MF362-04

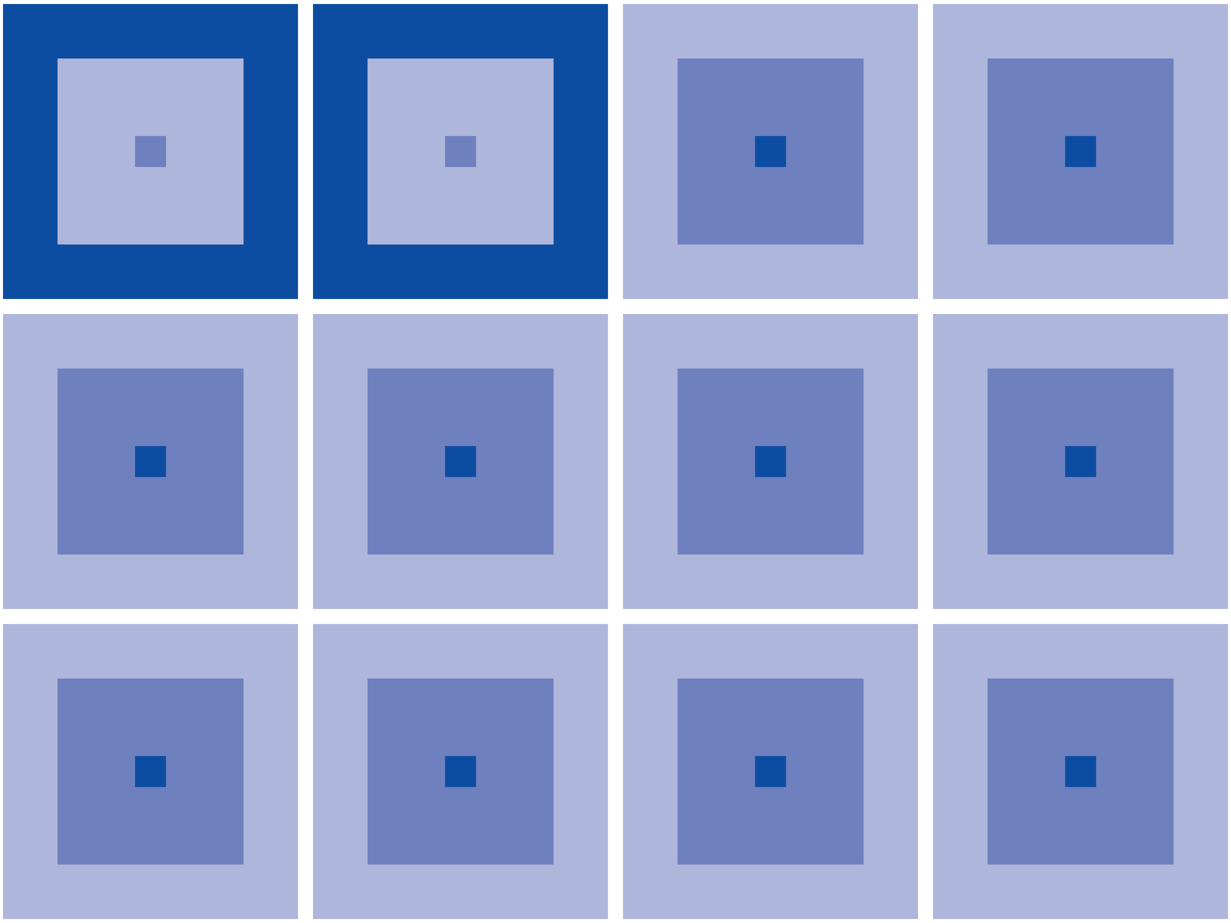
EPSON

CMOS 4-BIT SINGLE CHIP MICROCOMPUTER

S1C62N81

Technical Manual

S1C62N81 Technical Hardware/S1C62N81 Technical Software



ENERGY SAVING EPSON

SEIKO EPSON CORPORATION

NOTICE

Royalty on Copyrighted Musical Pieces
When a musical selection under copyright is created in the melody ROM section of EPSON's S1C62N81 and then marketed in your country or any other country, permission to use the copyright is required in accordance with the Copyright Law.
For such purpose, in connection with the contract we have concluded with the Japan Music Copyright Association regarding copyrights, customers using the S1C62N81 are required to apply with us before starting any software developments, regardless of whether the melody ROM section will be used or not. We shall process the necessary copyrights based on said application.
Due to the above-stated reasons, we shall bear no responsibility whatsoever in the following cases:

- *When the musical selection applied with us differs from the actual musical selection used;*
- *When no application has been made with us in spite of the fact that musical selection has been incorporated in the ROM section (this also applies to pirated musical pieces).*

Moreover, please take note that there are exceptional cases in which processing anew of copyrights may be required in accordance with the laws of the country of destination of the marketed product(s).

PREFACE

This manual is individualy described about the hardware and the software of the S1C62N81.

I. S1C62N81 Technical Hardware

This part explains the function of the S1C62N81, the circuit configurations, and details the controlling method.

II. S1C62N81 Technical Software

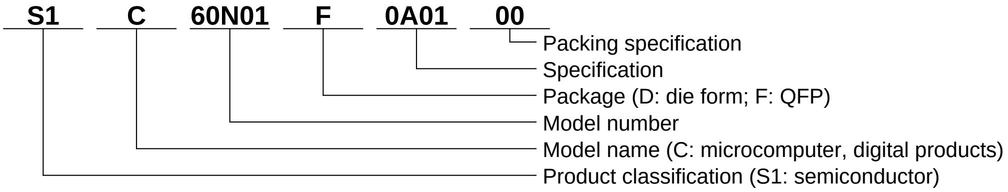
This part explains the programming method of the S1C62N81.

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

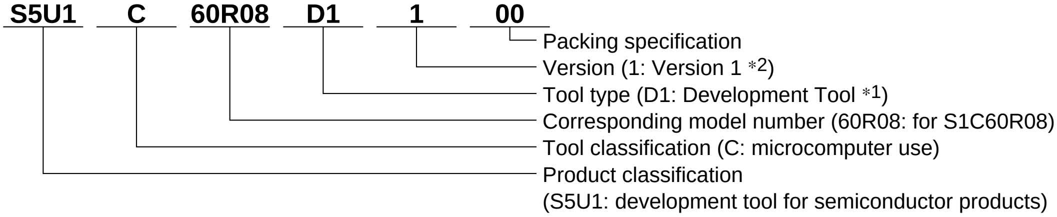
Devices



S1 C 60N01 F 0A01 00

- Packing specification
- Specification
- Package (D: die form; F: QFP)
- Model number
- Model name (C: microcomputer, digital products)
- Product classification (S1: semiconductor)

Development tools



S5U1 C 60R08 D1 1 00

- Packing specification
- Version (1: Version 1 *2)
- Tool type (D1: Development Tool *1)
- Corresponding model number (60R08: for S1C60R08)
- Tool classification (C: microcomputer use)
- Product classification (S5U1: development tool for semiconductor products)

*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. S1C62N81 Technical Hardware

CONTENTS

Hardware

CHAPTER 1 INTRODUCTION

Each member of the S1C62N81 Series of single chip microcomputers feature a 4-bit S1C6200 core CPU, 1,024 words of ROM (12 bits per word), 96 words of RAM (4 bits per word), an LCD driver, 5 bits for input ports (K00–K03 and K10), 7 bits for output ports (R00–R03 and R10–R12), one 4-bit I/O port (P00–P03), two timer (clock timer and stopwatch timer), and a melody generator.

Because of their low voltage operation and low power consumption, the S1C62N81 Series are ideal for a wide range of applications, and are especially suitable for battery-driven systems with a melody.

1.1 Configuration

The S1C62N81 Series are configured as follows, depending on the supply voltage and oscillation circuits.

Table 1.1.1 Configuration of the S1C62N81 Series

Model	Supply Voltage	Oscillation Circuits
S1C62N81	3.0 V	Crystal
S1C62L81	1.5 V	Crystal
S1C62A81	3.0 V	CR
S1C62B81	1.5 V	CR

1.2 Features

Built-in oscillation circuit	Crystal or CR oscillation circuit, 32.768 kHz (typ.)	
Instruction set	100 instructions	
ROM capacity	1,024 words ×12 bits	
RAM capacity (data RAM)	96 words × 4 bits	
Input port	5 bits (Supplementary pull-down resistors may be used *1)	
Output port	7 bits (*2), 1 bit melody output (Piezo buzzer can be driven directry by mask option)	
Input/output port	4 bits	
LCD driver	26 segments × 4 common duty (or 3 common duty)	
Timer	2 systems: clock timer/stopwatch timer	
Analog comparator	1 channel	
Melody generation circuit	Single-tone source generation, 15 tone intervals among 3 octaves, 8 notes, 2 tempos among 16 types Optional number of melodies, within ROM capacity (80 words) Supplementary envelope output may be used (*1) Piezo buzzer can be driven directry (*1)	
Battery voltage low detection circuit (BLD)	1.2 V / 2.4 V (*1)	
Interrupts:		
External interrupt	Input port interrupt	2 systems
Internal interrupt	Timer interrupt	2 systems
	Melody interrupt	1 system
Supply voltage	1.5 V (0.9–3.5 V) S1C62L81, S1C62B81 3.0 V (1.8–3.5 V) S1C62N81, S1C62A81	
Current consumption (typ.)	1.0 µA (CLK = 32.768 kHz, when halted) 3.0 µA (CLK = 32.768 kHz, when executing)	
Supply form	64-pin QFP (plastic) or chip	

Note **1. May be selected with mask option*

**2. FOUT output is possible through mask option selections*

1.3 Block Diagram

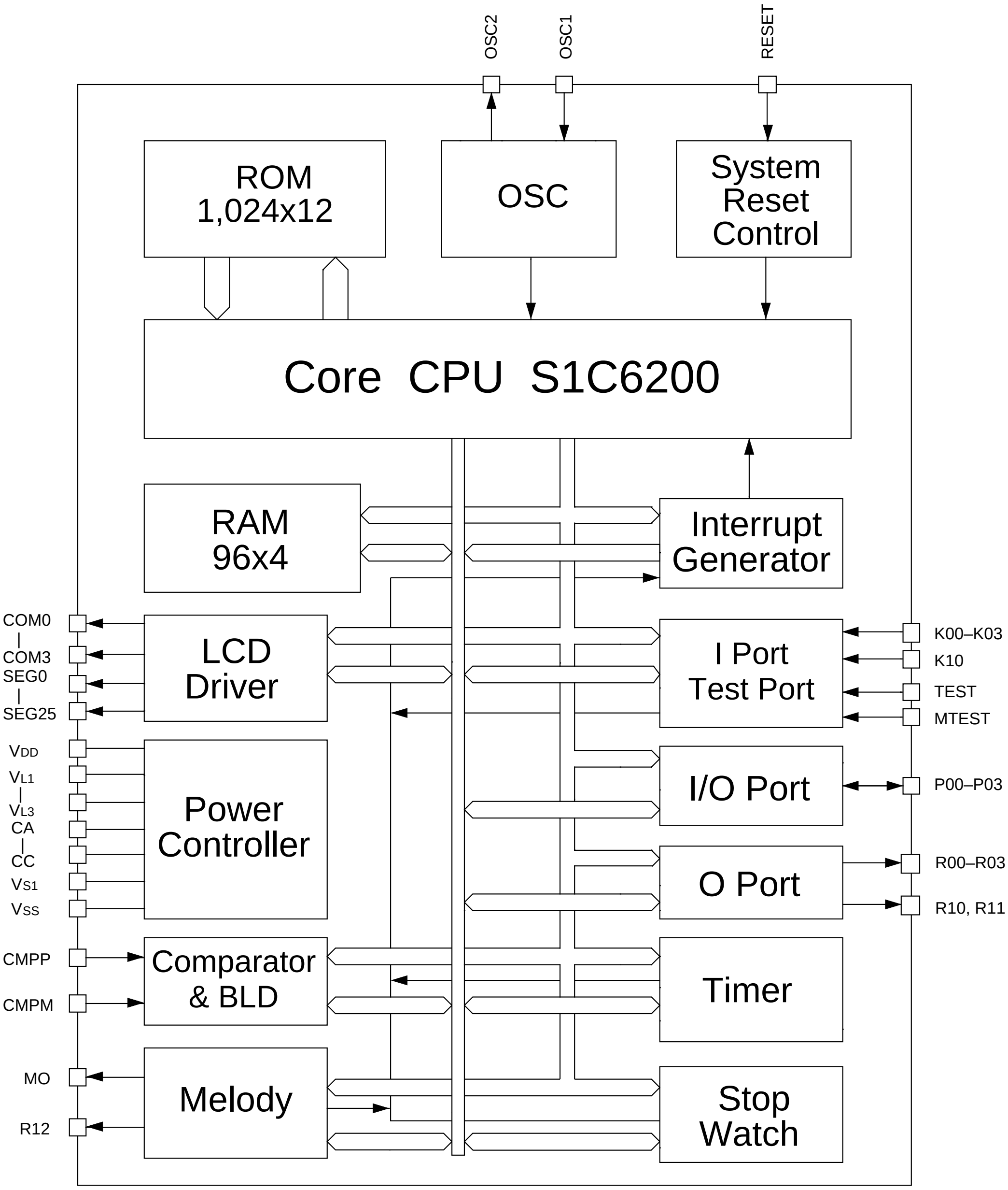


Fig. 1.3.1
Block diagram

1.4 Pin Layout Diagram

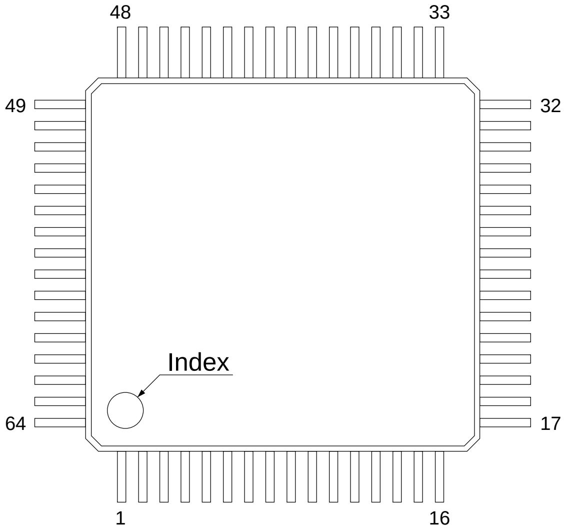


Fig. 1.4.1
Pin assignment

Pin No	Pin Name	Pin No	Pin Name	Pin No	Pin Name	Pin No	Pin Name
1	COM1	17	TEST	33	P01	49	R00
2	COM2	18	SEG12	34	P02	50	R01
3	COM3	19	SEG13	35	P03	51	R02
4	SEG25	20	SEG14	36	CMPM	52	R03
5	SEG0	21	SEG15	37	CMPP	53	V_{S1}
6	SEG1	22	SEG16	38	MTEST	54	V_{DD}
7	SEG2	23	SEG17	39	RESET	55	V_{SS}
8	SEG3	24	SEG18	40	K00	56	OSC2
9	SEG4	25	SEG19	41	K01	57	OSC1
10	SEG5	26	SEG20	42	K02	58	V_{L3}
11	SEG6	27	SEG21	43	K03	59	V_{L2}
12	SEG7	28	SEG22	44	K10	60	V_{L1}
13	SEG8	29	SEG23	45	R10	61	CC
14	SEG9	30	SEG24	46	R11	62	CB
15	SEG10	31	N.C.	47	R12	63	CA
16	SEG11	32	P00	48	MO	64	COM0

(N.C. = No Connection)

1.5 Pin Description

Table 1.5.1 Pin description

Terminal Name	Pin No.	Input/Output	Function
V_{DD}	54	(I)	Power source (+) terminal
V_{SS}	55	(I)	Power source (-) terminal
V_{S1}	53	O	Oscillation and internal logic system regulated voltage output terminal
V_{L1}	60	O	LCD system regulated voltage output terminal (approx. -1.05V)
V_{L2}	59	O	LCD system booster output terminal (V_{L1} × 2)
V_{L3}	58	O	LCD system booster output terminal (V_{L1} × 3)
CA–CC	61, 62, 63	–	Booster capacitor connecting terminal
OSC1	57	I	Crystal or CR oscillation input terminal
OSC2	56	O	Crystal or CR oscillation output terminal
K00–K10	40–44	I	Input terminal
P00–P03	32–35	I/O	I/O terminal
R00–R03	49–52	O	Output terminal
R10	45	O	Output terminal (FOUT output available through mask option selection)
R11	46	O	Ouput terminal
R12	47	O	Output terminal (melody inverted output and envelope function available through mask option selection)
MO	48	O	Melody signal output terminal
CMPP	37	I	Analog comparator non-inverted input terminal
CMPM	36	I	Analog comparator inverted input terminal
SEG0–25	4–17 19–30	O	LCD segment output terminal (convertible to DC output terminal by mask option)
COM0–3	64, 1–3	O	LCD common output terminal
RESET	39	I	Initial setting input terminal
TEST	18	I	Test input terminal
MTEST	38	I	Melody test input terminal

CHAPTER 2 POWER SUPPLY AND INITIAL RESET

2.1 Power Supply

With a single external power supply (*1) supplied to V_{DD} through V_{SS}, the S1C62N81 Series generate the necessary internal voltages with the regulated voltage circuit (<V_{S1}> for oscillators and internal circuit, <V_{L1}> for LCDs) and the voltage booster circuit (<V_{L2}, V_{L3}> for LCDs).
Figure 2.1.1 shows the power supply configuration.

*1 Supply voltage: S1C62N81/62A81...3.0 V
S1C62L81/62B81...1.5 V

Note

- *External loads cannot be driven by the output voltage of the regulated voltage circuit and voltage booster circuit.*
- *See Chapter 6, "ELECTRICAL CHARACTERISTICS", for voltage values.*

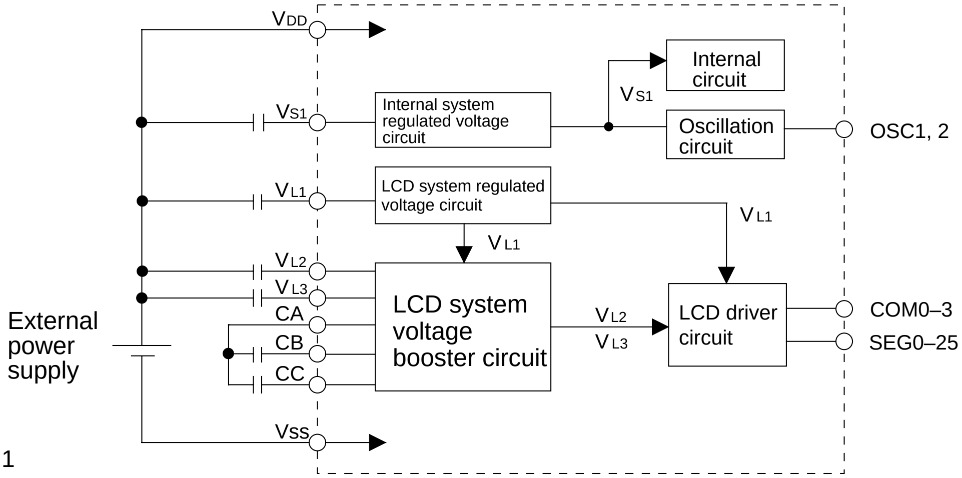


Fig. 2.1.1 Configuration of power supply

2.2 Initial Reset

To initialize the S1C62N81 Series circuits, an initial reset must be executed. There are three ways of doing this.

(1) Initial reset by the oscillation detection circuit
(2) External initial reset via the RESET pin
(3) External initial reset by simultaneous high input to pins K00–K03 (depending on mask option)

Figure 2.2.1 shows the configuration of the initial reset circuit.

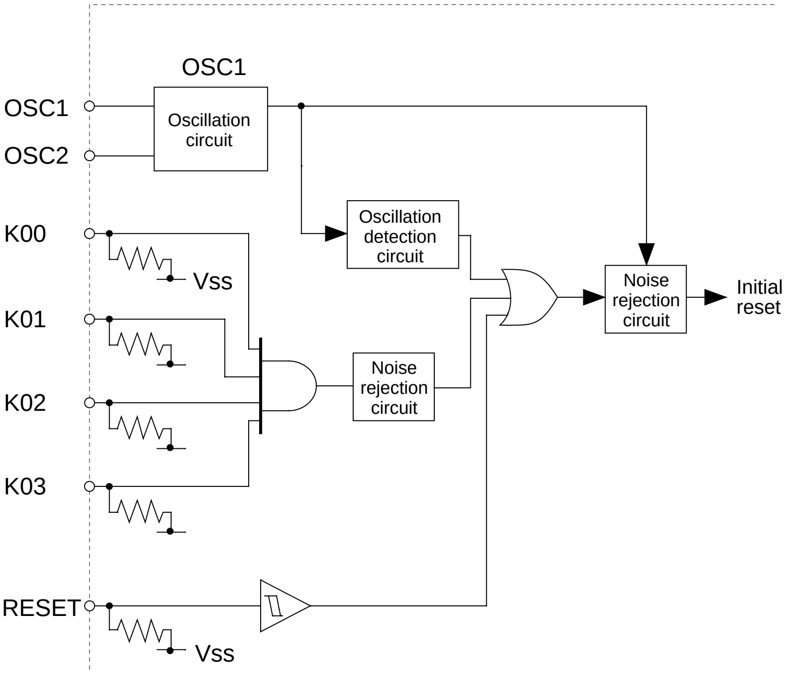


Fig. 2.2.1
Configuration of
initial reset circuit

Oscillation detection circuit

The oscillation detection circuit outputs the initial reset signal at power-on until the crystal oscillation circuit starts oscillating, or when the crystal oscillation circuit stops oscillating for some reason.

