the wavelength of the system clock, it is necessary to limit the potential at the pin to within 0.5 times the wavelength. If this condition cannot be satisfied, an externally connected pull-up resistor must be used or the read operation repeated as many times as necessary.

3.7 LCD Driver (COM0–COM3, SEG0–SEG31)

The S1C621A0 has four common pins and 32 segment pins, which can drive a maximum of 128 (32 × 4) segments LCD. The LCD driving power is generated on-chip and need not be supplied by an external source (see section 3.2 "Power Supply"). The LCDs are dynamically driven with a 1/4 duty using four voltages (V_{DD}, V_{L1}, V_{L2} and V_{L3}). (A 1/3 duty can also be used by using mask option.) The frame frequency is 1/512 times the OSC1 oscillation frequency in the 1/4 duty mode, and 1/384 times in the 1/3 duty mode. Table 3.7.1 shows the differences LCD driving mode options. Figures 3.7.1 and 3.7.2 show 1/3 and 1/4 duty mode driving waveforms.

Table 3.7.1
LCD Drive Mode Options

Duty	COM pins used	Max. number of segments	Frame frequency	When f_{OSC1} = 32 kHz
1/4	COM0–COM3	128 (32 × 4)	f_{OSC1} / 512	64 Hz
1/3	COM0–COM2	96 (32 × 3)	f_{OSC1} / 384	85 Hz

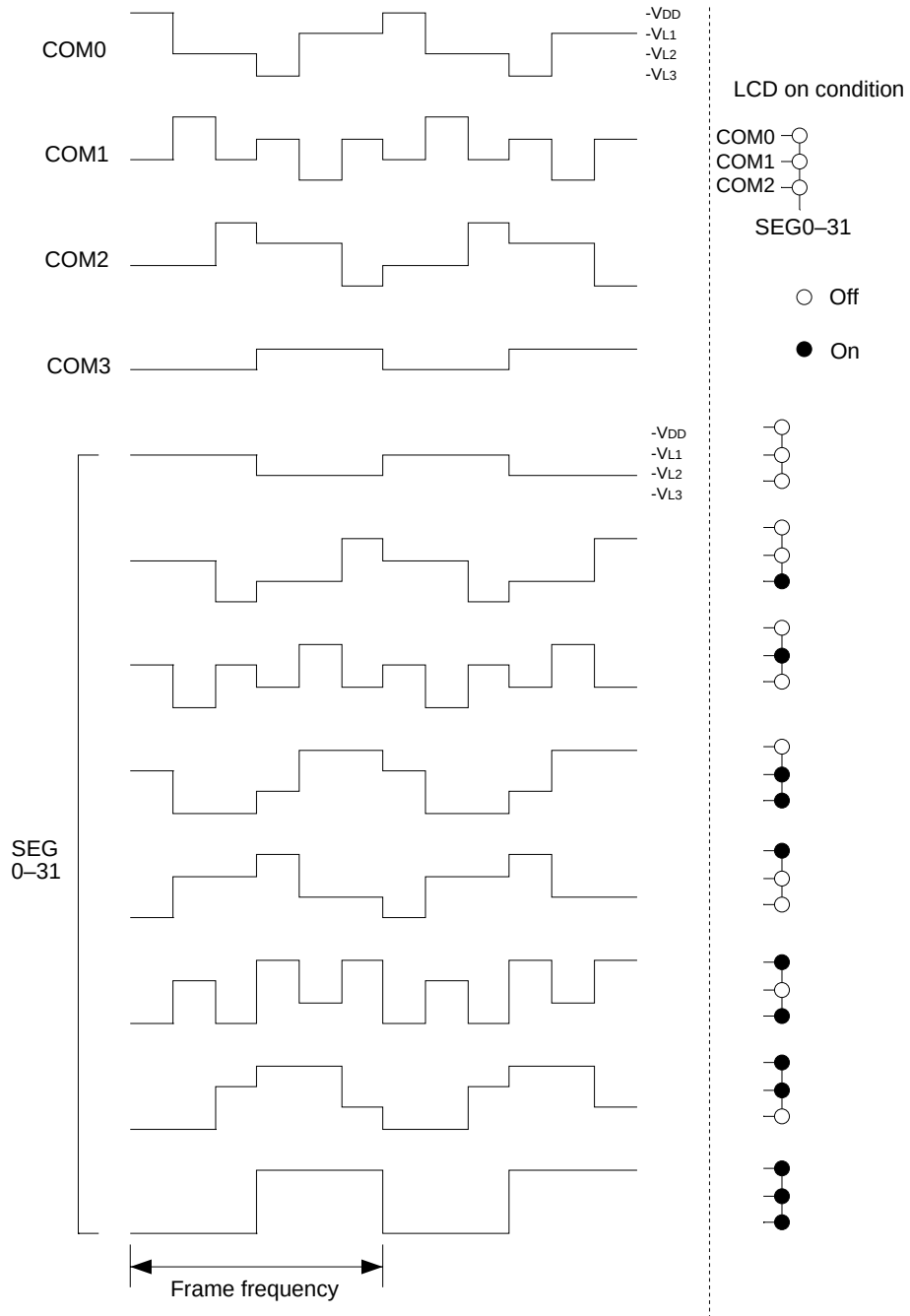


Fig. 3.7.1
1/3 Duty Mode Driving
Waveforms

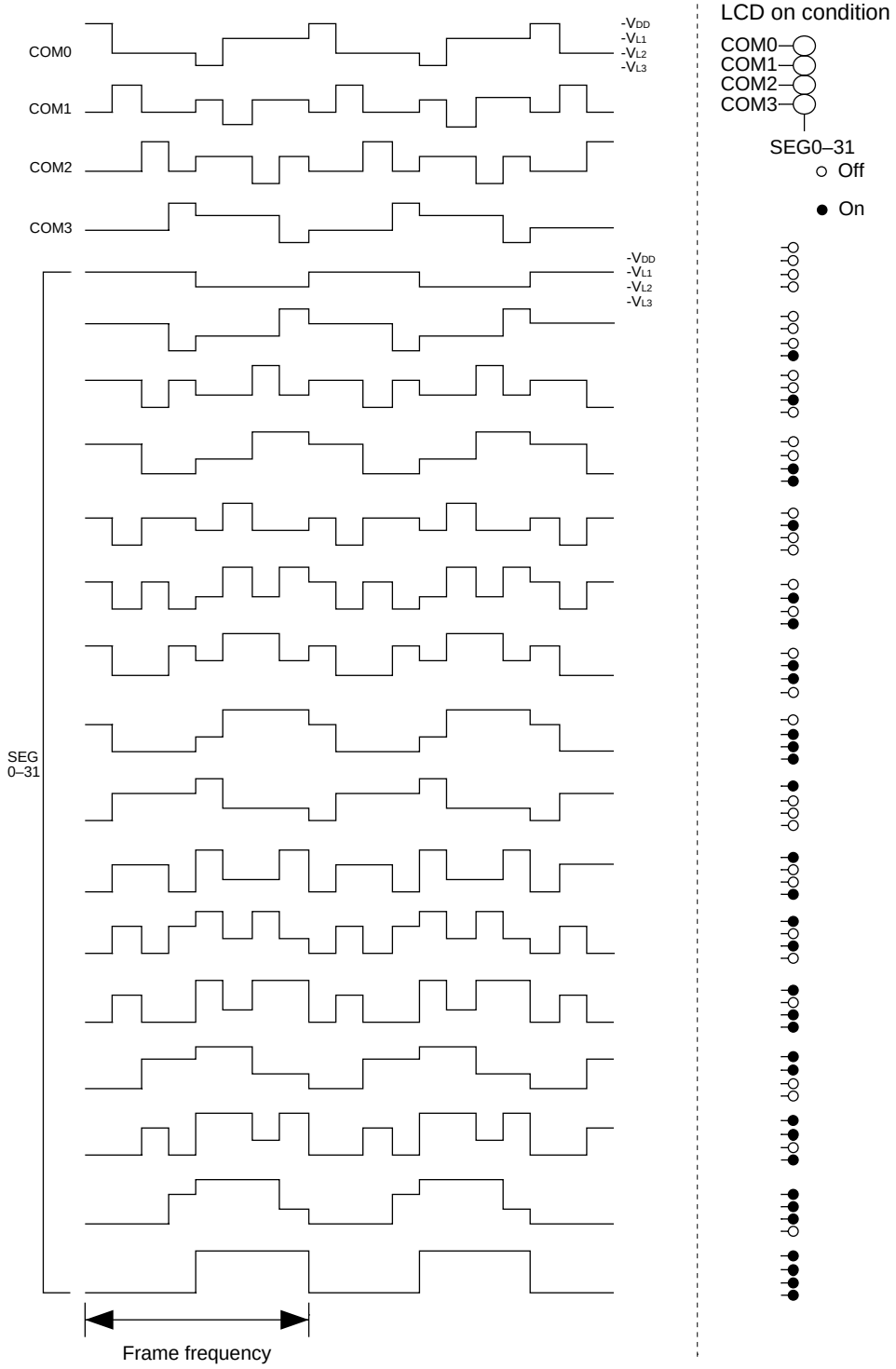


Fig. 3.7.2
1/4 Duty
Mode Driving
Waveforms

Segment assignment

Segment data is programmed at the display memory (write-only) addresses from 0D0H to 0EFH on the memory map. Writing 1 to the display memory turns on the associated LCD segment, and writing 0 turns off the associated LCD segment.

Any address and data and data bit can be assigned to a segment pin (SEG0–SEG31) by using mask option as long as they are written to within the above memory space. This allows design of a very flexible LCD panel. Figure 3.7.3 shows the relation between LCD segments and memory for 1/3 duty operation, for example.

Address	Data			
	D3	D2	D1	D0
0ECH	d	c	b	a
0EDH	p	g	f	e
0EEH	d'	c'	b'	a'
0EFH	p'	g'	f'	e'

Display memory assignment table

	Common 0	Common 1	Common 2
SEG10	EC, D0 (a)	ED, D1 (f)	ED, D0 (e)
SEG11	EC, D1 (b)	ED, D2 (g)	EC, D3 (d)
SEG12	EF, D1 (f')	EC, D2 (c)	ED, D3 (p)

Pin assignment table

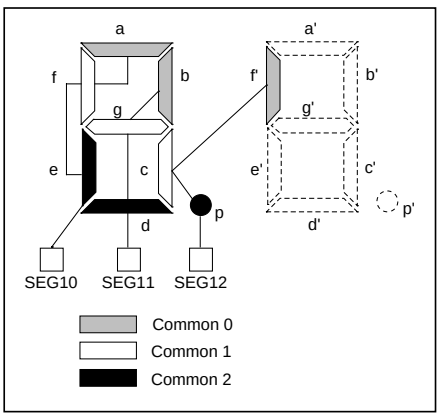


Fig. 3.7.3
Segment Assignment
— LCD Panel Example

The segment pins (SEG0–SEG31) can be DC output in units of two segment pins (binary output of VDD and VSS) by using the mask option. The output data corresponds to the common 0 of each segment pin.

3.8 Timer

The S1C621A0 has a built-in timer using the OSC1 clock as its source oscillation frequency. The timer can be controlled by the data memory at addresses from 0F3H to 0F5H.

The information given in this section is based on f_{OSC1} = 32.768 kHz. For a system which uses an oscillator having any other frequency at OSC1, substitute the appropriate value for 32.768 kHz throughout this section.

Timer memory map

Table 3.8.1 I/O Memory Map

Address	Register D3	D2	D1	D0	Name	Init	1	0	Comment
0F1H	WDRST	TI2	TI8	TI32	WDRST	Clear	Clear	–	Watchdog timer reset
	W	R			TI2	0	Yes	No	Interrupt factor flag (Timer 2 Hz) Clear to 0 after read
					TI8	0	Yes	No	Interrupt factor flag (Timer 8 Hz) Clear to 0 after read
					TI32	0	Yes	No	Interrupt factor flag (Timer 32 Hz) Clear to 0 after read
0F3H	TMRUN	ETI2	ETI8	ETI32	TMRUN	0	Run	Clear & Stop	Timer Run/Clear & Stop
	R/W				ETI2	0	Enable	Mask	Interrupt mask register (Timer 2 Hz)
					ETI8	0	Enable	Mask	Interrupt mask register (Timer 8 Hz)
					ETI32	0	Enable	Mask	Interrupt mask register (Timer 32 Hz)
0F4H	TM03	TM02	TM01	TM00	TM03	0			Timer data (low-order) 16 Hz
	R				TM02	0			Timer data (low-order) 32 Hz
					TM01	0			Timer data (low-order) 64 Hz
					TM00	0			Timer data (low-order) 128 Hz
0F5H	TM13	TM12	TM11	TM10	TM13	0			Timer data (High-order) 1 Hz
	R				TM12	0			Timer data (High-order) 2 Hz
					TM11	0			Timer data (High-order) 4 Hz
					TM10	0			Timer data (High-order) 8 Hz

Note: Init = value at initial reset, x = undefined, – = not set on the circuit

• **ETI32:** Interrupt mask register (Timer 32 Hz)

D0 = 0 Disable a 32 Hz interrupt request
D0 = 1 Enable a 32 Hz interrupt request

• **ETI8:** Interrupt mask register (Timer 8 Hz)

D1 = 0 Disable a 8 Hz interrupt request
D1 = 1 Enable a 8 Hz interrupt request

• **ETI2:** Interrupt mask register (Timer 2 Hz)

D2 = 0 Disable a 2 Hz interrupt request
D2 = 1 Enable a 2 Hz interrupt request

• **TMRUN:** Timer circuit control

D3 = 0 Reset state (Clear and stop)
D3 = 1 Increment state (run)

• **TM03–TM00:** Timer data (low-order)

TM00 128 Hz
TM01 64 Hz
TM02 32 Hz
TM03 16 Hz

• **TM13–TM10:** Timer data (high-order)

TM10 8 Hz
TM11 4 Hz
TM12 2 Hz
TM13 1 Hz

Timer data

The timer is an 8-bit binary counter. Software can read signals of 128 Hz to 1 Hz from the timer. The four bits of 128 Hz to 16 Hz are read through the data memory at address 0F4H, and the four bits of 8 Hz to 1 Hz through the data memory at address 0F5H. Figure 3.8.1 shows the timer timing chart.

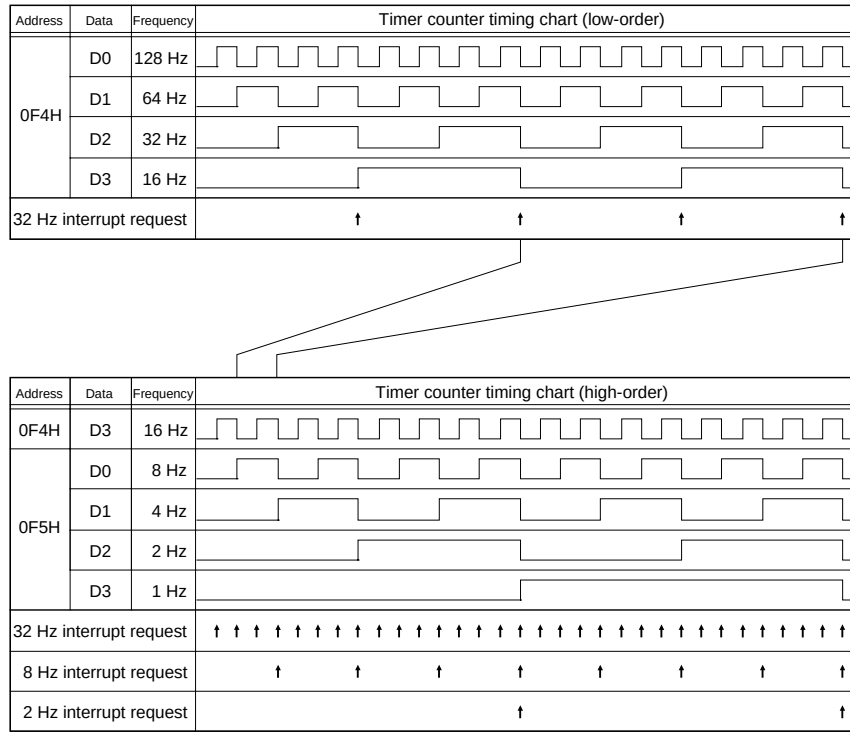


Fig. 3.8.1
Timing Chart of Timer

Timer interrupt request

The timer sets an interrupt factor flag (the low-order three bits of address 0F1H: TI32, TI8, TI2) to 1 at the falling edge of a 32 Hz, 8 Hz or 2 Hz signal, thus generating an interrupt request signal to the CPU. The interrupt request signal can be masked for each frequency using the low-order three bits of address 0F3H. In other words, ETI32, ETI8 and ETI2 enable or disable a 32 Hz interrupt request, an 8 Hz interrupt request and a 2 Hz interrupt request, respectively. The interrupt factor flag is a read-only register. When it is read, its contents are cleared to 0. Read this register in DI status (interrupt flag = 0). Reading of interrupt factor flags is available at EI, but be careful in the following cases.
If the interrupt mask register value corresponding to the interrupt factor flags to be read is set to 1, an interrupt request will be generated by the interrupt factor flags set timing, or an interrupt request will not be generated. Be very careful when interrupt factor flags are in the same address.
At initial reset, all of the 32 Hz, 8 Hz and 2 Hz interrupt requests are disabled.

Timer reset

The timer can be reset by using D3 (TMRUN) of data memory as address 0F3H.

With TMRUN at 0, the timer is in reset state (all time outputs are 0). With TMRUN at 1, the timer is in the increment state. At initial reset, TMRUN is 0 and the timer is in reset state.

Note *The timer is a read-only register, in which no data can be stored. It uses the OSC1 output as its source oscillation frequency, and is unaffected by the OSC3 oscillation output.*

3.9 Remote LED Control (R33, REM)

The S1C621A0 has an REM output pin which drives remote-controlled LEDs. The pin provides a low-level output while the REM output is off. (See Figure 3.9.1.)

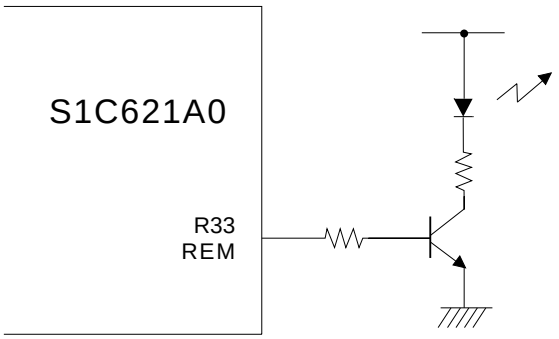


Fig. 3.9.1
Remote LED Control Circuit

In the infrared remote control generally used with household equipment, the transmission code is modulated in PPM process and the resulting signal is used for amplitude modulation (AM) of a carrier of about 38 kHz before transmission on infrared rays.

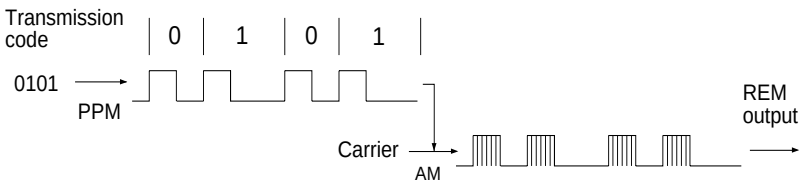


Fig. 3.9.2
Remote-controlled Transmission

Lower remote-control current consumption is ensured with a carrier duty ratio less than 1/2.

The S1C621A0 supports the following functions to implement remote-controlled transmission:

- A carrier frequency(fcarrier) can be set to a frequency 1/8 or 1/12 times the ceramic oscillator frequency (OSC3) by software. (division ratio 1)

Note *If an option is selected without use of OSC3, the OSC1 clock (instead of the OSC3 clock) is introduced into the REM circuit. For an option selected without using OSC3, the term "OSC3" should read "OSC1" in the description that follows.*

- A carrier duty can be selected by software from the four different values.
- Software supports two timer control modes: named soft-timer mode and hard-timer mode. In the soft-timer mode (software timer control mode), the ON/OFF timing of carriers is controlled by software. In the hard-timer mode (hardware timer control mode), software sets the ON-time, then hardware generates a waveform and an interrupt (REM interrupt) each time one bit is transmitted.

Table 3.9.1 summarizes the differences between the soft-timer mode and the hard-timer mode.

Table 3.9.1 Characteristics of Soft-timer and Hard-timer Control Modes

Item	Soft-timer mode	Hard-timer mode
Processing of other routines during REM output	Difficult	Possible
Reference cycle (τ) sway during REM transmission	Source oscillation sway and errors caused by instruction cycles	Source oscillation sway only
Setting of REM output width	Variable to any width	Fixed to several width
Relation between REM reference cycle and moduration frequency cycle	Variable	Fixed to several cycles
Carrier waveform	Duty slightly disturbed before and after ON time	Stabilized at setting

REM control registers

Table 3.9.2 REM Control Registers

Address	Data bit	R/W	Name	Controlled by: Software timer mode	Controlled by: Hardware timer mode	Contents
0F0H	D3	R/W	REMSO	○		0: Forced REM output off. (at initial reset) 1: Forced REM output on. Writing 1: Carrier counter is forced to be reset and started (for hardware timer, must be fixed at 0).
0F0H	D2	R	IREM		○	REM interrupt factor flag (undefined at initial reset) 1: interrupt request
0F2H	D3	R/W	REMC	○	○	0: Turns REM carrier circuit off. 1: Turns REM carrier circuit on. (at initial reset)
0F2H	D2	R/W	EIREM		○	REM interrupt mask register 0: Disables (masks) interrupt 1: Enables interrupt
0F7H	D3 D2	R/W R/W	RCDIV RCDUTY	○	○	Sets REM carrier interval and duty ratio. RCDIV / RCDUTY / Frequency division ratio 1 / Duty 0 / 0 / 1/8 / 1/4 0 / 1 / 1/8 / 3/8 1 / 0 / 1/12 / 1/3 1 / 1 / 1/12 / 1/4 (Division ratio 1 = fcarrier/fOSC3)
0F7H	D1 D0	R/W R/W	RT1 RT0		○	Sets REM reference cycle (τ). RT1 / RT0 / Frequency division ratio 2 0 / 0 / 1/12 0 / 1 / 1/16 1 / 0 / 1/20 1 / 1 / 1/32 (Division ratio 2 = fτ/fcarrier)
0F8H	D3 D2 D1 D0	W W W W	RIC3 RIC2 RIC1 RIC0		○	Sets count in interrupt counter. Counts on each time data is written. Writing 0FH is prohibited. Writing in software timer mode is prohibited.
0F9H	D3 D2	R/W R/W	ROUT1 ROUT0		○	Sets REM output width (τ count). Operateds only once each time data is written. Writing in software timer mode is prohibited.

Soft-timer mode

The soft-timer mode supported by the S1C621A0 uses a timer program which directly controls the duration, etc. of REM output modulation carriers generated by hardware.

The control registers used to implement the soft-timer mode are REMC, RCDIV, RCDUTY and REMSO. (See Figure 3.9.3.)

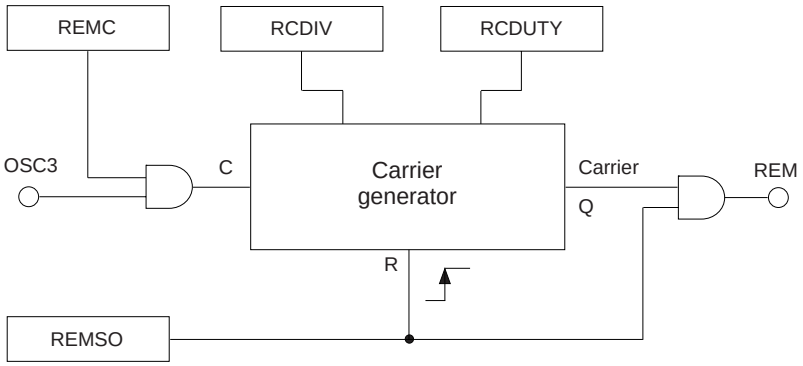


Fig. 3.9.3
Block Diagram of REM Circuit (Soft-timer Mode)

- **REMC Register (0F2H, D3)**

  Setting the REMC bit to 0 causes the REM circuit to stop to save power. With the REMC bit at 1, the OSC3 clock is introduced into the carrier generator circuit making the REM circuit operational. At initial reset, the REMC bit is set to 1 for initializing the carrier generator. The bit must not be set to 0 until after initialization (within 32 machine cycles).

- **RCDIV and RCDUTY Registers (0F7H, D2, D3)**

  A REM carrier is generated by dividing the OSC3 clock. The RCDIV and RCDUTY registers are used to set a division ratio and a duty. (See Table 3.9.3.)

  The registers can be written to only while the REM circuit is OFF (except for initialization). An error will result if they are set at any other time.

Once data has been written into the RCDIV and RCDUTY registers, carriers will be generated with the same division ratio and the same duty ratio. Figure 3.9.4 shows the waveforms of the OSC3 clock and carriers with different division ratios and duties.

Table 3.9.3
Division Ratio and Duty

RCDIV	RCDUTY	Division ratio 1	Duty
0	0	1 / 8	1/4
	1		3/8
1	0	1 / 12	1/3
	1		1/4

• REMSO Register (0F0H, D3)

The REMSO register provides REM output ON/OFF control. It is set to 0 at initial reset. Writing 1 in the REMSO bit causes the carrier generator to restart from a reset state and a carrier wave to be output at the REM pin. Writing 0 in the bit makes the REM pin go low.

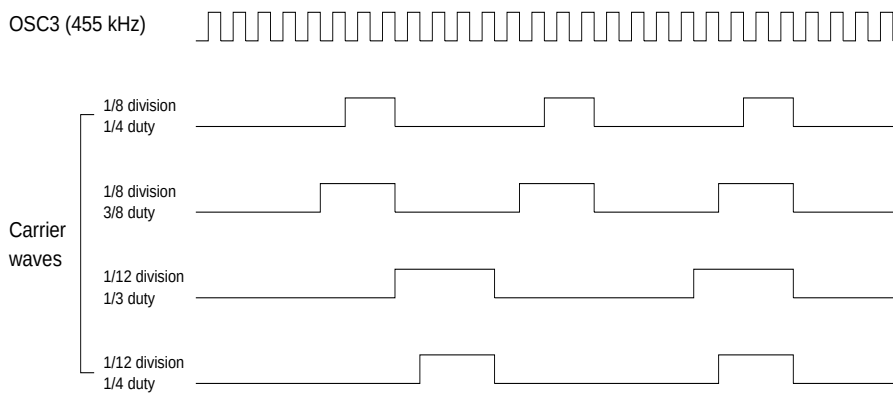


Fig. 3.9.4
OSC3 and Carriers

Hard-timer mode

With REM ON/OFF operations under soft-timer mode, the CPU is forced to devote the greater part of its time to them so that it has no flexibility for execution of other routines. To alleviate this problem, the S1C621A0 supports hardware REM control mode.

In the hard-timer mode, hardware automatically generates REM according to the REMOUT time t_{RO} (REM-ON time) and REM interrupt time t_{RI} which are defined by software. The hardware uses a REM interrupt to inform the timer that transmission of one bit is completed.

While the REM circuit is ON, the software timer sets the next REMOUT time. This assures that REM can be output in stable cycles. (See Figure 3.9.5.)

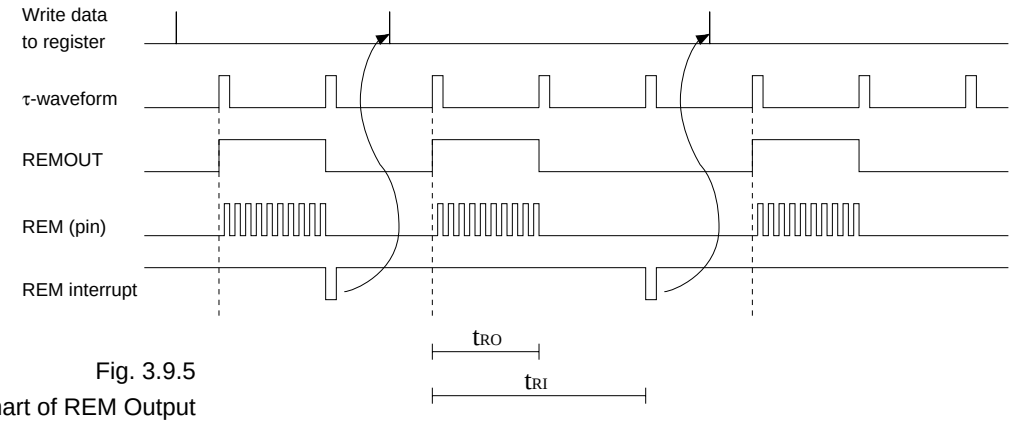


Fig. 3.9.5
Timing Chart of REM Output
(Hard-timer Mode)

Figure 3.9.6 shows a block diagram of the REM circuit used under hard-timer mode.

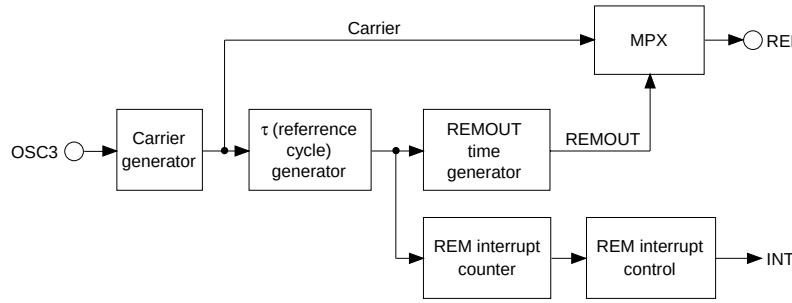


Fig. 3.9.6
Block Diagram of REM Circuit (Hard-timer Mode)

Following is a description of the associated registers (see the block diagram):

• Carrier Generator

This block is the same circuit as used in the soft-timer mode. The associated registers are REMC, RCDIV and RCDUTY all of which are set in the same way as in the soft-timer mode. The difference is that the REMSO register must be fixed at 0.

• τ (Reference Cycle) Generator

This block generates the cycle τ which gives a reference time for REMOUT and REM interrupts. The reference cycle τ is produced by dividing the REM carrier frequency.

The division ratio is defined by RT1 (0F7H, D1) and RT0 (0F7H, D0). (See Table 3.9.4.)

Table 3.9.4
Division Ratio

RT1	RT0	Division ratio 2
0	0	1 / 12
0	1	1 / 16
1	0	1 / 20
1	1	1 / 32

Once values have been set in these registers, subsequent cycles will be generated with the same division ratio. Note that they can be set only while the REM circuit is OFF (except for initialization). (An error will result if the registers are set at any other time.) Figure 3.9.7 shows the τ-waveform with the division ratio 2 set at 1/12.

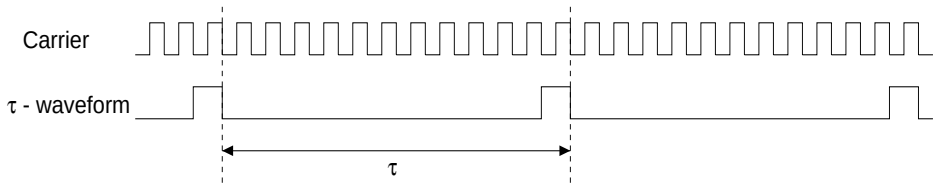


Fig. 3.9.7
τ-waveform

When the REMC is set to 0, the REM circuit stops synchronously with τ. This timing is shown in Figure 3.9.8.

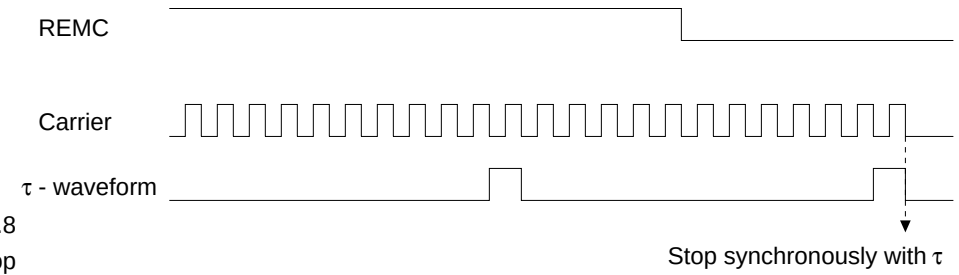


Fig. 3.9.8
REM Circuit Stop Timing

Maximum of 384 machine cycles* are required until the REM circuit stops after the REMC is set to 0. Even if the CPU clock is changed from OSC3 to OSC1 after the REMC = 0, OSC3 must not be turned OFF before the REM circuit stops.

* This time depends on the value of the set τ cycle. If a shorter τ cycle is set, the maximum time required for the REM circuit to stop after REMC = 0 is shortened.

• REMOUT Time Generator

This hardware block generates REMOUT time t_{RO}. This circuit starts operating at the rise of the τ-waveform each time data is written into the REMOUT time setting registers ROUT1 (0F9H, D3) and ROUT0 (0F9H, D2). The circuit automatically stops after the set time elapses.

The REMOUT time t_{RO} is defined by the following equation:

$$t_{RO} = (ROUT1 \times 2 + ROUT0) \times \tau$$

The values set in the ROUT1 and ROUT0 registers are taken into the REM time generator synchronously with the rise of a τ-waveform. For this reason, avoid writing data into the ROUT1 and ROUT0 registers during one carrier cycle immediately before and after the rise of the τ-waveform.

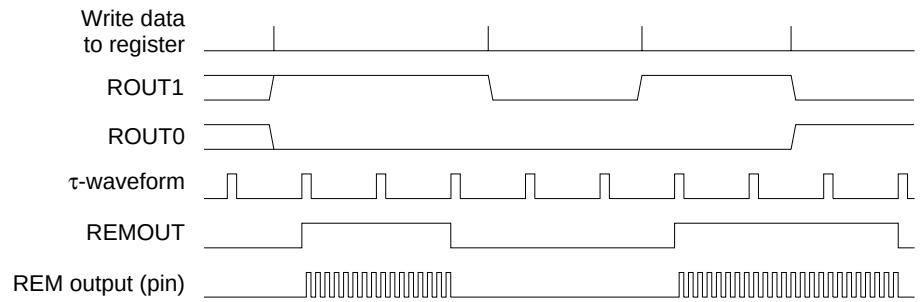


Fig. 3.9.9
REM Circuit
Restart Timing 1

If the REM circuit is restarted from OFF state with REMC = 0, the timing of the τ-waveform rises one carrier before the set division ratio 2. (See Figures 3.9.9 and 3.9.10.)

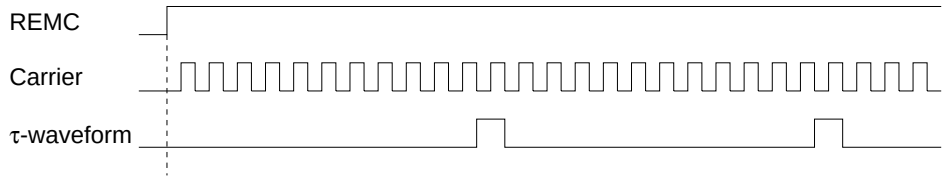


Fig. 3.9.10 REM Circuit Restart Timing 2

- **REM Interrupt Counter and REM Interrupt Control**

  The REM circuit can make an interrupt request to inform the software of how REM operations are progressing. Interrupt timing is set by writing data into the count setting register (0F8H, D3–D0) of the REM interrupt counter (RIC).

  The time until an interrupt request occurs (t_{RI}) is given by:

  $t_{RI} = t_{RIC} + (1 \pm 1 \text{ instruction cycle})$

  Where t_{RIC} is the time set by the RIC register ($n \times \tau$). The relation between the RIC register and t_{RIC} is:

  $t_{RIC} = (RIC3 \times 2^3 + RIC2 \times 2^2 + RIC1 \times 2 + RIC0) \times \tau$

  As with the REMOUT time circuit, the REM interrupt counter starts counting synchronously with the rising edge of the τ-waveform. The interrupt control circuit generates a REM interrupt synchronously with the τ-wave when the count is completed. (See Figure 3.9.11.)

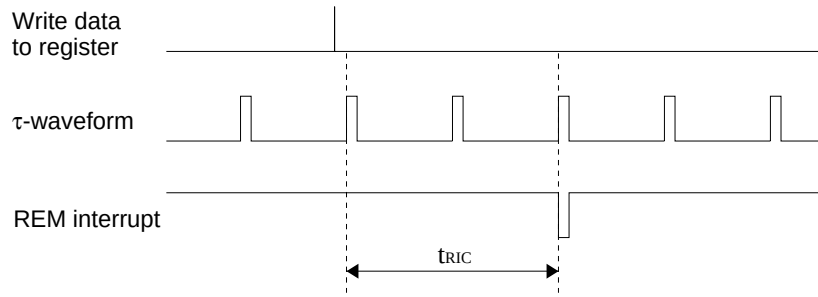


Fig. 3.9.11
REM Interrupt Timing

Once data has been written in the RIC register, avoid writing other data into the register before a REM interrupt occurs (which would otherwise cause an invalid interrupt).

The values allowed for the RIC register are 0 to EH.

• Other Registers

EIREM (0F2H, D2) and IREM (0F0H, D2) are also the registers that are used with the hardware timer control mode.

The EIREM register controls REM interrupt masking. It masks (invalidates) an interrupt request when it is 0, and validates an interrupt request when it is 1.

The IREM register is read to have software check the presence or absence of a REM interrupt. An interrupt is absent when IREM is 0, and present when it is 1. The REM interrupt factor flag is cleared when the register is read. Read the IREM register in DI status (interrupt flag = 0). Reading of interrupt factor flags is available at EI, but be careful in the following cases.
If the interrupt mask register value corresponding to the interrupt factor flags to be read is set to 1, an interrupt request will be generated by the interrupt factor flags set timing, or an interrupt request will not be generated. Be very careful when interrupt factor flags are in the same address.

Initializing REM circuit

The following programming steps are needed to initialize the REM circuit (τ clock, REM interrupt circuit) under software control:

(1) Write data at addresses 0F7H and 0F8H in that order within 80 machine clocks (equivalent to eleven 7-clock instructions) after release from initial reset.

(2) With REMC = 0 (0F2H, D3), the REM circuit must not be stopped within cycle of 1τ after data has been written at address 0F8H.

(3) To initialize the REM interrupt circuit, read the REM interrupt factor flag (address 0F0H) to clear it at least an interval of 2τ after data has been written at address 0F8H.

REM circuit programming notes

In programming the REM circuit, the following precautions should be observed:

(1) After initial reset, the REMC register stays at 1 to initialize the carrier generation circuit. The REMC register can only be reset to 0 after initialization (at least 32 machine cycles later).

(2) The REM circuit does not stop immediately after the REMC register is reset to 0. It stops synchronously with the interval τ, during which OSC3 must be held ON.

(3) With the REM circuit in operation, do not write data at addresses 0F8H and 0F9H (REM interrupt counter and REMOUT time setting register) during an interval of one carrier before and after the rise of τ.

(4) With the REM circuit in operation, do not write data at addresses 0F7H (τ-setting register).

(5) During the operation under hard-timer mode, the REMSO register must be fixed at 0.

(6) Read the IREM (REM interrupt factor flag) in DI status (interrupt flag = 0). Reading of interrupt factor flags is available at EI, but be careful in the following cases.
If the interrupt mask register value corresponding to the interrupt factor flags to be read is set to 1, an interrupt request will be generated by the interrupt factor flags set timing, or an interrupt request will not be generated. Be very careful when interrupt factor flags are in the same address.

(7) If the RIC register is set again before a REM interrupt occurs with the RIC register set, an invalid interrupt may occur.

(8) The values that can be set in the REM interrupt counter (0F8H) are from 0 to 0EH. Remember, writing 0FH into the counter may cause an error.

(9) Soft-timer mode cannot coexist with hard-timer mode. To use them in combination, stop the REM circuit before selecting either.

3.10 AMP and SVD

The S1C621A0 contains a general purpose analog comparator (AMP) and a supply voltage detector (SVD). To save power, these circuits can be powered individually ON/OFF under software control. Also, the output data of the circuits can be read by software. Table 3.10.1 shows a memory map involving the relation between the AMP and SVD.

Table 3.10.1 I/O Memory Map

Address	Register								Comment
	D3	D2	D1	D0	Name	Init	1	0	
0F6H	SVDDT	SVDON	CLKCHG	OSCC	SVDDT	–	Normal	Low voltage	SVD data (1 when SVDON = 0)
	R	R/W			SVDON	0	On	Off	SVD circuit on/off
					CLKCHG	0	OSC1	OSC3	CLK change OSC1/OSC3
					OSCC	1	On	Off	OSC3 oscillation on/off
0FFH	OPDT	OPON	IOC	R20	OPDT	–	High	Low	Analog comparator data (0 or 1)
	R	R/W			OPON	0	On	Off	Analog comparator on/off
					IOC	0	Out	In	I/O port control out/in
					R20	0	High	Low	Output port

Note: Init = value at initial reset, Low = low voltage, – = not set on the circuit

AMP (MOS input analog comparator)

As shown in Figure 3.10.1, the MOS input general-purpose analog comparator contained in the S1C621A0 has three pins: two inputs (inverted: AMPM, non-inverted: AMPP) and one output pin (AOUT).

• OPON

This register (OPON) controls analog comparator power circuit ON/OFF to save power. It is assigned to 0FFH, D2 of the memory space. Writing 1 to the register turns ON the AMP circuit, and writing 0 turns OFF the circuit. At initial reset, the register is set to 0. Data can be written to or read from the register. A waiting time of at least 2 ms is required for the AMP circuit to become stable after its power is turned ON. The comparator response time (t_{dA}) depend on difference of input voltage between AMPP and AMPM. Then it is necessary to wait mode time to defect small signal difference than electrical characteristics condition on t_{dA}.
for example:
t_{dA} (max.) = 5 ms at $V_{AMPM} = V_{AMPP} \pm 5$ mV 25°C

• OPDT

When the AMP circuit is used in such a way that the comparator output takes binary form (0 or 1 only), the data can be read by software. The data (read-only) is assigned to 0FFH, D3 of the memory space.

With the AMP circuit OFF (OPON = 0), AOUT goes to 1 and the read data (OPDT) also goes to 1.

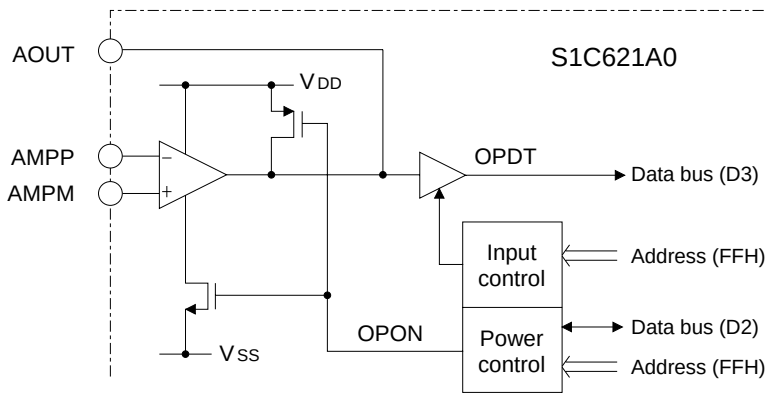


Fig. 3.10.1
AMP Circuit Block Diagram

SVD (Supply voltage detector)

The S1C621A0 contains a supply voltage detector (SVD). Low battery voltage can be detected by software. If viewed from the CPU, the SVD circuit has the same configuration as the AMP circuit as shown in Figure 3.10.2.
When SVDON = 1, $|V_{SS}| \geq |V_{SVD}|$ and SVDDT = 0 can be read.
When SVDON = 1, $|V_{SS}| < |V_{SVD}|$ and SVDDT = 1 can be read.
When SVDON = 0, SVDDT = 1 can be read.

• SVDON

This register (0F6H, D2) controls SVD circuit power ON/OFF switching to save power. Data can be written to or read from the SVDON register. Writing 1 into the register supplies SVD power. Writing 0 into the register turns SVD power off (power-save mode). At initial reset, SVDON is set to 0.

• SVDDT

When the supply voltage is less than the threshold $|V_{SVD}|$ with SVDON = 1, the SVD circuit output goes to 0. When the supply voltage exceeds the threshold, the SVD circuit output goes to 1. Software can read these valued to detect any low voltage. The data (read-only) is assigned to 0F6H, D3 of the memory space. With SVDON = 0, the SVD circuit output is fixed to 1 and the read data (SVDDT) also goes to 1.

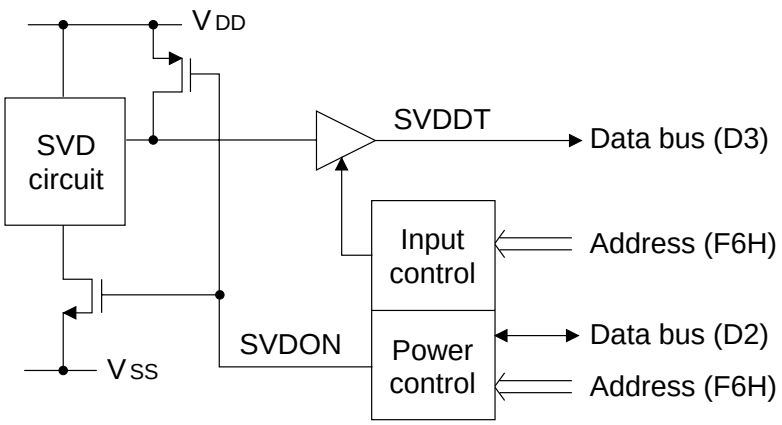


Fig. 3.10.2
SVD Block Diagram

3.11 Interrupt and Halt

The S1C621A0 has a total of six interrupt functions: two external input interrupts, three internal timer interrupts, and one remote control (REM) interrupt. Each of them can be masked. To enable an interrupt, the interrupt flag must be enabled (set to 1). Upon occurrence of an interrupt, the flag is disabled (set to 0).

When an interrupt occurs, the count of the next program to execute is saved into the stack (RAM) and the program counter is set to the interrupt vector (page 1, steps 01H to 0FH) depending on the interrupt factor. (These processes require a time of 12 clocks.) All subsequent processing is controlled by the software written in the interrupt vector.

Execution of the HALT instruction stops the CPU clock of the S1C621A0 to halt the CPU. An interrupt enables it to restart from the halt state. If the CPU can not restart, because dose not detect an interrupt, it restarts from initial reset state under watchdog timer control.

Table 3.11.1 I/O Memory Map

Address	Register								Comment
	D3	D2	D1	D0	Name	Init	1	0	
0F0H	REMSO	IREM	IK1	IK0	REMSO	0	On	Off	Forced REM output (on/off)
	R/W	R			IREM	x	Yes	No	Interrupt factor flag (REM) Clear to 0 after read
					IK1	0	Yes	No	Interrupt factor flag (K10–K13) Clear to 0 after read
					IK0	0	Yes	No	Interrupt factor flag (K00–K03) Clear to 0 after read
0F1H	WDRST	TI2	TI8	TI32	WDRST	Clear	Clear	–	Watchdog timer reset
	W	R			TI2	0	Yes	No	Interrupt factor flag (Timer 2 Hz) Clear to 0 after read
					TI8	0	Yes	No	Interrupt factor flag (Timer 8 Hz) Clear to 0 after read
					TI32	0	Yes	No	Interrupt factor flag (Timer 32 Hz) Clear to 0 after read
0F2H	REMC	EIREM	EIK1	EIK0	REMC	1	On	Off	REM carrier generation on/off
	R/W				EIREM	0	Enable	Mask	Interrupt mask register (REM)
					EIK1	0	Enable	Mask	Interrupt mask register (K10–K13)
					EIK0	0	Enable	Mask	Interrupt mask register (K00–K03)
0F3H	TMRUN	ETI2	ETI8	ETI32	TMRUN	0	Run	Clear & Stop	Timer Run/Clear & Stop
	R/W				ETI2	0	Enable	Mask	Interrupt mask register (Timer 2 Hz)
					ETI8	0	Enable	Mask	Interrupt mask register (Timer 8 Hz)
					ETI32	0	Enable	Mask	Interrupt mask register (Timer 32 Hz)

Note: Init = value at initial reset, x = undefined, – = not set on the circuit

Interrupt request

An interrupt request is caused by one of the following factors:

Table 3.11.2
Interrupt Factor and Interrupt Factor Flag

Interrupt factor	Interrupt factor flag	
Falling edge of 2 Hz timer signal	TI2	(0F1H•D2)
Falling edge of 8 Hz timer signal	TI8	(0F1H•D1)
Falling edge of 32 Hz timer signal	TI32	(0F1H•D0)
REM control	IREM	(0F0H•D2)
Falling edge of input (K10–K13)	IK1	(0F0H•D1)
Falling edge of input (K00–K03)	IK0	(0F0H•D0)

An interrupt factor sets the corresponding interrupt factor flag (read-only register) to 1. When its data is read, the register is reset to 0. At initial reset, it is reset to 0. (In this case, the value of the REM interrupt factor flag (IREM) is undefined.)

When the interrupt factor flag is set to 1 with both the corresponding interrupt mask register and interrupt flag set at 1, an interrupt request to the CPU is generated. (See Figure 3.11.3.)

Read the interrupt factor flag in DI status (interrupt flag = 0). Reading of interrupt factor flags is available at EI, but be careful in the following cases.
If the interrupt mask register value corresponding to the interrupt factor flags to be read is set to 1, an interrupt request will be generated by the interrupt factor flags set timing, or an interrupt request will not be generated. Be very careful when interrupt factor flags are in the same address.

• Timer Interrupt

As described in section 3.8 "Timer", the timer signal of 2 Hz, 8 Hz or 32 Hz can request a timer interrupt (when f_{OSC1} = 32.768 kHz). As the timer continues to operate even with the CPU in the halt state, the CPU can be restarted under timer control. (See Figure 3.11.1.)

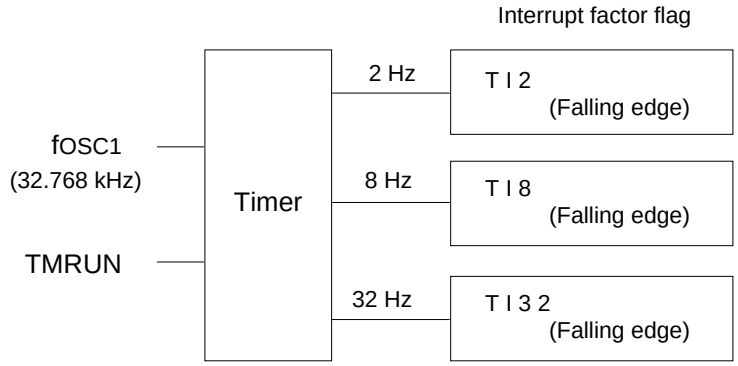


Fig. 3.11.1
Timer Interrupt Request Circuit

• REM Interrupt

As described in section 3.9 "Remote LED Control", a REM interrupt can be invoked during operation of the REM circuit or synchronously with a REM carrier. Note that the operation of the REM circuit is assured only with a CPU clock being supplied from OSC3.

• **Input Interrupt**

An input interrupt can be invoked by the IK1 group (K10–K13) or the IK0 group (K00–K03). As each pin contains an fOSC1/8 (4 kHz) clock noise reject circuit, the input must be held at low level for at least 16/fOSC1 (0.5 ms) to assure an input interrupt.

For the K10–K13 group, the interrupt factor flag can be set with the noise reject circuit bypassed by using the mask option. In this case, the input must be held at low level for at least 5 machine clocks (equivalent to 0.16 ms in the 32 kHz mode or 11 µs in the 455 kHz mode) to get an assured input interrupt.

Since the noise reject circuit continues operating with the CPU in the halt state, the CPU can be restarted by an input interrupt. Note that, if this is impossible, initial resetting under watchdog timer control is required.

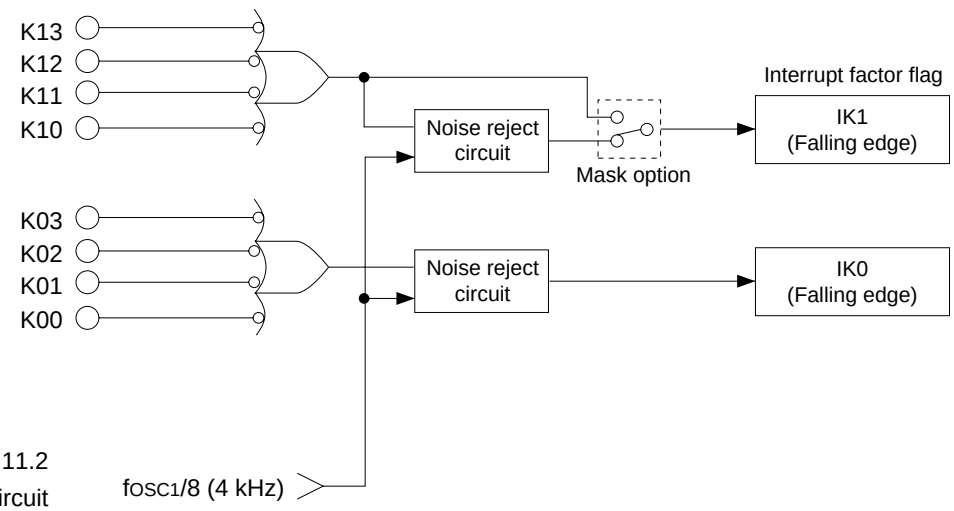
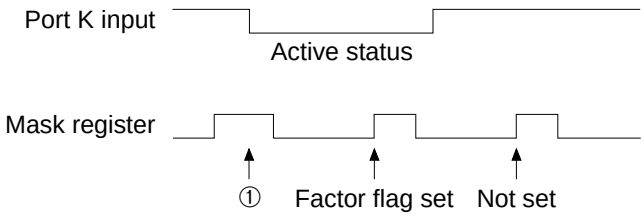


Fig. 3.11.2
Input Interrupt Circuit

<Input interrupt programing related precautions>



When the content of the mask register is rewritten, while the port K input is in the active status. The input interrupt factor flag is set at ①.

Fig. 3.11.3
Input Interrupt Timing

When using an input interrupt, if you rewrite the content of the mask register, when the value of the input terminal which becomes the interrupt input is in the active status (input terminal = low status), the factor flag for input interrupt may be set.
For example, a factor flag is set with the timing of ① shown in Figure 3.11.3. However, when clearing the content of the mask register with the input terminal kept in the low status and then setting it, the factor flag of the input interrupt is again set at the timing that has been set.
Consequently, when the input terminal is in the active status (low status), do not rewrite the mask register (clearing, then setting the mask register), so that a factor flag will only set at the falling edge in this case. When clearing, then setting the mask register, set the mask register, when the input terminal is not in the active status (high status).

Interrupt mask register

One interrupt mask register is available to each interrupt factor flag to mask an interrupt request. Data can be written to or read from the mask register. An interrupt request is enabled with 1 set in the register, and masked with 0 set in the register. (See Figure 3.11.4.) At initial reset, the mask register is reset to 0.