Reset pin (RESET)

An initial reset can be invoked externally by making the reset pin high. This high level must be maintained for at least 5 ms (when oscillating frequency, fosc = 32 kHz), because the initial reset circuit contains a noise rejection circuit. When the reset pin goes low the CPU begins to operate.

Simultaneous high input to input ports (K00–K03)

Another way of invoking an initial reset externally is to input a high signal simultaneously to the input ports (K00–K03) selected with the mask option. The specified input port pins must be kept high for 2–4 sec (when oscillating frequency fosc = 32 kHz), because of the noise rejection circuit. Table 2.2.1 shows the combinations of input ports (K00–K03) that can be selected with the mask option.

Table 2.2.1
Input port combinations

A	Not used
B	K00*K01
C	K00*K01*K02
D	K00*K01*K02*K03

When, for instance, mask option D (K00*K01*K02*K03) is selected, an initial reset is executed when the signals input to the four ports K00–K03 are all high at the same time.

If you use this function, make sure that the specified ports do not go high at the same time during normal operation.

Internal register following initialization

An initial reset initializes the CPU as shown in the table below.

Table 2.2.2
Initial values

CPU Core			
Name	Signal	Number of Bits	Setting Value
Program counter step	PCS	8	00H
Program counter page	PCP	4	1H
New page pointer	NPP	4	1H
Stack pointer	SP	8	Undefined
Index register X	X	8	Undefined
Index register Y	Y	8	Undefined
Register pointer	RP	4	Undefined
General register A	A	4	Undefined
General register B	B	4	Undefined
Interrupt flag	I	1	0
Decimal flag	D	1	Undefined
Zero flag	Z	1	Undefined
Carry flag	C	1	Undefined

Peripheral Circuits		
Name	Number of Bits	Setting Value
RAM	96 × 4	Undefined
Display memory	26 × 4	Undefined
Other peripheral circuit	–	*1

*1: See section 4.1, "Memory Map"

2.3 Test Pin (TEST, MTEST)

This pin is used when IC is inspected for shipment.
During normal operation connect it to V_{SS}.

CHAPTER 3 CPU, ROM, RAM

3.1 CPU

The S1C62N81 Series employs the S1C6200 core CPU, so that register configuration, instructions, and so forth are virtually identical to those in other processors in the family using the S1C6200. Refer to the "S1C6200/6200A Core CPU Manual" for details of the S1C6200.

Note the following points with regard to the S1C62N81 Series:

(1) The SLEEP operation is not provided, so the SLP instruction cannot be used.

(2) Because the ROM capacity is 1,024 words, 12 bits per word, bank bits are unnecessary, and PCB and NBP are not used.

(3) The RAM page is set to 0 only, so the page part (XP, YP) of the index register that specifies addresses is invalid.

```
PUSH   XP          PUSH   YP
POP    XP          POP    YP
LD     XP,r        LD     YP,r
LD     r,XP        LD     r,YP
```

3.2 ROM

The built-in ROM, a mask ROM for the program, has a capacity of 1,024 × 12-bit steps. The program area is 4 pages (0–3), each consisting of 256 steps (00H–FFH). After an initial reset, the program start address is page 1, step 00H. The interrupt vector is allocated to page l, steps 02H–0BH.

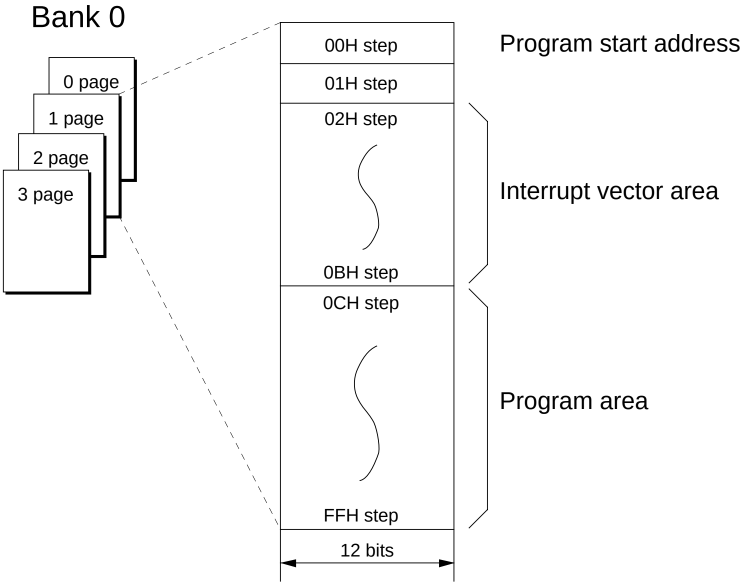


Fig. 3.2.1
ROM configuration

3.3 RAM

The RAM, a data memory for storing a variety of data, has a capacity of 96 words, 4-bit words. When programming, keep the following points in mind:

(1) Part of the data memory is used as stack area when saving subroutine return addresses and registers, so be careful not to overlap the data area and stack area.

(2) Subroutine calls and interrupts take up three words on the stack.

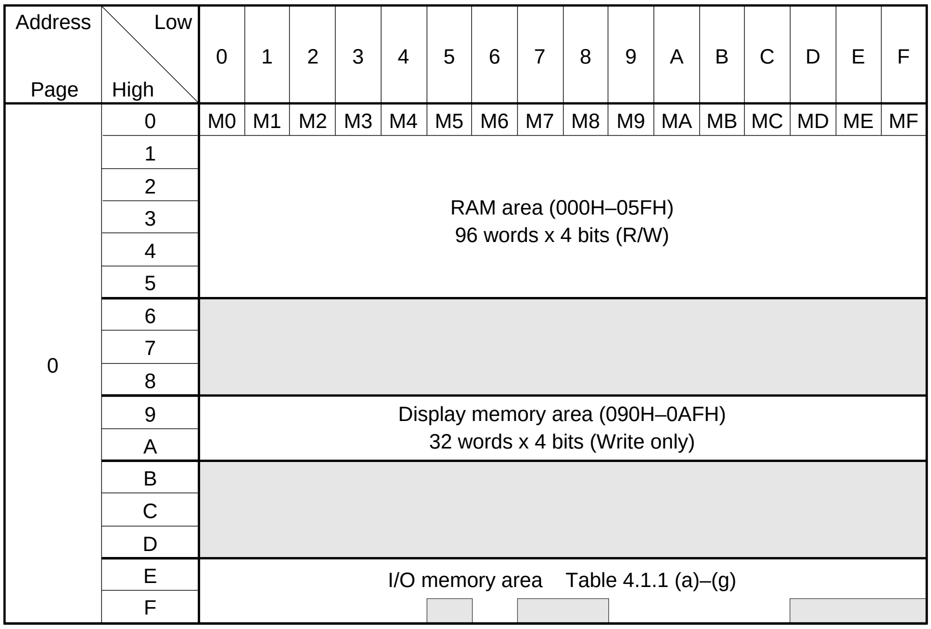
(3) Data memory 000H–00FH is the memory area pointed by the register pointer (RP).

CHAPTER 4 PERIPHERAL CIRCUITS AND OPERATION

Peripheral circuits (timer, I/O, and so on) of the S1C62N81 Series are memory mapped. Thus, all the peripheral circuits can be controlled by using memory operations to access the I/O memory. The following sections describe how the peripheral circuits operate.

4.1 Memory Map

The data memory of the S1C62N81 Series has an address space of 154 words, of which 32 words are allocated to display memory and 26 words, to I/O memory. Figure 4.1.1 show the overall memory mas for the S1C62N81 Series, and Tables 4.1.1 (a)–(g), the memory maps for the peripheral circuits (I/O space).



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 area (000H–05FH) 96 words x 4 bits (R/W)															
	2																
	3																
	4																
	5																
	6	▨															
	7																
	8																
	9	Display memory area (090H–0AFH) 32 words x 4 bits (Write only)															
	A																
	B	▨															
	C																
	D																
	E	I/O memory area Table 4.1.1 (a)–(g)															
	F						▨		▨	▨					▨	▨	▨

▨ Unused area

Fig. 4.1.1 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 4.1.1 (a) I/O memory map (0E0H–0E3H)

Address	Register								Comment
	D3	D2	D1	D0	Name	SR [1]	1	0	
0E0H	K03	K02	K01	K00	K03	–[2]	High	Low	Input port (K00–K03)
	R				K02	–[2]	High	Low	
					K01	–[2]	High	Low	
					K00	–[2]	High	Low	
0E1H	0	0	0	K10	0[5]				
	R				0[5]				
					0[5]				
					K10	–[2]	High	Low	Input port (K10)
0E2H	SWL3	SWL2	SWL1	SWL0	SWL3	0			MSB
	R				SWL2	0			Stopwatch timer 1/100 sec (BCD)
					SWL1	0			
					SWL0	0			LSB
0E3H	SWH3	SWH2	SWH1	SWH0	SWH3	0			MSB
	R				SWH2	0			Stopwatch timer 1/10 sec (BCD)
					SWH1	0			
					SWH0	0			LSB

[1] Initial value following initial reset

[2] Not set in the circuit

[3] Undefined

[4] Reset (0) immediately after being read

[5] Constantly 0 when being read

[6] Refer to main manual

Table 4.1.1 (b) I/O memory map (0E4H–0E7H)

Address	Register								Comment
	D3	D2	D1	D0	Name	SR *1	1	0	
0E4H	TM3	TM2	TM1	TM0	TM3	–	High	Low	Timer data (clock timer 2 Hz)
	R				TM2	–	High	Low	Timer data (clock timer 4 Hz)
					TM1	–	High	Low	Timer data (clock timer 8 Hz)
					TM0	–	High	Low	Timer data (clock timer 16 Hz)
0E5H	KCP03	KCP02	KCP01	KCP00	KCP03	0	Falling	Rising	Input comparison register (K03)
	R/W				KCP02	0	Falling	Rising	Input comparison register (K02)
					KCP01	0	Falling	Rising	Input comparison register (K01)
					KCP00	0	Falling	Rising	Input comparison register (K00)
0E6H	0	0	0	KCP10	0 *5				
	R			R/W	0 *5				
					0 *5				
					KCP10	0	Falling	Rising	Input comparison register (K10)
0E7H	0	0	0	EIMEL	0 *5				
	R			R/W	0 *5				
					0 *5				
					EIMEL	0	Enable	Mask	Interrupt mask register (melody)

* 1 Initial value following initial reset
* 2 Not set in the circuit
* 3 Undefined
* 4 Reset (0) immediately after being read
* 5 Constantly 0 when being read
* 6 Refer to main manual

Table 4.1.1 (c) I/O memory map (0E8H–0EBH)

Address	Register								Comment
	D3	D2	D1	D0	Name	SR *1	1	0	
0E8H	EIK03	EIK02	EIK01	EIK00	EIK03	0	Enable	Mask	Interrupt mask register (K03)
	R/W				EIK02	0	Enable	Mask	Interrupt mask register (K02)
					EIK01	0	Enable	Mask	Interrupt mask register (K01)
					EIK00	0	Enable	Mask	Interrupt mask register (K00)
0E9H	0	0	0	EIK10	0 *5				
	R			R/W	0 *5				
					0 *5				
					EIK10	0	Enable	Mask	Interrupt mask register (K10)
0EAH	0	0	EISW1	EISW0	0 *5				
	R		R/W		0 *5				
					EISW1	0	Enable	Mask	Interrupt mask register (stopwatch 1 Hz)
					EISW0	0	Enable	Mask	Interrupt mask register (stopwatch 10 Hz)
0EBH	0	EIT2	EIT8	EIT32	0 *5				
	R	R/W			EIT2	0	Enable	Mask	Interrupt mask register (clock timer 2 Hz)
					EIT8	0	Enable	Mask	Interrupt mask register (clock timer 8 Hz)
					EIT32	0	Enable	Mask	Interrupt mask register (clock timer 32 Hz)

*1 Initial value following initial reset
*2 Not set in the circuit
*3 Undefined
*4 Reset (0) immediately after being read
*5 Constantly 0 when being read
*6 Refer to main manual

Table 4.1.1 (d) I/O memory map (0ECH–0EFH)

Address	Register								Comment
	D3	D2	D1	D0	Name	SR *1	1	0	
0ECH	0	0	0	IMEL	0 *5				
	R				0 *5				
					0 *4				
					IMEL *4	0	Yes	No	Interrupt factor flag (melody)
0EDH	0	0	IK1	IK0	0 *5				
	R				0 *5				
					IK1 *4	0	Yes	No	Interrupt factor flag (K10)
					IK0 *4	0	Yes	No	Interrupt factor flag (K00–K03)
0EEH	0	0	ISW1	ISW0	0 *5				
	R				0 *5				
					ISW1 *4	0	Yes	No	Interrupt factor flag (stopwatch 1 Hz)
					ISW0 *4	0	Yes	No	Interrupt factor flag (stopwatch 10 Hz)
0EFH	0	IT2	IT8	IT32	0 *5				
	R				IT2 *4	0	Yes	No	Interrupt factor flag (clock timer 2 Hz)
					IT8 *4	0	Yes	No	Interrupt factor flag (clock timer 8 Hz)
					IT32 *4	0	Yes	No	Interrupt factor flag (clock timer 32 Hz)

* 1 Initial value following initial reset
* 2 Not set in the circuit
* 3 Undefined
* 4 Reset (0) immediately after being read
* 5 Constantly 0 when being read
* 6 Refer to main manual

Table 4.1.1 (e) I/O memory map (0F0H–0F3H)

Address	Register								Comment
	D3	D2	D1	D0	Name	SR *1	1	0	
0F0H	MAD3	MAD2	MAD1	MAD0	MAD3	0	High	Low	Melody ROM address (AD3)
	R/W				MAD2	0	High	Low	Melody ROM address (AD2)
					MAD1	0	High	Low	Melody ROM address (AD1)
					MAD0	0	High	Low	Melody ROM address (AD0, LSB)
0F1H	0	MAD6	MAD5	MAD4	0 *5				
	R	R/W			MAD6	0	High	Low	Melody ROM address (AD6, MSB)
					MAD5	0	High	Low	Melody ROM address (AD5)
					MAD4	0	High	Low	Melody ROM address (AD4)
0F2H	CLKC1	CLKC0	TEMPC	MELC	CLKC1	0	High	Low	CLKC1(0)&CLKC0(0) : melody speed × 1 CLKC1(0)&CLKC0(1) : melody speed × 8
	R/W				CLKC0	0	High	Low	CLKC1(1)&CLKC0(0) : melody speed × 16 CLKC1(1)&CLKC0(1) : melody speed × 32
					TEMPC	0	High	Low	Tempo change control
					MELC	0	ON	OFF	Melody control ON/OFF
0F3H	R03	R02	R01	R00	R03	0	High	Low	Output port data (R00–R03)
	R/W				R02	0	High	Low	
					R01	0	High	Low	
					R00	0	High	Low	

*1 Initial value following initial reset

*2 Not set in the circuit

*3 Undefined

*4 Reset (0) immediately after being read

*5 Constantly 0 when being read

*6 Refer to main manual

Table 4.1.1 (f) I/O memory map (0F4H, 0F6H, 0F9H–0FAH)

Address	Register				Comment				
	D3	D2	D1	D0	Name	SR *1	1	0	
0F4H	*3	R12 $\overline{\text{MO}}$ ENV	R11	R10 FOUT	R12	0	High	Low	Output port data (R12)
	R/W				$\overline{\text{MO}}$	1	–	–	Inverting melody output
					ENV	Hz	–	–	Melody envelope control
					R11	0	High	Low	Output port data (R11)
					R10	0	High	Low	Output port data (R10)
					FOUT		ON	OFF	Frequency output
0F6H	P03	P02	P01	P00	P03	– *2	High	Low	I/O port (P00–P03)
	R/W				P02	– *2	High	Low	
					P01	– *2	High	Low	
					P00	– *2	High	Low	
0F9H	0	TMRST	SWRUN	SWRST	0 *5				
	R	W	R/W	W	TMRST *5	Reset	Reset	–	Clock timer reset
					SWRUN	0	Run	Stop	Stopwatch timer RUN/STOP
					SWRST *5	Reset	Reset	–	Stopwatch timer reset
0FAH	HLMOD	0	BLDDT	BLDON	HLMOD	0	Heavy load	Normal load	Heavy load protection mode register
	R/W	R		R/W	0 *5				
					BLDDT	0	Battery voltage low	Battery voltage normal	Battery voltage detector data
					BLDON	0	ON	OFF	Battery voltage detector ON/OFF

* 1 Initial value following initial reset

* 2 Not set in the circuit

* 3 Undefined

* 4 Reset (0) immediately after being read

* 5 Constantly 0 when being read

* 6 Refer to main manual

Table 4.1.1 (g) I/O memory map (0FBH–0FCH)

Address	Register								Comment
	D3	D2	D1	D0	Name	SR *1	1	0	
0FBH	CSDC	0	CMPDT	CMPON	CSDC	0	Static	Dynamic	LCD drive switch
	R/W	R		R/W	0 *5				
					CMPDT	1	+ > -	- > +	Comparator's voltage condition: 1 = CMPP(+)input > CMPM(-)input, 0 = CMPM(-)input > CMPP(+)input
					CMPON	0	On	Off	Analog voltage comparator ON/OFF
0FCH	0	0	0	IOC	0 *5				
	R			R/W	0 *5				
					0 *5				
					IOC	0	Output	Input	I/O port P00–P03 Input/Output

*1 Initial value following initial reset
*2 Not set in the circuit
*3 Undefined
*4 Reset (0) immediately after being read
*5 Constantly 0 when being read
*6 Refer to main manual

4.2 Oscillation Circuit

Crystal oscillation circuit

The S1C62N81 Series have a built-in crystal oscillation circuit. This circuit generates the operating clock for the CPU and peripheral circuit on connection to an external crystal oscillator (typ. 32.768 kHz) and trimmer capacitor (5–25 pF).

Figure 4.2.1 is the block diagram of the crystal oscillation circuit.

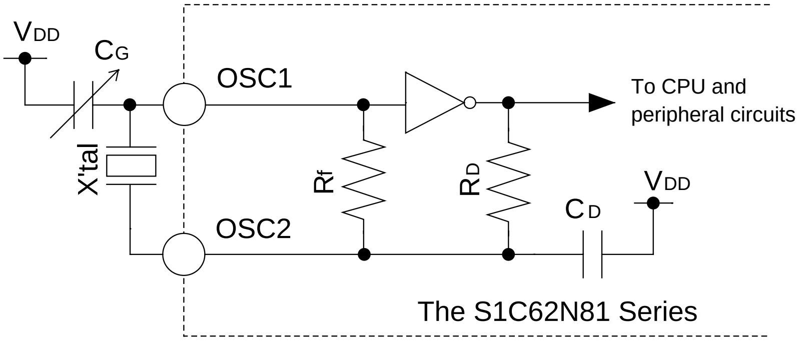


Fig. 4.2.1
Crystal oscillation circuit

As Figure 4.2.1 indicates, the crystal oscillation circuit can be configured simply by connecting the crystal oscillator (X'tal) between the OSC1 and OSC2 pins and the trimmer capacitor (CG) between the OSC1 and VDD pins.

CR oscillation circuit

For the S1C62N81 Series, CR oscillation circuit (typ. 32.768 kHz) may also be selected by a mask option. Figure 4.2.2 is the block diagram of the CR oscillation circuit.

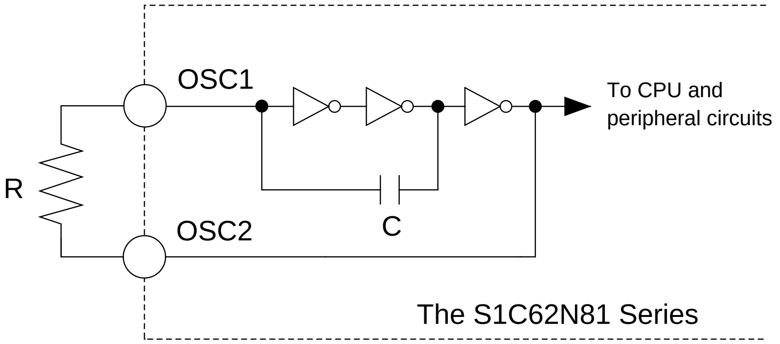


Fig. 4.2.2
CR oscillation circuit

As Figure 4.2.2 indicates, the CR oscillation circuit can be configured simply by connecting the register (R) between pins OSC1 and OSC2 since capacity (C) is built-in.
See Chapter 6, "ELECTRICAL CHARACTERISTICS" for R value.