Table 3.11.3
Interrupt Mask Register

Interrupt mask register			Interrupt factor flag		
Name	Address	Data bit	Name	Address	Data bit
ETI2	0F3H	D2	TI2	0F1H	D2
ETI8		D1	TI8		D1
ETI32		D0	TI32		D0
EIREM	0F2H	D2	IREM	0F0H	D2
EIK1		D1	IK1		D1
EIK0		D0	IK0		D0

Interrupt vector

In response to an interrupt request, the CPU starts interrupt processing. The CPU saves the PC into the stack and makes a jump to the interrupt address, that is the interrupt handling routine. The interrupt address is indirectly specified by an interrupt factor. Interrupt addresses are assigned to page 1, steps 01H to 0FH of the PC. In other words, the low-order 4 bits of the PC are indirectly addressed by interrupt factors, as follows:

Table 3.11.4
Interrupt Vector

PC	Value	Interrupt factor
PCS3	1	Timer interrupt requested
	0	Timer interrupt not requested
PCS2	1	REM interrupt requested
	0	REM interrupt not requested
PCS1	1	K10–K13 interrupt requested
	0	K10–K13 interrupt not requested
PCS0	1	K00–K03 interrupt requested
	0	K00–K03 interrupt not requested

Examples
- Only timer interrupt requested
  — Jump to page 1, step 08H
- Both timer interrupt and REM interrupt requested
  — Jump to page 1, step 0CH

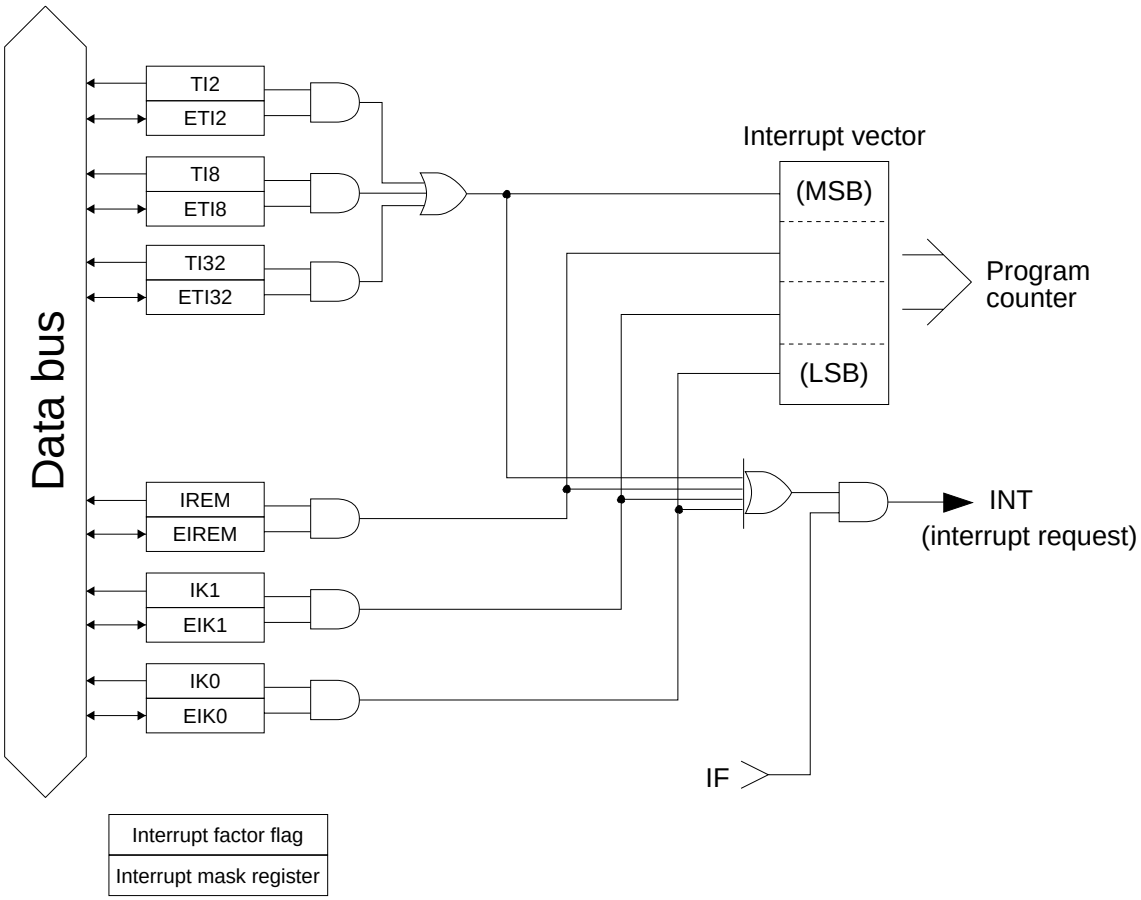


Fig. 3.11.4
Interrupt Request/Interrupt Vector Generation Circuit

3.12 Initial Reset

The S1C621A0 must be initially reset to initialize its circuits. Initial reset is triggered by an external reset ($\overline{\text{RESET}}$) signal, oscillation detector signal, or watchdog timer signal. The $\overline{\text{RESET}}$ input is needed for initialization at power-on.

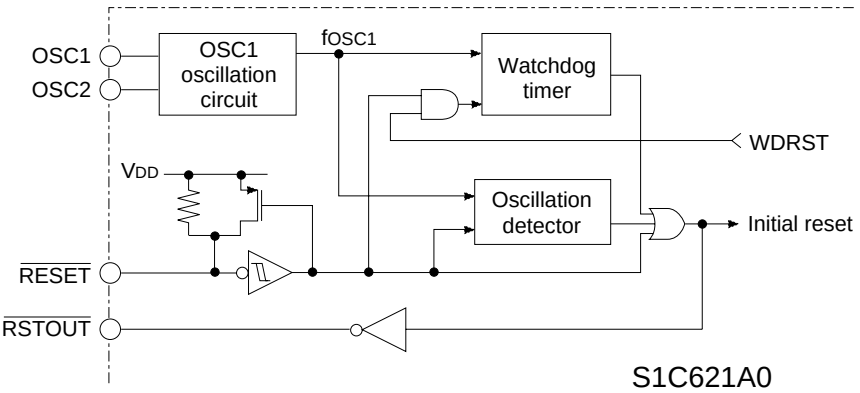


Fig. 3.12.1 Initial Reset Circuit Configuration

Reset at power-on

At power-on, the initial reset signal has two functions. One function is to initialize a circuit and the other to sustain the initializing function until the OSC3 oscillation is stabilized. Thus, the $\overline{\text{RESET}}$ input must be held at low level for at least 0.5 second after power-on.

After the $\overline{\text{RESET}}$ input reaches the high level and the OSC1 oscillation circuit starts operating, several milliseconds later, the system is released from internal reset and starts to operate.

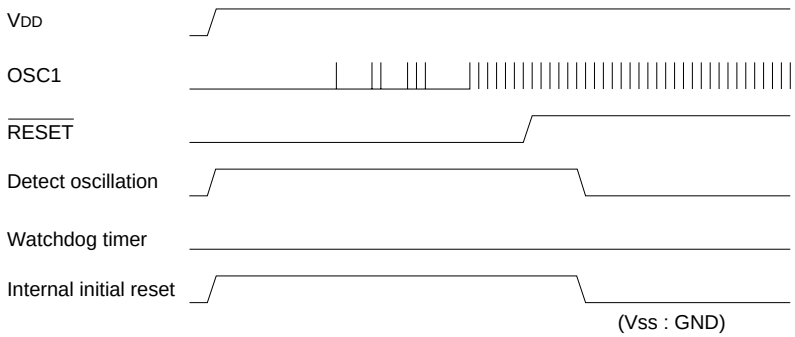


Fig. 3.12.2
Initial Reset Sequence at Power-on

$\overline{\text{RESET}}$

The $\overline{\text{RESET}}$ signal directly initializes the S1C621A0. The system is reset when $\overline{\text{RESET}}$ = 0, and released from the reset state when $\overline{\text{RESET}}$ = 1. As the $\overline{\text{RESET}}$ pin is pulled up and receives a Schmitt trigger input, it can be used as a power-on clear circuit if the $\overline{\text{RESET}}$ pin is connected with the VSS pin via a capacitor as shown in Figure 3.12.3. A reset switch must be provided (see Figure 3.12.3) to obtain an assured reset effect at power-on without being influenced by possible power. This is especially important for a reset operation without the use of the OSC3 oscillation circuit, in which case the system clock (OSC1) must be ON before the system is released from the reset state.

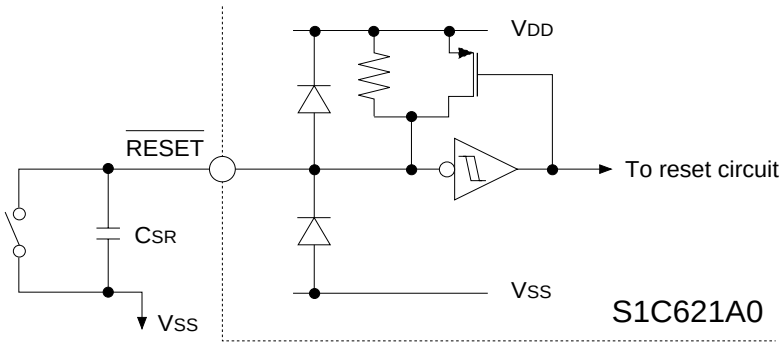


Fig. 3.12.3
Power-on Clear Circuit

Oscillation detection circuit

With $\overline{RESET}$ = 1, the oscillation detection circuit receives f_{OSC1} and makes a f-V conversion. If the OSC1 frequency is greater than a certain value, the oscillation output goes to 0 to clear the reset state. The time required for f-V conversion depends on f_{OSC1}, and is several milliseconds with f_{OSC1} = 32 kHz. This time gives a delay for clearing the reset state from the $\overline{RESET}$ input going to 1.

If f_{OSC1} stops for some reason, the oscillation detection circuit triggers a reset operation.

Watchdog timer

The watchdog timer guards the CPU against an unexpected overrun. It uses the OSC1 clock as the source oscillation frequency to perform the increment operation. If the watchdog timer fails to be reset in 3–4 seconds with f_{OSC1} = 32 kHz, the CPU will be initialized at initial reset.

To reset the watchdog timer, write 1 in the 0F1H, D3 bit (WDRST) on the memory space. Writing 0 in the bit causes no reset.

• WDRST (0F1H, D3)

Write 1:	Reset and restart watchdog timer
Write 0:	No operation
Read:	Always 0

A program overrun can be detected by programming the watchdog timer to be reset in the main routine.

The watchdog timer continues counting even with the CPU in the halt state. If the CPU stays in the halt state for 3–4 seconds, the watchdog timer triggers the CPU to be reset.

The watchdog timer function can be nullified by using the mask option.

$\overline{\text{RSTOUT}}$

The $\overline{\text{RSTOUT}}$ pin delivers all internal reset operations (triggered by $\overline{\text{RESET}}$, oscillation detector, and watchdog timer signals) to outside the system.

Initialization by initial reset

When the S1C621A0 is initially reset, its internal registers are set as follows:

• CPU core

Table 3.12.1
CPU Core Initialization

CPU core			
Internal circuits		Bit length	Status
Program counter step	PCS	8	0
Program counter page	PCP	4	1
New page pointer	NPP	4	1
Stack pointer	SP	8	Undefined
Index register	X	8	Undefined
Index register	Y	8	Undefined
Register pointer	RP	4	Undefined
General register	A	4	Undefined
General register	B	4	Undefined
Interrupt flag	I	1	0
Decimal flag	D	1	0
Zero flag	Z	1	Undefined
Carry flag	C	1	Undefined

• Peripheral circuits

Table 3.12.2
Peripheral Circuits Initialization

Peripheral circuits		
Internal circuits	Bit length	Status
RAM data	4 × 208	Undefined
Segment data	4 × 32	Undefined
Other peripheral circuits	See Tables 3.1.1(a)–(d)	

3.13 Test Input Pin ($\overline{\text{TEST}}$)

This pin is used when the S1C621A0 is shipped from the factory. It must be kept connected to VDD throughout normal operation mode.

3.14 Lower Current Dissipation

The S1C621A0 contains a control register for each circuit block to realize lower current consumption. The registers are programmed so as to operate each circuit with a minimum current. For reference in programming, the following table summarizes the circuits that can be controlled for lower current consumption and the in associated registers:

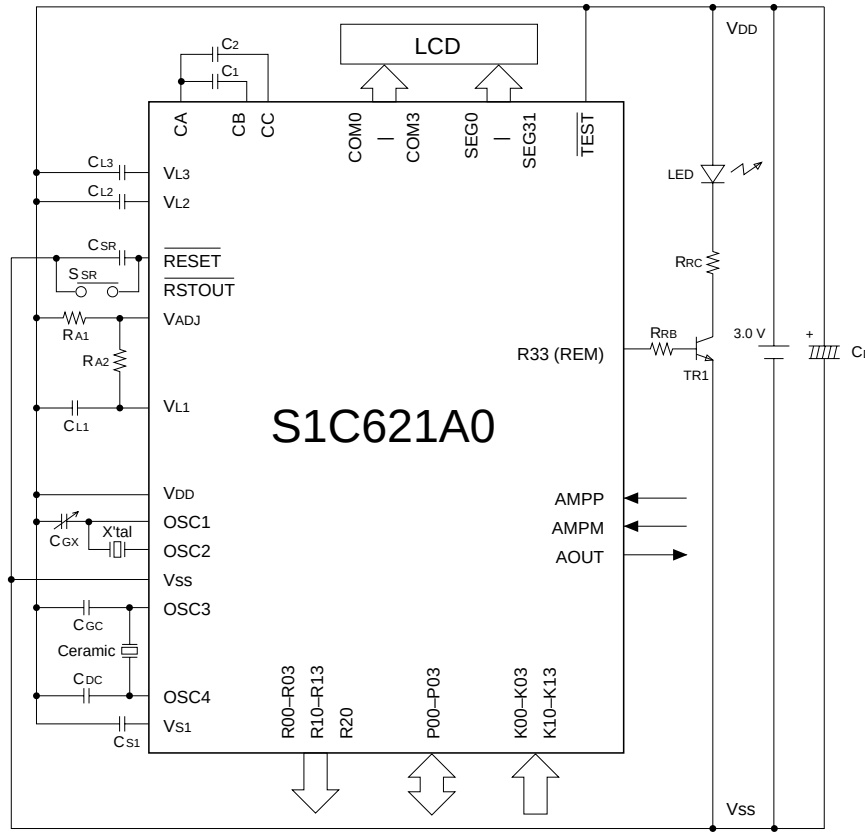
Table 3.14.1
Order of Current Consumption

Circuit block	Control register	Order of current consumption
CPU	HALT instruction	See section 5 "ELECTRICAL CHARACTERISTICS"
CPU operating frequency	CLKCHG	
Ceramic (CR) oscillation circuit	OSCC	Several tens µA
REM circuit	REMC	Several µA (in OSC3 mode) Several hundreds nA (OSC3 not used, option used)
Timer	TMRUN	Several hundreds nA
SVD circuit	SVDON	Several tens µA
AMP circuit	OPON	Several tens µA

At initial setting with the CPU in operation mode, the CPU is ready with OSC3 clock (CLKCHG = 0, in high-speed mode), the ceramic (CR) oscillation circuit is ON (OSCC = 1), the REM circuit is ON (REMC = 1), the timer is OFF (TMRUN = 0), the SVD circuit is OFF (SVDON = 0), and the AMP circuit is OFF (OPON = 0).

It should be noted that various factors affecting current consumption. For example, a system in which a resistor is connected to the V_{ADJ} pin to control the LCD power (V_{L1}) differ from a system in which the V_{ADJ} pin is shorted to V_{L1}. Also, characteristics of the LCD panel used will produce a difference in power consumption to the order of several micro-amperes.

CHAPTER 4 TYPICAL EXTERNAL CONNECTION



X'tal	Crystal oscillator	32.768 kHz CI (max) = 35 kΩ
C_{GX}	Trimmer capacitor	5–25 pF
C_R	Ceramic oscillator	455 kHz
C_{GC}		100 pF
C_{DC}		100 pF
C_{SR}		0.33 µF
R_{A1}		2 MΩ
R_{A2}		1 MΩ
C_{L1}		0.1 µF
C_{L2}		0.1 µF
C_{L3}		0.1 µF
C_1		0.1 µF
C_2		0.1 µF
C_{S1}		0.2 µF
C_D		6.8 µF

Fig. 4.1
Typical External Connection

Note The values given above are examples only, and do not guarantee system operation.

CHAPTER 5 ELECTRICAL CHARACTERISTICS

5.1 Absolute Maximum Ratings

(V_{DD} = 0 V)

Parameter	Symbol	Rated values	Unit
Supply voltage	V_{SS}	-5.2 to 0.5	V
Input pin voltage	V_I	V_{SS} -0.3 to 0.3	V
	V_{IOSC}	V_{S1} -0.3 to 0.3	V
Operating temperature	Topr	-20 to 70	°C
Storage temperature	Tstg	-65 to 150	°C
Soldering temperature and time (Leads)	Tsol	260°C, 10 sec (at lead)	–
Allowable power dissipation *	P_D	250	mW

* 80-pin plastic package only.

5.2 Recommended Operating Conditions

(Ta = -20 to 70°C)

Parameter	Symbol	Condition	Min	Typ	Max	Unit
Supply voltage	V_{SS}	V_{DD} = 0 V	-3.5	-3.0	-2.2	V
Oscillation frequency	f_{OSC1}		–	32.768	–	kHz
	f_{OSC3}	duty : 50 ± 5%	50	455	600	kHz
LCD voltage output	V_{L1}		-1.6	-1.03	–	V
CR oscillation external resistor	R_{CR}		100	140	500	kΩ

5.3 DC Characteristics

(V_{DD} = 0 V, V_{SS} = -2.2 to -3.5 V, V_{L3} = -3.0 V, Ta = -20 to 70°C)

Parameter	Symbol	Condition		Min	Typ	Max	Unit
High level input voltage (1)	V_{IH1}		K00–K13, P00–P03	0.2V_{SS}		0	V
Low level input voltage (1)	V_{IL1}		K00–K13, P00–P03	V_{SS}		0.8V_{SS}	V
High level input voltage (2)	V_{IH2}		$\overline{\text{RESET}}$	0.1V_{SS}		0	V
Low level input voltage (2)	V_{IL2}		$\overline{\text{RESET}}$	V_{SS}		0.9V_{SS}	V
High level input current	I_{IH}	V_{IH} = V_{DD}				1	µA
Low level input current	I_{IL1}	V_{IL1} = V_{SS}	No pull up, K00–K13	-1			µA
	I_{IL2}	V_{IL2} = V_{SS}	Pull up, K00–K13	-5		-0.35	µA
	I_{IL3}	V_{IL3} = V_{SS}	$\overline{\text{RESET}}$ pin	-5		-0.35	µA
	I_{IL4}	V_{IL4} = 0.2V_{SS}	Pull up, K00–K13	-30			µA
	I_{IL5}	V_{IL5} = 0.2V_{SS}	$\overline{\text{RESET}}$ pin	-40			µA
	I_{IL6}	V_{IL6} = V_{SS}	P00–P03 *	-15		-2	µA
High level output current (1)	I_{OH1}	V_{OH1} = 0.1V_{SS}	R00–R03, R10–R13, $\overline{\text{RSTOUT}}$			-250	µA
Low level output current (1)	I_{OL1}	V_{OL1} = 0.9V_{SS}	R00–R03, R10–R13, $\overline{\text{RSTOUT}}$	1.0			mA
High level output current (2)	I_{OH2}	V_{OH2} = 0.1V_{SS}	R20			-1.8	mA
Low level output current (2)	I_{OL2}	V_{OL2} = 0.9V_{SS}	R20	1.0			mA
High level output current (3)	I_{OH3}	V_{OH3} = 0.1V_{SS}	P00–P03			-250	µA
Low level output current (3)	I_{OL3}	V_{OL3} = 0.9V_{SS}	P00–P03	1.0			mA
High level output current (4)	I_{OH4}	V_{OH4} = 0.1V_{SS}	R33 (REM)			-1.8	mA
Low level output current (4)	I_{OL4}	V_{OL4} = 0.9V_{SS}	R33 (REM)	1.0			mA
Common output current	I_{OH5}	V_{OH5} = -0.05V (COM0–COM3)				-3.0	µA
	I_{OL5}	V_{OL5} = V_{L3} + 0.05V		3.0			µA
Segment output current (in LCD output mode)	I_{OH6}	V_{OH6} = -0.05V (SEG0–SEG31)				-3.0	µA
	I_{OL6}	V_{OL6} = V_{L3} + 0.05V		3.0			µA
Segment output current (in DC output mode)	I_{OH7}	V_{OH7} = 0.1V_{SS} (SEG0–SEG31)				-50	µA
	I_{OL7}	V_{OL7} = 0.9V_{SS}		70			µA

* Only at read cycle using internal program.

5.4 Analog Circuit Characteristics and Current Dissipation

(V_{DD} = 0 V, V_{SS} = -2.2 to -3.5 V, Ta = 25°C)

Parameter	Symbol	Condition	Min	Typ	Max	Unit
Internal voltage	V_{L1}	V_{ADJ} = V_{L1}, I_{L1} = 5 µA	-1.11	-1.03	-0.95	V
	V_{L2}	1 MΩ load connected between V_{DD} and V_{L2}	$2V_{L1}$		$2V_{L1}$ +0.1	V
	V_{L3}	1 MΩ load connected between V_{DD} and V_{L3}	$3V_{L1}$		$3V_{L1}$ +0.3	V
SVD voltage	V_{SVD}		-2.4	-2.3	-2.2	V
Response time	t_{Sd}				100	µs
AMP		V_{SS} = -3.0 V				
Maximum output voltage amplitude	V_{PP}	AMP operation			2.4	V
Through rate	SR		0.06			V/µs
Offset voltage	V_{OF}	Comparator operation [Condition Area] V_{OF} : V_I = V_{DD} - 0.9 V to Vss + 0.3 V			10	mV
Response time	t_{dA}	t_{dA} : V_{AMPP} = -1.5 V, V_{AMPM} = V_{AMPP} ± 15 mV			2.0	ms
High level output current	I_{OHA}	I_{OHA} : V_{OHA} = -3.0 V			-4.0	µA
Low level output current	I_{OLA}	I_{OLA} : V_{OLA} = -2.7 V	0.1			mA
Current consumption	I_{OP}	HALT mode OSCC = 0, V_{ADJ} = V_{L1}, No panel load		2	5	µA
		OSC1 mode *1 OSCC = 0, V_{ADJ} = V_{L1}, No panel load		9	18	µA
		OSC3 mode *1, *2 V_{ADJ} = V_{L1}, No panel load		130	250	µA

*1 The SVD circuit and analog comparator are in the OFF status.

*2 OSC3 mode: Ceramic oscillation (455 kHz) or CR oscillation (R = 140 kΩ)

5.5 Oscillation Characteristics

Oscillation characteristics are affected by various conditions (board patterns, parts used, etc.). The following values are given for reference only.

OSC1 and OSC2

Test conditions unless otherwise specified:
(VSS = -3.0 V, Crystal oscillator: Q13MC146, CG = 25 pF, CD = Built-in, Ta = 25°C)

Parameter	Symbol	Condition	Min	Typ	Max	Unit
Oscillation start time	tstal	VSS = -2.2 to -3.5 V	–	–	5	sec
Internal capacitance	CD	Package as assembled	–	25	–	pF
		Bare chip	–	24	–	pF
Frequency / voltage deviation	f / V	Vss = -2.2 to -3.5 V	–	–	5	ppm
Frequency / IC deviation	f / IC		-10	–	10	ppm
Frequency adjustment range	f / CG	CG = 5–25 pF	40	–	–	ppm
Higher-harmonic oscillation	VSS	CG = 5 pF	–	–	-3.5	V
Allowable leakage resistance		Between OSC1 pin and other pins	200	–	–	MΩ

OSC3 and OSC4

Ceramic oscillation

Test condition unless otherwise specified:
(VSS = -3.0 V, Ta = 25°C, Ceramic resonator: CSB455E*, CGC = CDC = 100 pF)

Parameter	Symbol	Condition	Min	Typ	Max	Unit
Oscillation start voltage	Vsta		-2.2	–	–	V
Oscillation start time	tsta	Vss = -2.2 to -3.5 V	–	3	–	ms

* CSB455E .. made by Murata Mfg. Co.

CR oscillation

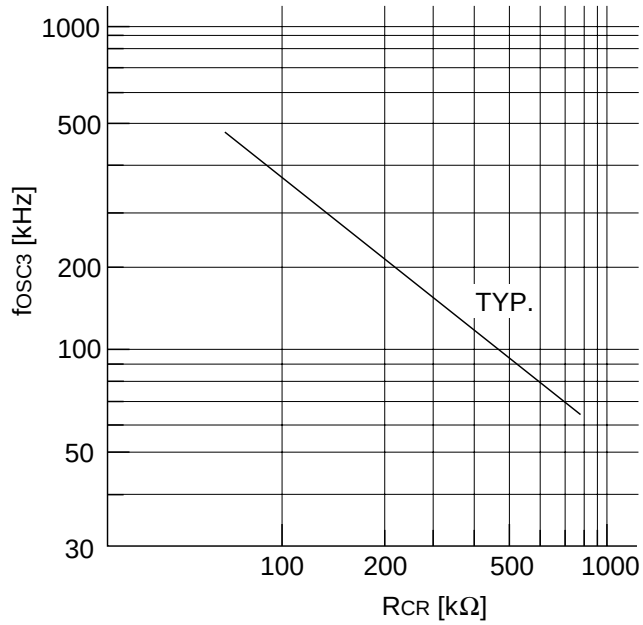
Test condition unless otherwise specified:
(V_{SS} = -3.0 V, Ta = 25°C, External resistor: R_{CR} = 140 kΩ,
Non external capacitor)

Parameter	Symbol	Condition	Min	Typ	Max	Unit
Oscillation frequency	f_{OSC3} (CR)		–	280	–	kHz
Oscillation start voltage	V_{sta3} (CR)		-2.2	–	–	V
Oscillation start time	t_{sta3} (CR)	Vss = -2.2 to -3.5 V	–	3	–	ms

• S1C621A0 oscillation characteristics — f_{OSC3} vs R_{CR} — (for reference)

Condition: Ta = 25°C, V_{DD} = GND, V_{SS} = -3.0 V,
Non board and package capacitance

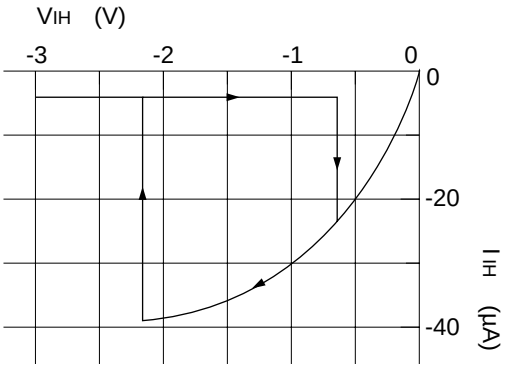
Note: Oscillation characteristics are affected by various conditions (board pattern, parts used, etc.).