4.3 Input Ports (K00–K03, K10)

Configuration of input ports

The S1C62N81 Series have a general-purpose input (4 bits + 1 bit). Each of the input port pins (K00–K03, K10) has an internal pull-down resistance. The pull-down resistance can be selected for each bit with the mask option.
Figure 4.3.1 shows the configuration of input port.

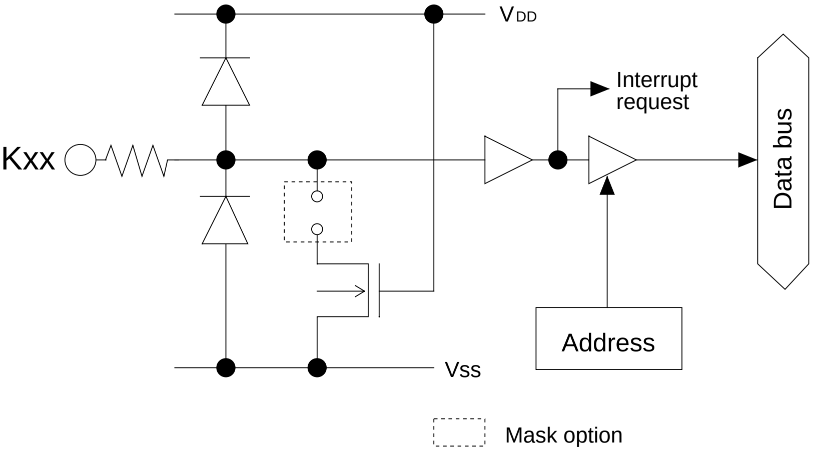


Fig. 4.3.1
Configuration of input port

Selecting "pull-down resistance enabled" with the mask option allows input from a push button, key matrix, and so forth. When "pull-down resistance disabled" is selected, the port can be used for slide switch input and interfacing with other LSIs.

Input comparison registers and interrupt function

All five input port bits (K00–K03, K10) provide the interrupt function. The conditions for issuing an interrupt can be set by the software for the five bits. Also, whether to mask the interrupt function can be selected individually for all five bits by the software. Figure 4.3.2 shows the configuration of K00–K03 and K10.

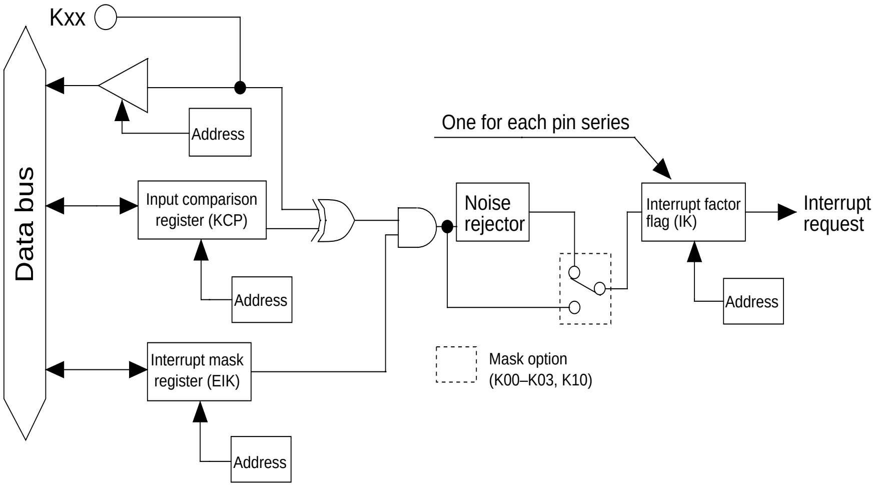

Kxx
Data bus
Address
Input comparison register (KCP)
Address
Interrupt mask register (EIK)
Address
One for each pin series
Noise rejector
Interrupt factor flag (IK)
Interrupt request
Address
Mask option (K00–K03, K10)

Fig. 4.3.2
Input interrupt circuit configuration (K00–K03, K10)

The input interrupt timing for K00–K03 and K10 depends on the value set in the input comparison registers (KCP00–KCP03 and KCP10). An interrupt can be set to occur on the rising or falling edge of the input.

The interrupt mask registers (EIK00–EIK03, EIK10) enable the interrupt mask to be selected individually for K00–K03 and K10. An interrupt occurs when the input value which are not masked change so they no longer match those of the input comparison register. An interrupt for K10 can be generated by setting the same conditions individually.

When an interrupt is generated, the interrupt factor flag (IK0 and IK1) is set to 1.

Figure 4.3.3 shows an example of an interrupt for K00–K03.

Note *Writing to the interrupt mask registers (EIK00–EIK03, EIK10) should be done only in the DI status (interrupt flag = 0). Otherwise, it causes malfunction.*

Interrupt mask registers

EIK03	EIK02	EIK01	EIK00
1	1	1	0

Input comparison registers

KCP03	KCP02	KCP01	KCP00
1	0	1	0

With the above setting, an interrupt for K00–K03 occurs under the following conditions.

Input ports

(1)

K03	K02	K01	K00
1	0	1	0

(Initial value)

↓

(2)

K03	K02	K01	K00
1	0	1	1

↓

(3)

K03	K02	K01	K00
0	0	1	1

→ Interrupt generated

K00 is masked, so the three bits of K01–K03 cease to match those of the input comparison register KCP01–KCP03, and an interrupt occurs.

↓

(4)

K03	K02	K01	K00
0	1	1	1

↓

(5)

K03	K02	K01	K00
1	0	1	1

Fig. 4.3.3
Example of interrupt
of K00–K03

K00 is masked by the interrupt mask register (EIK00), so an interrupt does not occur at (2). At (3), K03 changes to 0; the data of the pin that is interrupt-enabled no longer matches the data of the input comparison register, so an interrupt occurs. As already explained, the condition for the interrupt to occur is the change in the port data and contents of the input comparison register so they no longer match. Hence, in (4) or (5), when the nonmatching pattern changes to another nonmatching pattern or matching pattern, an interrupt does not occur. Also, pins that have been masked for interrupt do not affect the conditions for interrupt generation.

Mask option

The contents that can be selected with the input port mask option are as follows:

(1) An internal pull-down resistance can be selected for each of the five bits of the input ports (K00–K03, K10). Having selected "pull-down resistance disabled", take care that the input does not float. Select "pull-down resistance enabled" for input ports that are not being used.

(2) The input interrupt circuit contains a noise rejection circuit to prevent interrupts form occurring through noise. The mask option enables selection of the noise rejection circuit for each separate pin series. When "use" is selected, a maximum delay of 0.5 ms (fosc = 32 kHz) occurs from the time an interrupt condition is established until the interrupt factor flag (IK) is set to 1.

Control of input ports

Tables 4.3.1 (a) and 4.3.1 (b) list the input port control bits and their addresses.

Table 4.3.1 (a) Input port control bits (1)

Address	Register								Comment
	D3	D2	D1	D0	Name	SR	1	0	
0E0H	K03	K02	K01	K00	K03	–	High	Low	Input port (K00–K03)
	R				K02	–	High	Low	
					K01	–	High	Low	
					K00	–	High	Low	
0E1H	0	0	0	K10	0				
	R				0				
					0				
					K10	–	High	Low	Input port (K10)
0E5H	KCP03	KCP02	KCP01	KCP00	KCP03	0	Falling	Rising	Input comparison register (K03)
	R/W				KCP02	0	Falling	Rising	Input comparison register (K02)
					KCP01	0	Falling	Rising	Input comparison register (K01)
					KCP00	0	Falling	Rising	Input comparison register (K00)
0E6H	0	0	0	KCP10	0				
	R			R/W	0				
					0				
					KCP10	0	Falling	Rising	Input comparison register (K10)

Table 4.3.1 (b) Input port control bits (2)

Address	Register								Comment
	D3	D2	D1	D0	Name	SR	1	0	
0E8H	EIK03	EIK02	EIK01	EIK00	EIK03	0	Enable	Mask	Interrupt mask register (K03)
	R/W				EIK02	0	Enable	Mask	Interrupt mask register (K02)
					EIK01	0	Enable	Mask	Interrupt mask register (K01)
					EIK00	0	Enable	Mask	Interrupt mask register (K00)
0E9H	0	0	0	EIK10	0				
	R			R/W	0				
					0				
					EIK10	0	Enable	Mask	Interrupt mask register (K10)
0EDH	0	0	IK1	IK0	0				
	R				0				
					IK1	0	Yes	No	Interrupt factor flag (K10)
					IK0	0	Yes	No	Interrupt factor flag (K00–K03)

K00–K03, K10 Input port data (0E0H, 0E1H D0)

The input data of the input port pins can be read with these registers.

When 1 is read:	High level
When 0 is read:	Low level
Writing:	Invalid

The value read is 1 when the pin voltage of the five bits of the input ports (K00–K03, K10) goes high (V_{DD}), and 0 when the voltage goes low (V_{SS}). These bits are reading, so writing cannot be done.

KCP00–KCP03, KCP10 Input comparison registers (0E5H, 0E6H D0)

The interrupt conditions for pins K00–K03 and K10 can be set with these registers.

When 1 is read:	Falling edge
When 0 is read:	Rising edge
Reading:	Valid

Of the five bits of the input ports, the interrupt conditions can be set for the rising or falling edge of the input for each of the five bits (K00–K03 and K10) through the input comparison registers (KCP00–KCP03 and KCP10).
After an initial reset, these registers are set to 0.

EIK00–EIK03, EIK10 Interrupt mask registers (0E8H, 0E9H D0)

Masking the interrupt of the input port pins can be done with these registers.

When 1 is written:	Enable
When 0 is written:	Mask
Reading:	Valid

With these registers, masking of the input port bits can be done for each of the five bits. After an initial reset, these registers are all set to 0.
Writing to these registers should be done only in the DI status (interrupt flag = 0). Otherwise, it causes malfunction.

K0, IK1 Interrupt factor flags (0EDH D0 and D1)

These flags indicate the occurrence of an input interrupt.

When 1 is read:	Interrupt has occurred
When 0 is read:	Interrupt has not occurred
Writing:	Invalid

The interrupt factor flags IK0 and IK1 are associated with K00–K03 and K10, respectively. From the status of these flags, the software can decide whether an input interrupt has occurred.
These flags are reset when the software has read them.
Reading should be done only in the DI status (interrupt flag = 0). Otherwise, it causes malfunction.
After an initial reset, these flags are set to 0.

4.4 Output Ports (R00–R03, R10–R12)

Configuration of output ports

The S1C62N81 Series have 7 bits for general output ports (R00–R03 and R10–R12).
Output specifications of the output ports can be selected individually with the mask option. Two kinds of output specifications are available: complementary output, and Pch open drain output. Also, the mask option enables the output ports R10 and R12 to be used as special output ports. Figure 4.4.1 shows the configuration of the output ports.

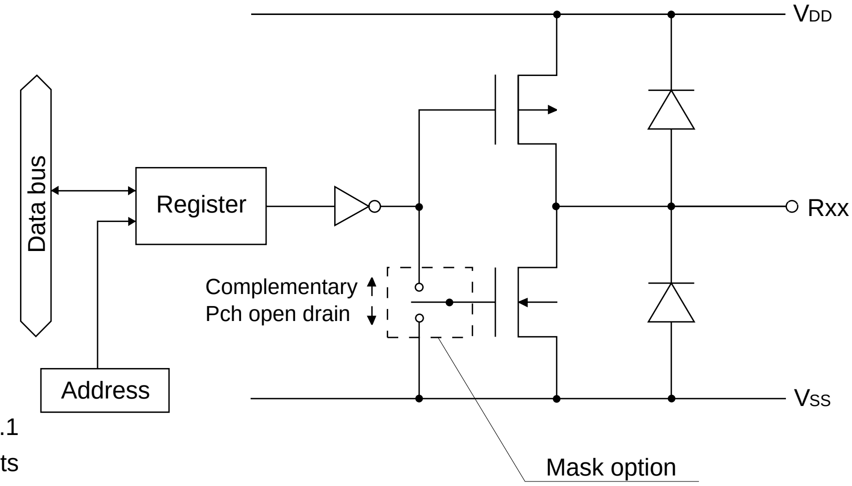


Fig. 4.4.1
Configuration of output ports

Mask option

The mask option enables the following output port selection.

(1) Output specifications of output ports

The output specifications for the output ports (R00–R03, R10–R12) may be either complementary output or Pch open drain output for each of the seven bits. However, even when Pch open drain output is selected, a voltage exceeding the source voltage must not be applied to the output port.

(2) Special output

In addition to the regular DC output, special output can be selected for output ports R10 and R12, as shown in Table 4.4.1. Figure 4.4.2 shows the structure of output ports R10–R12.

Table 4.4.1
Special output

Pin Name	When Special Output is Selected
R12	$\overline{\text{MO}}$ or ENV
R10	FOUT

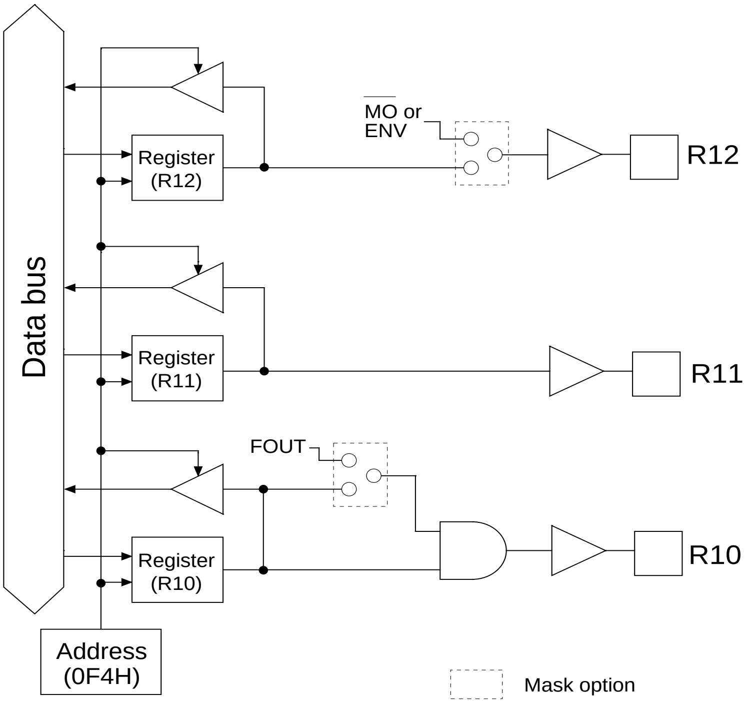


Fig. 4.4.2
Structure of output port
R10–R12

FOUT (R10)

When output port R10 is set for FOUT output, it outputs the clock of fosc or the divided fosc. The clock frequency is selectable by mask option from the frequencies listed in Table 4.4.2.

Table 4.4.2
FOUT clock frequency

Setting Value	Clock Frequency (Hz)
fosc/1	32,768
fosc/2	16,384
fosc/4	8,192
fosc/8	4,096
fosc/16	2,048
fosc/32	1,024
fosc/64	512
fosc/128	256

(fosc = 32,768 Hz)

Note *A hazard may occur when the FOUT signal is turned on or off.*

$\overline{\text{MO}}$, ENV (R12)

R12 can select the following two functions using the mask option as special output.

(1) Inverse output ($\overline{\text{MO}}$) of melody output (MO)

Using the $\overline{\text{MO}}$ and MO terminals together, piezoelectric buzzer may be driven directly. This means the minimum number of external parts is necessary to play melodies.

(2) Envelope function

An envelope can be added when playing a melody by connecting the play sound pressure damping capacitor to terminal R12.

For details, see Chapter 5, "BASIC EXTERNAL WIRING DIAGRAM", and Section 4.11, "Melody Generator".

Control of output ports

Table 4.4.3 lists the output port control bits and their addresses.

Table 4.4.3 Control bits of output ports

Address	Register				Name	SR	1	0	Comment
	D3	D2	D1	D0					
0F3H	R03	R02	R01	R00	R03	0	High	Low	Output port data (R00–R03)
	R/W				R02	0	High	Low	
					R01	0	High	Low	
					R00	0	High	Low	
0F4H		R12 $\overline{\text{MO}}$ ENV	R11	R10 FOUT	R12	0	High	Low	Output port data (R12)
		R/W			$\overline{\text{MO}}$	1	–	–	Inverting melody output
					ENV	Hz	–	–	Melody envelope control
					R11	0	High	Low	Output port data (R11)
					R10	0	High	Low	Output port data (R10)
					FOUT		ON	OFF	Frequency output

R00–R03, R10–R12 (DC output)

Output port data (0F3H, 0F4H D0–D2)

Sets the output data for the output ports.

When 1 is written:	High output
When 0 is written:	Low output
Reading:	Valid

The output port pins output the data written to the corresponding registers (R00–R03, R10–R12) without changing it. When 1 is written to the register, the output port pin goes high (V_{DD}), and when 0 is written, the output port pin goes low (V_{SS}). After an initial reset, all registers are set to 0.

R12 (when $\overline{\text{MO}}$ or ENV is selected)

Special output port data (0F4H D2)

This bit will not affect the melody ($\overline{\text{MO}}$) or envelope (ENV) signal at Rl2. R12 register is a general purpose register which can be read and written.

When 1 is written:	No effect at R12
When 0 is written:	No effect at R12
Reading:	Valid

R10 (when FOUT is selected)

Special output port data (0F4H D0)

Controls the FOUT (clock) output.

When 1 is written:	Clock output
When 0 is written:	Low level (DC) output
Reading:	Valid

FOUT output can be controlled by writing data to R10. After an initial reset, this register is set to 0.

Figure 4.4.3 shows the output waveform for FOUT output.

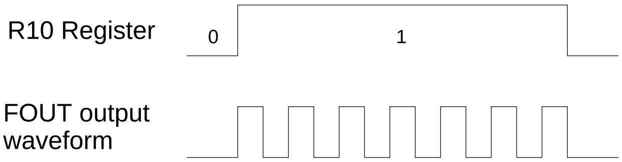


Fig. 4.4.3
FOUT output waveform

4.5 I/O Ports (P00–P03)

Configuration of I/O port

The S1C62N81 Series have a 4-bit general-purpose I/O port. Figure 4.5.1 shows the configuration of the I/O port. The four bits of the I/O port P00–P03 can be set to either input mode or output mode. The mode can be set by writing data to the I/O control register (IOC).

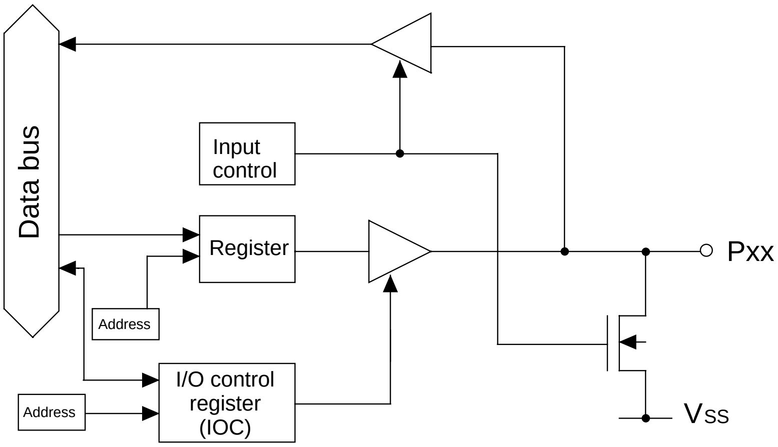


Fig. 4.5.1
Configuration of I/O port

I/O control register and I/O mode

Input or output mode can be set for the four bits of I/O port P00–P03 by writing data into I/O control register IOC.
To set the input mode, 0 is written to the I/O control register. When an I/O port is set to input mode, its impedance becomes high and it works as an input port. However, the input line is pulled down when input data is read.

The output mode is set when 1 is written to the I/O control register (IOC). When an I/O port set to output mode works as an output port, it outputs a high signal (VDD) when the port output data is 1, and a low signal (VSS) when the port output data is 0.
After an initial reset, the I/O control register is set to 0, and the I/O port enters the input mode.

Mask option

The output specification during output mode (IOC = 1) of the I/O port can be set with the mask option for either complementary output or Pch open drain output. This setting can be performed for each bit of the I/O port. However, when Pch open drain output has been selected, voltage in excess of the supply voltage must not be applied to the port.

Control of I/O port

Table 4.5.1 lists the I/O port control bits and their addresses.