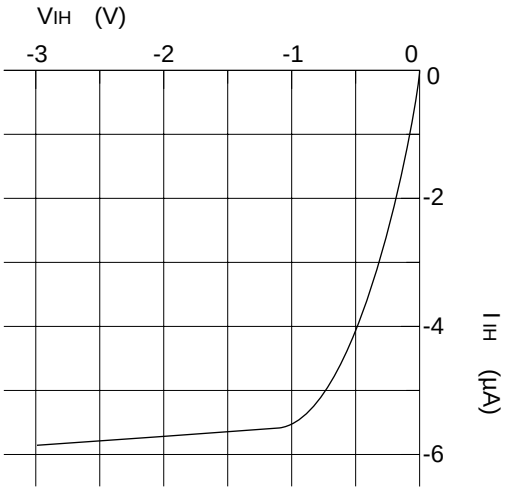
5.6 Input Current Characteristics (For Reference)

Condition: Ta = 25°C, V_{DD} = 0 V, V_{SS} = -3.0 V

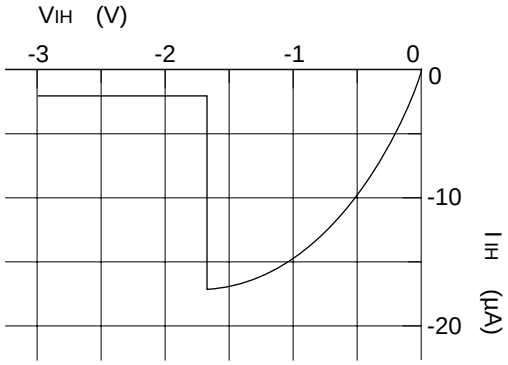
• **$\overline{\text{RESET}}$**



• **P****



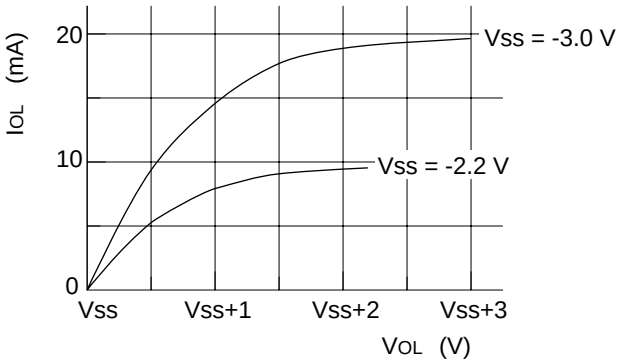
• **K****



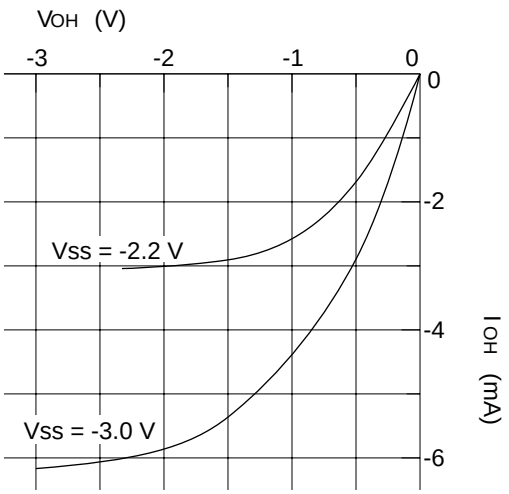
5.7 Output Current Characteristics (For Reference)

Condition: Ta = 25°C, V_{DD} = 0 V

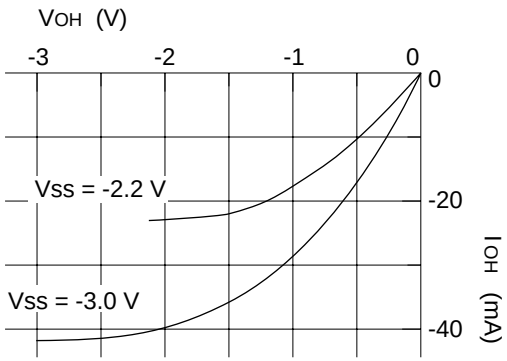
• R*, P*, RSOUT



• R0*, R1*, P*, RSOUT



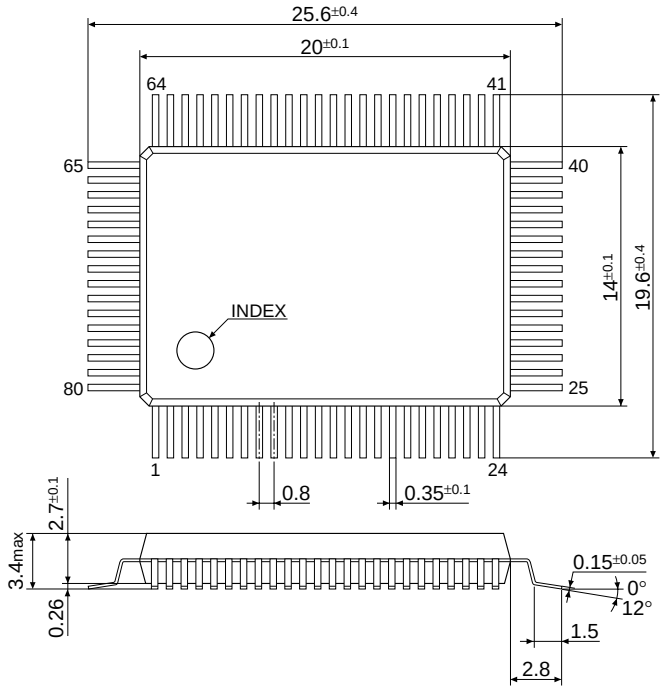
• R20, R33



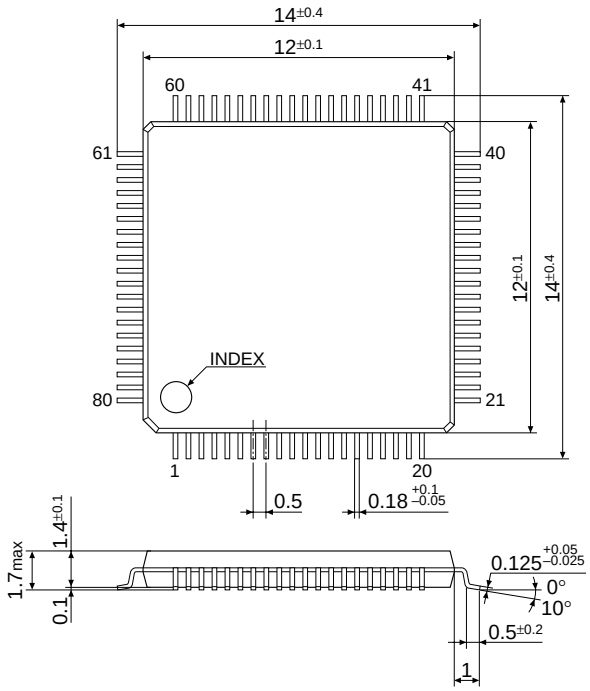
CHAPTER 6 PACKAGE

6.1 Plastic Package

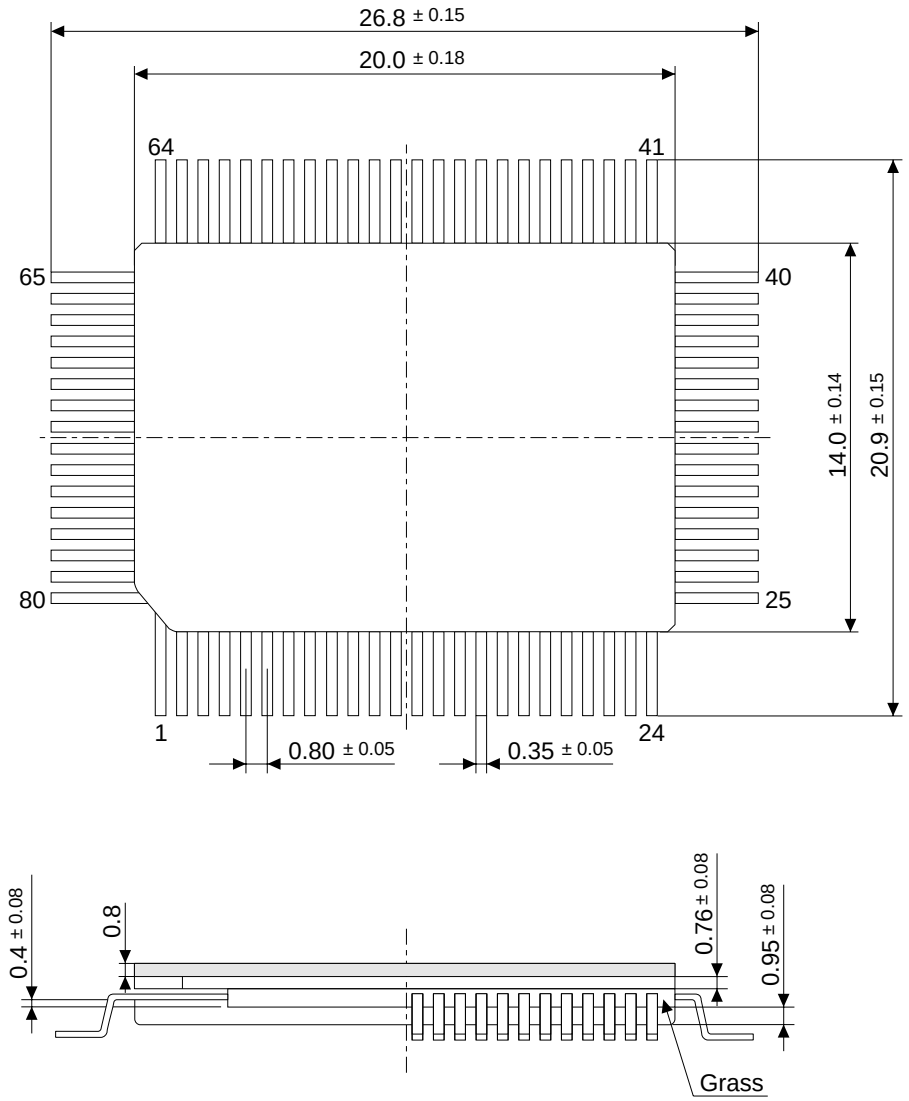
QFP5-80pin (Unit: mm)



QFP14-80pin (Unit: mm)

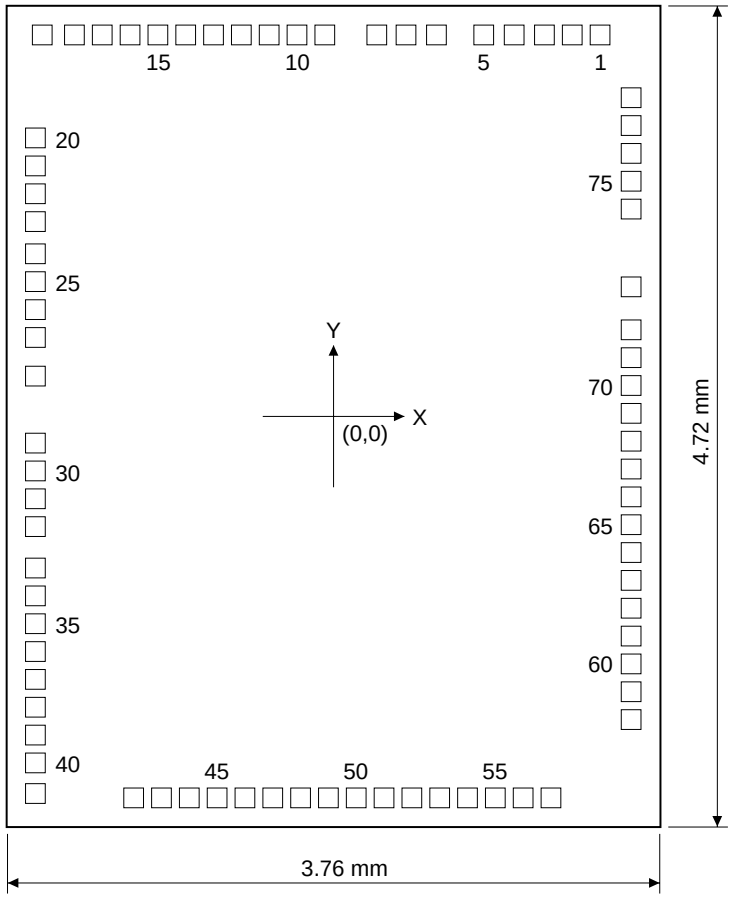


6.2 Ceramic Package for Test Sample



CHAPTER 7 PAD LAYOUT

7.1 Pad Layout Diagram



Chip thickness: 0.4 mm

Pad open size: 95 µm

7.2 Pad Coordinates

Chip size: 3.76 mm × 4.72 mm Unit: µm

Pad No.	Pad name	Coordinate		Pad No.	Pad name	Coordinate	
		X	Y			X	Y
1	CB	1,531	2,194	40	K13	-1,713	-1,993
2	CA	1,371	2,194	41	$\overline{\text{TEST}}$	-1,713	-2,169
3	V_{L3}	1,211	2,194	42	SEG31	-1,148	-2,194
4	V_{L2}	1,037	2,194	43	SEG30	-988	-2,194
5	$\overline{\text{RESET}}$	862	2,194	44	SEG29	-828	-2,194
6	V_{ADJ}	590	2,194	45	SEG28	-668	-2,194
7	V_{L1}	415	2,194	46	SEG27	-508	-2,194
8	R33(REM)	247	2,194	47	SEG26	-348	-2,194
9	$\overline{\text{RSTOUT}}$	-52	2,194	48	SEG25	-188	-2,194
10	AMPP	-212	2,194	49	SEG24	-28	-2,194
11	AMPM	-372	2,194	50	SEG23	132	-2,194
12	AOUT	-532	2,194	51	SEG22	292	-2,194
13	V_{DD}	-692	2,194	52	SEG21	452	-2,194
14	OSC1	-852	2,194	53	SEG20	612	-2,194
15	OSC2	-1,012	2,194	54	SEG19	772	-2,194
16	V_{SS}	-1,172	2,194	55	SEG18	932	-2,194
17	OSC3	-1,332	2,194	56	SEG17	1,092	-2,194
18	OSC4	-1,492	2,194	57	SEG16	1,252	-2,194
19	V_{S1}	-1,677	2,194	58	SEG15	1,713	-1,744
20	R00	-1,713	1,599	59	SEG14	1,713	-1,584
21	R01	-1,713	1,438	60	SEG13	1,713	-1,424
22	R02	-1,713	1,278	61	SEG12	1,713	-1,264
23	R03	-1,713	1,118	62	SEG11	1,713	-1,104
24	R10	-1,713	933	63	SEG10	1,713	-944
25	R11	-1,713	773	64	SEG9	1,713	-784
26	R12	-1,713	612	65	SEG8	1,713	-624
27	R13	-1,713	452	66	SEG7	1,713	-464
28	R20	-1,713	230	67	SEG6	1,713	-304
29	P00	-1,713	-155	68	SEG5	1,713	-144
30	P01	-1,713	-315	69	SEG4	1,713	16
31	P02	-1,713	-475	70	SEG3	1,713	176
32	P03	-1,713	-635	71	SEG2	1,713	336
33	K00	-1,713	-873	72	SEG1	1,713	496
34	K01	-1,713	-1,033	73	SEG0	1,713	743
35	K02	-1,713	-1,193	74	COM3	1,713	1,190
36	K03	-1,713	-1,353	75	COM2	1,713	1,350
37	K10	-1,713	-1,513	76	COM1	1,713	1,510
38	K11	-1,713	-1,673	77	COM0	1,713	1,670
39	K12	-1,713	-1,833	78	CC	1,713	1,830

CHAPTER 8 DIFFERENCES BETWEEN S1C621A0 AND S1C6214 (previous number, discontinued product)

8.1 Description

S1C621A0 is a micro controller which is an improved version of S1C6214. Since the basic functions, memory maps, and other basic aspects have not been changed, in most cases the program which uses the conventional S1C6214 may be used without modifications. Note, however, that since there are slight changes in some functions and characteristics, be sure to check the program used with S1C6214 as well as the external circuit before using. Also, in S1C621A0, pin configuration of package has changed from S1C6214. Refer to the "1.3 Pin Configuration" for details. The differences between S1C621A0 and S1C6214 are listed below:

8.2 Differences

Differences in function

S1C621A0 uses an improved CPU Core "S1C6200A". Its difference with S1C6214 in terms of functions lies only in this aspect.

Table 8.2.1 Functional Differences between S1C621A0 and S1C6214

Function	S1C621A0	S1C6214
Value of D (decimal) flag setting at initial reset	0	Undefined
Writing on the interrupt mask register at EI (enable interrupt flag)	Possible	Not possible
Reading the interrupt factor flag at EI (enable interrupt flag)	Possible	Not possible

In addition, after issuing an interrupt request, the time until the completion of interrupt processing by hardware has slightly changed. For details, refer to the "S1C6200/6200A Core CPU Manual".

(1) Writing on the interrupt mask register at EI

The operation during the instruction execution for writing "0" (i.e., to mask the interrupt factor) on the interrupt mask register at EI is shown in Figure 8.2.1. At this point, the interrupt is masked 0.5 clock before the start of the instruction execution through the 0.5 clock advance operation. Moreover, during the instruction execution for writing "1" (i.e., to cancel the interrupt mask) on the mask register at EI, it is the same as the ordinary interrupt timing as shown in Figure 8.2.1. In other words, if the interrupt factor flag value is set to "1", the interrupt processing by hardware will start in the next instruction execution cycle 0.5 clock before the completion of the instruction execution.

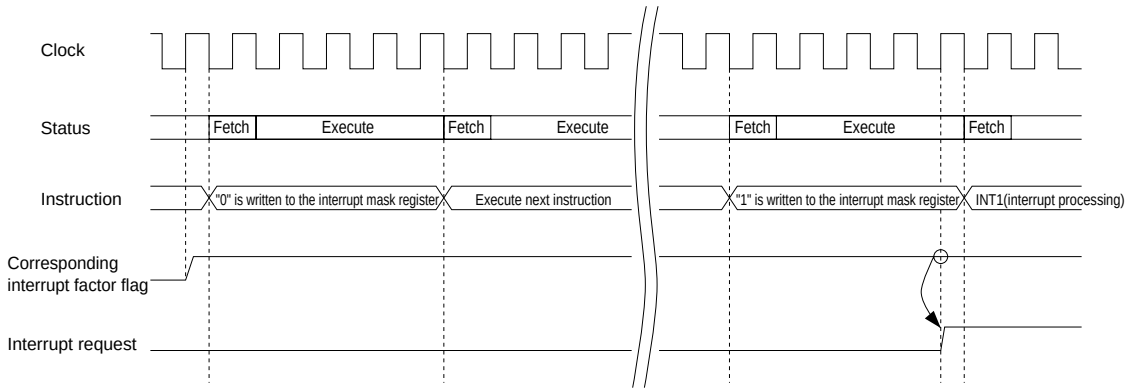


Fig. 8.2.1
Writing on the Interrupt Mask Register and Interrupt Request Generation

(2) Reading the interrupt factor flag at EI

At EI, reading the interrupt factor flag is possible but caution must be observed in the following case: when the value of the interrupt mask register corresponding to the interrupt factor flag which is to be read is set to "1" (unmasked). In this case, interrupt request may be issued to the CPU due to the timing by which the interrupt factor flag is set to "1", or the interrupt factor flag may be cleared by reading it and hence interrupt request will not be issued.
Particularly when there are multiple interrupt factor flags in the same address, extra caution is required.

Quasi-functional difference

Although it is actually a difference in characteristics, the following may be cited as quasi functional difference:

Table 8.2.2 Quasi Functional Difference between S1C621A0 and S1C6214

Quasi function	S1C621A0	S1C6214
AMP circuit: comparator operation input voltage condition (V_I)	$V_I = V_{DD}-0.9 \sim V_{SS}+0.3V$	$V_I = V_{DD}-1.1 \sim V_{SS}+0.1V$

Differences in characteristics

Differences in characteristics between S1C621A0 and S1C6214 are summarized in Table 8.2.3. For details, refer to their respective hardware manuals.

Table 8.2.3 Differences in Characteristics between S1C621A0 and S1C6214

Characteristics			S1C621A0	S1C6214
Absolute maximum rating	Operating temperature (Topr)		-20~70°C	-20~60°C
Recommended operating condition	Oscillation frequency (f_{OSC3}) MAX.		600 kHz	500 kHz
DC characteristics	High level output current (3) P00~P03		-250 µA	-100 µA
	Low level output current (3) P00~P03		1 mA	140 µA
	High level output current (4) R33		-1.8 mA (V_{OH4}=0.1Vss)	-1.0 mA (V_{OH4}=0.5Vss)
	Low level input current RESET		-40 µA	-30 µA
Analog circuit characteristics	AMP through rate (SR)		0.06 V/µsec	0.004 V/µsec
	High level output current (I_{OHA})		-4 µA	-0.5 µA
Current dissipation	HALT mode	TYP.	2 µA	3 µA
	OSCC = 0	MAX.	5 µA	7.5 µA
	OSC1 mode	TYP.	9 µA	13 µA
	OSCC = 0	MAX.	18 µA	23 µA
	OSC3 mode	TYP.	130 µA	150 µA
		MAX.	250 µA	300 µA
	AMP circuit	TYP.	35 µA	Several µA
Analog circuit	Internal voltage (V_{L1})	MIN.	-1.11 V	-1.13 V
		TYP.	-1.03 V	-1.05 V
		MAX.	-0.95 V	-0.98 V
OSC1	Internal capacitance (C_{DX})	Package	25 pF	22 pF
		Chip	24 pF	21 pF

II. S1C621A0 Technical Software

CONTENTS

Software

CHAPTER 1 INITIAL RESET

1.1 Internal Register Status at Initial Reset

At initial reset, the internal registers of the S1C621A0 are initialized as shown in Tables 1.1.1(a)–(b) and 1.1.2(a)–(d).

Table 1.1.1(a) Internal Status at Initial Reset (CPU Core)

CPU core			
Internal circuits		Bit length	Status
Program counter step	PCS	8	0
Program counter page	PCP	4	1
New page pointer	NPP	4	1
Stack pointer	SP	8	Undefined
Index register	X	8	Undefined
Index register	Y	8	Undefined
Register pointer	RP	4	Undefined
General register	A	4	Undefined
General register	B	4	Undefined
Interrupt flag	I	1	0
Decimal flag	D	1	0
Zero flag	Z	1	Undefined
Carry flag	C	1	Undefined

Table 1.1.1(b) Internal Status at Initial Reset (Peripheral Circuit)

Peripheral circuits		
Internal circuits	Bit length	Status
RAM data	4 × 208	Undefined
Segment data	4 × 32	Undefined
Other peripheral circuits	See Tables 1.1.2(a)–(d)	

When the internal registers are initialized as shown above program execution starts with page 1, step 0.

The internal circuits whose status is undefined must be defined by the program.

Table 1.1.2.a I/O Memory Map (0F0H–0F3H)

Address	Register D3	D2	D1	D0	Name	Init	1	0	Comment
0F0H	REMSO	IREM	IK1	IK0	REMSO	0	On	Off	Forced REM output (on/off)
	R/W	R			IREM	x	Yes	No	Interrupt factor flag (REM) Clear to 0 after read
					IK1	0	Yes	No	Interrupt factor flag (K10–K13) Clear to 0 after read
					IK0	0	Yes	No	Interrupt factor flag (K00–K03) Clear to 0 after read
0F1H	WDRST	TI2	TI8	TI32	WDRST	Clear	Clear	–	Watchdog timer reset
	W	R			TI2	0	Yes	No	Interrupt factor flag (Timer 2 Hz) Clear to 0 after read
					TI8	0	Yes	No	Interrupt factor flag (Timer 8 Hz) Clear to 0 after read
					TI32	0	Yes	No	Interrupt factor flag (Timer 32 Hz) Clear to 0 after read
0F2H	REMC	EIREM	EIK1	EIK0	REMC	1	On	Off	REM carrier generation on/off
	R/W				EIREM	0	Enable	Mask	Interrupt mask register (REM)
					EIK1	0	Enable	Mask	Interrupt mask register (K10–K13)
					EIK0	0	Enable	Mask	Interrupt mask register (K00–K03)
0F3H	TMRUN	ETI2	ETI8	ETI32	TMRUN	0	Run	Clear & Stop	Timer Run/Clear & Stop
	R/W				ETI2	0	Enable	Mask	Interrupt mask register (Timer 2 Hz)
					ETI8	0	Enable	Mask	Interrupt mask register (Timer 8 Hz)
					ETI32	0	Enable	Mask	Interrupt mask register (Timer 32 Hz)

Note: Init = value at initial reset, x = undefined, – = not set on the circuit

Table 1.1.2.b I/O Memory Map (0F4H–0F7H)

Address	Register								Comment
	D3	D2	D1	D0	Name	Init	1	0	
0F4H	TM03	TM02	TM01	TM00	TM03	0			Timer data (low-order) 16 Hz
	R				TM02	0			Timer data (low-order) 32 Hz
					TM01	0			Timer data (low-order) 64 Hz
					TM00	0			Timer data (low-order) 128 Hz
0F5H	TM13	TM12	TM11	TM10	TM13	0			Timer data (High-order) 1 Hz
	R				TM12	0			Timer data (High-order) 2 Hz
					TM11	0			Timer data (High-order) 4 Hz
					TM10	0			Timer data (High-order) 8 Hz
0F6H	SVDDT	SVDON	CLKCHG	OSCC	SVDDT	–	Normal	Low voltage	SVD data (1 when SVDON = 0)
	R	R/W			SVDON	0	On	Off	SVD circuit on/off
					CLKCHG	0	OSC1	OSC3	CLK change OSC1/OSC3
					OSCC	1	On	Off	OSC3 oscillation on/off
0F7H	RCDIV	RCDUTY	RT1	RT0	RCDIV	x			REM carrier cycle set
	R/W				RCDUTY	x			REM carrier duty set
					RT1	x			REM τ-cycle set
					RT0	x			REM τ-cycle set

Note: Init = value at initial reset, x = undefined, – = not set on the circuit

Table 1.1.2.c I/O Memory Map (0F8H–0F9H, 0FAH–0FBH)

Address	Register D3	D2	D1	D0	Name	Init	1	0	Comment
0F8H	RIC3	RIC2	RIC1	RIC0	RIC3	x			REM interrupt counter set (τ clock) (all :1 after count comleted)
	W				RIC2	x			
					RIC1	x			
					RIC0	x			
0F9H	ROUT1	ROUT0	MF91	MF90	ROUT1	0			REM output-on time set (0 τ – 3 τ)
	R/W				ROUT0	0			REM output-on time set (0 τ – 3 τ)
					MF91	x			General-purpose register (bit)
					MF90	x			General-purpose register (bit)
0FAH	K03	K02	K01	K00	K03	–	High	Low	Input port (fall : interrupt factor)
	R				K02	–	High	Low	Input port (fall : interrupt factor)
					K01	–	High	Low	Input port (fall : interrupt factor)
					K00	–	High	Low	Input port (fall : interrupt factor)
0FBH	K13	K12	K11	K10	K13	–	High	Low	Input port (fall : interrupt factor)
	R				K12	–	High	Low	Input port (fall : interrupt factor)
					K11	–	High	Low	Input port (fall : interrupt factor)
					K10	–	High	Low	Input port (fall : interrupt factor)

Note: Init = value at initial reset, x = undefined, – = not set on the circuit

Table 1.1.2.d I/O Memory Map (0FCH–0FFH)

Address	Register								Comment
	D3	D2	D1	D0	Name	Init	1	0	
0FCH	R03	R02	R01	R00	R03	1	High	Low	Output port
	R/W				R02	1	High	Low	Output port
					R01	1	High	Low	Output port
					R00	1	High	Low	Output port
0FDH	R13	R12	R11	R10	R13	1	High	Low	Output port
	R/W				R12	1	High	Low	Output port
					R11	1	High	Low	Output port
					R10	1	High	Low	Output port
0FEH	P03	P02	P01	P00	P03	1	High	Low	I/O port (used as input port after initial reset)
	R/W				P02	1	High	Low	I/O port (used as input port after initial reset)
					P01	1	High	Low	I/O port (used as input port after initial reset)
					P00	1	High	Low	I/O port (used as input port after initial reset)
0FFH	OPDT	OPON	IOC	R20	OPDT	–	High	Low	Analog comparator data (0 or 1)
	R	R/W			OPON	0	On	Off	Analog comparator on/off
					IOC	0	Out	In	I/O port control out/in
					R20	0	High	Low	Output port

Note: Init = value at initial reset, x = undefined, – = not set on the circuit

1.2 Watchdog Timer

The watchdog timer detects any unexpected malfunction in the CPU.

Watchdog timer control

The S1C621A0 contains a watchdog timer using OSC1 (32.768 kHz) as the source oscillation frequency. If the software fails to reset the timer for 3–4 seconds, it automatically provides the CPU with an initial reset signal.

Following is a description of the address of the bit used for watchdog timer control:

Table 1.2.1 I/O Memory Map

Address	Register								Comment
	D3	D2	D1	D0	Name	Init	1	0	
0F1H	WDRST	TI2	TI8	TI32	WDRST	Clear	Clear	–	Watchdog timer reset
	W	R			TI2	0	Yes	No	Interrupt factor flag (Timer 2 Hz) Clear to 0 after read
					TI8	0	Yes	No	Interrupt factor flag (Timer 8 Hz) Clear to 0 after read
					TI32	0	Yes	No	Interrupt factor flag (Timer 32 Hz) Clear to 0 after read

Note: Init = value at initial reset, – = not set on the circuit

- **WDRST:** Watchdog timer reset

  Writing 1 to this bit causes the watchdog timer to be reset. If 0 is written in the WDRST bit, the CPU goes into the 'no-operation' state.

  During the Halt state, the timer continues to operate. If the Halt state continues for 3–4 seconds, the CPU is automatically initialized and must restart from the initial reset state.

Watchdog timer control program sample

• Resetting the watchdog timer

Two program steps are needed to reset the watchdog timer:

```
        LD      Y,0F1H    ;Sets the address of the watchdog timer
        LD      MY,1000B  ;Resets the watchdog timer
```

LD instructions must be used to reset the timer because the watchdog timer reset bit (WDRST) is at the same address as the timer interrupt factor flag (TI). (If OR instructions were used, the interrupt factor flag would be cleared.)

The WDRST bit can be addressed by using the X register instead of the Y register.

CHAPTER 2 PERIPHERAL CIRCUITS

This section describes how the S1C621A0 software controls each of the peripheral circuits.

2.1 Oscillation Circuit

Oscillation circuit control

Either the OSC1 (32.768 kHz) crystal oscillation circuit or the OSC3 (455 kHz) ceramic oscillation circuit can be selected by software.
Following is a description of the address of the bits for oscillation circuit control:

Table 2.1.1 I/O Memory Map

Address	Register								Comment
	D3	D2	D1	D0	Name	Init	1	0	
0F6H	SVDDT	SVDON	CLKCHG	OSCC	SVDDT	–	Normal	Low voltage	SVD data (1 when SVDON = 0)
	R	R/W			SVDON	0	On	Off	SVD circuit on/off
					CLKCHG	0	OSC1	OSC3	CLK change OSC1/OSC3
					OSCC	1	On	Off	OSC3 oscillation on/off

Note: Init = value at initial reset, – = not set on the circuit

- **OSCC:** OSC3 oscillation circuit drive control

  The OSCC bit controls the alternative of turning the OSC3 oscillation circuit on or off. OSC3 oscillation stays on with OSCC at 1, and off with OSCC at 0. The oscillation turns on at initial reset.