Table 4.5.1 I/O port control bits

Address	Register								Comment
	D3	D2	D1	D0	Name	SR	1	0	
0F6H	P03	P02	P01	P00	P03	–	High	Low	I/O port (P00–P03)
	R/W				P02	–	High	Low	
					P01	–	High	Low	
					P00	–	High	Low	
0FCH	0	0	0	IOC	0				
	R			R/W	0				
					0				
					IOC	0	Output	Input	I/O port P00–P03 Input/Output

P00–P03 I/O port data (0F6H)

I/O port data can be read and output data can be written through the port.

- When writing data

When 1 is written:	High level
When 0 is written:	Low level

When an I/O port is set to the output mode, the written data is output from the I/O port pin unchanged. When 1 is written as the port data, the port pin goes high (VDD), and when 0 is written, the level goes low (VSS). Port data can also be written in the input mode.

- When reading data

When 1 is read:	High level
When 0 is read:	Low level

The pin voltage level of the I/O port is read. When the I/O port is in the input mode the voltage level being input to the port pin can be read; in the output mode the output voltage level can be read. When the pin voltage is high (VDD) the port data read is 1, and when the pin voltage is low (VSS) the data is 0. Also, the built-in pull-down resistance functions during reading, so the I/O port pin is pulled down.

Note - When the I/O port is set to the output mode and a low-impedance load is connected to the port pin, the data written to the register may differ from the data read.

- When the I/O port is set to the input mode and a low-level voltage (Vss) is input by the built-in pull-down resistance, an erroneous input results if the time constant of the capacitive load of the input line and the built- in pull-down resistance load is greater than the read-out time. When the input data is being read, the time that the input line is pulled down is equivalent to 0.5 cycles of the CPU system clock. Hence, the electric potential of the pins must settle within 0.5 cycles. If this condition cannot be met, some measure must be devised, such as arranging a pull-down resistance externally, or performing multiple read-outs.

IOC I/O control register (0FCH D0)

The input or output I/O port mode can be set with this register.

When 1 is written:	Output mode
When 0 is written:	Input mode
Reading:	Valid

The input or output mode of the I/O port is set in units of four bits. For instance, IOC sets the mode for P00–P03. Writing 1 to the I/O control register makes the I/O port enter the output mode, and writing 0, the input mode. After an initial reset, the IOC register is set to 0, so the I/O port is in the input mode.

4.6 LCD Driver (COM0–COM3, SEG0–SEG25)

Configuration of LCD driver

The S1C62N81 Series have four common pins and 26 (SEG0–SEG25) segment pins, so that an LCD with a maximum of 104 (26 × 4) segments can be driven. The power for driving the LCD is generated by the CPU internal circuit, so there is no need to supply power externally.

The driving method is 1/4 duty (or 1/3 duty by mask option) dynamic drive, adopting the four types of potential, V_{DD}, V_{L1}, V_{L2} and V_{L3}. The frame frequency is 32 Hz for 1/4 duty, and 42.7 Hz for 1/3 duty (in the case of fosc = 32.768 kHz). Figure 4.6.1 shows the drive waveform for 1/4 duty, and Figure 4.6.2 shows the drive waveform for 1/3 duty.

Note *fosc indicates the oscillation frequency of the oscillation circuit.*

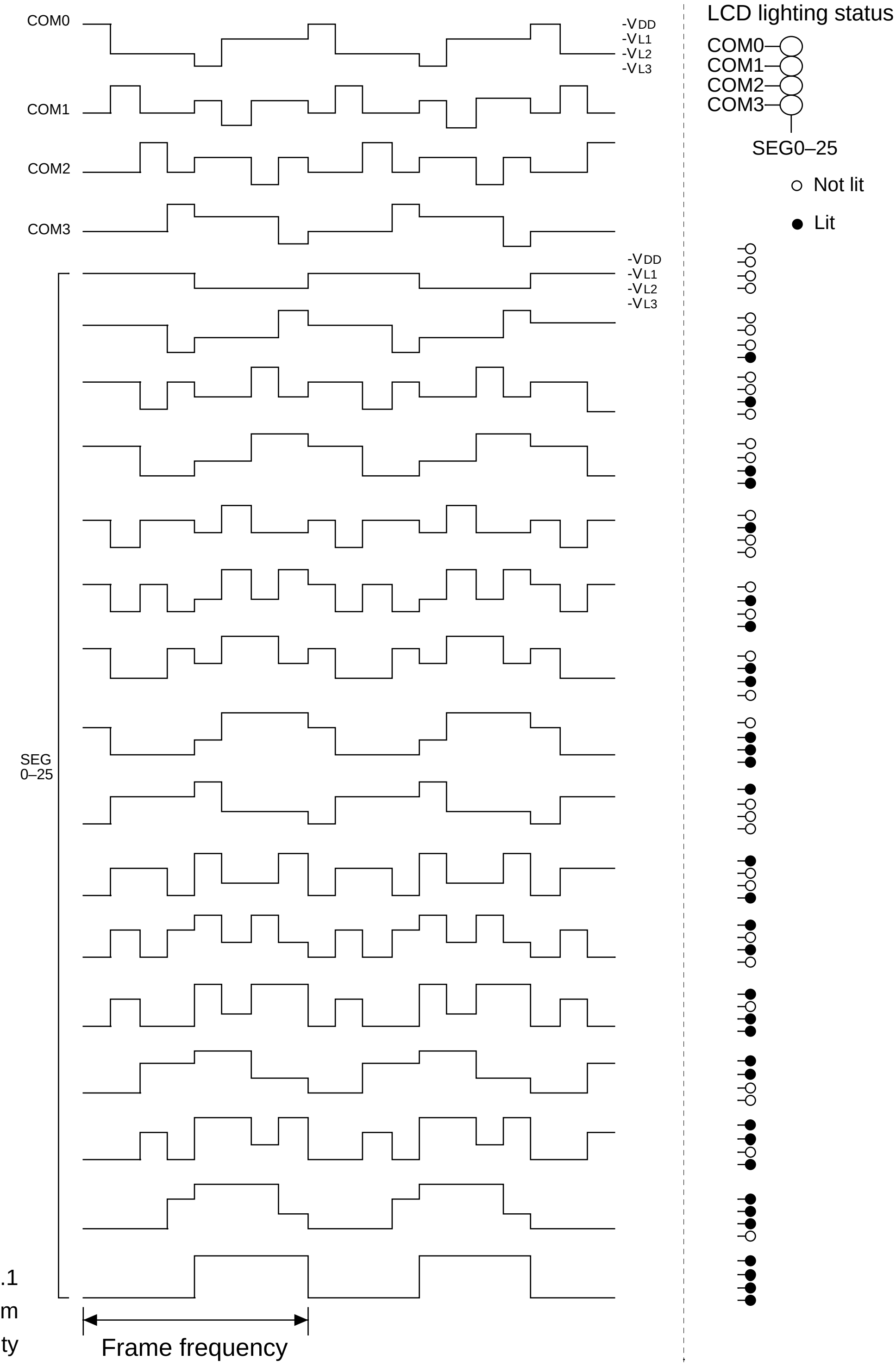


Fig. 4.6.1
Drive waveform
for 1/4 duty

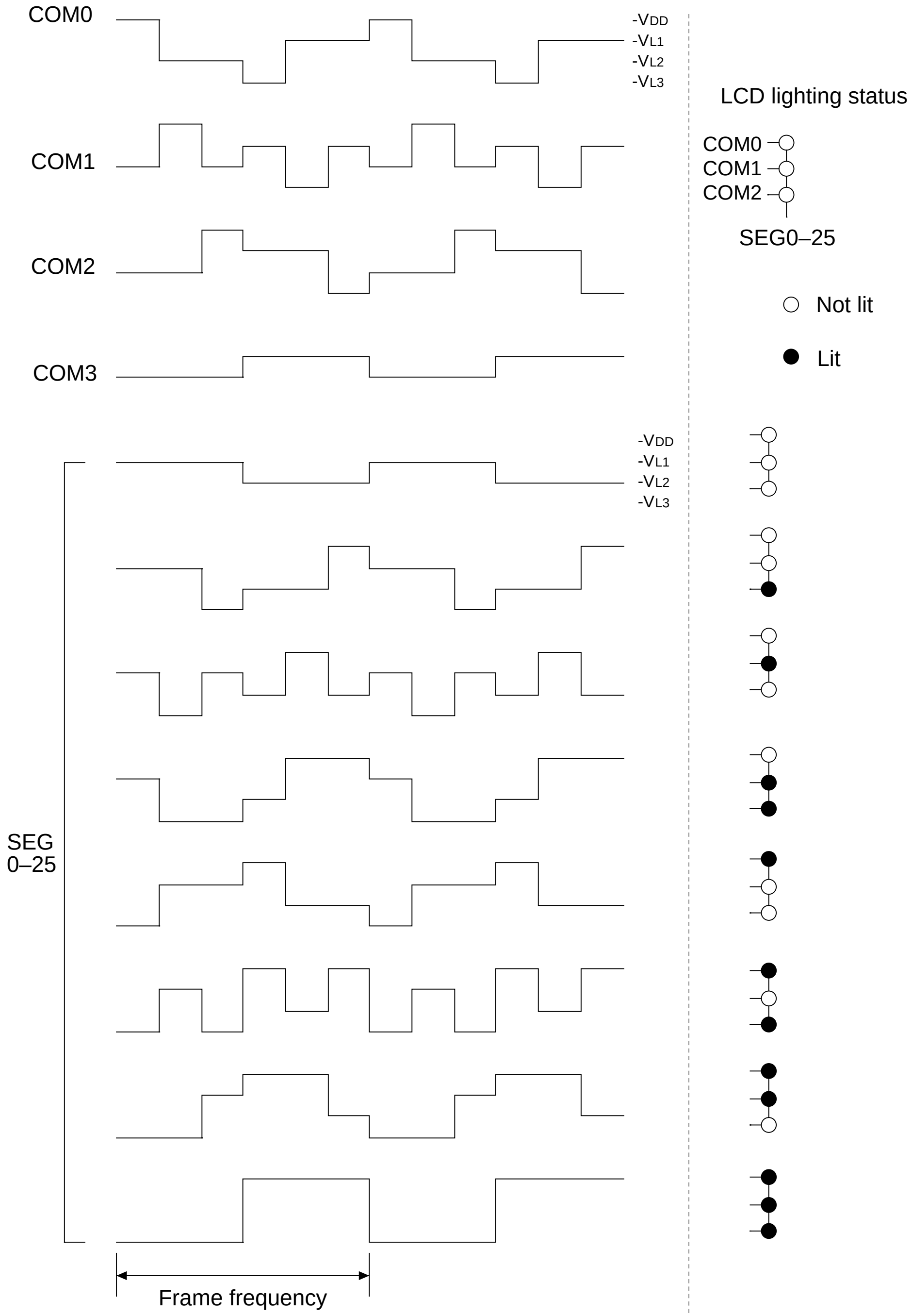


Fig. 4.6.2
Drive waveform
for 1/3 duty

Switching between dynamic and static drive

The S1C62N81 Series members allow software setting of the LCD static drive. This function enables easy adjustment (cadence adjustment) of the oscillation frequency of the OSC circuit.

The procedure for executing of the LCD static drive is as follows:

① Write 1 to the CSDC register at address 0FBH D3.

② Write the same value to all registers corresponding to COM0–COM3 of the display memory.

Note - *Even when l/3 duty is selected, the display data corresponding to COM3 is valid for static drive. Hence, for static drive, set the same value to all display memory corresponding COM0–COM3.*

- *For cadence adjustment, set the display data including display data corresponding to COM3, so that all the LCD segments go on.*

Figure 4.6.3 shows the drive waveform for static drive.

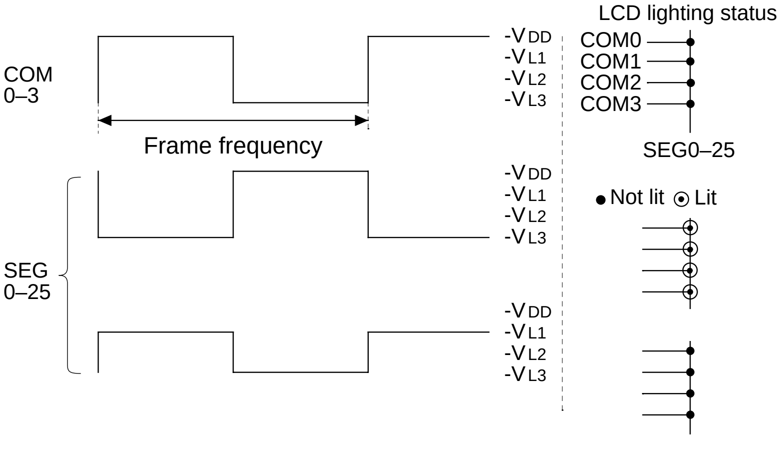


Fig. 4.6.3
LCD static drive waveform

Mask option (segment allocation)

(1) Segment allocation

As shown in Figure 4.l.1, the S1C62N81 Series display data is decided by the display data written to the display memory (write-only) at address 090H–0AFH.

The address and bits of the display memory can be made to correspond to the segment pins (SEG0–SEG25) in any combination through mask option. This simplifies design by increasing the degree of freedom with which the liquid crystal panel can be designed.

Figure 4.6.4 shows an example of the relationship between the LCD segments (on the panel) and the display memory in the case of 1/3 duty.

Address	Data			
	D3	D2	D1	D0
09AH	d	c	b	a
09BH	p	g	f	e
09CH	d'	c'	b'	a'
09DH	p'	g'	f'	e'

Display data memory allocation

	Common 0	Common 1	Common 2
SEG10	9A, D0 (a)	9B, D1 (f)	9B, D0 (e)
SEG11	9A, D1 (b)	9B, D2 (g)	9A, D3 (d)
SEG12	9D, D1 (f')	9A, D2 (c)	9B, D3 (p)

Pin address allocation

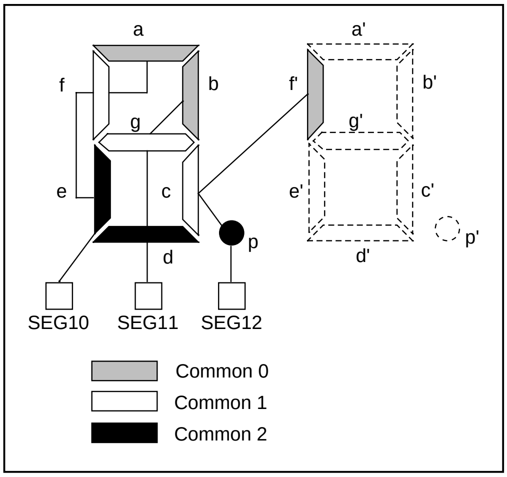


Fig. 4.6.4
Segment allocation

(2) Drive duty

According to the mask option, either 1/4 or 1/3 duty can be selected as the LCD drive duty.
Table 4.6.1 shows the differences in the number of segments according to the selected duty.

Table 4.6.1 Differences according to selected duty

Duty	Pins Used in Common	Maximum Number of Segments	Frame Frequency (when fosc = 32 kHz)
1/4	COM0–3	104 (26 × 4)	32 Hz
1/3	COM0–2	78 (26 × 3)	42.7 Hz

(3) Output specification

① The segment pins (SEG0–SEG25) are selected by mask option in pairs for either segment signal output or DC output (V_{DD} and V_{SS} binary output). When DC output is selected, the data corresponding to COM0 of each segment pin is output.

② When DC output is selected, either complementary output or Pch open drain output can be selected for each pin by mask option.

Note *The pin pairs are the combination of SEG (2*n) and SEG (2*n + 1) (where n is an integer from 0 to 12).*

Control of LCD driver

Table 4.6.2 shows the control bits of the LCD driver and their addresses. Figure 4.6.5 shows the display memory map.

Table 4.6.2 Control bits of LCD driver

Address	Register								Comment
	D3	D2	D1	D0	Name	SR	1	0	
0FBH	CSDC	0	CMPDT	CMPON	CSDC	0	Static	Dynamic	LCD drive switch
	R/W	R		R/W	0				
					CMPDT	1	+ > -	- > +	Comparator's voltage condition: 1 = CMPP(+)input > CMPM(-)input, 0 = CMPM(-)input > CMPP(+)input
					CMPON	0	On	Off	Voltage comparator ON/OFF

Address	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
090	Display memory (Write only) 32 words x 4 bits															
0A0																

Fig. 4.6.5 Display memory map

CSDC LCD drive switch (0FBH D3)

The LCD drive format can be selected with this switch.

When 1 is written:	Static drive
When 0 is written:	Dynamic drive
Reading:	Valid

After an initial reset, dynamic drive (CSDC = 0) is selected.

Display memory (090H–0AFH)

The LCD segments are turned on or off according to this data.

When 1 is written:	On
When 0 is written:	Off
Reading:	Invalid

By writing data into the display memory allocated to the LCD segment (on the panel), the segment can be turned on or off. After an initial reset, the contents of the display memory are undefined.

4.7 Clock Timer

Configuration of clock timer

The S1C62N81 Series have a built-in clock timer driven by the source oscillator. The clock timer is configured as a seven-bit binary counter that serves as a frequency divider taking a 256 Hz source clock from a prescaler. The four high-order bits (16 Hz–2 Hz) can be read by the software. Figure 4.7.1 is the block diagram of the clock timer.

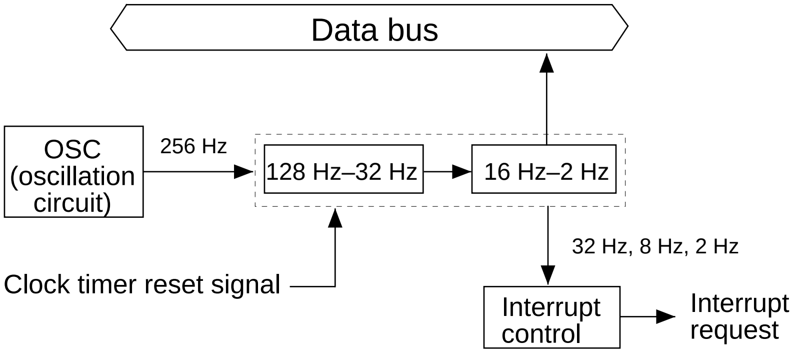


Fig. 4.7.1
Block diagram of clock timer

Normally, this clock timer is used for all kinds of timing purpose, such as clocks.

Interrupt function

The clock timer can interrupt on the falling edge of the 32 Hz, 8 Hz, and 2 Hz signals. The software can mask any of these interrupt signals.

Figure 4.7.2 is the timing chart of the clock timer.

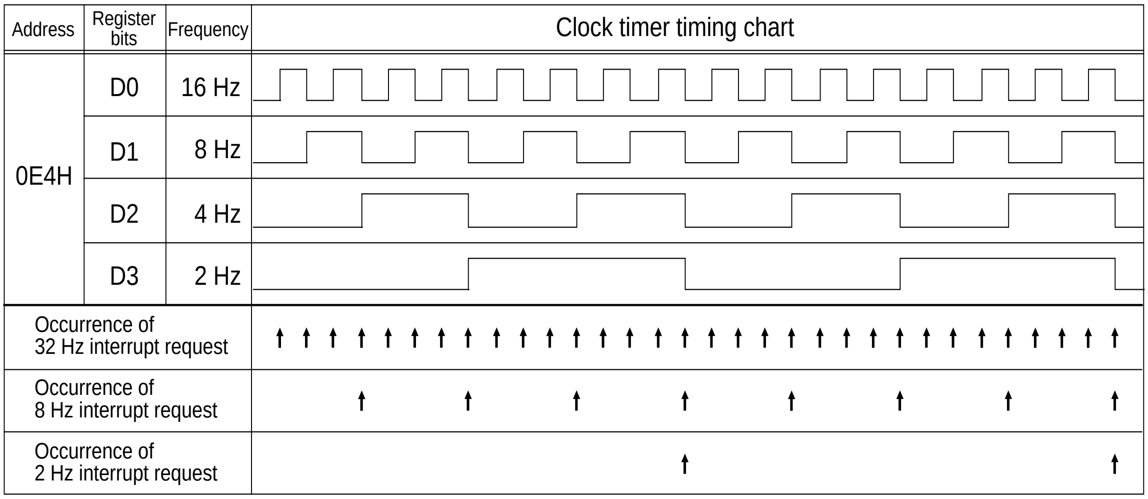

Address
Register bits
Frequency
Clock timer timing chart
0E4H
D0
16 Hz
D1
8 Hz
D2
4 Hz
D3
2 Hz
Occurrence of 32 Hz interrupt request
Occurrence of 8 Hz interrupt request
Occurrence of 2 Hz interrupt request

Fig. 4.7.2 Timing chart of the clock timer

As shown in Figure 4.7.2, an interrupt is generated on the falling edge of the 32 Hz, 8 Hz, and 2 Hz frequencies. When this happens, the corresponding interrupt event flag (IT32, IT8, IT2) is set to 1. Masking the separate interrupts can be done with the interrupt mask register (EIT32, EIT8, EIT2). However, regardless of the interrupt mask register setting, the interrupt event flags will be set to 1 on the falling edge of their corresponding signal (e.g. the falling edge of the 2 Hz signal sets the 2 Hz interrupt factor flag to 1).