  The OSC3 oscillation circuit must be kept off for power-saving purposes, except where a remote-control carrier signal must be output or the CPU operated at high speed.

- **CLKCHG:** CPU clock frequency change

  The CLKCHG bit controls the alternative of using OSC1 or OSC3 as the CPU operation clock. OSC3 is selected with CLKCHG at 0, and OSC1 with CLKCHG at 1. OSC3 is selected at initial reset.

Oscillation circuit control program samples

• Changing clock at power-on (OSC3 → OSC1):

```
        LD      Y,0F1H  ;Set watchdog timer address
        LD      A,0EH   ;┐
        LD      B,5     ;│
        LD      M3,B    ;│ Set loop counter
LOOP4   LD      M2,B    ;│
LOOP3   LD      M1,A    ;│
LOOP2   LD      M0,A    ;┘
LOOP1   LD      X,0D0H  ;Set segment memory address
LOOP0   LBPX    MX,0FFH ;┐
        ADD     A,0FH   ;│ All LCDs on
        JP      NZ,LOOP0;┘
        LD      MY,1000B;Reset watchdog timer
        DEC     M0      ;┐ Loop 1
        JP      NZ,LOOP1;┘
        DEC     M1      ;┐ Loop 2
        JP      NZ,LOOP2;┘
        DEC     M2      ;┐ Loop 3
        JP      NZ,LOOP3;┘
        DEC     M3      ;┐ Loop 4
        JP      NZ,LOOP4;┘
        LD      Y,0F6H  ;Set oscillation circuit address
①       OR      MY,0010B;Change CPU clock from OSC3 to OSC1
②       AND     MY,1110B;Turn OSC3 oscillation off
```

A time of at least 3 seconds is required until OSC1 oscillation becomes stable after a voltage has supplied. This means that when you change the CPU clock from OSC3 to OSC1, you must have a delay of at least 3 seconds after an initial reset has been invoked.

In the above sample, the CPU clock is changed about 3.2 seconds after all LCDs have turned on.

To stop OSC3 oscillation, execute steps ① to ② in the program. Using only one instruction step for changing the CPU clock from OSC3 to OSC1 and turning OSC3 oscillation off will result in a CPU malfunction.

- **Changing clock from OSC1 to OSC3:**

```
        LD      Y,0F6H      ;Set oscillation circuit address
        OR      MY,0001B    ;Turn OSC3 oscillation on
        LD      A,0         ;┐
LOOP    ADD     A,0FH       ;│ The CPU waits about 6 ms
        JP      NZ,LOOP     ;┘
        AND     MY,1101B    ;Change CPU clock from OSC1 to OSC3
```

With the 455 kHz ceramic oscillator, a time of at least 5 ms is required until the selected OSC3 oscillation becomes stable. This means that when you change the CPU clock from OSC1 to OSC3, you must have a delay of at least 5 ms after OSC3 oscillation has turned on.

In the above sample, the CPU clock is changed about 6 ms after OSC3 oscillation has turned on.

Remember, changing the CPU clock while OSC3 oscillation remains unstable will cause a CPU malfunction.

The OSCC and CLKCHG bits can be addressed by using the X register instead of the Y register.

2.2 Input Ports

Input port control

The S1C621A0 contains 8 bits of general-purpose input ports (4 bits × 2). Data of input port pins can be read on a 4-bit basis (K00–K03, K10–K13).

Following is a description of the addresses of the input ports:

Table 2.2.1 I/O Memory Map

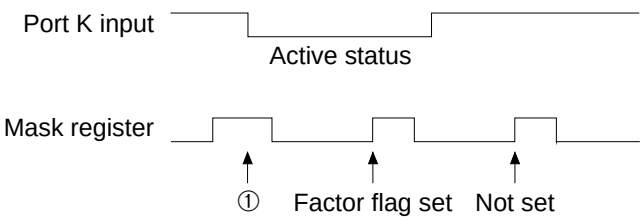
Address	Register								Comment
	D3	D2	D1	D0	Name	Init	1	0	
0FAH	K03	K02	K01	K00	K03	–	High	Low	Input port (fall : interrupt factor)
	R				K02	–	High	Low	Input port (fall : interrupt factor)
					K01	–	High	Low	Input port (fall : interrupt factor)
					K00	–	High	Low	Input port (fall : interrupt factor)
0FBH	K13	K12	K11	K10	K13	–	High	Low	Input port (fall : interrupt factor)
	R				K12	–	High	Low	Input port (fall : interrupt factor)
					K11	–	High	Low	Input port (fall : interrupt factor)
					K10	–	High	Low	Input port (fall : interrupt factor)

Note: Init = value at initial reset, – = not set on circuit

The input state of each bit is determined by the data at the corresponding address, i.e., K00–K03 on the data at 0FAH and K10–K13 on the data at 0FBH.

The input port pin can give the CPU an interrupt request by the falling edge of an input. For details, see Section 2.9, "Interrupt and Halt".

• Input interrupt programing related precautions



When the content of the mask register is rewritten, while the port K input is in the active status. The input interrupt factor flag is set at ①.

Fig. 2.2.1
Input Interrupt Timing

When using an input interrupt, if you rewrite the content of the mask register, when the value of the input terminal which becomes the interrupt input is in the active status (input terminal = low status), the factor flag for input interrupt may be set.
For example, a factor flag is set with the timing of ① shown in Figure 2.2.1. However, when clearing the content of the mask register with the input terminal kept in the low status and then setting it, the factor flag of the input interrupt is again set at the timing that has been set.
Consequently, when the input terminal is in the active status (low status), do not rewrite the mask register (clearing, then setting the mask register), so that a factor flag will only set at the falling edge in this case. When clearing, then setting the mask register, set the mask register, when the input terminal is not in the active status (high status).

Input port control program samples

• Loading K00–K03 to A register

```
LD      Y,0FAH      ;Set port address
LD      A,MY        ;A register ← K00–K03
```

Execution of these two program steps loads the data of the input port to the A register, like this:

Fig. 2.2.2
Contents of A Register

	D3	D2	D1	D0
A register	K03	K02	K01	K00

The data of the input port can be loaded to the B register or MX instead of the A register.

- **Bit-unit checking of input port pins**

```
        DI                  ;Disable interrupt
        LD      Y,0FBH      ;Set port address
INPUT1  FAN     MY,0010B    ;┐
        JP      NZ,INPUT1   ;┘ Loop until K11 turns to 0
INPUT2  FAN     MY,0010B    ;┐
        JP      Z,INPUT2    ;┘ Loop until K11 turns to 1
```

This program is looped until a rising edge is input to the K11 input port pin.

The input port can be addressed by using the X register instead of the Y register.

Note *When the input port is changed from low level to high level through a pull-up resistor, the wave rises with a certain delay caused by the time constants of the pull-up resistance and the input gate capacitance. It is therefore necessary to provide a proper waiting time before the input port data is read.*

2.3 Output Ports

Output port control

The S1C621A0 contains 9 bits of general-purpose output ports (4 bits × 2 + 1). The output port is a read/write register; its pins provide the contents of the register.

Following is a description of the address of each output port.

Table 2.3.1 I/O Memory Map

Address	Register								Comment
	D3	D2	D1	D0	Name	Init	1	0	
0FCH	R03	R02	R01	R00	R03	1	High	Low	Output port
	R/W				R02	1	High	Low	Output port
					R01	1	High	Low	Output port
					R00	1	High	Low	Output port
0FDH	R13	R12	R11	R10	R13	1	High	Low	Output port
	R/W				R12	1	High	Low	Output port
					R11	1	High	Low	Output port
					R10	1	High	Low	Output port
0FFH	OPDT	OPON	IOC	R20	OPDT	–	High	Low	Analog comparator data (0 or 1)
	R	R/W			OPON	0	On	Off	Analog comparator on/off
					IOC	0	Out	In	I/O port control out/in
					R20	0	High	Low	Output port

Note: Init = value at initial reset, – = not set on the circuit

The output state of each bit is determined by the data at the corresponding address, i.e., R00–R03 depends on the data at 0FCH, R10–R13 on the data at 0FDH, and bit R20 on the data at 0FFH, D3.

At initial reset, bits R00–R13 are set to 1 and R20 to 0.

Output port control program samples

• Loading data of B register to R10–R13

```
LD      Y,0FDH    ;Set port address
LD      MY,B      ;R10–R13 ← B register
```

Execution of these two program steps loads the data of the B register to the output port pins, like this:

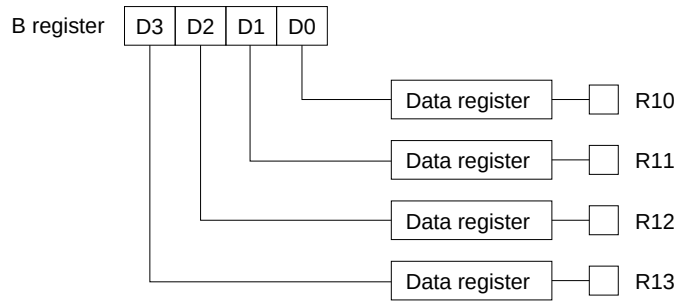


Fig. 2.3.1
Correspondence of Write Data and Output Port

The output data can be taken from the A register, MX, or immediate data instead of the B register.

• Bit-unit operation of output port pins

```
LD      Y,0FCH    ;Set port address
OR      MY,0010B  ;R01 ← 1
AND     MY,1011B  ;R02 ← 0
```

Execution of this program causes the output port pins to be set as follows:

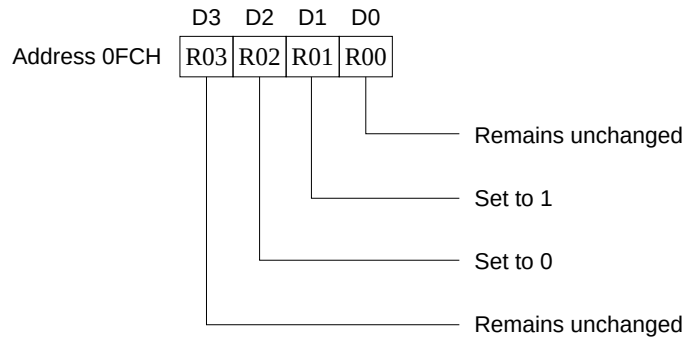


Fig. 2.3.2
Output Result

- **FOUT output**

  When FOUT is selected in R12 by using a mask option, its output turns on if 0 is set in the register and to high level DC if 1 is set in the register.

```
        LD      Y,0FDH    ;Set port address
        AND     MY,1011B  ;FOUT on
        LD      X,COUNT   ;┐
        LBPX    MX,5BH    ;│
        LD      B,1       ;│
LOOP    LD      X,COUNT   ;│ Wait about 100 ms
        SUB     MX,B      ;│ (in the 32.768 kHz clock mode)
        LDPX    A,MX      ;│
        SBC     MX,0      ;│
        OR      A,MX      ;│
        JP      NZ,LOOP   ;┘
        OR      MY,0100B  ;FOUT off
```

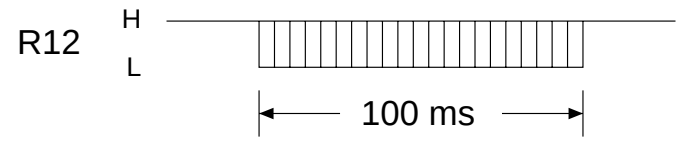


Fig. 2.3.3
FOUT Waveform

In the above program example, the FOUT output stays on for about 100 ms as shown in Figure 2.3.3. The output port can be addressed by using the X register instead of the Y register.

Note *When FOUT turns on or off asynchronously of timing signal controlled by software, its waveform contains "hazard". Keep this in mind when using FOUT as the source oscillation for any other device.*

2.4 I/O Ports

I/O port control

The S1C621A0 contains 4 bits of general-purpose I/O port (4 bits × 1). The contents of the I/O control register determines whether the port is used as input or output. The port works as input with the register set at 0, and output with the register set at 1.

Following is a description of the address of the I/O port.

Table 2.4.1 I/O Memory Map

Address	Register								Comment
	D3	D2	D1	D0	Name	Init	1	0	
0FEH	P03	P02	P01	P00	P03	1	High	Low	I/O port (used as input port after initial reset)
	R/W				P02	1	High	Low	I/O port (used as input port after initial reset)
					P01	1	High	Low	I/O port (used as input port after initial reset)
					P00	1	High	Low	I/O port (used as input port after initial reset)
0FFH	OPDT	OPON	IOC	R20	OPDT	–	High	Low	Analog comparator data (0 or 1)
	R	R/W			OPON	0	On	Off	Analog comparator on/off
					IOC	0	Out	In	I/O port control out/in
					R20	0	High	Low	Output port

Note: Init = value at initial reset, – = not set on the circuit

• Use as input port:

The I/O port works as input with the I/O control register (address 0FFH, D1) set at 0. The input state of the I/O port pins (P00–P03) is determined by reading the data at address 0FEH.

• Use as output port:

The I/O port works as output with the I/O control register set at 1. The output state of the I/O port pins is determined by the data at address 0FEH. The data is held by the register, and can be set independently of the contents of the I/O control register. (Out put data can be set even with the port set in the input mode.)

At initial reset, the I/O control register is set to 0. This means that the I/O port turns to the input mode and all data registers are set to 1.

I/O port control program samples

• Loading P00–P03 input data to A register

```
LD      Y,0FFH      ;Set I/O control register address
AND     MY,1101B    ;Set port to input mode
LD      Y,0FEH      ;Set port address
LD      A,MY        ;A register ← P00–P03
```

Execution of these four program steps loads the data of the I/O port pins to the A register, like this:

Fig. 2.4.1
Correspondence of I/O Port (Input) and A Register

	D3	D2	D1	D0
A register	P03	P02	P01	P00

• Loading P00–P03 output data to A register

```
LD      Y,0FFH      ;Set I/O control register address
OR      MY,0010B    ;Set port to output mode
LD      Y,0FEH      ;Set port address
LD      A,MY        ;A register ← P00–P03 (register output)
```

Execution of these four program steps loads the contents of the data register of the I/O port to the A register, like this:

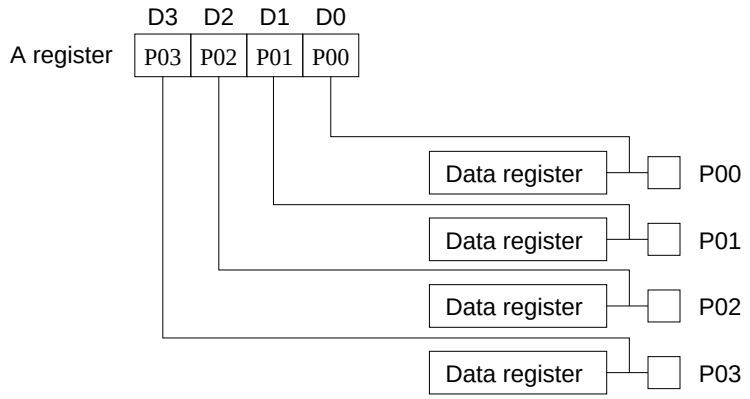


Fig. 2.4.2
Correspondence of I/O Port
(Output) and A Register

Data can be loaded from the I/O port to the B register or MX instead of the A register.

• Loading contents of B register to P00–P03

```
LD      Y,0FFH      ;Set I/O control register address
OR      MY,0010B    ;Set port to output mode
LD      Y,0FEH      ;Set port address
LD      MY,B        ;P00–P03 ← B register
```

Execution of these four program steps loads the data of the B register to the I/O port pins, like this:

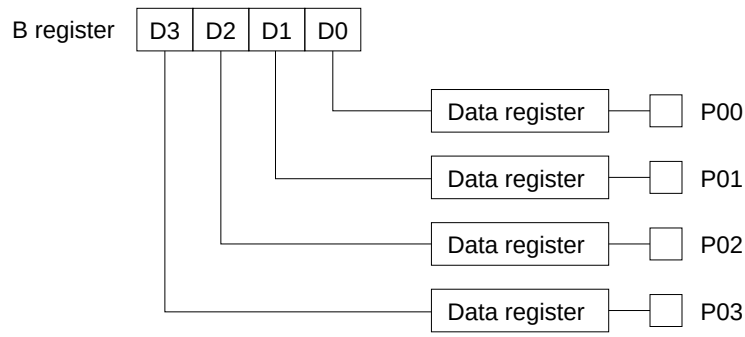


Fig. 2.4.3
Correspondence of I/O Port (Output) and B Register

The output data can be taken from the A register, MX, or immediate data instead of the B register.

The I/O port can be addressed by using the X register instead of the Y register.

Bit-unit operation of the I/O port is identical to that of the input ports (K00–K03, K10–K13) or output ports (R00–R03, R10–R13, R20).

Note *The pull-up operation of the I/O port is turned on by the internal memory read signal. Remember, this will cause data to be taken in during a transient pull-up condition if OSC3 stays on.*

2.5 LCD Driver

LCD driver control

The S1C621A0 contains LCD drivers that can drive up to 128 segments. Each segment memory can be assigned to any bit at addresses 0D0H to 0EFH by using a mask option. The LCD segment is on with 1 set in the segment memory, and off with 0 set in the segment memory. Note that the segment memory is a write-only memory.

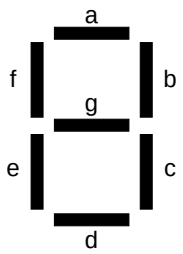


Fig. 2.5.1
Segment Memory Assignment

Address	Data			
	D3	D2	D1	D0
0D0H	d	c	b	a
0D1H		g	f	e

In the assignment shown above, display of 7 segments is controlled by writing data at segment memory addresses 0D0H and 0D1H.

LCD driver control program samples

• **Displaying 7 segments**

```
        ORG     000H
        RETD    3FH         ;Display "0"
        RETD    06H         ;Display "1"
        RETD    5BH         ;Display "2"
        RETD    4FH         ;Display "3"
        RETD    66H         ;Display "4"
        RETD    6DH         ;Display "5"
        RETD    7DH         ;Display "6"
        RETD    27H         ;Display "7"
        RETD    7FH         ;Display "8"
        RETD    6FH         ;Display "9"
;
SEVENS  LD      B,0         ;Set jump address
        LD      X,0D0H      ;Set segment memory address
        JPBA
```

When this routine (the above example) is called (by the CALL or CALZ instruction) with any number from 0 to 9 set in the A register for the assignment of Figure 2.5.1, seven segments are displayed according to the contents of the A register:

Table 2.5.1
Diagram of Characters

ACC	Display	ACC	Display
0	0	5	5
1	1	6	6
2	2	7	7
3	3	8	8
4	4	9	9

The RETD instruction can be used to write data to the segment memory only if it is addressed by using the X register. (Addressing by using the Y register is invalid.)

Note that the stack pointer must be set to a proper value before the CALL (CALZ) instruction is executed.

• Bit-unit operation of segment memory

Address	Data			
	D3	D2	D1	D0
0E0H			▲	●

SEG–A

SEG–B

Fig. 2.5.2
Segment Memory Assignment

```
LD      X,SEGBUF  ;Set segment memory buffer address
LD      Y,0E0H    ;Set segment memory address
LD      MX,3      ;Set buffer data
LD      MY,MX     ;SEG-A and B on (○, △)
AND     MX,1110B  ;Change buffer data
LD      MY,MX     ;SEG-A off (●, △)
AND     MX,1101B  ;Change buffer data
LD      MY,MX     ;SEG-B off (●, ▲)
```

For manipulation of the segment memory in bit-units for the assignment of Figure 2.5.2, a buffer must be provided in RAM to hold data. Note that, since the segment memory is a write-only memory, data cannot be changed directly by using an ALU instruction (for example, AND or OR).

2.6 Timer

Timer control

The S1C621A0 contains a timer using OSC1 (32.768 kHz) as the source oscillation frequency. It is an 8-bit binary counter, the data of which can be read in 4-bit units (the high-order 4 bits and the low-order 4 bits).

Following is a description of the address of the timer:

Table 2.6.1 I/O Memory Map

Address	Register D3	D2	D1	D0	Name	Init	1	0	Comment
0F3H	TMRUN	ETI2	ETI8	ETI32	TMRUN	0	Run	Clear & Stop	Timer Run/Clear & Stop
	R/W				ETI2	0	Enable	Mask	Interrupt mask register (Timer 2 Hz)
					ETI8	0	Enable	Mask	Interrupt mask register (Timer 8 Hz)
					ETI32	0	Enable	Mask	Interrupt mask register (Timer 32 Hz)
0F4H	TM03	TM02	TM01	TM00	TM03	0			Timer data (low-order) 16 Hz
	R				TM02	0			Timer data (low-order) 32 Hz
					TM01	0			Timer data (low-order) 64 Hz
					TM00	0			Timer data (low-order) 128 Hz
0F5H	TM13	TM12	TM11	TM10	TM13	0			Timer data (High-order) 1 Hz
	R				TM12	0			Timer data (High-order) 2 Hz
					TM11	0			Timer data (High-order) 4 Hz
					TM10	0			Timer data (High-order) 8 Hz

Note: Init = value at initial reset

The state of the timer is determined by the low-order 4-bit data at address 0F4H and the high-order 4-bit data at address 0F5H. With 32.768 kHz used at OSC1, each bit is assigned a frequency as specified in Table 2.6.1.

The timer can be reset and stopped by setting 0 at 0F3H, D3 (TMRUN). With 1 set at this address, the timer is counted up. At initial reset, the TMRUN bit is set to 0 and the timer cleared.

The timer can invoke an interrupt factor to the CPU at the falling edge of a 32 Hz, 8 Hz or 2 Hz signal. For details, see Section 2.9, "Interrupt and Halt".

Figure 2.6.1 shows the timing of each bit of the timer and the timing of each interrupt request.

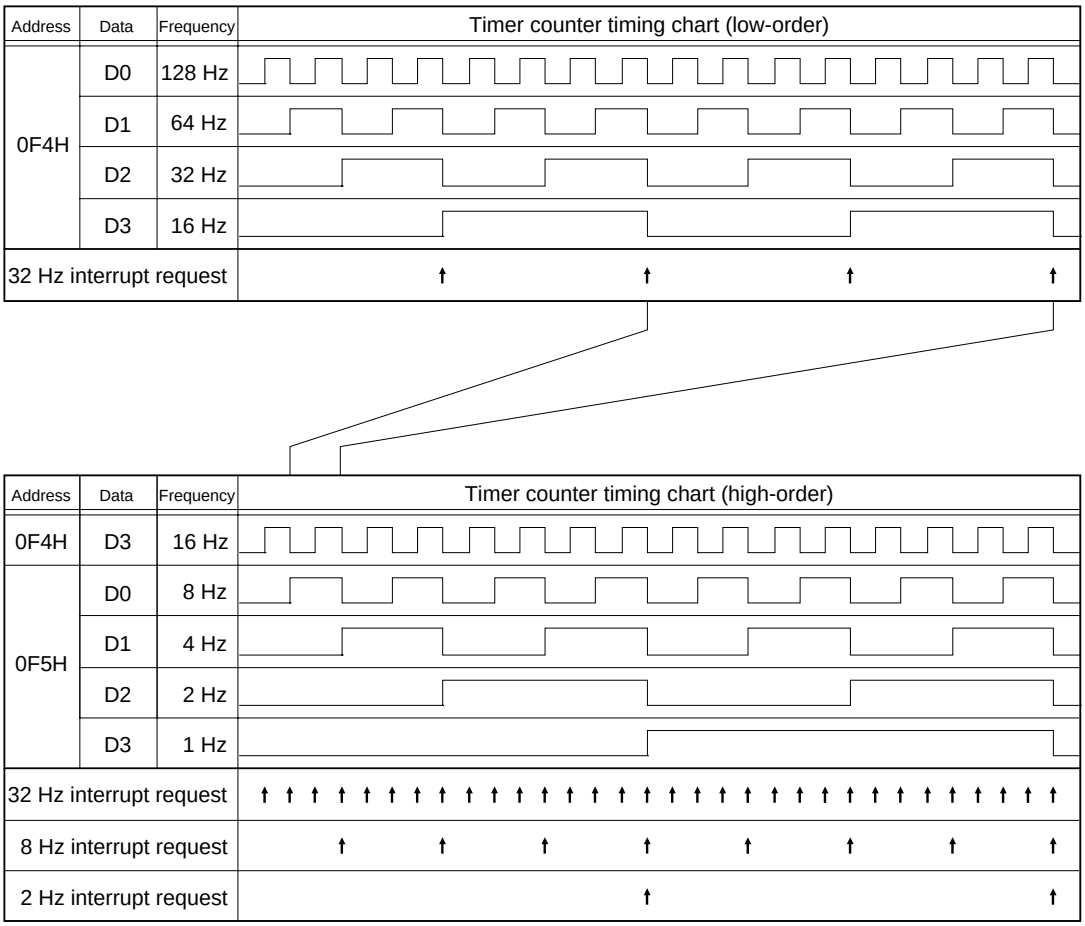


Fig. 2.6.1
Timing Chart of Timer

Timer control program samples

• Initializing the timer

```
LD      Y,0F3H    ;Set timer reset bit address
AND     MY,0111B  ;Reset and stop the timer
OR      MY,1000B  ;Start the timer
```

Execution of these three program steps resets (i.e., clears all of TM00–TM03 and TM10–TM13 to 0) and starts the timer.

• Loading the timer

```
LD      Y,0F4H    ;Set timer low-order address
LDPY    A,MY      ;A register ← TM00–TM03
LD      B,MY      ;B register ← TM10–TM13
```

Execution of these three program steps loads the low-order 4 bits to the A register and the high-order 4 bits to the B register:

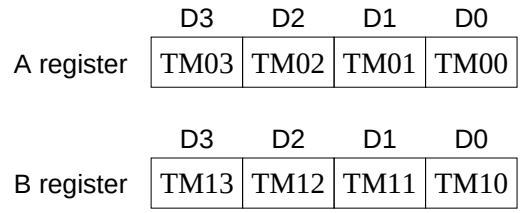


Fig. 2.6.2
Correspondence of Timer Data and Registers

The timer can be loaded to MX instead of the A and B registers.

• Checking timer edge

```
        LD      X,TMSTAT  ;Set timer edge counter address
        CP      MX,0      ;Check if MX is zero
        JP      Z,RETURN  ;Jump if Z flag = 1
        LD      Y,0F5H    ;Set timer high-order address
        LD      A,MY      ;A register ← TM13–TM10
        LD      Y,TMDTBF  ;Set timer data buffer address
        XOR     MY,A      ;Exclusive-OR
        FAN     MY,0010B  ;Check D1 bit of timer data buffer
        LD      MY,A      ;Timer data buffer A register
        JP      Z,RETURN  ;Jump if Z flag = 1
        ADD     MX,0FH    ;Decrement timer edge counter
;
RETURN  RET               ; Return
```

This program takes a subroutine form. It is called at short intervals, and decrements the data at address TMSTAT at an apparent rate of once every 125 ms until the data reaches 0. The timing chart is shown in Figure 2.6.3.

TMSTAT and TMDTBF may be arbitrary addresses in RAM and do not involve a hardware function.

The timer can be addressed by using the X register instead of the Y register.

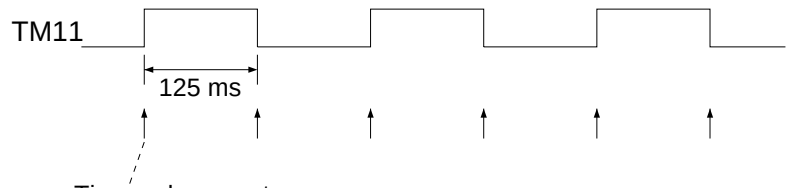


Fig. 2.6.3
Timing Chart

2.7 Remote-control LEDs

Remote LED control

The S1C621A0 contains a control circuit that drives remote-control LEDs. Typically, either PPM or PWM waves are used for remote-control transmission. The pulse width of these waves can be determined by software or by hardware (data set by software and waves generated by hardware).