Note *Write to the interrupt mask register (EIT32, EIT8, EIT2) and read the interrupt factor flags (IT32, IT8, IT2) only in the DI status (interrupt flag = 0). Otherwise, it causes malfunction.*

Control of clock timer

Table 4.7.1 shows the clock timer control bits and their addresses.

Table 4.7.1 Control bits of clock timer

Address	Register								Comment
	D3	D2	D1	D0	Name	SR	1	0	
0E4H	TM3	TM2	TM1	TM0	TM3	–	High	Low	Timer data (clock timer 2 Hz)
	R				TM2	–	High	Low	Timer data (clock timer 4 Hz)
					TM1	–	High	Low	Timer data (clock timer 8 Hz)
					TM0	–	High	Low	Timer data (clock timer 16 Hz)
0EBH	0	EIT2	EIT8	EIT32	0				
	R	R/W			EIT2	0	Enable	Mask	Interrupt mask register (clock timer 2 Hz)
					EIT8	0	Enable	Mask	Interrupt mask register (clock timer 8 Hz)
					EIT32	0	Enable	Mask	Interrupt mask register (clock timer 32 Hz)
0EFH	0	IT2	IT8	IT32	0				
	R				IT2	0	Yes	No	Interrupt factor flag (clock timer 2 Hz)
					IT8	0	Yes	No	Interrupt factor flag (clock timer 8 Hz)
					IT32	0	Yes	No	Interrupt factor flag (clock timer 32 Hz)
0F9H	0	TMRST	SWRUN	SWRST	0				
	R	W	R/W	W	TMRST	Reset	Reset	–	Clock timer reset
					SWRUN	0	Run	Stop	Stopwatch timer RUN/STOP
					SWRST	Reset	Reset	–	Stopwatch timer reset

TM0–TM3 Timer data (0E4H)

The 16 Hz to 2 Hz timer data of the clock timer can be read from this register. These four bits are read-only, and write operations are invalid.

After an initial reset, the timer data is initialized to 0H.

EIT32, EIT8, EIT2 Interrupt mask registers (0EBH D0–D2)

These registers are used to mask the clock timer interrupt.

When 1 is written:	Enabled
When 0 is written:	Masked
Reading:	Valid

The interrupt mask register bits (EIT32, EIT8, EIT2) mask the corresponding interrupt frequencies (32 Hz, 8 Hz, 2 Hz). Writing to the interrupt mask registers should be done only in the DI status. Otherwise, it causes malfunction.

After an initial reset, these registers are all set to 0.

IT32, IT8, IT2 Interrupt factor flags (0EFH D0–D2)

These flags indicate the status of the clock timer interrupt.

When 1 is read:	Interrupt has occurred
When 0 is read:	Interrupt has not occurred
Writing:	Invalid

The interrupt factor flags (IT32, IT8, IT2) correspond to the clock timer interrupts (32 Hz, 8 Hz, 2 Hz). The software can determine from these flags whether there is a clock timer interrupt. However, even if the interrupt is masked, the flags are set to 1 on the falling edge of the signal. These flags can be reset when the register is read by the software. Also, the flags should be read only in the DI status (interrupt flag = 0). Otherwise, it causes malfunction.

After an initial reset, these flags are set to 0.

TMRST Clock timer reset (0F9H D2)

This bit resets the clock timer.

When 1 is written:	Clock timer reset
When 0 is written:	No operation
Reading:	Always 0

The clock timer is reset by writing 1 to TMRST. The clock timer starts immediately after this. No operation results when 0 is written to TMRST.
This bit is write-only, and so is always 0 when read.

4.8 Stopwatch Timer

Configuration of stopwatch timer

The S1C62N81 Series incorporate a 1/100 sec and 1/10 sec stopwatch timer. The stopwatch timer is configured as a two-stage, four-bit BCD timer serving as the clock source for an approximately 100 Hz signal (obtained by approximately dividing the 256 Hz signal output from the prescaler). Data can be read out four bits at a time by the software.
Figure 4.8.1 is the block diagram of the stopwatch timer.

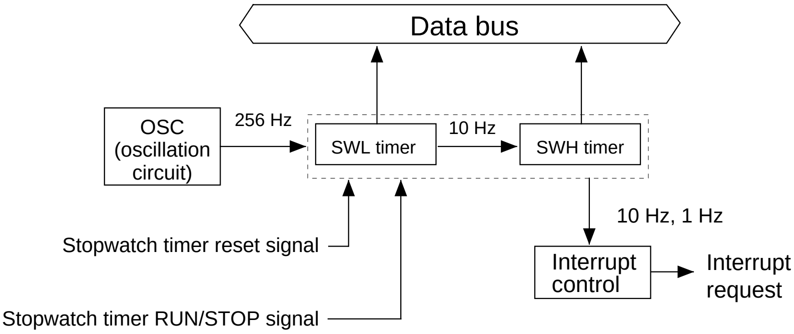

Data bus
OSC (oscillation circuit)
256 Hz
SWL timer
10 Hz
SWH timer
10 Hz, 1 Hz
Stopwatch timer reset signal
Stopwatch timer RUN/STOP signal
Interrupt control
Interrupt request

Fig. 4.8.1
Block diagram of stopwatch timer

The stopwatch timer can be used separately from the clock timer. In particular, digital stopwatch functions can be easily realized by software.

Count-up pattern

The stopwatch timer is configured as two four-bit BCD timers, SWL and SWH. The SWL timer, at the stage preceding the stopwatch timer, has an approximate l00 Hz signal as its input clock. It counts up every 1/100 sec and generates an approximate 10 Hz signal. The SWH timer has an approximate 10 Hz signal generated by the SWL timer for its input clock. It counts up every 1/10 sec and generates a 1 Hz signal.

Figure 4.8.2 shows the count-up pattern of the stopwatch timer.

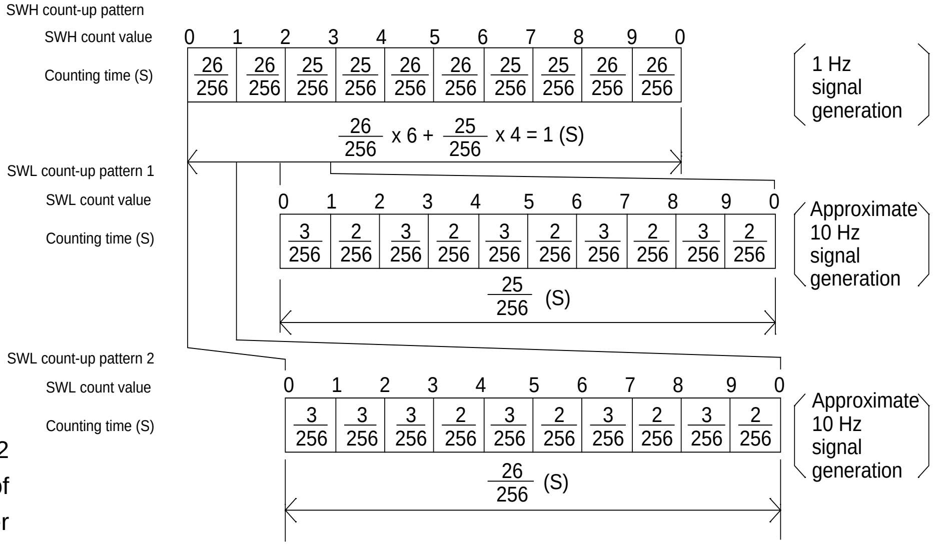


Fig. 4.8.2
Count-up pattern of stopwatch timer

SWL generates an approximate 10 Hz signal from the 256 Hz based signal. The count-up intervals are 2/256 sec and 3/256 sec, so that two final patterns are generated: a 25/256 sec interval and a 26/256 sec interval. Consequently, the count-up intervals are 2/256 sec and 3/256 sec, which do not amount to an accurate 1/100 sec. SWH counts the approximate 10 Hz signals generated by the 25/256 sec, and 26/256 sec intervals in the ratio of 4:6 to generate a l Hz signal. The count-up intervals are 25/256 sec and 26/256 sec, which do not amount to an accurate 1/10 sec.

Interrupt function

The 10 Hz (approximate 10 Hz) and 1 Hz interrupts can be generated by the overflow of the SWL and SWH stopwatch timers, respectively. Also, software can separately mask the frequencies as described earlier.
Figure 4.8.3 is the timing chart for the stopwatch timer.

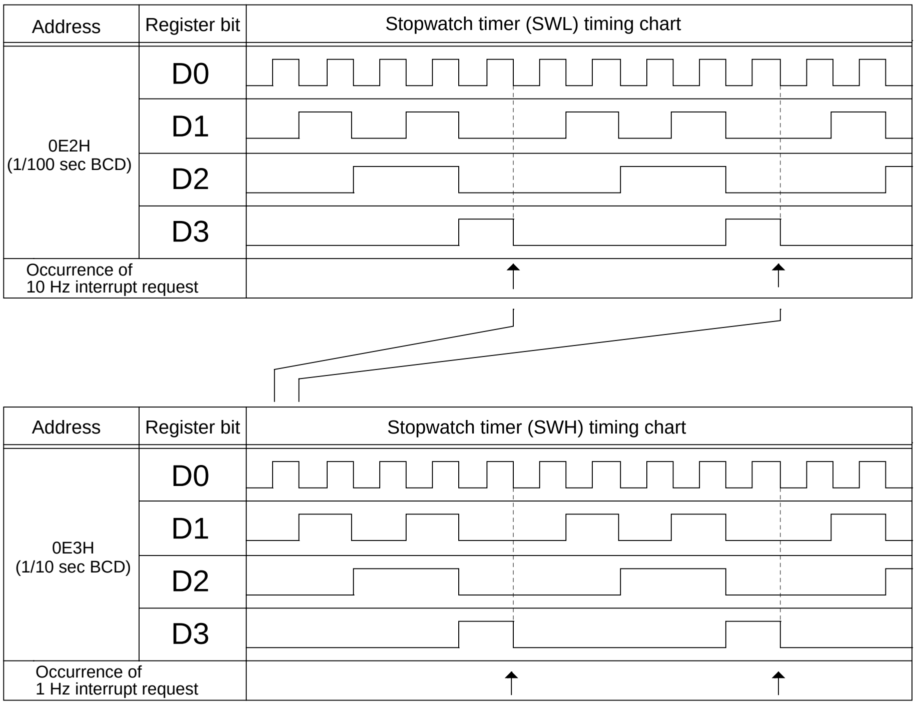


Fig. 4.8.3
Timing chart for
stopwatch timer

As shown in Figure 4.8.3, the interrupts are generated by the overflow of the respective timers (9 changing to 0). Also when this happens, the corresponding interrupt factor flags (ISW0, ISW1) are set to 1. The respective interrupts can be masked separately with the interrupt mask registers (EISW0, EISW1). However, regardless of the setting of the interrupt mask registers, the interrupt factor flags are set to 1 by the overflow of the corresponding timers.

Note *Write to the interrupt mask registers (EISW0, EISW1) and read the interrupt factor flags (ISW0, ISW1) only in the DI status (interrupt flag = 0). Otherwise, it causes malfunction.*

Control of stopwatch timer

Table 4.8.1 shows the stopwatch timer control bits and their addresses.

Table 4.8.1 Stopwatch timer control bits

Address	Register								Comment
	D3	D2	D1	D0	Name	SR	1	0	
0E2H	SWL3	SWL2	SWL1	SWL0	SWL3	0			MSB
	R				SWL2	0			Stopwatch timer 1/100 sec (BCD)
					SWL1	0			
					SWL0	0			LSB
0E3H	SWH3	SWH2	SWH1	SWH0	SWH3	0			MSB
	R				SWH2	0			Stopwatch timer 1/10 sec (BCD)
					SWH1	0			
					SWH0	0			LSB
0EAH	0	0	EISW1	EISW0	0				
	R		R/W		0				
					EISW1	0	Enable	Mask	Interrupt mask register (stopwatch 1 Hz)
					EISW0	0	Enable	Mask	Interrupt mask register (stopwatch 10 Hz)
0EEH	0	0	ISW1	ISW0	0				
	R				0				
					ISW1	0	Yes	No	Interrupt factor flag (stopwatch 1 Hz)
					ISW0	0	Yes	No	Interrupt factor flag (stopwatch 10 Hz)
0F9H	0	TMRST	SWRUN	SWRST	0				
	R	W	R/W	W	TMRST	Reset	Reset	–	Clock timer reset
					SWRUN	0	Run	Stop	Stopwatch timer RUN/STOP
					SWRST	Reset	Reset	–	Stopwatch timer reset

SWL0–SWL3 1/100 sec stopwatch timer (0E2H)

Data (BCD) of the 1/100 sec column of the stopwatch timer can be read. These four bits are read-only, and cannot be written to.
After an initial reset, the timer data is set to 0H.

SWH0–SWH3 1/10 sec stopwatch timer (0E3H)

Data (BCD) of the 1/10 sec column of the stopwatch timer can be read. These four bits are read-only, and cannot be written to.
After an initial reset, the timer data is set to 0H.

EISW0, EISW1 Interrupt mask register (0EAH D0 and D1)

These registers mask the stopwatch timer interrupt.

When 1 is written:	Enabled
When 0 is written:	Masked
Reading:	Valid

The interrupt mask register bits (EISW0, EISW1) are used to mask the 10 Hz and 1Hz interrupts, respectively. Writing to the interrupt mask registers should be done only in the DI status (interrupt flag = 0). Otherwise, it causes malfunction.
After an initial reset, these registers are both set to 0.

ISW0, ISW1 Interrupt factor flags (0EEH D0 and D1)

These flags indicate the status of the stopwatch timer interrupt.

When 1 is read:	Interrupt has occurred
When 0 is read:	Interrupt has not occurred
Writing:	Invalid

The interrupt factor flags (ISW0, ISW1) correspond to the 10 Hz and 1 Hz interrupts, respectively. With these flags, the software can determine whether a stopwatch timer interrupt has occurred. However, regardless of the interrupt mask register setting, these flags are set to 1 by the timer overflow. They are reset when the register is read by the software. Also, reading should be done only in the DI status (interrupt flag = 0). Otherwise, it causes malfunction.
After an initial reset, these flags are set to 0.

SWRST Stopwatch timer reset (0F9H D0)

This bit resets the stopwatch timer.

When 1 is written:	Stopwatch timer reset
When 0 is written:	No operation
Reading:	Always 0

The stopwatch timer is reset when 1 is written to SWRST. When the stopwatch timer is reset while running, operation restarts immediately. Also,while stopped, the reset data is maintained.
This bit is write-only, and is always 0 when read.

SWRUN Stopwatch timer run/stop (0F9H D1)

This bit controls run/stop of the stopwatch timer.

When 1 is written:	Run
When 0 is written:	Stop
Reading:	Valid

The stopwatch timer runs when 1 is written to SWRUN, and stops when 0 is written.
When stopped, the timer data is maintained until the timer next Run or is reset. Also, when the timer runs after being stopped, the data that was maintained can be used to resume the count.
If the timer data is read while running, a correct read may be impossible because of the carry from the low-order bit (SWL) to the high-order bit (SWH). This occurs if reading has extended over the SWL and SWH bits when the carry occurs. To prevent this, read after stopping, and then continue running. Also, the stopped duration must be within 976 µs (256 Hz, 1/4 cycle).
After an initial reset, this register is set to 0.

4.9 Battery Voltage Low Detection (BLD) Circuit and Heavy Load Protection Function

Configuration of BLD circuit and heavy load protection function

The S1C62N81 Series have a built-in battery voltage low detection (BLD) circuit and a heavy load protection function. Figure 4.9.1 shows the configuration of the circuit.

BLD circuit

The BLD circuit monitors the conditions of the battery voltage (supply voltage), and software can check whether the battery voltage has dropped below the detecting voltage level of the BLD circuit: 2.4 V when supply voltage is 3.0 V (S1C62N81), or 1.2 V when supply voltage is 1.5 V (S1C62L81). Registers BLDON (BLD control on/off) and BLDDT (BLD data) are used for the BLD circuit. The software can turn BLD operation on and off. When BLD is on, the IC draws a large current, so keep BLD off unless it is. Since battery voltage detection is automatically performed by the hardware every 2 Hz (0.5 second) when the heavy load protection function operates, do not permit the operation of the BLD circuit by the software in order to minimize power current consumption.

Heavy load protection function circuit

When using the S1C62N81, the melody, lamp, and other features impose a heavy load on the battery. Therefore, a heavy load protection function is incorporated in case of a voltage drop. Software-initiated switching can be effected in heavy load protection mode. The HLMOD register controls the heavy load protection function. Conversely, when the BLD circuit detects a voltage drop below 1.2 V (S1C62L81), or 2.4 V (S1C62N81), switching to heavy load protection mode is carried out automatically.
This function enables 0.9 V operation (S1C62L81/62B81).

In the heavy load protection mode, the BLD circuit is activated intermittently by hardware. The cycle is 2 Hz and the operating time is 122 µs (when the oscillation frequency, fosc, of the oscillation circuit is 32.768 kHz). If the source voltage is reduced by a heavy load while in the heavy load protection mode, the rate of decrease can be detected by hardware. The result is that the heavy load is lost. Even when the heavy load protection mode is released by software, the mode continues until the source voltage exceeds the voltage detected by the BLD circuit. Therefore, malfunctioning due to a reduced source voltage can be prevented completely.

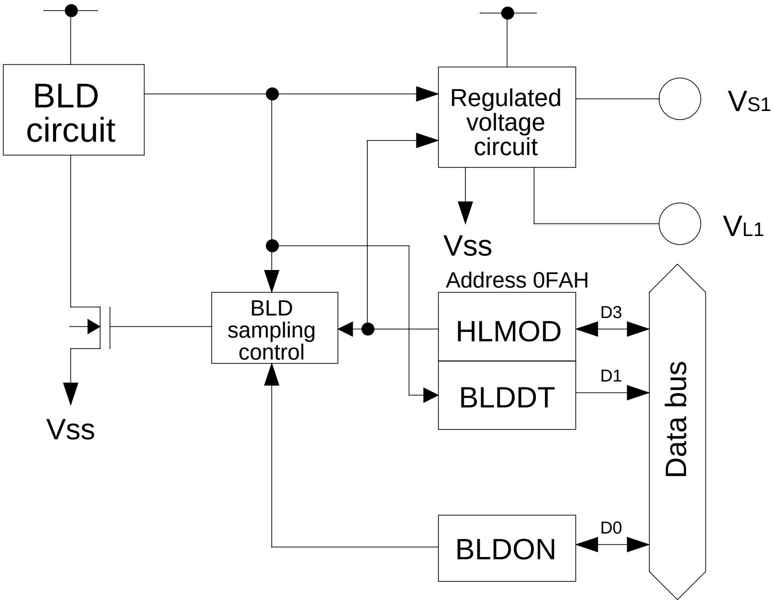


Fig. 4.9.1
Configuration of BLD and heavy load protection circuits

Operation of BLD detection timing

The following explains the timing when the BLD circuit writes the result of battery voltage detection to the BLDDT register.
The result of battery voltage detection is written to the BLDDT register by the BLD circuit, and this data can be read by the software to determine the battery voltage.
There are two methods, explained below, for executing the detection by the BLD circuit.

(1) Sampling with HLMOD set to 1

When HLMOD is set to 1 and BLD sampling is executed, the detection results can be written to the BLDDT register with the following timing:

Immediately after sampling with the 2 Hz cycle output by the oscillation circuit while HLMOD = 1 (sampling time is 122 µs in the case of fosc = 32.768 kHz).

Consequently, after HLMOD has been set to 1, the new detection result is written in a 2 Hz.

(2) Sampling with BLDON set to 1

When BLDON is set to 1, BLD detection is executed. As soon as BLDON is reset to 0, the result is loaded to in the BLDDT register. To obtain a stable BLD detection result, the BLD circuit must be on for at least 100 µs. So, to obtain the BLD detection result, follow the programming sequence below.

① Set BLDON to 1
② Maintain for 100 µs minimum
③ Set BLDON to 0
④ Read BLDDT

However, at 32 kHz for the S1C62N81 and S1C62L81, the instruction cycles are long enough, so there is no need to worry about maintaining 100 µs for BLDON = 1 in the software.

Notice that even if the BLD circuit detects a drop in the supply voltage (1.2 V/2.4 V or less) and invokes the heavy load protection mode, this will be the same as when the software invokes the heavy load protection mode, in that the BLD circuit will be sampled with a timing synchronized to the 2 Hz output from the prescaler. If the BLD circuit detects a voltage drop and enters the heavy load protection mode, it will return to the normal mode once the supply voltage recovers and the BLD circuit determines that the supply voltage is 1.2 V/2.4 V or more.

Operation of heavy load protection function

The S1C62N81 has a heavy load protection function for when the battery load becomes heavy and the supply voltage drops, such as when a melody is played or an external lamp lights. This functions works in the heavy load protection mode. The normal mode changes to the heavy load protection mode in the following two cases:

① When the software changes the mode to the heavy load protection mode

② When the BLD circuit detects a supply voltage less than 2.4 V (S1C62N81) or 1.2 V (S1C62L81), in which case the mode is automatically changed to the heavy load protection mode

Based on the operation of the BLD circuit and the heavy load protection function, the S1C62L81 obtains an operation supply voltage as low as 0.9 V. See the electrical characteristics for the precision of voltage detection by the BLD circuit.

In the heavy load protection mode, the internally regulated voltage is generated by the liquid crystal driver supply output, V_{L2}, in order to operate the internal circuit. Consequently, more current is consumed in the heavy load protection mode than in the normal mode. Unless necessary, do not select the heavy load protection mode with the software.

Note *Activation of the BLD circuit by software in the heavy load protection mode causes a malfunction. Avoid such activation if possible.*

Control of BLD circuit and heavy load protection function

Table 4.9.1 shows the control bits and their addresses for the BLD circuit and the heavy load protection function.

Table 4.9.1 Control bits for BLD circuit and heavy load protection function

Address	Register								Comment
	D3	D2	D1	D0	Name	SR	1	0	
0FAH	HLMOD	0	BLDDT	BLDON	HLMOD	0	Heavy load	Normal load	Heavy load protection mode register
	R/W	R		R/W	0				
					BLDDT	0	Battery voltage low	Battery voltage normal	BLD data
					BLDON	0	ON	OFF	BLD ON/OFF

HLMOD Heavy load protection mode on/off (0FAH D3)

When 1 is written:	Heavy load protection mode on
When 0 is written:	Heavy load protection mode off
Reading:	Valid

When HLMOD is set to 1, the IC enters the heavy load protection mode, and sampling control is executed for the time the BLD circuit is on. The sampling timing is as follows:

Sampling in cycles of 2 Hz output by the oscillation circuit while HLMOD = 1 (sampling time is 122 µs in the case of fosc = 32.768 kHz).

When BLD sampling is done with HLMOD set to 1, the results are written to the BLDDT register with the as following timing:

Immediately on completion of sampling in cycles of 2 Hz output by the oscillation circuit while HLMOD = 1.

Consequently, after HLMOD is set to 1, the new detected result is written in 2 Hz.
In the heavy load protection mode, the consumed current becomes larger. Unless necessary, do not select the heavy load protection mode with the software.

BLDON BLD control on/off (0FAH D0)

When 0 is written:	BLD detection off
When 1 is written:	BLD detection on
Reading:	Valid

When this bit is written, the BLD detection on/off operation is controlled. Large current is drawn during BLD detection, so keep BLD detection off except when necessary. When BLDON is set to 1, BLD detection is executed. As soon as BLDON is reset to 0, the detected result is loaded into the BLDDT register.

BLDDT BLD data (0FAH D1)

When 0 is read:	Supply voltage ≥ Criteria voltage
When 1 is read:	Supply voltage < Criteria voltage

When BLDDT is 1, the S1C62N81 enters the heavy load protection mode. In this mode, the detection operation of the BLD circuit is sampled in 2 Hz cycles and the respective detection results are written to the BLDDT register.

4.10 Analog Voltage Comparator

Configuration of analog voltage comparator

The S1C62N81 Series have a built-in analog voltage comparator that compares two analog input voltages to produce result data 0 or 1 in register CMPDT, according to the compared voltages, CMPP and CMPM. The configuration of the analog voltage comparator circuit is shown in Figure 4.10.1. The voltage comparator has two analog voltage inputs, CMPP (non-inverting input, +) and CMPM (inverting input, -). When the voltage comparator is turned on by control register CMPON, the result of comparing CMPP and CMPM will be stored in register CMPDT. Therefore, the result in the register will indicate whether CMPP is greater than CMPM (when CMPDT = 1) or smaller than CMPM (when CMPDT = 0).

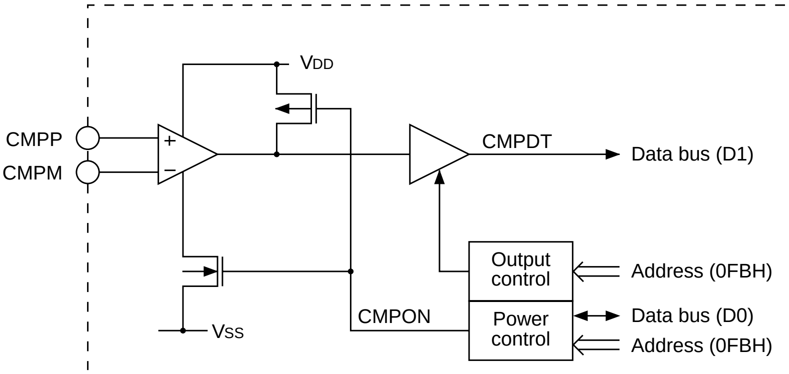


Fig. 4.10.1
Configuration of analog voltage comparator circuit

Operation of analog voltage comparator

Two registers, CMPON and CMPDT, are used in the analog voltage comparator. The CMPON register switches the analog voltage comparator on or off to reduce power consumption. The CMPDT register indicates the result of comparison of the CMPP and CMPM pins.
Writing 1 to the CMPON register turns on the comparator circuit. After an initial reset, this bit is set to 0. Data in the CMPON register is read-accessible or write-accessible. A wait time of at least 3 ms is required for analog voltage comparator to become stable after its power is turned on. The comparator response time depends on the potential difference between the CMPP and CMPM inputs.

When analog voltage comparator is turned on, the circuit compares the two analog voltages from the CMPP and CMPM inputs, then outputs the result as binary 0 (CMPM>CMPP) or 1 (CMPP>CMPM). The result of the comparison is read from the CMPDT register. Writing to the CMPDT register is prohibited.

Note *Data in the CMPDT register becomes 1 when CMPON is 0 (analog voltage comparator circuit is off), and undefined when the CMPP and / or CMPM input is disconnected. Avoid reading operation under those conditions.*

Control of analog voltage comparator

Table 4.10.1 lists the control bits of the analog voltage comparator and their addresses.

Table 4.10.1 Control bits of analog voltage comparator

Address	Register								Comment
	D3	D2	D1	D0	Name	SR	1	0	
0FBH	CSDC	0	CMPDT	CMPON	CSDC	0	Static	Dynamic	LCD drive switch
	R/W	R		R/W	0				
					CMPDT	1	+ > -	- > +	Comparator's voltage condition: 1 = CMPP(+)input > CMPM(-)input, 0 = CMPM(-)input > CMPP(+)input
					CMPON	0	On	Off	Voltage comparator ON/OFF

CMPON Comparator on/off control (0FBH D0)

Switches the analog voltage comparator circuit to on or off.

When 1 is written:	Comparator turns on
When 0 is written:	Comparator turns off
Reading:	Valid

After an initial reset, this bit is set to 0.

Note *While analog voltage comparator is ON, the consumed current becomes large. Unless necessary, do not turn on the analog comparator.*

CMPDT Comparator data (0FBH D1)

Shows the result of comparing CMPP and CMPM.

When 1 is read:	CMPP voltage is greater than CMPM voltage
When 0 is read:	CMPP voltage is smaller than CMPM voltage
Writing:	Invalid

This bit is undefined when the CMPP and/or CMPM input pin is disconnected, and is 1 when CMPON is 0.
After an initial reset, this bit is set to 1.

4.11 Melody Generator

Outline of melody generator

The S1C62N81 Series has built-in melody generator. Outputs related to the melody function are generated from MO terminal or R12 terminal. The following 3 types of melody playing may be selected through the mask option:

(1) Piezo buzzer single terminal driving through the MO terminal

The R12 output is set to DC output through the mask option. Melody is output from the MO terminal alone. This setting increases the number of externally fitted parts to play the melody but since the R12 output may be used as a common high-power current output, it is useful when high-power current driving common output is required.

(2) Piezo buzzer direct driving through the MO and R12 outputs

The R12 output is set to piezo direct driving through the mask option. Reversed signal of the MO terminal output signal is output from the R12 terminal. This allows the piezo buzzer direct driving to materialize. This setting makes it possible to keep the number of externally fitted parts to the minimum.

(3) Envelope driving

The R12 output is set to the envelope function through the mask option. Sound pressure of the playing is attenuated with time, making it possible to implement a fully expressive playing.

Refer to Chapter 5, "BASIC EXTERNAL WIRING DIAGRAM" for the respective external wirings.

The characteristics of the melody generator are as follows:

(1) **Size of the Melody ROM:** 80 words

Basically, one note is equivalent to one word. Any number of melodies may be written as long as it is within 80 words. Data such as note length, intervals and end of melody may be written.

(2) **Size of Scale ROM:** 15 scales

C3–C6$^{\#}$ (without frequency booster) or C4–C7$^{\#}$ (with frequency booster) may be selected from among 15 scales. The use of frequency booster may also be selected by the mask option.

(3) **Playing mode:**

There are 3 playing modes.

① One shot mode (only 1 melody is played)

② Level hold mode (the same or a different melody is continuously played)

③ Retrigger mode (forced change or termination of melody)

(4) **Tempo:**

2 types may be selected from among 16 types through the mask option.

(5) **Playing speed:**

Aside from the normal speed mode, 8 times, 16 times, and 32 times speed mode may be controlled through software. This function allows the generation of sound effects.

The block diagram of the melody generator is shown in Figure 4.11.1. The note and interval data of the melody to be played is pre-written on the melody ROM. The interval data of the melody ROM is used to specify the scale ROM address and according to the scale ROM data read from it, the interval generating circuit generates the interval. The output is controlled at the melody output control circuit and is output at the MO and R12 terminals. The note generator is generated according to the melody ROM data. The output is entered in the melody ROM address counter; every time the playing of a note is completed, one address is incremented. This results in continuous melody being automatically played. The playing tempo is created by the tempo generator based on the signal which divided the oscillation frequency in the oscillation circuit. Through the mask option, 2 types of tempo may be selected from among 16 types. Moreover, the division ratio of the divider may be modified by software and 4 types of playing speed can be implemented. Envelope function may also be added to the output melody and R12 output may be implemented by setting it to correspond with the envelope.

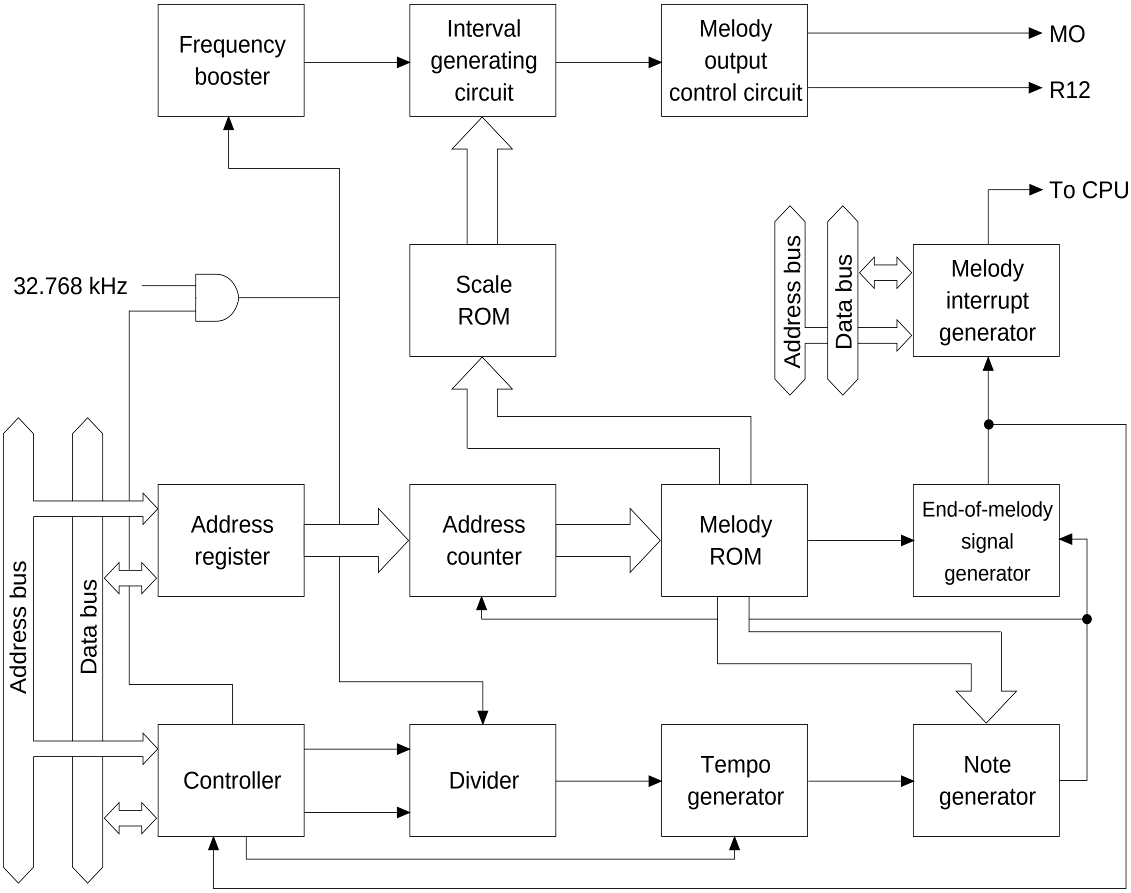


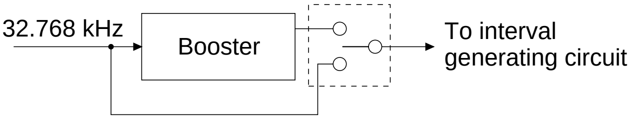
Fig. 4.11.1
Melody generator
block diagram

A detailed description of the circuits which form the melody generator is provided below.

(a) Frequency booster

The configuration of the frequency booster is shown in Figure 4.11.2. It is a circuit which raises the input frequency (32.768 kHz) for the melody generator to 2 times the frequency. The output of this frequency booster is provided with a switch through the mask option; by selecting this switch, scale which can be output may be changed. In other words, if frequency booster output were selected for input to interval generating circuit, interval can be created between C4 to C7# and if 32.768 kHz were selected as is, interval can be created between C3 to C6#.

Fig. 4.11.2
Frequency booster



(b) Controller

The configuration of the controller is shown in Figure 4.11.3. The controller consists of a 4-bit register located in the I/O RAM space and an ON/OFF control circuit and controls the melody's ON/OFF, tempo selection, playing speed selection. The ON/OFF control circuit controls the turning ON/OFF of the melody playing by entering the MELC register output and the signal from the end-of-melody signal generator. The address of the 4-bit register is "0F2H" and the meaning of each bit is as follows:

D0 (MELC):

This is the bit that controls the turning ON/OFF of the melody playing. The controlling function of this bit makes it possible to control the above-described 3 types of playing. Refer to "Playing mode" regarding the method of control.

D1 (TEMPC):

This is the bit that selects the tempo. 2 types of tempo selected by mask option may be changed. The timing of tempo change is not done when data is written on this bit but rather, when the next melody begins.

D2 and D3 (CLKC0 and CLKC1):

This is the bit that changes playing speed. By the combination of CLKC0 and CLKC1, 4 types of playing speed may be selected. The playing speed for the selectable tempo listed in Table 4.11.7 is the normal speed; playing speeds which are 8, 16 and 32 times the normal speed may also be selected. This is useful in generating sound effects. For details, see "Playing tempo".

Note *Since playing speed is modified simultaneously with data writing on these bits, caution must be observed when operating these bits in the middle of a playing.*

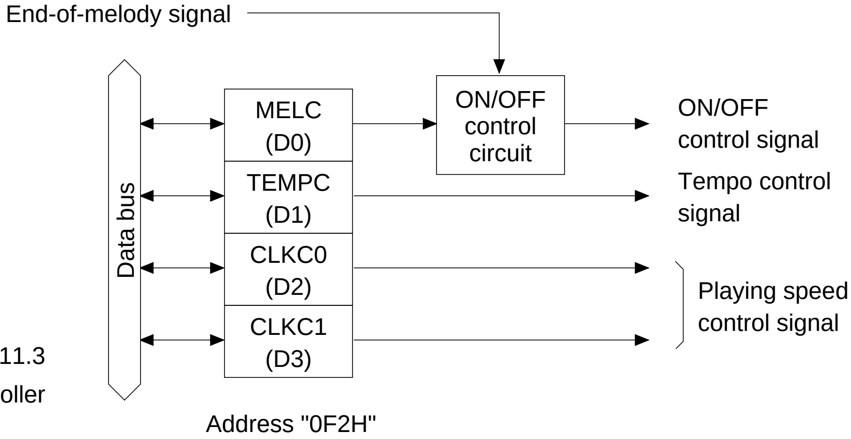


Fig. 4.11.3
Controller

(c) Address register

The configuration of the melody ROM address registers is shown in Figure 4.11.4. It consists of the 7-bit register in the I/O RAM space. The addresses are "0F0H" and "0F1H". The data of these registers indicate the addresses of the melody ROM which become the addresses of the melody ROM when the melody is started. These melody ROM addresses are written to the melody ROM address counter when the melody playing begins, i.e., before the the melody playing begins, the desired melody may be played from among the melodies written in the melody ROM by setting data on these registers.

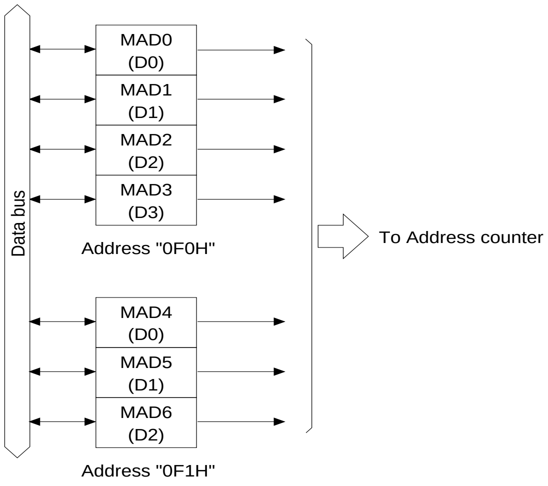


Fig. 4.11.4
Address register

Note *Caution must be observed because when an address from "50H" to "7FH" is set on the address register, since the address does not exist in the melody ROM hardware-wise, all melody ROM output will become "1" and silent notes equivalent to 32 notes will be played.*

(d) Address counter

The configuration of the melody ROM address counters is shown in Figure 4.11.5. It consists of a counter in which note playing end signal generated from the note generator is entered and which increases the melody ROM addresses by 1 address every time a note playing is completed. Moreover, when a melody playing begins, address register data (MAD0 to MAD6) are set on these counters. This causes the address set in the address register to specify the melody ROM address.

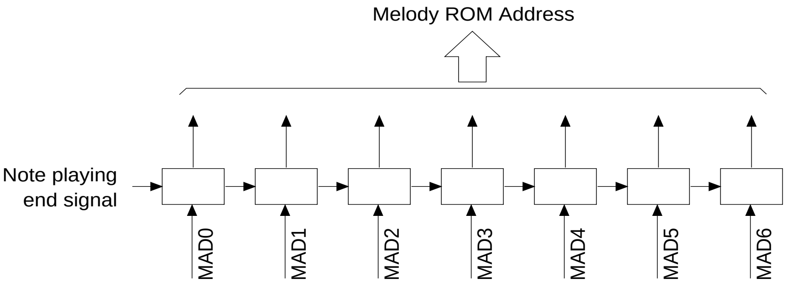


Fig. 4.11.5
Address counters

(e) Melody ROM

The melody ROM is a mask ROM with 80 words × 9 bits capacity in which data of the melody to be played (note, interval, end-of-melody, etc.) is stored beforehand. Any number of melodies may be stored as long as the total number of notes is within 80 words (basically, 1 note/ word). Details regarding the melody ROM configuration, etc., can be found in next ection, "Melody data".

Note *When the melody ROM is set with a non-existent address ("50H" and above), all of its output will become "1" (i.e., 32 silent notes) by the hardware.*

(f) Divider

The configuration of the divider is shown in Figure 4.11.6. It is a circuit that divides the clock (32.768 kHz) which is input in the melody generator and inputs the divided clock into the tempo generator. The dividing ratio may be controlled by software. The data of the "CLKC0" and "CLKC1" registers in the above-mentioned controller is input and the dividing ratio will differ according to the value of the input data. The dividing ratio and playing speed for the combinations of CLKC0 and CLKC1 values are shown in Table 4.11.1. The "normal" speed in the playing speed column refers to the playing speed by which the tempo listed in Table 4.11.7 may be implemented. playing speeds 8 times (the normal speed) or more are useful for generating sound effects.