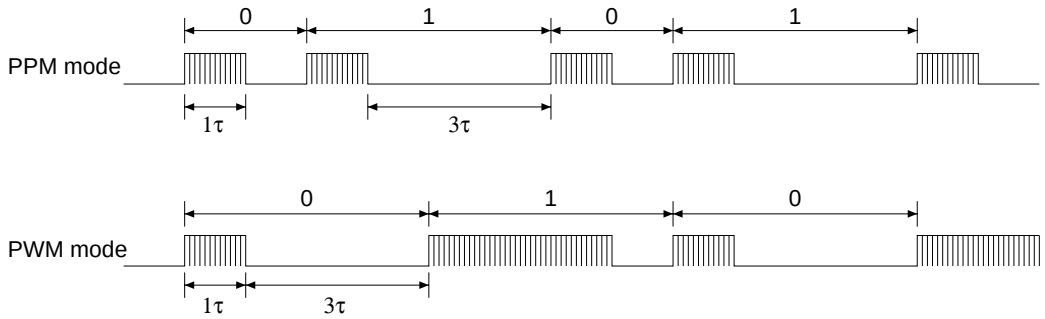


Fig. 2.7.1
Remote-controlled
Transmission

The carrier of the waveforms shown in Figure 2.7.1 has a frequency 1/8 or 1/12 the OSC3 ceramic oscillation frequency. The ratio is selected by software. Two different duty ratios are available for each of these two division ratios.

Table 2.7.1 shows the registers that control the remote-control transmission waves (REM). It is followed by a detailed description of the individual registers. The Technical Hardware Manual describes the features of each of the two methods for determining the REM pulse width (software or hardware). Select whichever method is convenient to your programming.

Table 2.7.1 REM Control Registers

<table>
<tr><th rowspan="2">Address</th><th rowspan="2">Data bit</th><th rowspan="2">R/W</th><th rowspan="2">Name</th><th colspan="2">Method</th><th rowspan="2">Contents</th></tr>
<tr><th>Software timer</th><th>Hardware timer</th></tr>
<tr><td>0F0H</td><td>D3</td><td>R/W</td><td>REMSO</td><td>○</td><td></td><td>0: Forced REM output off.
1: Forced REM output on.
Writing 1: Carrier counter is forced to be reset and started (for hardware timer, must be fixed at 0).</td></tr>
<tr><td>0F0H</td><td>D2</td><td>R</td><td>IREM</td><td></td><td>○</td><td>REM interrupt factor flag
1: interrupt request</td></tr>
<tr><td>0F2H</td><td>D3</td><td>R/W</td><td>REMC</td><td>○</td><td>○</td><td>0: Turns REM carrier circuit off.
1: Turns REM carrier circuit on.</td></tr>
<tr><td>0F2H</td><td>D2</td><td>R/W</td><td>EIREM</td><td></td><td>○</td><td>REM interrupt mask register
0: Disables (masks) interrupt
1: Enables interrupt</td></tr>
<tr><td>0F7H</td><td>D3
D2</td><td>R/W
R/W</td><td>RCDIV
RCDUTY</td><td>○</td><td>○</td><td>Sets REM carrier interval and duty ratio.
<table>
<tr><th>RCDIV</th><th>RCDUTY</th><th>Frequency division ratio</th><th>Duty</th></tr>
<tr><td>0</td><td>0</td><td>1/8</td><td>1/4</td></tr>
<tr><td>0</td><td>1</td><td>1/8</td><td>3/8</td></tr>
<tr><td>1</td><td>0</td><td>1/12</td><td>1/3</td></tr>
<tr><td>1</td><td>1</td><td>1/12</td><td>1/4</td></tr>
</table></td></tr>
<tr><td>0F7H</td><td>D1
D0</td><td>R/W
R/W</td><td>RT1
RT0</td><td></td><td>○</td><td>Sets REM reference cycle (τ).
<table>
<tr><th>RT1</th><th>RT0</th><th>Frequency division ratio 2</th></tr>
<tr><td>0</td><td>0</td><td>1/12</td></tr>
<tr><td>0</td><td>1</td><td>1/16</td></tr>
<tr><td>1</td><td>0</td><td>1/20</td></tr>
<tr><td>1</td><td>1</td><td>1/32</td></tr>
</table></td></tr>
<tr><td>0F8H</td><td>D3
D2
D1
D0</td><td>W
W
W
W</td><td>RIC3
RIC2
RIC1
RIC0</td><td></td><td>○</td><td>Sets count in interrupt counter.
Counts one each time data is written.
Writing 0FH is prohibited.
Writing in software timer mode is prohibited.</td></tr>
<tr><td>0F9H</td><td>D3
D2</td><td>R/W
R/W</td><td>ROUT1
ROUT0</td><td></td><td>○</td><td>Sets REM output width (τ count).
Operates only once each time data is written.
Writing in software timer mode is prohibited.</td></tr>
</table>

Detailed description of REM control registers—Part 1 (operation under software timer control)

• REMC

The REMC register turns the REM circuit on or off. Setting the register to 1 causes the OSC3 clock to be introduced into the carrier generator circuit, thus turning the REM circuit on. If the register is set to 0, the REM circuit turns off; this synchronizes with the reference cycle (τ). OSC3 must be turned off at least the time τ after the REMC register has been set to 0. The REMC register can be set to 1 only with OSC3 on.

At Initial reset, the REMC register is set to 1 and initialized by hardware. This dictates that at least 32 machine cycles be required before the REMC register can be set to 0.

If you need no REM output, hold the REMC register at 0 to save power.

• RCDIV and RCDUTY

The REM carrier is generated by dividing the OSC3 frequency. The RCDIV and RCDUTY registers are used to determine a division ratio and a duty ratio. Once set, these registers generate carriers with the set division ratio and duty ratio. Except for initialization the REMC register should be set at 0 when the RCDIV and RCDUTY registers are set. (Setting RCDIV and RCDUTY with the REMC at 1 will cause a malfunction.)

• REMSO

The REMSO register is used when a pulse width is determined by software, A carrier is on at the REM pin while the REMSO register is set at 1. Thus the pulse width of the REM output is determined by the timing of when data 1 or 0 is set in the REMSO register.
If you use hardware to determine a pulse width, the REMSO register must be fixed at 0.

These four registers are used to determine REM output under software timer control.

Sample of REM output program under software timer control

- **PPM waveform output (OSC3 = 455 kHz)**

Main routine

```
        LD      X,0F6H    ;Set oscillation circuit address
        OR      MX,0001B  ;OSC3 oscillation on
        CALL    TWAIT     ;Wait 7.7 ms
        AND     MX,1101B  ;Change from OSC1 to OSC3
        LD      Y,0F7H    ;Set RCDIV and RCDUTY address
        OR      MY,1000B  ;Set 1/12 division ratio
        AND     MY,1011B  ;Set 1/3 duty ratio
        LD      Y,0F2H    ;Set REMC address
        OR      MY,1000B  ;REMC on
        LD      Y,0F0H    ;Set REMSO address
(1)     LD      MY,1000B  ;REMSO on
        CALL    TWAIT     ;Wait 0.56 ms
(2)     LD      MY,0000B  ;REMSO off
        CALL    TWAIT     ;Wait 0.56 ms
(3)     LD      MY,1000B  ;REMSO on
        CALL    TWAIT     ;Wait 0.56 ms
(4)     LD      MY,0000B  ;REMSO off
        CALL    TWAIT     ;┐
        CALL    TWAIT     ;│
        CALL    TWAIT     ;│ Wait 1.69 ms
        NOP7              ;│
        NOP5              ;┘
(5)     LD      MY,1000B  ;REMSO on
        CALL    TWAIT     ;Wait 0.56 ms
(6)     LD      MY,0000B  ;REMSO off
        LD      Y,0F2H    ;Set REMC address
        AND     MY,0111B  ;REMC off
        CALL    TWAIT     ;┐ Wait 1.12 ms
        CALL    TWAIT     ;┘
        OR      MX,0010B  ;Change from OSC3 to OSC1
        AND     MX,1110B  ;OSC3 oscillation off
```

Subroutine

```
TWAIT   LD      A,0CH     ;┐
LOOP1   NOP7              ;│
        ADD     A,0FH     ;│ OSC1: Wait 7.7 ms
        JP      NZ,LOOP1  ;│ OSC3: Wait 0.56 ms
        NOP5              ;│
        RET               ;┘
```

In the above sample program of REM output under software timer control, the data is output in PPM format and in the order 0 to 1. The timing chart is shown in Figure 2.7.2.

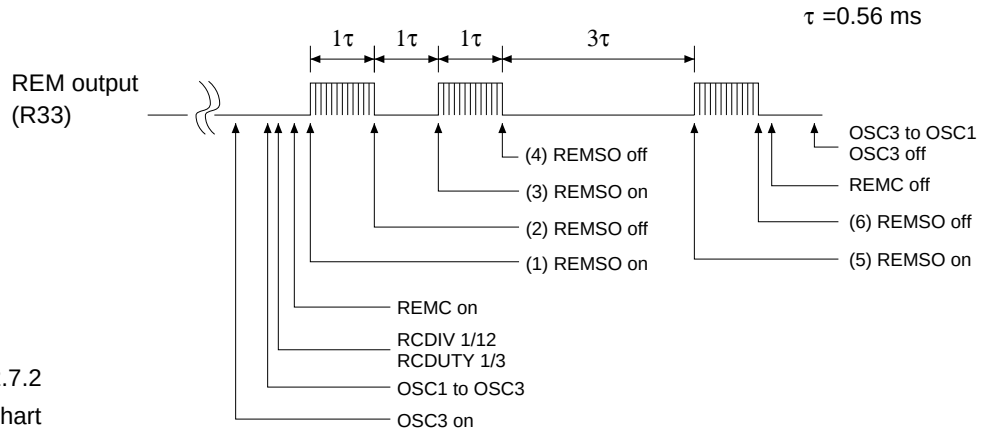


Fig. 2.7.2
Timing Chart

This sample program is not practical because the output data 1 and 0 are fixed. A practical program must have the length of waiting time selected by the 1 and 0 transmission data and the waiting time defined in the program must take into account the time required for such selection. It is generally recommended that the 1 and 0 transmission data be selected from the contents of a transmission data table stored in RAM.

As the REM output operation under software timer control involves difficulties in time control, hardware timer control is the better choice if the transmission format meets the hardware timer specifications.

Detailed description of REM control registers—Part 2 (operation under hardware timer control)

• RT1 and RT0

The RT1 and RT0 registers define the timing at which the reference cycle τ is generated in the hardware timer control mode. Once these registers are set, a waveform will be generated at the same reference τ interval.

The registers must be set with REMC at 0, except for initialization. (Setting the registers with REMC at 1 results in a malfunction.)

Figure 2.7.3 shows the timing chart with a carrier division ratio of 1/12.

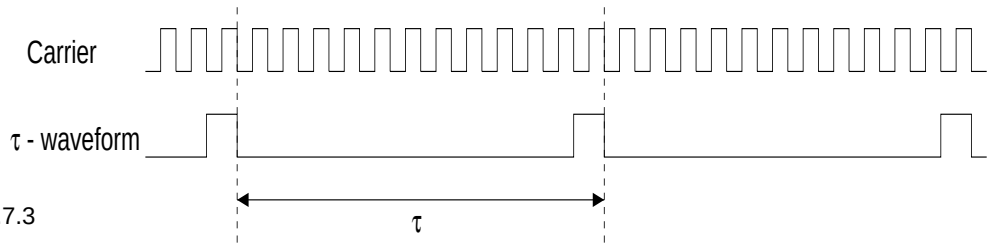


Fig. 2.7.3
τ-waveform

• REMC

The REMC register is described in "Detailed description of REM control registers—Part 1". Its timing chart is shown in Figure 2.7.4.

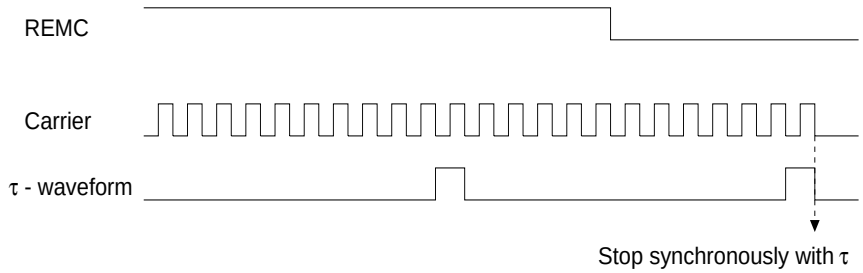


Fig. 2.7.4
REM Circuit Stop Timing

A maximum 384 machine cycles* are required until the REM circuit stops after REMC is set to 0. Even if the CPU clock is changed from OSC3 to OSC1 with REMC = 0, OSC3 must not be turned off before the REM circuit stops.

* This time depends on the interval τ that has been set. If a shorter interval is set, the maximum time required until the REM circuit stops after REMC = 0 is shorter.

• ROUT1 and ROUT0

The ROUT1 and ROUT0 registers define the REM output time. REM data is automatically output at τ intervals in accordance with the timing data set in the ROUT1 and ROUT0 registers. The REM output is automatically turned off after the elapse of the set time. The REM output time t_{RO} is given by the following equation:

$$t_{RO} = (ROUT1 \times 2 + ROUT0) \times \tau$$

The values set in the ROUT1 and ROUTO registers are taken into the REM output length generation circuit synchronously with the rising edge of a τ-waveform. For this reason, avoid writing data into the ROUT1 and ROUT0 registers during one carrier cycle before and after the rise of the τ-waveform.

If 0 is set in both registers (ROUT1 and ROUT0), no REM data is output. Figure 2.7.5 shows the timing of REM output based on the data set in the ROUT1 and ROUT0 registers.

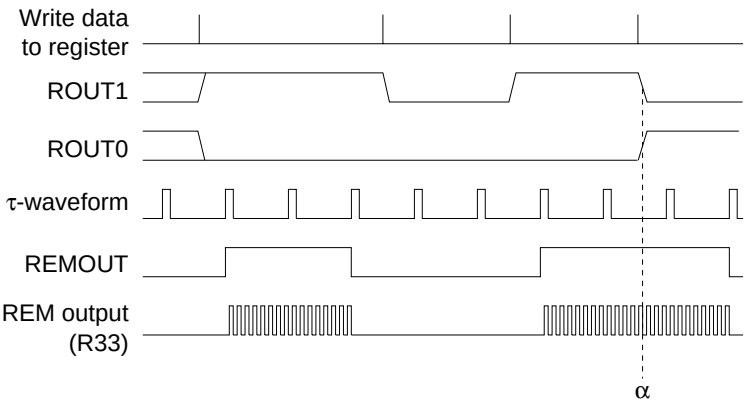


Fig. 2.7.5
REM Circuit Restart Timing

Under hardware timer control, REM output occurs only once after data is written to the ROUT1 and ROUT0 registers. It is therefore necessary that data be written into these registers acceding to the required timing to generate a transmission waveform.

For a one-time output of REM data covering 4τ or more cycles, data must be written into the ROUT1 and ROUT0 registers before the timing at which the REM output turns off (see the timing α shown in Figure 2.7.5).

If 0 is set in both the ROUT1 and ROUT0 registers at the timing α in Figure 2.7.5, the REM output will be turned of at the next τ timing.

• RIC0–RIC3

This 4-bit register (down-counter) determines the timing of a REM interrupt under hardware timer control. It is decremented synchronously with τ. When reaching 0FH, the register stops down-counting and invokes an interrupt.

The time length t_{RIC}, which depends on the RIC0–RIC3 data, is given by the following equation:

$$t_{RIC} = (RIC3 \times 2^3 + RIC2 \times 2^2 + RIC1 \times 2 + RIC0) \times \tau$$

t_{RI}, the time required until an interrupt is invoked, is calculated by:

$$t_{RI} = t_{RIC} + (1 \pm 1 \text{ instruction cycle})$$

- t_{RIC}: Time set by RIC register ($n \times \tau$)
- (1 ± 1 instruction cycle): Interrupt response time

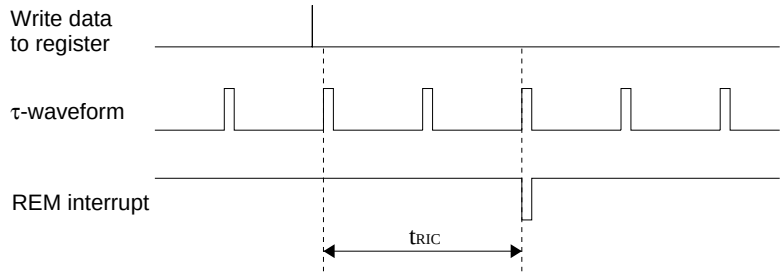


Fig. 2.7.6
REM Interrupt Timing

In Figure 2.7.6, a REM interrupt occurs at the timing TRIC set in the RIC register.

Once data has been set in the RIC register, avoid writing data again into the register before a REM interrupt occurs, as this may cause an invalid interrupt.

The values allowed for the RIC register are 0 to 0EH.

• EIREM

The EIREM register controls REM interrupt masking. To output REM data using an interrupt under hardware timer control, set the EIREM register to 1. Set it to 0 after completion of the REM output.

• IREM

This register is the flag which is set when a REM interrupt occurs. Whether or not a REM interrupt has occurred can be identified by software. The flag is reset by reading it.

Read the IREM flag must be read in DI status (interrupt flag = 0). Reading of interrupt factor flags is available at EI, but be careful in the following cases.
If the interrupt mask register value corresponding to the interrupt factor flags to be read is set to 1, an interrupt request will be generated by the interrupt factor flags set timing, or an interrupt request will not be generated. Be very careful when interrupt factor flags are in the same address.
For details on the EIREM and IREM, see section 2.9, "Interrupt and Halt".

The registers described above are used under hardware timer control. To program remote-control transmission waveforms under hardware timer control, set ROUT and RIC with the REM interrupt routine and repeat operations as many times as the total bits making up the transmission data.

Sample of REM output program under hardware timer control

- **PPM waveform output (OSC3 = 455 kHz)**

Main routine

```
        DI              ;Disable interrupt
        LD      B,0001B ;Initialize bit pointer
        LD      Y,DATA  ;Set transmission data address
        LD      MY,0101B;Set transmission data
        LD      X,0F6H  ;Set oscillation circuit address
        OR      MX,0001B;OSC3 oscillation on
        CALL    TWAIT   ;Wait 7.3 ms
        AND     MX,1101B;Change from OSC1 to OSC3
        LD      X,0F7H  ;Set REM frequency division ratio
                        ;and duty address
        LD      MX,1010B;Division ratio = 1/12, duty = 1/3,
                        ;τ = 20 carriers
        LD      X,0F2H  ;Set REMC address
        LBPX    MX,8CH  ;REMC on, enable REM interrupt,
                        ;disable input interrupt and timer
                        ;interrupt
        CALL    REMINT  ;Call REM interrupt routine
;
LOOP1   FAN     A,1111B ;Check transmission and flag
        JP      Z,LOOP1 ;Jump if transmission end flag = 0
        CALL    TWAIT   ;Wait 0.527 ms, REMC off
        LD      X,0F6H  ;Set oscillation circuit address
        OR      MX,0010B;Change from OSC3 to OSC1
        AND     MX,1110B;OSC3 oscillation off
        EI              ;Enable interrupt
```

Subroutine

```
TWAIT   LD      A,0BH   ;┐
LOOP2   NOP7            ;│
        ADD     A,0FH   ;│ OSC1: Wait 7.3 ms
        JP      NZ,LOOP2;│ OSC3: Wait 0.527 ms
        NOP5            ;│
        NOP7            ;┘
        RET
```

REM interrupt routine

```
   REMIT   PUSH    F           ;Store flag
           FAN     B,1111B     ;Check bit pointer for zero
           JP      Z,REMEND    ;Jump if Z flag = 1
           FAN     MY,B        ;Check transmission data for 1 or 0
           LD      X,0F8H      ;Set RIC address
           JP      NZ,PPM1     ;Jump if transmission data = 1
(2)        LBPX    MX,41H      ;Set ROUT1 τ and RIC1 τ
           JP      BITSFT      ;Jump to BITSFT
(1) PPM1   LBPX    MX,43H      ;Set ROUT1 τ and RIC3 τ
   BITSFT  RCF                 ;Reset C flag
           RLC     B           ;Shift bit pointer left
           LD      X,0F0H      ;Set REM interrupt factor flag
                               ;address
           LD      A,MX        ;Reset REM interrupt factor flag
           LD      A,0         ;Transmission end flag ← 0
           POP     F           ;Restore flag
           EI                  ;Enable interrupt
           RET                 ;Interrupt routine return
   ;
   REMEND  LD      X,0F9H      ;Set ROUT address
(3)        LD      MX,0100B    ;Set ROUT1 τ
           CALL    TWAIT       ;┐
           CALL    TWAIT       ;┘Wait 1.055 ms 2τ
           LD      X,0F2H      ;Set REMC address
           LBPX    MX,0F3H     ;REMC off, disable REM interrupt,
                               ;enable input interrupt and
                               ;timer interrupt
           LD      X,0F0H      ;Set REM interrupt factor flag
                               ;address
           LD      A,MX        ;Reset REM interrupt factor flag
           LD      A,1         ;Transmission end flag ← 1
           POP     F           ;Restore flag
           RET                 ;Interrupt routine return
```

Given above is a sample of an REM output program under hardware timer control. The contents of address "DATA" are sequentially output in PPM format on a 4-bit, LSB first basis.

Figure 2.7.7 shows the timing chart.

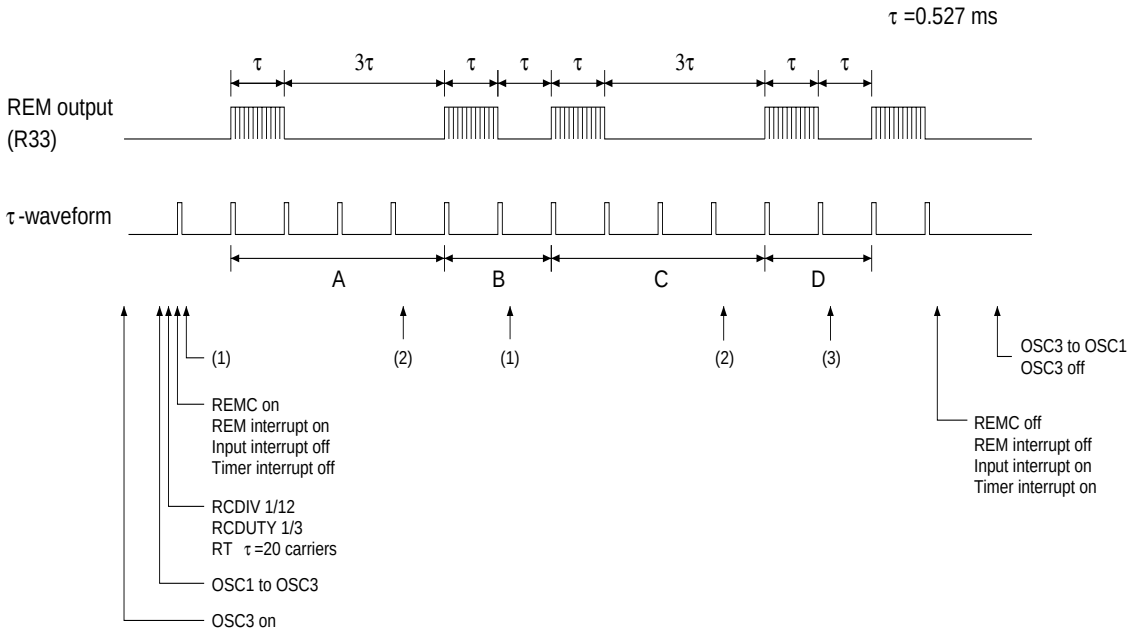


Fig. 2.7.7
Timing Chart

In the hardware timer control mode, REM data is output synchronously with the τ waveform according to the data set in the associated registers. This releases the CPU temporarily from the REM output procedure after the data has been set in the registers. Use of the REM interrupt function allows the REM output procedure and other operations to be virtually paralleled.

Notes on REM circuit

• Initialization of REM circuit

The following programming steps are needed to initialize the REM circuit (τ-clock, REM interrupt circuit):

1. Write data at addresses 0F7H and 0F8H in that order within 80 machine clocks (equivalent to eleven 7-clock instructions) after release from initial reset.

2. With REMC = 0 (0F2H, D3), the REM circuit must not be stopped within an interval of 1τ after data has been written at address 0F8H.

3. To initialize the REM interrupt circuit, read the REM interrupt factor flag (address 0F0H) to clear it at least an interval of 2τ after data has been written at address 0F8H.

• Programming REM circuit

In programming the REM circuit the following precautions should be observed:

1. After initial reset, the REMC register stays at 1 to initialize the carrier generator circuit. The REMC register can only be reset to 0 after initialization (at least 32 machine clocks later).

2. The REM circuit does not stop immediately after the REMC register is reset to 0. It stops synchronously with the cycle τ, until which time OSC3 must be held on.

3. With the REM circuit in operation, do not write data at addresses 0F8H and 0F9H (REM interrupt counter and REMOUT width setting register) during an interval or one carrier before and after the rise of τ.

4. With the REM circuit in operation, do not write data at address 0F7H (τ-setting register).

5. During operation under hardware timer control, the REMSO register must be fixed at 0.

6. Read the IREM (REM interrupt factor flag) in DI status (interrupt flag = 0). Reading of interrupt factor flags is available at EI, but be careful in the following cases.
   If the interrupt mask register value corresponding to the interrupt factor flags to be read is set to 1, an interrupt request will be generated by the interrupt factor flags set timing, or an interrupt request will not be generated. Be very careful when interrupt factor flags are in the same address.

7. The values that can be set in the REM interrupt counter (0F8H) are from 0 to 0EH. Remember, writing 0FH into the counter causes an error.

8. Once data has been written into the REM interrupt counter (0F8H), writing data into the counter again results in an interrupt error if it is done before a REM interrupt occurs.

2.8 AMP and SVD

AMP and SVD control

The S1C621A0 contains an analog comparator (AMP) and a supply voltage detector (SVD), the data of which can be read by software. These circuits can be individually turned on/off to save power.

Following is a description of the addresses of the AMP/SVD control bits:

Table 2.8.1 I/O Memory Map

Address	Register								Comment
	D3	D2	D1	D0	Name	Init	1	0	
0F6H	SVDDT	SVDON	CLKCHG	OSCC	SVDDT	–	Normal	Low voltage	SVD data (1 when SVDON = 0)
	R	R/W			SVDON	0	On	Off	SVD circuit on/off
					CLKCHG	0	OSC1	OSC3	CLK change OSC1/OSC3
					OSCC	1	On	Off	OSC3 oscillation on/off
0FFH	OPDT	OPON	IOC	R20	OPDT	–	High	Low	Analog comparator data (0 or 1)
	R	R/W			OPON	0	On	Off	Analog comparator on/off
					IOC	0	Out	In	I/O port control out/in
					R20	0	High	Low	Output port

Note: Init = value at initial reset, – = not set on the circuit

(1) AMP

• OPON

This bit controls analog comparator (AMP) power on/off. The AMP circuit stays on with OPON at 1, and off with OPON at 0. At initial reset, the AMP circuit is off. While the circuit is not in use, hold it at 0 to save power.

• OPDT

When the AMP is used as a comparator, the output data appears in OPDT. This bit is 1 with AMPP > AMPM, and 0 with AMPP < AMPM. If the OPON bit is 0, the OPDT bit is fixed at 1.

(2) SVD

• SVDON

This bit controls supply voltage detector (SVD) power on/off. The SVD circuit stays on with SVDON at 1, and off with SVDON at 0. At initial reset, the SVD circuit is off. While the circuit is not in use, hold it at 0 to save power.

• SVDDT

The output data from the SVD circuit appears in SVDDT. This bit is 1 if the supply voltage is normal, and 0 if it becomes low. When the SVDON bit is 0, the SVDDT bit is fixed at 1.

AMP and SVD control program samples

• Loading AMP output data to A register

```
        LD      Y,0FFH      ;Set AMP circuit address
        OR      MY,0100B    ;AMP circuit on
        LD      A,8         ;┐
LOOP    ADD     A,0FH       ;│ Wait about 3 ms
        JP      NZ,LOOP     ;┘
        LD      A,MY        ;A register, D3 ← OPDT
        AND     MY,1011B    ;AMP circuit off
```

Execution of the above program loads the AMP output data "OPDT" to D3 of the A register.

It takes 3 ms for the AMP output to become stable when the circuit is turned from off to on. Therefore, the program must include a waiting time of at least 3 ms before the output data is loaded after the AMP circuit has been turned on.

The OPDT data can be loaded to the B register or MX instead of the A register.

These two bits can be addressed by using the X register instead of the Y register.

- **Loading SVD output data to B register**

```
        LD      Y,0F6H      ;Set SVD circuit address
        OR      MY,0100B    ;SVD circuit on
        LD      A,8         ;┐
LOOP    ADD     A,0FH       ; │ Wait about 3 ms
        JP      NZ,LOOP     ;┘
        LD      B,MY        ;B register, D3 ← SVDDT
        AND     MY,1011B    ;SVD circuit off
```

Execution of the above program loads the SVD output data "SVDDT" to D3 of the B register.