Table 4.11.1
Dividing ratio

CLKC1	CLKC0	Dividing Ratio	Playing Speed
0	0	1/512	Normal
0	1	1/64	8 times
1	0	1/32	16 times
1	1	1/16	32 times

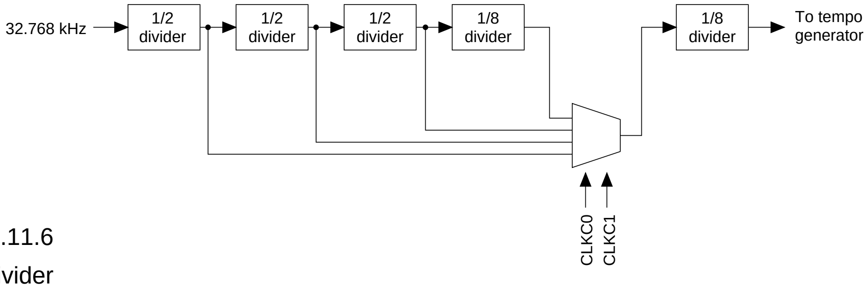


Fig. 4.11.6
Divider

(g) Tempo generator

The configuration of the tempo generator is shown in Figure 4.11.7. The tempo generator is a circuit which generates the 2 types of tempo selected by mask option and consists of the 4-bit counter in which the output signal from the divider is input and the 4 switches which set their respective bit. The 4-bit counter output serves as the note generator input. The 4 switches are automatically set to generate the 2 types of tempo selected by mask option. Bit settings and the corresponding tempo generated are shown in Table 4.11.2. On the other hand, the relationship between the 2 types of tempo selected by mask option and switch settings are shown in Table 4.11.3. For example, if the respective bit values of the 2 types of tempo selected by mask option are "1" for TEMPC = 0 and "0" for TEMPC = 1, the switch setting for this bit combination will be $\overline{\text{TEMPC}}$ (reverse signal of the TEMPC register output).

Table 4.11.2
Counter setting and tempo

TS3	TS2	TS1	TS0	♩ ≒
0	0	0	0	30
0	0	0	1	32
0	0	1	0	34.3
0	0	1	1	36.9
0	1	0	0	40
0	1	0	1	43.6
0	1	1	0	48
0	1	1	1	53.3
1	0	0	0	60
1	0	0	1	68.6
1	0	1	0	80
1	0	1	1	96
1	1	0	0	120
1	1	0	1	160
1	1	1	0	240
1	1	1	1	480

Table 4.11.3
Tempo and switch setting

TEMPC=0	TEMPC=1	Switch Setting
0	0	Pull down
1	0	$\overline{\text{TEMPC}}$
0	1	TEMPC
1	1	Pull up

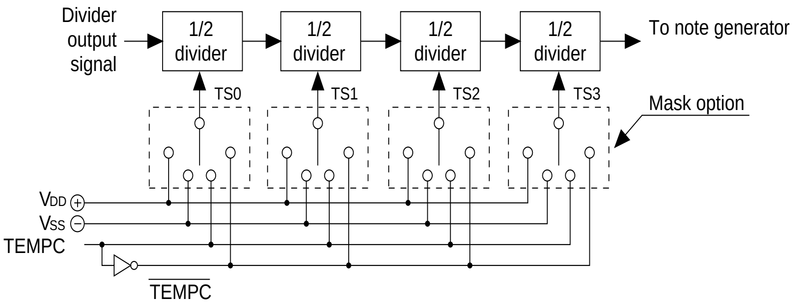


Fig. 4.11.7
Tempo generator

(h) Note generator

This is a generator which counts the tempo generator output and creates various notes. Its configuration is shown in Figure 4.11.8. It consists of counters in which 3 bits can be set. Each counter is set by the 3 bits (D5–D7) from the melody ROM causing the counter dividing ratio to change and hence various notes are generated. The bit settings and the corresponding notes generated are shown in Table 4.11.4. The counter output becomes the note playing end signal and the address of the melody ROM is incremented 1 step at a time.

Table 4.11.4
Note data and notes

D7	D6	D5	Note
0	0	0	𝅗𝅥
0	0	1	♩..
0	1	0	♩.
0	1	1	♩+♪
1	0	0	♩
1	0	1	♪.
1	1	0	♪
1	1	1	𝅘𝅥𝅯

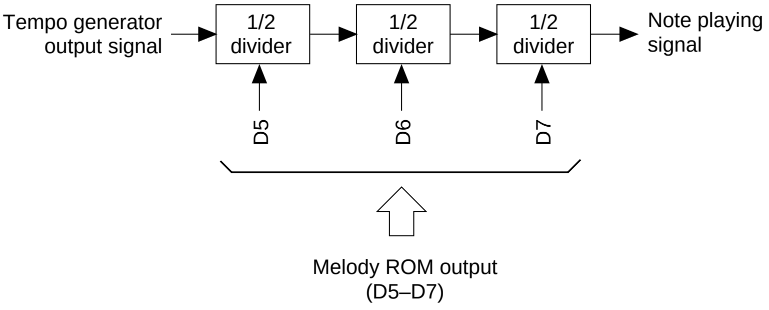


Fig. 4.11.8
Note generator

(i) Scale ROM

This is a mask ROM in which 15 scale types which have been optionally selected and created from either C3–C6# (available output frequency range: 4,096 Hz–125.5 Hz; without frequency booster) or C4–C7# (available output frequency range: 8,192 Hz–251.1 Hz; with frequency booster) are stored beforehand. The 15 available addresses are "00H"–"0EH". Word length is 8 bits; the data written on them and the corresponding scale (frequency) generated are shown in Tables 4.11.5 (a) and (b). The maximum value which may be written as a data is "FDH". The address is specified by the melody ROM output and the output is entered in the interval generating circuit.

Note *Bear in mind that the range of the data which can be written on the scale ROM is from "00H" to "FDH". If any data beyond this range is written, the interval generating circuit will not function normally.*

Table 4.11.5 (a) Scale ROM data and interval (without frequency booster)

Scale Data	Frequency (Hz)	Scale ROM Code									Dividing Ratio	Absolute Error (%)	Standard Frequency (Hz)
		S7	S6	S5	S4	S3	S2	S1	S0	Hex.			
C3	128	0	0	0	0	0	1	0	0	04	1/128 x 1/2	0	128
C3#	135.405	0	0	0	1	0	0	1	0	12	1/121 x 1/2	-0.152	135.611
D3	143.719	0	0	1	0	0	0	0	0	20	1/114 x 1/2	0.031	143.675
D3#	152.409	0	0	1	0	1	1	1	1	2F	1/107 + 103	0.024	152.218
E3	161.419	0	0	1	1	1	0	1	1	3B	1/101 + 102	0.092	161.270
F3	170.667	0	1	0	0	0	1	0	0	44	1/96 x 1/2	-0.113	170.860
F3#	181.039	0	1	0	1	0	0	0	1	51	1/90 + 91	0.010	181.019
G3	191.626	0	1	0	1	1	0	1	1	5B	1/85 + 86	-0.030	191.783
G3#	203.528	0	1	1	0	0	1	0	1	65	1/80 + 81	0.167	203.187
A3	215.579	0	1	1	0	1	1	0	0	6C	1/76 x 1/2	0.143	215.270
A3#	227.556	0	1	1	1	0	1	0	0	74	1/72 x 1/2	-0.226	228.070
B3	240.941	0	1	1	1	1	1	0	0	7C	1/68 x 1/2	-0.287	241.632
C4	256	1	0	0	0	0	1	0	0	84	1/64 x 1/2	0	256
C4#	270.810	1	0	0	0	1	1	0	1	8D	1/60 + 61	-0.153	271.222
D4	287.439	1	0	0	1	0	0	1	0	92	1/57 x 1/2	0.031	287.350
D4#	303.407	1	0	0	1	1	0	0	0	98	1/54 x 1/2	-0.339	304.436
E4	321.255	1	0	0	1	1	1	1	0	9E	1/51 x 1/2	-0.400	322.540
F4	341.333	1	0	1	0	0	1	0	0	A4	1/48 x 1/2	-0.113	341.720
F4#	360.088	1	0	1	0	1	0	1	1	AB	1/45 + 46	-0.542	362.038
G4	385.506	1	0	1	1	0	0	0	1	B1	1/42 + 43	0.503	383.566
G4#	404.543	1	0	1	1	0	1	0	1	B5	1/40 + 41	-0.453	406.374
A4	431.158	1	0	1	1	1	0	0	0	B8	1/38 x 1/2	0.144	430.540
A4#	455.111	1	0	1	1	1	1	0	0	BC	1/36 x 1/2	-0.226	456.140
B4	481.882	1	1	0	0	0	0	0	0	C0	1/34 x 1/2	-0.287	483.264
C5	512	1	1	0	0	0	1	0	0	C4	1/32 x 1/2	0	512
C5#	546.133	1	1	0	0	1	0	0	0	C8	1/30 x 1/2	0.675	542.444
D5	574.877	1	1	0	0	1	1	0	1	CD	1/28 + 29	0.031	574.700
D5#	606.815	1	1	0	0	1	1	1	0	CE	1/27 x 1/2	-0.339	608.872
E5	642.510	1	1	0	1	0	0	1	1	D3	1/25 + 26	-0.400	645.080
F5	682.667	1	1	0	1	0	1	0	0	D4	1/24 x 1/2	-0.113	683.440
F5#	728.178	1	1	0	1	1	0	0	1	D9	1/22 + 23	0.563	724.076
G5	762.047	1	1	0	1	1	0	1	1	DB	1/21 + 22	-0.668	767.132
G5#	819.200	1	1	0	1	1	1	0	0	DC	1/20 x 1/2	0.787	812.748
A5	862.316	1	1	0	1	1	1	1	0	DE	1/19 x 1/2	0.144	861.080
A5#	910.222	1	1	1	0	0	0	0	0	E0	1/18 x 1/2	-0.226	912.280
B5	963.765	1	1	1	0	0	0	1	0	E2	1/17 x 1/2	-0.287	966.528
C6	1024	1	1	1	0	0	1	0	0	E4	1/16 x 1/2	0	1024
C6#	1092.267	1	1	1	0	0	1	1	0	E6	1/15 x 1/2	0.675	1084.888

Table 4.11.5 (b) Scale ROM data and interval (with frequency booster)

Scale Data	Frequency (Hz)	Scale ROM Code									Dividing Ratio	Absolute Error (%)	Standard Frequency (Hz)
		S7	S6	S5	S4	S3	S2	S1	S0	Hex.			
C4	256	0	0	0	0	0	1	0	0	04	1/128 x 1/2	0	256
C4#	270.810	0	0	0	1	0	0	1	0	12	1/121 x 1/2	-0.152	271.222
D4	287.439	0	0	1	0	0	0	0	0	20	1/114 x 1/2	0.031	287.350
D4#	304.819	0	0	1	0	1	1	1	1	2F	1/107 + 103	2.448	304.436
E4	322.837	0	0	1	1	1	0	1	1	3B	1/101 + 102	0.092	322.540
F4	341.333	0	1	0	0	0	1	0	0	44	1/96 x 1/2	-0.113	341.720
F4#	362.077	0	1	0	1	0	0	0	1	51	1/90 + 91	0.011	362.038
G4	383.251	0	1	0	1	1	0	1	1	5B	1/85 + 86	-0.082	383.566
G4#	407.056	0	1	1	0	0	1	0	1	65	1/80 + 81	0.168	406.374
A4	431.158	0	1	1	0	1	1	0	0	6C	1/76 x 1/2	0.143	430.540
A4#	455.111	0	1	1	1	0	1	0	0	74	1/72 x 1/2	-0.226	456.140
B4	481.882	0	1	1	1	1	1	0	0	7C	1/68 x 1/2	-0.287	483.264
C5	512	1	0	0	0	0	1	0	0	84	1/64 x 1/2	0	512
C5#	541.620	1	0	0	0	1	1	0	1	8D	1/60 + 61	-0.152	542.444
D5	574.877	1	0	0	1	0	0	1	0	92	1/57 x 1/2	0.031	574.700
D5#	606.815	1	0	0	1	1	0	0	0	98	1/54 x 1/2	-0.339	608.872
E5	642.510	1	0	0	1	1	1	1	0	9E	1/51 x 1/2	-0.400	645.080
F5	682.667	1	0	1	0	0	1	0	0	A4	1/48 x 1/2	-0.113	683.440
F5#	720.176	1	0	1	0	1	0	1	1	AB	1/45 + 46	-0.541	724.076
G5	771.012	1	0	1	1	0	0	0	1	B1	1/42 + 43	0.503	767.132
G5#	809.086	1	0	1	1	0	1	0	1	B5	1/40 + 41	-0.453	812.748
A5	862.316	1	0	1	1	1	0	0	0	B8	1/38 x 1/2	0.143	861.080
A5#	910.222	1	0	1	1	1	1	0	0	BC	1/36 x 1/2	-0.226	912.280
B5	963.765	1	1	0	0	0	0	0	0	C0	1/34 x 1/2	-0.287	966.528
C6	1024	1	1	0	0	0	1	0	0	C4	1/32 x 1/2	0	1024
C6#	1092.267	1	1	0	0	1	0	0	0	C8	1/30 x 1/2	0.676	1084.888
D6	1149.754	1	1	0	0	1	1	0	1	CD	1/28 + 29	0.031	1149.400
D6#	1213.630	1	1	0	0	1	1	1	0	CE	1/27 x 1/2	-0.339	1217.748
E6	1285.020	1	1	0	1	0	0	1	1	D3	1/25 + 26	-0.399	1290.160
F6	1365.333	1	1	0	1	0	1	0	0	D4	1/24 x 1/2	-0.113	1366.880
F6#	1456.356	1	1	0	1	1	0	0	1	D9	1/22 + 23	0.563	1448.152
G6	1524.093	1	1	0	1	1	0	1	1	DB	1/21 + 22	-0.667	1534.264
G6#	1638.400	1	1	0	1	1	1	0	0	DC	1/20 x 1/2	0.788	1625.496
A6	1724.632	1	1	0	1	1	1	1	0	DE	1/19 x 1/2	0.143	1722.160
A6#	1820.444	1	1	1	0	0	0	0	0	E0	1/18 x 1/2	-0.226	1824.560
B6	1927.529	1	1	1	0	0	0	1	0	E2	1/17 x 1/2	-0.287	1933.056
C7	2048	1	1	1	0	0	1	0	0	E4	1/16 x 1/2	0	2048
C7#	2194.533	1	1	1	0	0	1	1	0	E6	1/15 x 1/2	0.676	2169.776

(j) Interval generating circuit

The interval generating circuit generates the interval (frequency) corresponding to the scale ROM output. Its configuration is shown in Figure 4.11.9. Using the input clock (32.768 kHz) to the melody generator or the 8-bit divider with the booster output (65.536 kHz) as input clock, dividing ratios (1/8–1/261) set by the scale ROM output (S0–S7) can be attained. The divider output passes through the output controller and becomes sound output. Scales which can be output are C3–C6# (available output frequency range: 4,096 Hz–125.5 Hz; without frequency booster) or C4–C7# (available output frequency range: 8,192 Hz–251.1 Hz; with frequency booster). The dividing ratio may be derived from S0–S7 values which are the scale ROM output using the following equation:

N (dividing ratio) = $(/S7 \times 2^6 + /S6 \times 2^5 + /S5 \times 2^4 + /S4 \times 2^3 + /S3 \times 2^2 + /S2 \times 2^1 + /S1 \times 2^0 + 3) \times 2 + S0$
(Note: /SX = reversed value of SX)

Example:

If
(S7, S6, S5, S4, S3, S2, S1, S0) = (1, 1, 1, 0, 0, 1, 0, 0),
then,
$N = (0 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 3) \times 2 + 0 = 32$

In other words, if the input clock were 32.768 kHz, the output will be 32,768/32 = 1,024 Hz (C6).

The selection of input clock may be done by changing the switch (by mask option) explained in the section on booster.

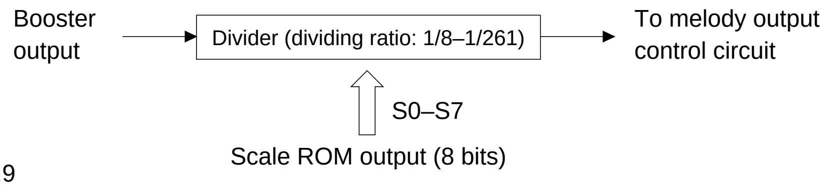

Booster output
Divider (dividing ratio: 1/8–1/261)
To melody output control circuit
S0–S7
Scale ROM output (8 bits)

Fig. 4.11.9
Interval generating circuit

(k) End-of-melody signal generator

This is a circuit that receives the end-of-melody data written on the melody ROM and generates the end-of-melody signal which synchronized with the end of a note playing. The output is entered into the controller and the melody interrupt generator and becomes the source signal which informs the end of a melody.

(l) Melody interrupt generator

The configuration of the melody interrupt generator is shown in Figure 4.11.10. It is a circuit that receives the end-of-melody signal from the end-of-melody signal generator and generates the melody interrupt signal which informs the CPU that a certain melody has been completed. At the same time, it sets an interrupt factor flag the timing of which is shown in Figure 4.11.11. The interrupt factor flag becomes valid approximately 7.8 ms (in case of normal speed) after the end-of-melody signal is generated. The interrupt factor flag may be read out by software and is reset simultaneously with the read out. The register address is "ECH D0". It can also be masked for the interrupt signal and masking can be controlled by software. The mask register address is "E7H D0".

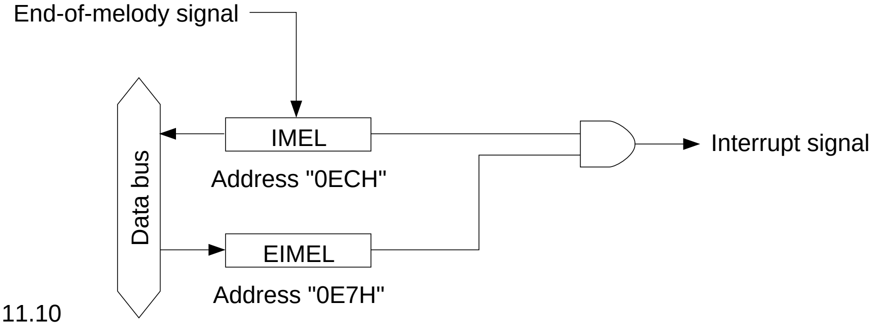


Fig. 4.11.10
Melody interrupt generator

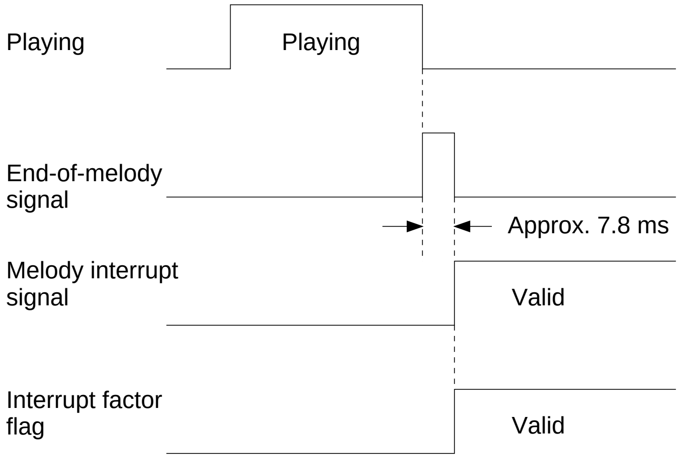


Fig. 4.11.11
Interrupt generation timing

Note *Reading out the interrupt request flag or writing on the mask register should always be performed in the "DI (interrupt prohibited)" state. Otherwise, misoperation may result.*

(m)Melody output control circuit

This is a circuit that determines the form of melody playing (piezo buzzer direct driving and addition of envelope function) according to the mask option selection.

(n) Melody output terminal (M0 and R12)

This is a terminal which produces melody during playing. Its output configuration and output waveform are shown in Figure 4.11.12. The configuration differs if the mask option selection were R12.

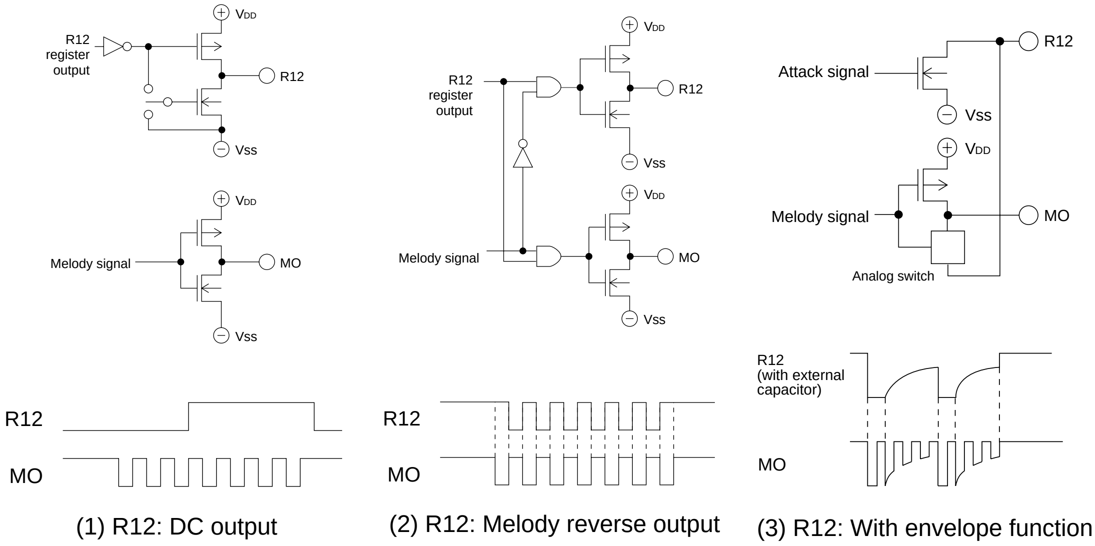


Fig. 4.11.12 Melody terminal output configuration and output waveform

(1)R12: DC output

Melody is output from the MO terminal and from the R12 terminal, data written on the "R12" register is output. The MO terminal is a complementary output terminal and goes high when melody is not played. Complementary output or Pch open-drain output may be selected for the R12 terminal by mask option.