As with the AMP circuit, it takes 3 ms for the SVD output to become stable when the circuit is turned from off to on.

The SVDDT data can be loaded to the A register or MX instead of the B register.

These two bits can be addressed by using the X register instead of the Y register.

2.9 Interrupt and Halt

Interrupt and halt control

The S1C621A0 supports a total of six interrupt functions: two input interrupts, three timer interrupts, and one remote control (REM) interrupt. These interrupts can be individually enabled or disabled. In addition, all interrupts can be disabled at one time by the DI instruction.

Each interrupt has a independent vector address, and its priority is decided by software. (Note that one common address is used for the three timer interrupts.)

An interrupt is used for restarting from a halt state dictated by the HALT instruction. Thus, if a halt state is entered with all interrupts disabled, the system will restart from the initial reset state under watchdog timer control.

The addresses relating to interrupt control are listed below.

Table 2.9.1 I/O Memory Map

Address	Register D3	D2	D1	D0	Name	Init	1	0	Comment
0F0H	REMSO	IREM	IK1	IK0	REMSO	0	On	Off	Forced REM output (on/off)
	R/W	R			IREM	x	Yes	No	Interrupt factor flag (REM) Clear to 0 after read
					IK1	0	Yes	No	Interrupt factor flag (K10–K13) Clear to 0 after read
					IK0	0	Yes	No	Interrupt factor flag (K00–K03) Clear to 0 after read
0F1H	WDRST	TI2	TI8	TI32	WDRST	Clear	Clear	–	Watchdog timer reset
	W	R			TI2	0	Yes	No	Interrupt factor flag (Timer 2 Hz) Clear to 0 after read
					TI8	0	Yes	No	Interrupt factor flag (Timer 8 Hz) Clear to 0 after read
					TI32	0	Yes	No	Interrupt factor flag (Timer 32 Hz) Clear to 0 after read
0F2H	REMC	EIREM	EIK1	EIK0	REMC	1	On	Off	REM carrier generation on/off
	R/W				EIREM	0	Enable	Mask	Interrupt mask register (REM)
					EIK1	0	Enable	Mask	Interrupt mask register (K10–K13)
					EIK0	0	Enable	Mask	Interrupt mask register (K00–K03)
0F3H	TMRUN	ETI2	ETI8	ETI32	TMRUN	0	Run	Clear & Stop	Timer Run/Clear & Stop
	R/W				ETI2	0	Enable	Mask	Interrupt mask register (Timer 2 Hz)
					ETI8	0	Enable	Mask	Interrupt mask register (Timer 8 Hz)
					ETI32	0	Enable	Mask	Interrupt mask register (Timer 32 Hz)

Note: Init = value at initial reset, x = undefined, – = not set on the circuit

(1) Interrupt Factor Flags

• IK0
Set to 1 by a fall of input port signals K00–K03. Software checks this flag to judge whether or not a K00–K03 input interrupt has occurred. Also, software resets the flag by reading its status.

• IK1
Set to 1 by a fall of input port signals K10–K13. Software checks this flag to judge whether or not a K10–K13 input interrupt has occurred. Also, software resets the flag by reading its status.

• IREM
Set to 1 by a borrow (0FH) of the REM interrupt counter RIC. Software checks this flag to judge whether or not a REM interrupt has occurred. Also, software resets the flag by reading its status.

• TI32
Set to 1 by a fall of the timer TM02 (32 Hz). Software checks this flag to judge whether or not a 32 Hz timer interrupt has occurred. Also, software resets the flag by reading its status.

• TI8
Set to 1 by a fall of the timer TM10 (8 Hz). Software checks this flag to judge whether or not an 8 Hz timer interrupt has occurred. Also, software resets the flag by reading its status.

• TI2
Set to 1 by a fall of the timer TM12 (2 Hz). Software checks this flag to judge whether or not a 2 Hz timer interrupt has occurred. Also, software resets the flag by reading its status.

Read the interrupt factor flag in the DI status (interrupt flag = 0). Reading of interrupt factor flags is available at EI, but be careful in the following cases.
If the interrupt mask register value corresponding to the interrupt factor flags to be read is set to 1, an interrupt request will be generated by the interrupt factor flags set timing, or an interrupt request will not be generated. Be very careful when interrupt factor flags are in the same address.

(2) Interrupt Mask Register

• EIK0

The EIK0 flag register enables or disables a K00–K03 input interrupt. When this register is set to 1 in the EI mode with the interrupt flag IK0 at 1, the CPU is interrupted.

• EIK1

The EIK1 flag register enables or disables a K10–K13 input interrupt. When this register is set to 1 in the EI mode with the interrupt flag IK1 at 1, the CPU is interrupted.

• EIREM

The EIREM flag register enables or disables a REM interrupt. When this register is set to 1 in the EI mode with the interrupt flag IREM at 1, the CPU is interrupted.

• ETI32

The ETI32 flag register enables or disables a 32 Hz timer interrupt. When this register is set to 1 in the EI mode with the interrupt flag TI32 at 1, the CPU is interrupted.

• ETI8

The ETI8 flag register enables or disables an 8 Hz timer interrupt. When this register is set to 1 in the EI mode with the interrupt flag TI8 at 1, the CPU is interrupted.

• ETI2

The ETI2 flag register enables or disables a 2 Hz timer interrupt. When this register is set to 1 in the EI mode with the interrupt flag TI2 at 1, the CPU is interrupted.

(3) Interrupt Vector Address

The S1C621A0 interrupt vector address is made up of the low-order 4 bits of the program counter (12 bits), each of which is assigned a specific function as shown in Figure 2.9.1.

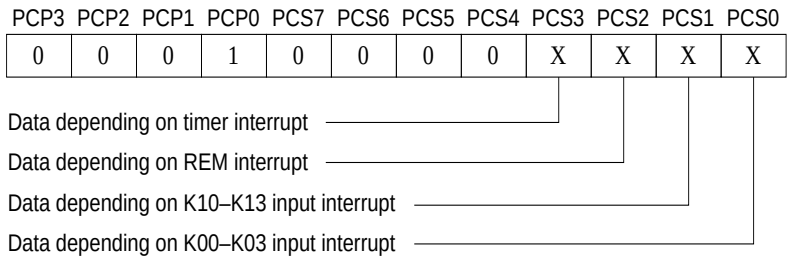


Fig. 2.9.1
Interrupt Vector Address

As shown in Figure 2.9.1, the data set in each of the low-order 4 bits of the program counter depends on whether or not an interrupt has occurred. Each of the bits is set to 1 if the corresponding interrupt occurs; otherwise, it is set to 0.

All of the three timer interrupts have the same vector address, and software must be used to judge whether or not a given timer interrupt has occurred.

The interrupt vector address is set to page 1, step 1-0FH depending on the mode in which an interrupt occurs.

Examples

- A K00–K03 input interrupt and a timer interrupt have occurred at the same time:

Fig. 2.9.2
Example of Interrupt Vector Setting 1

PCP3	PCP2	PCP1	PCP0	PCS7	PCS6	PCS5	PCS4	PCS3	PCS2	PCS1	PCS0
0	0	0	1	0	0	0	0	1	0	0	1

The program counter is set to page 1, step 9.

- Only a REM interrupt has occurred:

Fig. 2.9.3
Example of Interrupt Vector Setting 2

PCP3	PCP2	PCP1	PCP0	PCS7	PCS6	PCS5	PCS4	PCS3	PCS2	PCS1	PCS0
0	0	0	1	0	0	0	0	0	1	0	0

The program counter is set to page 1, step 4.

The occurrence of an interrupt automatically causes hardware to reset the interrupt flag, resulting in the DI mode. Use software to set the EI mode as required after the interrupt has been processed.

Setting the EI mode at the beginning of the interrupt routine allows programmable nesting levels.

Each interrupt factor flag must be reset in the associated interrupt routine before setting the EI mode. (The flag is reset by software when it read its status.) If the EI mode is set without resetting the interrupt factor flag and with the interrupt enable flag at 1, the preceding interrupt will occur again.

Figure 2.9.4 shows an interrupt logic diagram.

Read the interrupt factor flag in DI status (interrupt flag = 0). Reading of interrupt factor flags is available at EI, but be careful in the following cases.
If the interrupt mask register value corresponding to the interrupt factor flags to be read is set to 1, an interrupt request will be generated by the interrupt factor flags set timing, or an interrupt request will not be generated. Be very careful when interrupt factor flags are in the same address.

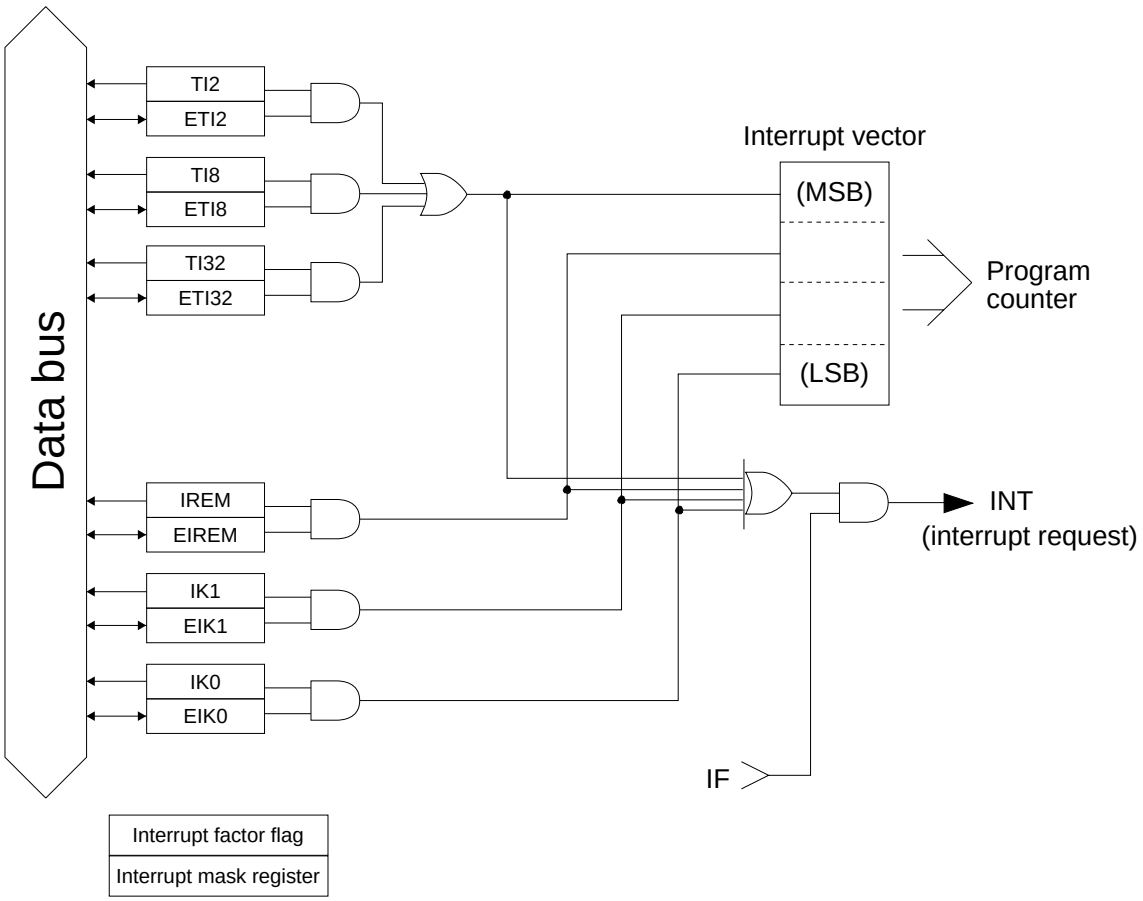


Fig. 2.9.4
Interrupt Logic Diagram

Interrupt and halt control program samples

• **Restarting from halt mode by interrupt**

Main routine

```
        LD      Y,0F2H      ;Set REM input interrupt mask
                            ;register address
        OR      MY,0111B    ;Enable REM input interrupt
        INC     Y           ;Set timer interrupt mask register
                            ;address
        OR      MY,0111B    ;Enable timer interrupt
;
MAIN    EI                  ;Set interrupt flag
        HALT                ;Halt mode
        JP      MAIN        ;Jump to "MAIN"
```

Interrupt vector routine

```
        ORG     101H
;
        JP      REMIN       ;Generate "IK0"
        JP      REMIN       ;Generate "IK1"
        JP      REMIN       ;Generate "IK0, IK1"
        JP      REMIN       ;Generate "IREM"
        JP      REMIN       ;Generate "IK0, IREM"
        JP      REMIN       ;Generate "IK1, IREM"
        JP      REMIN       ;Generate "IK0, IK1, IREM"
        JP      REMIN       ;Generate "TI"
        JP      REMIN       ;Generate "IK0, TI"
        JP      REMIN       ;Generate "IK1, TI"
        JP      REMIN       ;Generate "IK0, IK1, TI"
        JP      REMIN       ;Generate "IREM, TI"
        JP      REMIN       ;Generate "IK0, IREM, TI"
        JP      REMIN       ;Generate "IK1, IREM TI"
        JP      REMIN       ;Generate "IK0, IK1, IREM, TI"
;
REMIN   LD      Y,0F0H      ;Set REM,input interrupt factor
                            ;flag address
        LD      X,RIFSTK    ;Set REM,input interrupt factor
                            ;flag buffer address
        LD      MX,MY       ;Factor flag buffer ← factor
                            ;flag
        FAN     MX,0100B    ;Check REM interrupt factor
        JP      Z,NOREM     ;Jump if no interrupt factor
                            ;exists
        CALL    REMINT      ;Call REM interrupt routine
;
```

```
NOREM   LD      X,RIFSTK  ;Set REM input interrupt factor
                          ;flag buffer address
        FAN     MX,0010B  ;Check K10–K13 input interrupt
                          ;factor
        JP      Z,NOIK1   ;Jump if no interrupt factor
                          ;exists
        CALL    IK1INT    ;Call K10–K13 input interrupt
                          ;routine
;
NOIK1   LD      X,RIFSTK  ;Set REM input interrupt factor
                          ;flag buffer address
        FAN     MX,0001B  ;Check K00–K03 input interrupt
                          ;factor
        JP      Z,TIMER   ;Jump if no interrupt factor
                          ;exists
        CALL    IK0INT    ;Call K00–K03 input interrupt
                          ;routine
;
TIMER   LD      Y,0F1H    ;Set timer interrupt factor flag
                          ;address
        LD      X,TMFSTK  ;Set timer interrupt factor flag
                          ;buffer address
        LD      MX,MY     ;Factor flag buffer ← factor
                          ;flag
        FAN     MX,0100B  ;Check 2 Hz timer interrupt factor
        JP      Z,NO2HZ   ;Jump if no interrupt factor
                          ;exists
        CALL    TINT02    ;Call 2 Hz timer interrupt routine
;
NO02HZ  LD      X,TMFSTK  ;Set timer interrupt factor flag
                          ;buffer address
        FAN     MX,0010B  ;Check 8 Hz timer interrupt factor
        JP      Z,NO8HZ   ;Jump if no interrupt factor
                          ;exists
        CALL    TINT08    ;Call 8 Hz timer interrupt routine
;
NO08HZ  LD      X,TMFSRK  ;Set timer interrupt factor flag
                          ;buffer address
        FAN     MX,0001B  ;Check 32 Hz timer interrupt
                          ;factor
        JP      Z,NO32HZ  ;Jump if no interrupt factor
                          ;exists
        CALL    TINT32    ;Call 32 Hz timer interrupt routine
;
NO32HZ  RET               ;Interrupt routine return
```

In the above program, the CPU is cycled through starting from the halt mode by interrupt, processing the interrupt and returning to the halt mode again.

All interrupts are enabled. If they occur at the same time, they are processed in the following priority order set by software:

1) REM interrupt
2) K10–K13 input interrupt
3) K00–K03 input interrupt
4) 2 Hz timer interrupt
5) 8 Hz timer interrupt
6) 32 Hz timer interrupt

The sample program prohibits interrupt nesting and always proceeds in the priority order. The interrupt routines are located at the addresses (label: ??????) given by CALL instructions.

An interrupt factor flag is reset by software when it read its status. When simultaneously using interrupts that have interrupt factor flag at the same address, store flag checks in RAM before doing so. (Direct checking with the FAN instruction will reset all factor flags at the same address.)

Read the interrupt flag in DI status (interrupt flag = 0). Reading of interrupt factor flags is available at EI, but be careful in the following cases.
If the interrupt mask register value corresponding to the interrupt factor flags to be read is set to 1, an interrupt request will be generated by the interrupt factor flags set timing, or an interrupt request will not be generated. Be very careful when interrupt factor flags are in the same address.

CHAPTER 3 SYSTEM INITIALIZATION SAMPLE PROGRAM

```
        ORG     100H
        JP      INIT        ;Jump to "INIT"
;
        ORG     110H
INIT    RST     F,0011B     ;Interrupt disable and decimal adjust off
                            ;(For compatibility with the S1C6214)
        LD      A,0         ;┐
        LD      B,0DH       ; │ Initialize stack pointer
        LD      SPL,A       ; │ SP ← 0D0H
        LD      SPH,B       ;┘
        LD      X,0F2H      ;┐ Reset and stop timer
        LBPX    MX,8        ;┘ All interrupt enable flags off
        LD      X,0F0H      ;┐
        LDPX    MX,0        ;┘ REMSO off
        LD      A,MX        ;Reset timer interrupt factor flag
        LD      MX,1000B    ;Reset watchdog timer
        LD      X,0F7H      ;┐ Set frequency division ratio = 1/12,
                            ; │ duty = 1/3, τ = 16 carriers
        LBPX    MX,9        ;┘ Set RIC = 0 τ
;
        LD      Y,0F1H      ;Set watchdog timer address
        LD      A,0         ;┐
        LD      B,0FH       ; │
        LD      M2,B        ; │ Set loop counter
ALLON3  LD      M1,A        ; │
ALLON2  LD      M0,A        ;┘
ALLON1  LD      X,0D0H      ;Set segment memory address
ALLON0  LBPX    MX,0FFH     ;┐
        LD      MY,1000B    ; │ Reset watchdog timer
        ADD     A,0FH       ; │ All LCDs on
        JP      NZ,ALLON0   ;┘
        DEC     M0          ;┐ Loop 1
        JP      NZ,ALLON1   ;┘
        DEC     M1          ;┐ Loop 2
        JP      NZ,ALLON2   ;┘
```

```
        DEC     M2          ;┐ Loop 3
        JP      NZ,ALLON3   ;┘
;
        LD      X,0F2H      ;┐ REMC off
        LD      MX,0        ;┘
        LD      X,0F0H      ;┐ Reset REM interrupt, input interrupt
        LD      A,MX        ;┘ factor flag
;
        LD      X,0         ;┐
        LD      A,0DH       ;
RAMCR1  LD      B,0         ;
RAMCR0  LDPX    MX,0        ;
        ADD     B,0FH       ;  Clear RAM 00H–CFH
        JP      NZ,RAMCR0   ;
        ADD     A,0FH       ;
        JP      NZ,RAMCR1   ;┘
;
        LD      X,0F6H      ;┐ Change from OSC3 to OSC1
        OR      MX,0010B    ;  OSC3 oscillation off
        AND     MX,1110B    ;┘
        LD      X,0F9H      ;┐ Set ROUT = 0 τ
        LD      MX,0        ;┘
        LD      X,0F0H      ; Reset watchdog timer
        LBPX    MX,80H      ; Enable input interrupt and 2 Hz
                            ;  timer interrupt
        LBPX    MX,0C3H     ; Start timer
;
        LD      X,0         ;┐
        LD      Y,0         ;
        LD      A,0         ;  Clear each register flag
        LD      B,0         ;
        RST     F,0         ;┘
        EI                  ;Enable interrupt
```

Table 3.1
Result of Execution of
Initialization Program

Internal circuit		Setting value
General register	A	0
General register	B	0
Index register	X	0
Index register	Y	0
Stack pointer	SP	0D0H
Interrupt flag	I	1
Decimal flag	D	0
Zero flag	Z	0
Carry flag	C	0
RAM data (0–0CFH)		0
Segment data (0D0H–0EFH)		0FH
System clock change (OSC1)		
REM circuit stop		
REM frequency division ratio, duty and τ set		

The sample program above is a basic initialization program of the S1C621A0. Before using the program, add further information that may be needed for the intended application.

CHAPTER 4 SUMMARY OF PROGRAMMING NOTES

• Core CPU

Only the program counter, new page pointer, interrupt flag and decimal flag are set. Any undefined values for internal circuits must be programmed.

• Memory

Memory is not mounted in unused area within the memory map and in memory area not indicated in this manual. For this reason, normal operation cannot be assured for programs that have been prepared with access to these areas.

• Oscillation Circuit

(1) At least 3 seconds is required for OSC1 (crystal oscillation circuit) to become stable after a voltage has supplied. When changing the CPU clock from OSC3 to OSC1, have a delay of at least 3 seconds after release from a system reset.

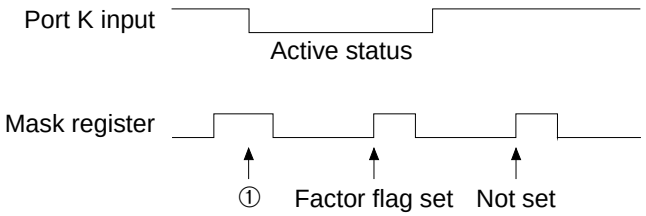
(2) If the 455 kHz ceramic oscillator is used for OSC3 oscillation, at least 5 ms is required for the OSC3 oscillation to become stable when the circuit is switched from off to on. The CPU clock must not be switched from OSC1 to OSC3 until 5 ms after the OSC3 oscillation has started.

(3) OSC3 oscillation can be turned off only when OSC1 is used as the CPU clock. Using only one instruction to change the clock from OSC3 to OSC1 and turn OSC3 oscillation off will result in a CPU malfunction.

• Input Port

(1) When an input port is changed from low to high level using a pull-up resistor, the waveform rises with a certain delay caused by the time constant of the pull-up resistance and input gate capacitance. Therefore, it is necessary to program the operation so that a proper waiting timer exists before the input port is read.

(2) Input interrupt programing related precautions



When the content of the mask register is rewritten, while the port K input is in the active status. The input interrupt factor flag is set at ①.

Fig. 4.1
Input Interrupt Timing

When using an input interrupt, if you rewrite the content of the mask register, when the value of the input terminal which becomes the interrupt input is in the active status (input terminal = low status), the factor flag for input interrupt may be set.
For example, a factor flag is set with the timing of ① shown in Figure 4.1. However, when clearing the content of the mask register with the input terminal kept in the low status and then setting it, the factor flag of the input interrupt is again set at the timing that has been set. Consequently, when the input terminal is in the active status (low status), do not rewrite the mask register (clearing, then setting the mask register), so that a factor flag will only set at the falling edge in this case. When clearing, then setting the mask register, set the mask register, when the input terminal is not in the active status (high status).

• I/O Port

The I/O port is pulled up by an internal memory read signal. Note that the I/O port is read during a transient pull-up state during operation of OSC3.

• LCD Driver

The segment memory is a write-only memory. Its data cannot be directly changed by an ALU instruction (AND, OR, etc.). (Bit manipulation requires a buffer to be provided in RAM for storage of data.)

• REM Circuit

See "Notes on REM circuit" in Section 2.7.

• AMP and SVD

At AMP or SVD power-on, at least 3 ms is required until the output becomes stable. The program must include a waiting time of at least 3 ms before the output is read.

• Interrupt

(1) An interrupt is used for restarting from the halt state. If a halt state is entered with all interrupts disabled, the system will rested from the initial reset state under watchdog timer control.

(2) When an interrupt occurs, hardware resets the interrupt flag placing it in DI mode. After the interrupt has been completed, use software to set the EI mode as necessary.

It is possible to make the nesting level programmable by setting the EI mode at the beginning of the interrupt routine.

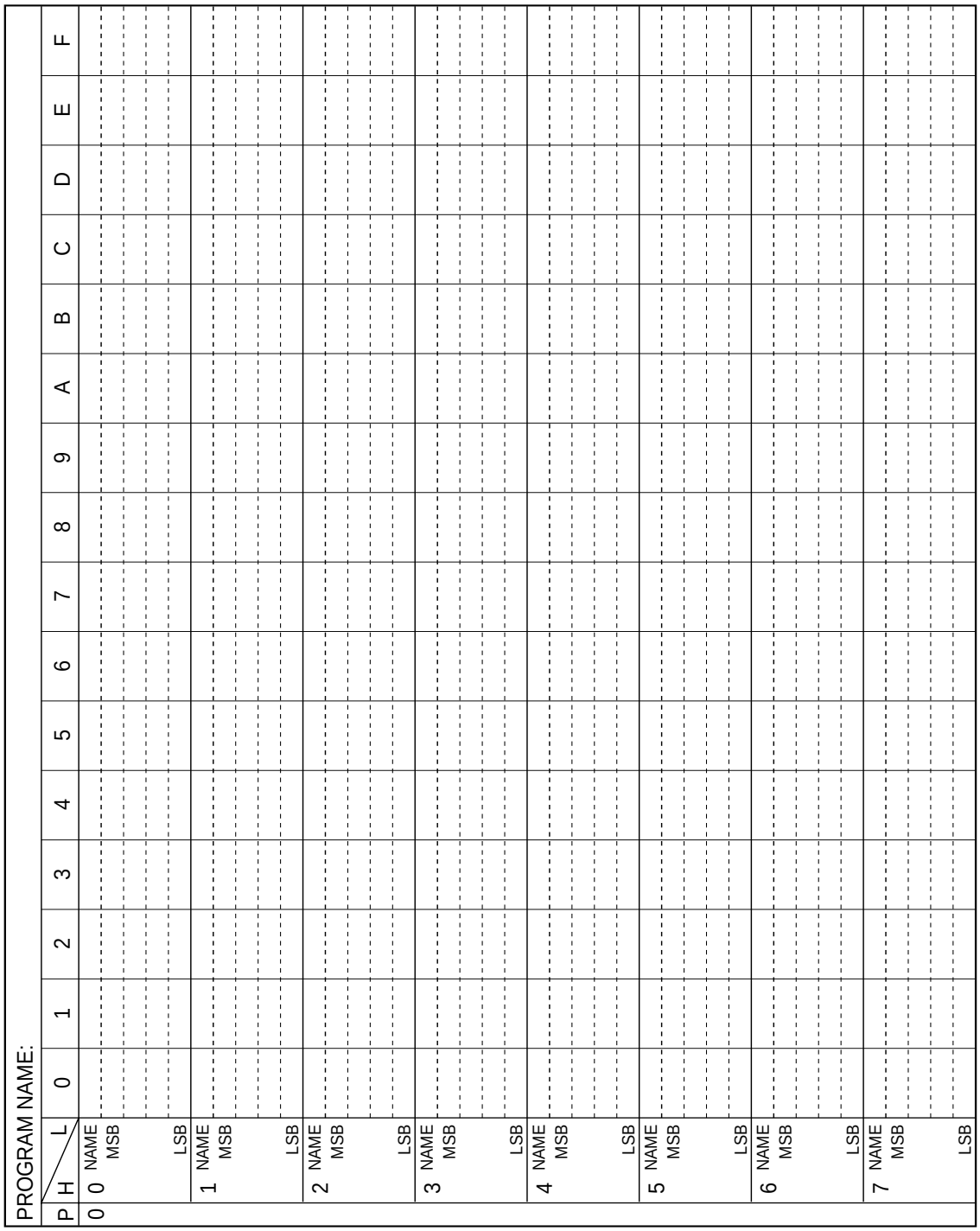
(3) Each interrupt factor flag must be reset before the EI mode is set in the associated interrupt handling routine. (The flag is reset by software when it reads its status.) If the EI mode is set with out resetting the interrupt factor flag and with the interrupt enable flag at 1, the preceding interrupt will occur again.

(4) An interrupt factor flag is reset by software when it reads its status. If simultaneously using interrupts that have interrupt factor flags at the same address, store the flag checks in RAM before doing so. (Checking directly with the FAN instruction will reset all the factor flags at the same address.)

(5) Read the interrupt factor flag in the DI status (interrupt flag = 0). Reading of interrupt factor flags is available at EI, but be careful in the following cases.
If the interrupt mask register value corresponding to the interrupt factor flags to be read is set to 1, an interrupt request will be generated by the interrupt factor flags set timing, or an interrupt request will not be generated. Be very careful when interrupt factor flags are in the same address.

(6) For an input interrupt, the noise reject circuit causes a delay of 0.65 ms (maximum) before the interrupt factor flag is set. To clear the flag after scanning keys with a key matrix, a proper waiting time must be set in the program.

APPENDIX A RAM Map



PROGRAM NAME:

P	H \ L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	8 NAME																
	MSB																
	LSB																
	9 NAME																
	MSB																
	LSB																
	A NAME																
	MSB																
	LSB																
	B NAME																
	MSB																
	LSB																
	C NAME																
	MSB																
	LSB																
	D NAME																
	MSB																
	LSB																
	E NAME																
	MSB																
	LSB																
	F NAME																
	MSB	REMSO	WDRST	REMC	TMRUN	TM03	TM13	SVDDT	RCDIV	RIC3	ROUT1	K03	K13	R03	R13	P03	OPDT
		IREM	TI2	EIREM	ETI2	TM02	TM12	SVDON	RCDUTY	RIC2	ROUT0	K02	K12	R02	R12	P02	OPON
		IK1	TI8	EIK1	ETI8	TM01	TM11	CLKCHG	RT1	RIC1	MF91	K01	K11	R01	R11	P01	IOC
	LSB	IK0	TI32	EIK0	ETI31	TM00	TM10	OSCC	RT0	RIC0	MF90	K00	K10	R00	R10	P00	R20

APPENDIX B Instruction Set

Classification	Mnemonic	Operand	Operation Code												Flag				Clock	Operation
			B	A	9	8	7	6	5	4	3	2	1	0	I	D	Z	C		
Branch instructions	PSET	p	1	1	1	0	0	1	0	p4	p3	p2	p1	p0					5	NBP ← p4, NPP ← p3~p0
	JP	s	0	0	0	0	s7	s6	s5	s4	s3	s2	s1	s0					5	PCB ← NBP, PCP ← NPP, PCS ← s7~s0
		C, s	0	0	1	0	s7	s6	s5	s4	s3	s2	s1	s0					5	PCB ← NBP, PCP ← NPP, PCS ← s7~s0 if C=1
		NC, s	0	0	1	1	s7	s6	s5	s4	s3	s2	s1	s0					5	PCB ← NBP, PCP ← NPP, PCS ← s7~s0 if C=0
		Z, s	0	1	1	0	s7	s6	s5	s4	s3	s2	s1	s0					5	PCB ← NBP, PCP ← NPP, PCS ← s7~s0 if Z=1
		NZ, s	0	1	1	1	s7	s6	s5	s4	s3	s2	s1	s0					5	PCB ← NBP, PCP ← NPP, PCS ← s7~s0 if Z=0
	JPBA		1	1	1	1	1	1	1	0	1	0	0	0					5	PCB ← NBP, PCP ← NPP, PCSH ← B, PCSL ← A
	CALL	s	0	1	0	0	s7	s6	s5	s4	s3	s2	s1	s0					7	M(SP-1) ← PCP, M(SP-2) ← PCSH, M(SP-3) ← PCSL+1 SP ← SP-3, PCP ← NPP, PCS ← s7~s0
	CALZ	s	0	1	0	1	s7	s6	s5	s4	s3	s2	s1	s0					7	M(SP-1) ← PCP, M(SP-2) ← PCSH, M(SP-3) ← PCSL+1 SP ← SP-3, PCP ← 0, PCS ← s7~s0
	RET		1	1	1	1	1	1	0	1	1	1	1	1					7	PCSL ← M(SP), PCSH ← M(SP+1), PCP ← M(SP+2) SP ← SP+3
	RETS		1	1	1	1	1	1	0	1	1	1	1	0					12	PCSL ← M(SP), PCSH ← M(SP+1), PCP ← M(SP+2) SP ← SP+3, PC ← PC+1
	RETD	*l*	0	0	0	1	*l*7	*l*6	*l*5	*l*4	*l*3	*l*2	*l*1	*l*0					12	PCSL ← M(SP), PCSH ← M(SP+1), PCP ← M(SP+2) SP ← SP+3, M(X) ← *l*3~*l*0, M(X+1) ← *l*7~*l*4, X ← X+2
System control instructions	NOP5		1	1	1	1	1	1	1	1	1	0	1	1					5	No operation (5 clock cycles)
	NOP7		1	1	1	1	1	1	1	1	1	1	1	1					7	No operation (7 clock cycles)
	HALT		1	1	1	1	1	1	1	1	1	0	0	0					5	Halt (stop clock)
Index operation instructions	INC	X	1	1	1	0	1	1	1	0	0	0	0	0					5	X ← X+1
		Y	1	1	1	0	1	1	1	1	0	0	0	0					5	Y ← Y+1
	LD	X, x	1	0	1	1	x7	x6	x5	x4	x3	x2	x1	x0					5	XH ← x7~x4, XL ← x3~x0
		Y, y	1	0	0	0	y7	y6	y5	y4	y3	y2	y1	y0					5	YH ← y7~y4, YL ← y3~y0
		XP, r*	1	1	1	0	1	0	0	0	0	0	r1	r0					5	XP ← r
		XH, r	1	1	1	0	1	0	0	0	0	1	r1	r0					5	XH ← r
		XL, r	1	1	1	0	1	0	0	0	1	0	r1	r0					5	XL ← r
		YP, r*	1	1	1	0	1	0	0	1	0	0	r1	r0					5	YP ← r
		YH, r	1	1	1	0	1	0	0	1	0	1	r1	r0					5	YH ← r
		YL, r	1	1	1	0	1	0	0	1	1	0	r1	r0					5	YL ← r
		r, XP*	1	1	1	0	1	0	1	0	0	0	r1	r0					5	r ← XP
		r, XH	1	1	1	0	1	0	1	0	0	1	r1	r0					5	r ← XH
		r, XL	1	1	1	0	1	0	1	0	1	0	r1	r0					5	r ← XL
		r, YP*	1	1	1	0	1	0	1	1	0	0	r1	r0					5	r ← YP
		r, YH	1	1	1	0	1	0	1	1	0	1	r1	r0					5	r ← YH
		r, YL	1	1	1	0	1	0	1	1	1	0	r1	r0					5	r ← YL
	ADC	XH, i	1	0	1	0	0	0	0	0	i3	i2	i1	i0			↕	↕	7	XH ← XH+i3~i0+C
		XL, i	1	0	1	0	0	0	0	1	i3	i2	i1	i0			↕	↕	7	XL ← XL+i3~i0+C
		YH, i	1	0	1	0	0	0	1	0	i3	i2	i1	i0			↕	↕	7	YH ← YH+i3~i0+C
		YL, i	1	0	1	0	0	0	1	1	i3	i2	i1	i0			↕	↕	7	YL ← YL+i3~i0+C

"*" mean "not in S1C621A0".

Classification	Mnemonic	Operand	Operation Code												Flag				Clock	Operation
			B	A	9	8	7	6	5	4	3	2	1	0	I	D	Z	C		
Index operation instructions	CP	XH, i	1	0	1	0	0	1	0	0	i3	i2	i1	i0			↕	↕	7	XH-i3~i0
		XL, i	1	0	1	0	0	1	0	1	i3	i2	i1	i0			↕	↕	7	XL-i3~i0
		YH, i	1	0	1	0	0	1	1	0	i3	i2	i1	i0			↕	↕	7	YH-i3~i0
		YL, i	1	0	1	0	0	1	1	1	i3	i2	i1	i0			↕	↕	7	YL-i3~i0
Data transfer instructions	LD	r, i	1	1	1	0	0	0	r1	r0	i3	i2	i1	i0					5	r←i3~i0
		r, q	1	1	1	0	1	1	0	0	r1	r0	q1	q0					5	r←q
		A, Mn	1	1	1	1	1	0	1	0	n3	n2	n1	n0					5	A←M(n3~n0)
		B, Mn	1	1	1	1	1	0	1	1	n3	n2	n1	n0					5	B←M(n3~n0)
		Mn, A	1	1	1	1	1	0	0	0	n3	n2	n1	n0					5	M(n3~n0)←A
		Mn, B	1	1	1	1	1	0	0	1	n3	n2	n1	n0					5	M(n3~n0)←B
	LDPX	MX, i	1	1	1	0	0	1	1	0	i3	i2	i1	i0					5	M(X)←i3~i0, X←X+1
		r, q	1	1	1	0	1	1	1	0	r1	r0	q1	q0					5	r←q, X←X+1
	LDPY	MY, i	1	1	1	0	0	1	1	1	i3	i2	i1	i0					5	M(Y)←i3~i0, Y←Y+1
		r, q	1	1	1	0	1	1	1	1	r1	r0	q1	q0					5	r←q, Y←Y+1
	LBPX	MX, *l*	1	0	0	1	*l*7	*l*6	*l*5	*l*4	*l*3	*l*2	*l*1	*l*0					5	M(X)←*l*3~*l*0, M(X+1)←*l*7~*l*4, X←X+2
Flag operation instructions	SET	F, i	1	1	1	1	0	1	0	0	i3	i2	i1	i0	↑	↑	↑	↑	7	F←F∨i3~i0
	RST	F, i	1	1	1	1	0	1	0	1	i3	i2	i1	i0	↓	↓	↓	↓	7	F←F∧i3~i0
	SCF		1	1	1	1	0	1	0	0	0	0	0	1				↑	7	C←1
	RCF		1	1	1	1	0	1	0	1	1	1	1	0				↓	7	C←0
	SZF		1	1	1	1	0	1	0	0	0	0	1	0			↑		7	Z←1
	RZF		1	1	1	1	0	1	0	1	1	1	0	1			↓		7	Z←0
	SDF		1	1	1	1	0	1	0	0	0	1	0	0		↑			7	D←1 (Decimal Adjuster ON)
	RDF		1	1	1	1	0	1	0	1	1	0	1	1		↓			7	D←0 (Decimal Adjuster OFF)
	EI		1	1	1	1	0	1	0	0	1	0	0	0	↑				7	I ←1 (Enables Interrupt)
	DI		1	1	1	1	0	1	0	1	0	1	1	1	↓				7	I ←0 (Disables Interrupt)
Stack operation instructions	INC	SP	1	1	1	1	1	1	0	1	1	0	1	1					5	SP← SP+1
	DEC	SP	1	1	1	1	1	1	0	0	1	0	1	1					5	SP← SP-1
	PUSH	r	1	1	1	1	1	1	0	0	0	0	r1	r0					5	SP← SP-1, M(SP)←r
		XP *	1	1	1	1	1	1	0	0	0	1	0	0					5	SP← SP-1, M(SP)←XP
		XH	1	1	1	1	1	1	0	0	0	1	0	1					5	SP← SP-1, M(SP)←XH
		XL	1	1	1	1	1	1	0	0	0	1	1	0					5	SP← SP-1, M(SP)←XL
		YP *	1	1	1	1	1	1	0	0	0	1	1	1					5	SP← SP-1, M(SP)←YP
		YH	1	1	1	1	1	1	0	0	1	0	0	0					5	SP← SP-1, M(SP)←YH
		YL	1	1	1	1	1	1	0	0	1	0	0	1					5	SP← SP-1, M(SP)←YL
		F	1	1	1	1	1	1	0	0	1	0	1	0					5	SP← SP-1, M(SP)←F
	POP	r	1	1	1	1	1	1	0	1	0	0	r1	r0					5	r←M(SP), SP←SP+1
		XP *	1	1	1	1	1	1	0	1	0	1	0	0					5	XP←M(SP), SP←SP+1
		XH	1	1	1	1	1	1	0	1	0	1	0	1					5	XH←M(SP), SP←SP+1
		XL	1	1	1	1	1	1	0	1	0	1	1	0					5	XL←M(SP), SP←SP+1
		YP *	1	1	1	1	1	1	0	1	0	1	1	1					5	YP←M(SP), SP←SP+1

"*" mean "not in S1C621A0".

Classification	Mnemonic	Operand	Operation Code												Flag				Clock	Operation
			B	A	9	8	7	6	5	4	3	2	1	0	I	D	Z	C		
Stack operation instructions	POP	YH	1	1	1	1	1	1	0	1	1	0	0	0					5	YH← M(SP), SP← SP+1
		YL	1	1	1	1	1	1	0	1	1	0	0	1					5	YL ← M(SP), SP← SP+1
		F	1	1	1	1	1	1	0	1	1	0	1	0	↕	↕	↕	↕	5	F← M(SP), SP← SP+1
	LD	SPH, r	1	1	1	1	1	1	1	0	0	0	r1	r0					5	SPH← r
		SPL, r	1	1	1	1	1	1	1	1	0	0	r1	r0					5	SPL ← r
		r, SPH	1	1	1	1	1	1	1	0	0	1	r1	r0					5	r← SPH
		r, SPL	1	1	1	1	1	1	1	1	0	1	r1	r0					5	r← SPL
Arithmetic instructions	ADD	r, i	1	1	0	0	0	0	r1	r0	i3	i2	i1	i0	★	↕	↕		7	r← r+i3~i0
		r, q	1	0	1	0	1	0	0	0	r1	r0	q1	q0	★	↕	↕		7	r← r+q
	ADC	r, i	1	1	0	0	0	1	r1	r0	i3	i2	i1	i0	★	↕	↕		7	r← r+i3~i0+C
		r, q	1	0	1	0	1	0	0	1	r1	r0	q1	q0	★	↕	↕		7	r← r+q+C
	SUB	r, q	1	0	1	0	1	0	1	0	r1	r0	q1	q0	★	↕	↕		7	r← r-q
	SBC	r, i	1	1	0	1	0	1	r1	r0	i3	i2	i1	i0	★	↕	↕		7	r← r-i3~i0-C
		r, q	1	0	1	0	1	0	1	1	r1	r0	q1	q0	★	↕	↕		7	r← r-q-C
	AND	r, i	1	1	0	0	1	0	r1	r0	i3	i2	i1	i0		↕			7	r← r∧ i3~i0
		r, q	1	0	1	0	1	1	0	0	r1	r0	q1	q0		↕			7	r← r∧ q
	OR	r, i	1	1	0	0	1	1	r1	r0	i3	i2	i1	i0		↕			7	r← r∨i3~i0
		r, q	1	0	1	0	1	1	0	1	r1	r0	q1	q0		↕			7	r← r∨q
	XOR	r, i	1	1	0	1	0	0	r1	r0	i3	i2	i1	i0		↕			7	r← r∀i3~i0
		r, q	1	0	1	0	1	1	1	0	r1	r0	q1	q0		↕			7	r← r∀q
	CP	r, i	1	1	0	1	1	1	r1	r0	i3	i2	i1	i0		↕	↕		7	r-i3~i0
		r, q	1	1	1	1	0	0	0	0	r1	r0	q1	q0		↕	↕		7	r-q
	FAN	r, i	1	1	0	1	1	0	r1	r0	i3	i2	i1	i0		↕			7	r∧i3~i0
		r, q	1	1	1	1	0	0	0	1	r1	r0	q1	q0		↕			7	r∧q
	RLC	r	1	0	1	0	1	1	1	1	r1	r0	r1	r0		↕	↕		7	d3← d2, d2← d1, d1← d0, d0 ←C, C← d3
	RRC	r	1	1	1	0	1	0	0	0	1	1	r1	r0		↕	↕		5	d3← C, d2 ←d3, d1 ←d2, d0 ←d1, C← d0
	INC	Mn	1	1	1	1	0	1	1	0	n3	n2	n1	n0		↕	↕		7	M(n3~n0) ←M(n3~n0)+1
	DEC	Mn	1	1	1	1	0	1	1	1	n3	n2	n1	n0		↕	↕		7	M(n3~n0) ←M(n3~n0)-1
	ACPX	MX, r	1	1	1	1	0	0	1	0	1	0	r1	r0	★	↕	↕		7	M(X) ←M(X)+r+C, X ←X+1
	ACPY	MY, r	1	1	1	1	0	0	1	0	1	1	r1	r0	★	↕	↕		7	M(Y) ←M(Y)+r+C, Y ←Y+1
	SCPX	MX, r	1	1	1	1	0	0	1	1	1	0	r1	r0	★	↕	↕		7	M(X) ←M(X)-r-C, X← X+1
	SCPY	MY, r	1	1	1	1	0	0	1	1	1	1	r1	r0	★	↕	↕		7	M(Y) ←M(Y)-r-C, Y ← Y+1
	NOT	r	1	1	0	1	0	0	r1	r0	1	1	1	1		↕			7	r←$\overline{r}$

Abbreviations used in the explanations have the following meanings.

Symbols associated with registers and memory

A ..... A register
B ..... B register
X ..... XHL register (low order eight bits of index register IX)
Y ..... YHL register (low order eight bits of index register IY)
XH ..... XH register (high order four bits of XHL register)
XL ..... XL register (low order four bits of XHL register)
YH ..... YH register (high order four bits of YHL register)
YL ..... YL register (low order four bits of YHL register)
XP ..... XP register (high order four bits of index register IX)
YP ..... YP register (high order four bits of index register IY)
SP ..... Stack pointer SP
SPH ..... High-order four bits of stack pointer SP
SPL ..... Low-order four bits of stack pointer SP
MX, M(X) .. Data memory whose address is specified with index register IX
MY, M(Y)... Data memory whose address is specified with index register IY
Mn, M(n) .. Data memory address 000H–00FH (address specified with immediate data n of 00H–0FH)
M(SP) ..... Data memory whose address is specified with stack pointer SP
r, q ..... Two-bit register code
r, q is two-bit immediate data; according to the contents of these bits, they indicate registers A, B, and MX and MY (data memory whose addresses are specified with index registers IX and IY)

r		q		Registers specified
r1	r0	q1	q0	
0	0	0	0	A
0	1	0	1	B
1	0	1	0	MX
1	1	1	1	MY

Symbols associated with program counter

NBP ..... New bank pointer
NPP ..... New page pointer
PCB ..... Program counter bank
PCP ..... Program counter page
PCS ..... Program counter step
PCSH .. Four high order bits of PCS
PCSL ... Four low order bits of PCS

Symbols associated with flags

F ..... Flag register (I, D, Z, C)
C ..... Carry flag
Z ..... Zero flag
D ..... Decimal flag
I ..... Interrupt flag
↓ ..... Flag reset
↑ ..... Flag set
↕ ..... Flag set or reset

Associated with immediate data

p ..... Five-bit immediate data or label 00H–1FH
s ..... Eight-bit immediate data or label 00H–0FFH
l ..... Eight-bit immediate data 00H–0FFH
i ..... Four-bit immediate data 00H–0FH

Associated with arithmetic and other operations

+ ..... Add
- ..... Subtract
∧ ..... Logical AND
∨ ..... Logical OR
∀ ..... Exclusive-OR
★ ..... Add-subtract instruction for decimal operation when the D flag is set

APPENDIX C Cross-assembler Pseudo-instruction List

Item No.	Pseudo-instruction	Meaning	Example of Use
1	EQU (Equation)	To allocate data to label	`ABC EQU 9` `BCD EQU ABC+1`
2	ORG (Origin)	To define location counter	`ORG 100H` `ORG 256`
3	SET (Set)	To allocate data to label (data can be changed)	`ABC SET 0001H` `ABC SET 0002H`
4	DW (Define Word)	To define ROM data	`ABC DW 'AB'` `BCD DW 0FFBH`
5	PAGE (Page)	To define boundary of page	`PAGE 1H` `PAGE 15`
6	SECTION (Section)	To define boundary of section	`SECTION`
7	END (End)	To terminate assembly	`END`
8	MACRO (Macro)	To define macro	`CHECK MACRO DATA`

9	LOCAL
	(Local)
		To make local specification of label during macro definition	`LOCAL LOOP`
			`LOOP CP MX,DATA`
			`JP NZ,LOOP`

10	ENDM (End Macro)	To end macro definition	`ENDM` `CHECK 1`

APPENDIX D Table of the ICE Commands

Item No.	Function	Command Format	Outline of Operation
1	Assemble	#A,a ↵	Assemble command mnemonic code and store at address "a"
2	Disassemble	#L,a1,a2 ↵	Contents of addresses a1 to a2 are disassembled and displayed
3	Dump	#DP,a1,a2 ↵	Contents of program area a1 to a2 are displayed
		#DD,a1,a2 ↵	Content of data area a1 to a2 are displayed
4	Fill	#FP,a1,a2,d ↵	Data d is set in addresses a1 to a2 (program area)
		#FD,a1,a2,d ↵	Data d is set in addresses a1 to a2 (data area)
5	Set Run Mode	#G,a ↵	Program is executed from the "a" address
		#TIM ↵	Execution time and step counter selection
		#OTF ↵	On-the-fly display selection
6	Trace	#T,a,n ↵	Executes program while displaying results of step instruction from "a" address
		#U,a,n ↵	Displays only the final step of #T,a,n
7	Break	#BA,a ↵	Sets Break at program address "a"
		#BAR,a ↵	Breakpoint is canceled
		#BD ↵	Break condition is set for data RAM
		#BDR ↵	Breakpoint is canceled
		#BR ↵	Break condition is set for Evaluation Board CPU internal registers
		#BRR ↵	Breakpoint is canceled
		#BM ↵	Combined break conditions set for program data RAM address and registers
		#BMR ↵	Cancel combined break conditions for program data ROM address and registers
		#BRES ↵	All break conditions canceled
		#BC ↵	Break condition displayed
		#BE ↵	Enter break enable mode
		#BSYN ↵	Enter break disable mode
		#BT ↵	Set break stop/trace modes
		#BRKSEL,REM ↵	Set BA condition clear/remain modes
8	Move	#MP,a1,a2,a3 ↵	Contents of program area addresses a1 to a2 are moved to addresses a3 and after
		#MD,a1,a2,a3 ↵	Contents of data area addresses a1 to a2 are moved to addresses a3 and after
9	Data Set	#SP,a ↵	Data from program area address "a" are written to memory
		#SD,a ↵	Data from data area address "a" are written to memory
10	Change CPU Internal Registers	#DR ↵	Display Evaluation Board CPU internal registers
		#SR ↵	Set Evaluation Board CPU internal registers
		#I ↵	Reset Evaluation Board CPU
		#DXY ↵	Display X, Y, MX and MY
		#SXY ↵	Set data for X and Y display and MX, MY

Item No.	Function	Command Format	Outline of Operation
11	History	#H,p1,p2 ↵	Display history data for pointer 1 and pointer 2
		#HB ↵	Display upstream history data
		#HG ↵	Display 21 line history data
		#HP ↵	Display history pointer
		#HPS,a ↵	Set history pointer
		#HC,S/C/E ↵	Sets up the history information acquisition before (S), before/after (C) and after (E)
		#HA,a1,a2 ↵	Sets up the history information acquisition from program area a1 to a2
		#HAR,a1,a2 ↵	Sets up the prohibition of the history information acquisition from program area a1 to a2
		#HAD ↵	Indicates history acquisition program area
		#HS,a ↵	Retrieves and indicates the history information which executed a program address "a"
		#HSW,a ↵ #HSR,a ↵	Retrieves and indicates the history information which wrote or read the data area address "a"
12	File	#RF,file ↵	Move program file to memory
		#RFD,file ↵	Move data file to memory
		#VF,file ↵	Compare program file and contents of memory
		#VFD,file ↵	Compare data file and contents of memory
		#WF,file ↵	Save contents of memory to program file
		#WFD,file ↵	Save contents of memory to data file
		#CL,file ↵	Load ICE set condition from file
		#CS,file ↵	Save ICE set condition to file
13	Coverage	#CVD ↵	Indicates coverage information
		#CVR ↵	Clears coverage information
14	ROM Access	#RP ↵	Move contents of ROM to program memory
		#VP ↵	Compare contents of ROM with contents of program memory
		#ROM ↵	Set ROM type
15	Terminate ICE	#Q ↵	Terminate ICE and return to operating system control
16	Command Display	#HELP ↵	Display ICE instruction
17	Self Diagnosis	#CHK ↵	Report results of ICE self diagnostic test

↵ means press the RETURN key.

EPSON International Sales Operations

AMERICA

EPSON ELECTRONICS AMERICA, INC.

- HEADQUARTERS -
150 River Oaks Parkway
San Jose, CA 95134, U.S.A.
Phone: +1-408-922-0200 Fax: +1-408-922-0238

- SALES OFFICES -

West
1960 E. Grand Avenue
El Segundo, CA 90245, U.S.A.
Phone: +1-310-955-5300 Fax: +1-310-955-5400

Central
101 Virginia Street, Suite 290
Crystal Lake, IL 60014, U.S.A.
Phone: +1-815-455-7630 Fax: +1-815-455-7633

Northeast
301 Edgewater Place, Suite 120
Wakefield, MA 01880, U.S.A.
Phone: +1-781-246-3600 Fax: +1-781-246-5443

Southeast
3010 Royal Blvd. South, Suite 170
Alpharetta, GA 30005, U.S.A.
Phone: +1-877-EEA-0020 Fax: +1-770-777-2637

EUROPE

EPSON EUROPE ELECTRONICS GmbH

- HEADQUARTERS -
Riesstrasse 15
80992 Munich, GERMANY
Phone: +49-(0)89-14005-0 Fax: +49-(0)89-14005-110

SALES OFFICE
Altstadtstrasse 176
51379 Leverkusen, GERMANY
Phone: +49-(0)2171-5045-0 Fax: +49-(0)2171-5045-10

UK BRANCH OFFICE
Unit 2.4, Doncastle House, Doncastle Road
Bracknell, Berkshire RG12 8PE, ENGLAND
Phone: +44-(0)1344-381700 Fax: +44-(0)1344-381701

FRENCH BRANCH OFFICE
1 Avenue de l' Atlantique, LP 915 Les Conquerants
Z.A. de Courtaboeuf 2, F-91976 Les Ulis Cedex, FRANCE
Phone: +33-(0)1-64862350 Fax: +33-(0)1-64862355

BARCELONA BRANCH OFFICE

Barcelona Design Center
Edificio Prima Sant Cugat
Avda. Alcalde Barrils num. 64-68
E-08190 Sant Cugat del Vallès, SPAIN
Phone: +34-93-544-2490 Fax: +34-93-544-2491

ASIA

EPSON (CHINA) CO., LTD.
28F, Beijing Silver Tower 2# North RD DongSanHuan
ChaoYang District, Beijing, CHINA
Phone: 64106655 Fax: 64107319

SHANGHAI BRANCH
4F, Bldg., 27, No. 69, Gui Jing Road
Caohejing, Shanghai, CHINA
Phone: 21-6485-5552 Fax: 21-6485-0775

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.
10F, No. 287, Nanking East Road, Sec. 3
Taipei
Phone: 02-2717-7360 Fax: 02-2712-9164
Telex: 24444 EPSONTB

HSINCHU OFFICE
13F-3, No. 295, Kuang-Fu Road, Sec. 2
HsinChu 300
Phone: 03-573-9900 Fax: 03-573-9169

EPSON SINGAPORE PTE., LTD.
No. 1 Temasek Avenue, #36-00
Millenia Tower, SINGAPORE 039192
Phone: +65-337-7911 Fax: +65-334-2716

SEIKO EPSON CORPORATION KOREA OFFICE
50F, KLI 63 Bldg., 60 Yoido-dong
Youngdeungpo-Ku, Seoul, 150-763, KOREA
Phone: 02-784-6027 Fax: 02-767-3677

SEIKO EPSON CORPORATION
ELECTRONIC DEVICES MARKETING DIVISION

Electronic Device Marketing Department
IC Marketing & Engineering Group
421-8, Hino, Hino-shi, Tokyo 191-8501, JAPAN
Phone: +81-(0)42-587-5816 Fax: +81-(0)42-587-5624

ED International Marketing Department Europe & U.S.A.
421-8, Hino, Hino-shi, Tokyo 191-8501, JAPAN
Phone: +81-(0)42-587-5812 Fax: +81-(0)42-587-5564

ED International Marketing Department Asia
421-8, Hino, Hino-shi, Tokyo 191-8501, JAPAN
Phone: +81-(0)42-587-5814 Fax: +81-(0)42-587-5110

In pursuit of **"Saving" Technology**, Epson electronic devices.
Our lineup of semiconductors, liquid crystal displays and quartz devices
assists in creating the products of our customers' dreams.
Epson IS energy savings.

S1C621A0
Technical Manual

SEIKO EPSON CORPORATION
ELECTRONIC DEVICES MARKETING DIVISION

■ EPSON Electronic Devices Website

http://www.epson.co.jp/device/

First issue March, 1991
Printed May, 2001 in Japan ⓂA