(2)R12: Melody reverse output

Using MO and R12 terminals, the piezo buzzer may be directly driven. During playing, reverse signal of the MO terminal is output from the R12 terminal. Both terminals go high when melody is not being played. The output configuration of both terminals becomes complementary.

(3) R12: With envelope function

Envelope function can be implemented by connecting an external capacitor to the R12 terminal. Melody is output from the MO terminal and the signal which will recharge the external capacitor will be output from the R12 terminal. The R12 electric potential will turn out supplying the negative electric potential of the MO terminal output and when the melody signal goes high, it will pass the analog switch and will be supplied to the MO terminal. For details regarding the envelope function, refer to "Envelope function".

Melody data

- **Melody ROM**

The melody ROM has an 80-word capacity, the length of a word being 9 bits. Basically, data of 1 note is stored in 1 word. These data are continuously read out by the hardware and melody is played. The 4 types of data which may be written as 1-note data are as follows:

(1) Interval data
(2) Note data
(3) End data
(4) Attack data

When melody playing starts, the start address is specified with the address written on the address register. The melody ROM address is then automatically increased by the address counter one step at a time and melody is played. The melody automatically stops at the point where the end-of-melody data written on the melody ROM is read out by the hardware. At the same time, interrupt flag is set and interrupt for the CPU is generated.

Fig. 4.11.13
Data format of
the melody ROM

D8	D7	D6	D5	D4	D3	D2	D1	D0
Attack data	Note data			Scale data				End data

Since only melody start address setting and melody start control may be controlled by software, optional melodies which have been written on the melody ROM can easily be played by lessening the load of the software.
The format of the data contained in a melody ROM word is shown in Figure 4.11.13. These melody data are explained in details below.

- **Note data (D5–D7)**
  Note data are data which indicate the notes to be used. As shown in Figure 4.11.13, note data are written on 3 bits: D5–D7. There are 8 types of notes which can be used in the S1C62N81 Series and the corresponding 3 note data bits are shown in Table 4.11.6. Although notes shorter than 32 notes may not be played, notes longer than 2 notes may be played by operating the above-mentioned attack note. This procedure is explained in the section on attack data.

Table 4.11.6
Note data and notes

D7	D6	D5	Note
0	0	0	𝅗𝅥
0	0	1	♩..
0	1	0	♩.
0	1	1	♩+♪
1	0	0	♩
1	0	1	♪.
1	1	0	♪
1	1	1	𝅘𝅥𝅯

- **Scale data (D1–D4)**
  Intervals to be used are pre-written on the scale ROM. There are 15 scale ROM addresses which can be used: "00H" to "0EH". The addresses are written on the 4 bits (D1–D4; see Figure 4.11.13) which serve as interval data area. Intervals written on the interval ROM address which has been specified with the interval data (refer to Table 4.11.5) are generated at the interval generating circuit. Although the scale ROM addresses are only from "00H" to "0EH", "0FH" also exists in the hardware and is set for silent notes. Because of this, writing "0FH" on the melody ROM interval data area will result in the playing of silent notes. The length of a silent note depends on the note data written on the same word.

- **Attack data (D8)**
  The attack data is a 1-bit data which determines whether or not to make the break between notes clear. In each melody first word, set this data to "1". Otherwise, there will be no melody play even if the user starts play.
  If envelope function is not available, writing "1" for this bit will produce an approximately 12 ms rest every time the melody ROM address increases by 1 step (i.e., at the break of the playing of different notes). This is particularly useful when the same notes follow one another. As a rule, "1" is written on the attack bit of all words. However, when long notes other than those listed in Table 4.11.6 are desired, they can be implemented by linking several words of the same interval to a continuous address and at the same time setting the attack bit to "0".
  On the other hand, when envelope function is available, setting this bit to "1" will cause the capacitor for the envelope function which is externally installed to be recharged when the playing starts and increase the sound pressure of the playing. Moreover, when this bit is set to "0", since the capacitor will be continuously discharged without being recharged, the sound pressure of the playing will continue to diminish. The principle of the envelope function is explained in details in the next section.

- **End data (D0)**
  This is 1-bit data which indicate the end of a set of played melody. If this bit were written with "1", when the word is played, end-of-melody signal will be generated at the end-of-melody signal generator and will then be input to the melody interrupt generator and the controller. This signal is received at the melody interrupt generator which issues interrupt request to the CPU and generates interrupt flag. Moreover, the controller stops the playing when the melody ON/OFF control register is set to "0" when the signal is received and either repeats the same melody or continuously plays new melodies when it is set to "1". By dividing the 80-word melody ROM with end-of-melody data, any number of melodies may be written as long as it is within the capacity. Also, a melody which will be repeatedly used need be written only once, i.e., there is no need to write the melody for as many number of times you wish to repeat it. Repeated playing can be easily accomplished by merely specifying the playing start address repeatedly through the software. Control of playing is explained in details in "Control of playing".

Playing of silent note

Silent note may be played by writing "0FH" on the melody ROM interval data. The length of the silent note is the same as the length of the note written on the same word. For details, refer to "Melody data".

Envelope function

The S1C62N81 Series may be added with envelope function for melody playing by mask option. The IC internal circuit when the envelope function is valid and the external circuit required is shown in Figure 4.11.14. The IC internal setting is done by mask option and the following need to be externally installed:

- piezo buzzer sounding body;
- booster coil for raising the sound pressure of the playing;
- PNP bipolar transistor to drive the sounding body (piezo buzzer);
- capacitor for implementing smooth sound pressure attenuation; and
- resistor for controlling the power current discharge of the capacitor.

The output waveform when envelope function is shown in Figure 4.11.15. The attack signal indicated in the diagram will go high ("H" level) when the playing of the word starts if the attack data written on the melody ROM were "1". The pulse width is approximately 12 ms. The ATK (attack) signal recharges the externally installed capacitor and the R12 terminal output level will be recharged up to the power voltage as shown in Figure 4.11.15. This will result in the MO terminal output amplitude becoming the power voltage since they (R12 and MO terminals) are wired together inside the IC as shown in Figure 4.11.14. The sound pressure of the melody played then will be maximum. Henceforth, because the capacitor connected to the R12 terminal is discharged as the base current of the externally installed transistor as time passes, the base current will drop and the playing sound pressure will attenuate with the passing of time. The MO terminal output waveform is shown in Figure 4.11.15. The MO terminal output amplitude will decrease with capacitor discharge. This is the principle of the envelope function.

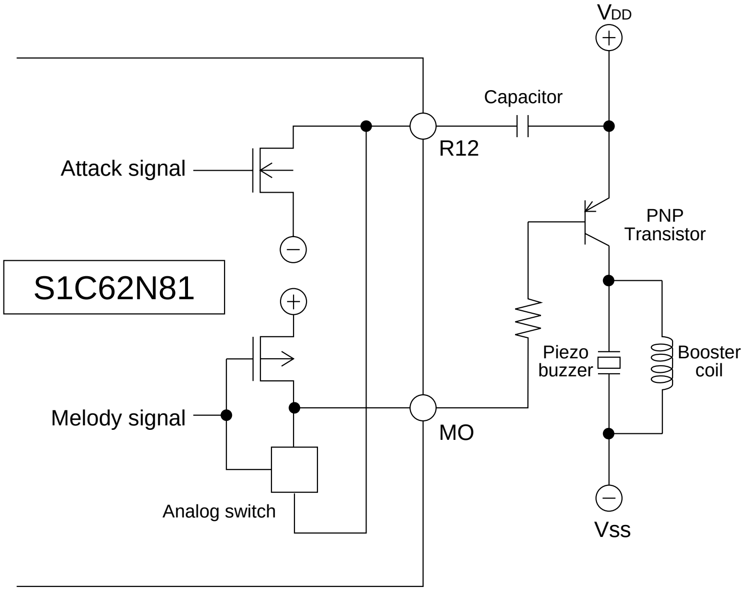


Fig. 4.11.14
Configuration of the
envelope function

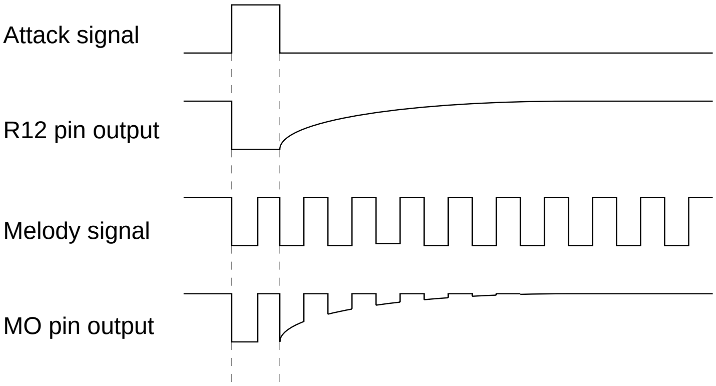


Fig. 4.11.15
Envelope output waveform

Playing tempo

In the S1C62N81 Series, 2 types of melody playing tempo may be selected from among 16 types by mask option. Tempos which may be selected are shown in Table 4.11.7 (see also "Tempo generator"). The proper use of the 2 types of tempo selected is specified through the software. The 2 types of tempo which may selected are: the tempo to be played when "0" is written on the TEMPC register of the controller and the tempo to be played when "1" is written on the said register.

Table 4.11.7
Tempos available
for selection

TS3	TS2	TS1	TS0	♩ ≒
0	0	0	0	30
0	0	0	1	32
0	0	1	0	34.3
0	0	1	1	36.9
0	1	0	0	40
0	1	0	1	43.6
0	1	1	0	48
0	1	1	1	53.3
1	0	0	0	60
1	0	0	1	68.6
1	0	1	0	80
1	0	1	1	96
1	1	0	0	120
1	1	0	1	160
1	1	1	0	240
1	1	1	1	480

Note *Changing the 2 types of tempo selected by mask option is not done on the spot when data is written on the TEMPC register but rather, the tempo is changed when a new melody is played after the data has been written, i.e., the tempo cannot be changed in the middle of a melody playing.*

Furthermore, 4 types of playing speed may be selected in the S1C62N81 Series. The selection can be done through the software and control is performed by writing data on CLKC0 and CLKC1 registers of the controller. The data written on the registers and the corresponding playing speed are shown in Table 4.11.8. By writing "0" on CLKC0 and CLKC1, normal speed tempo (i.e., tempo selected by mask option) may be played. playing at 8 times, 16 times and 32 times of the normal speed is useful for producing sound effects for games and animal sounds.

Table 4.11.8
Playing speed

CLKC1	CLKC0	Playing Speed
0	0	Normal
0	1	8 times
1	0	16 times
1	1	32 times

Note *Changing the playing speed is instantly accomplished by writing data on CLKC0 and CLKC1 registers. When speed need not be changed in the middle of a melody, write the playing speed data upon completion of a melody playing, i.e., during rest.*

Playing mode

The S1C62N81 Series have 3 modes for melody playing: one shot mode, level hold mode and retrigger mode. The control of these modes is done through operation of the MELC register of the controller.

(a) One shot mode

In this mode, only one specified melody is played; playing automatically stops when the melody ends. Control procedures are as follows:

(1) Set the melody ROM address (start address) of the desired melody in the address register (MAD0–MAD6).

(2) Immediately after writing "1" (before the melody playing ends), write "0" on the MELC register.

The above operation will allow only one melody to be played. Melody playing is started from the address written on the address register, by writing "1" on the MELC register. When playing of the last word of a melody (end-of-melody data is "1") ends, end-of-melody signal is generated and interrupt request to the CPU and interrupt flag are generated in the melody interrupt generator. At this point, since "0" has previously been written on the MELC register with the above operation (2), signal to halt playing is generated in the controller and hence, playing will stop.

The relationship between MELC register value and playing output is shown in Figure 4.11.16.

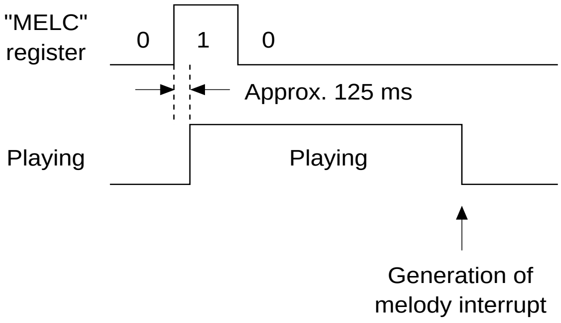


Fig. 4.11.16
One shot mode

Note Bear in mind that playing will start approximately 125 ms (in case of normal speed) after writing "1" on the MELC register.

(b) Level hold mode

Repetition of the same melody or continuous playing of different melodies is possible in this mode. The operating procedure are as follows:

(1) Set the melody ROM address (start address) of the desired melody in the address register (MAD0–MAD6).

(2) Write "1" on the MELC register.

(3) Immediately after procedure (2) above (before the melody being played ends), write the start address of the second melody on the address register (MAD0–MAD6). When repeating the same melody, there is no need to write anew on the address register.

(4) Since melody interrupt will be generated when the first melody ends, write the address for the third melody on the address register (MAD0–MAD6) with the interrupt routine. This operation must be completed before the second melody ends. When the same melody is to be repeatedly played, there is no need for this operation.

The optional melody in the melody ROM may continuously be played by repeating the above steps.

(5) To stop playing, write "0" on the MELC register while the last melody is being played. This will cause the playing to be automatically stopped when playing of the last melody is completed.

The relationship between MELC register value and playing output is shown in Figure 4.11.17.

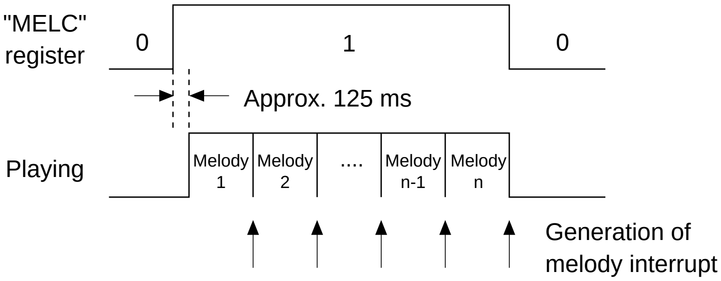


Fig. 4.11.17
Level hold mode

(c) Retrigger mode

This playing mode is for modifying or stopping the melody forcedly in the middle of playing. Its operating procedure is as follows:

(1) In the middle of a melody playing, write the melody ROM address of the next melody to be played on the address register (MAD0–MAD6).

(2) Change the MELC register setting from "0" to "1". At this point, the played melody will be forcedly changed.

(3) After this operation, the 3 types of playing mode may be selected freely again.

To stop a melody in the middle of its playing is also implemented by employing this mode. The operation is as follows:

(1) In the middle of a melody playing, set the melody ROM address written with silent notes on the address register (MAD0–MAD6).

(2) Change the MELC register setting from "0" to "1" and then to "0" again.

With the above operation, the melody being played will be forced to change into silent note playing; as soon as the playing of the silent notes is completed, the playing will automatically stop. In the above operation (2), writing operation for the last "0" must be done before the playing of silent notes ends.

The relationship between MELC register value and playing output is shown in Figure 4.11.18.

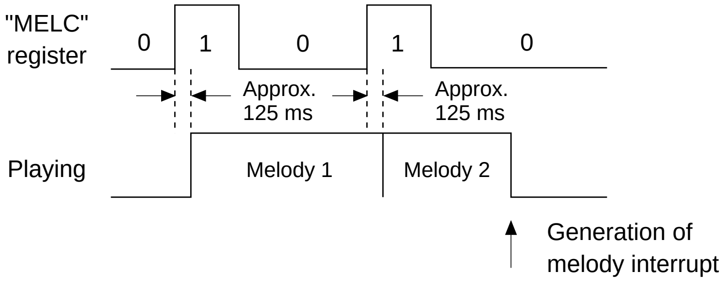


Fig. 4.11.18
Retrigger mode

Note *Bear in mind that when melody playing is forcedly modified with the above operations, playing of the modified melody will start approximately 125 ms (in case of normal speed) after "1" has been written on the MELC register.*

Control of the melody generator

Operation of registers for melody control is explained in this section.

Table 4.11.9 Control bits of melody generator

Address	Register								Comment
	D3	D2	D1	D0	Name	SR	1	0	
0E7H	0	0	0	EIMEL	0				
	R			R/W	0				
					0				
					EIMEL	0	Enable	Mask	Interrupt mask register (melody)
0ECH	0	0	0	IMEL	0				
	R				0				
					0				
					IMEL	0	Yes	No	Interrupt factor flag (melody)
0F0H	MAD3	MAD2	MAD1	MAD0	MAD3	0	High	Low	Melody ROM address (AD3)
	R/W				MAD2	0	High	Low	Melody ROM address (AD2)
					MAD1	0	High	Low	Melody ROM address (AD1)
					MAD0	0	High	Low	Melody ROM address (AD0, LSB)
0F1H	0	MAD6	MAD5	MAD4	0				
	R	R/W			MAD6	0	High	Low	Melody ROM address (AD6, MSB)
					MAD5	0	High	Low	Melody ROM address (AD5)
					MAD4	0	High	Low	Melody ROM address (AD4)
0F2H	CLKC1	CLKC0	TEMPC	MELC	CLKC1	0	High	Low	CLKC1(0)&CLKC0(0) : melody speed × 1 CLKC1(0)&CLKC0(1) : melody speed × 8
	R/W				CLKC0	0	High	Low	CLKC1(1)&CLKC0(0) : melody speed × 16 CLKC1(1)&CLKC0(1) : melody speed × 32
					TEMPC	0	High	Low	Tempo change control
					MELC	0	ON	OFF	Melody control ON/OFF

– MELC: Melody ON/OFF Control Register (F2H D0)

By operating this register, control of the melody playing ON/OFF and the 3 types playing modes—one shot mode, level hold mode and retrigger mode—can be performed.

When 1 is written:	Playing starts
When 0 is written:	Playing stops
Reading:	Valid

– TEMPC: Tempo Control Register (F2H D1)

By operating this register, 1 type of tempo may be selected from the 2 types previously selected by mask option.

When 1 is written:	Selects the tempo of TEMPC1 selected by mask option
When 0 is written:	Selects the tempo of TEMPC0 selected by mask option
Reading:	Valid

Note *Changing the tempo through this register is not possible in the middle of a melody playing even if this register is operated while a melody is being played. Change of melody will synchronize with the playing of a new melody.*

– CLKC0: Playing Speed Control Register (F2H D2)
CLKC1: Playing Speed Control Register (F2H D3)

By operating these registers, playing speed of a melody may be changed. The combination of CLKC0 and CLKC1 register values and playing speed are shown in Table 4.11.10.

When 1 is written:	1
When 0 is written:	0
Reading:	Valid

Table 4.11.10
Playing speed

CLKC1	CLKC0	Playing Speed
0	0	Normal
0	1	8 times
1	0	16 times
1	1	32 times

Note *Playing speeds are changed the moment these registers are operated. Take caution when operating these registers in the middle of a melody playing.*

– MAD0–MAD6: Address Registers (F0H D0–D3 and F1H D0–D2)

These registers are used to set the melody playing start. By operating the "MELC" register, when playing of a new melody starts, the addresses set in these registers are read by the melody ROM address counter and become the melody start addresses.

When 1 is written:	1
When 0 is written:	0
Reading:	Valid

Note *When these registers are written with "50H" –"7FH", since these addresses do not exist in the melody ROM, the hardware will cause silent notes equivalent to 32 notes to be played and so, caution must be observed.*

– EIMEL: Melody Interrupt Mask Register (E7H D0)

By operating this register, melody interrupt can be masked.

When 1 is written:	Interrupt is valid
When 0 is written:	Interrupt is invalid
Reading:	Valid

Note Be sure to operate this register in the "DI (interrupt not allowed)" state. Otherwise, it may result in misoperation.

– IMEL: Melody Interrupt Factor Flag (ECH D0)

The moment the melody playing (i.e., playing of the address the end-of-melody data in the melody ROM of which is "1") ends, a flag is set on this register. Due to this, the end of a melody playing can be known by reading out this register. This register is also reset by the hardware after the readout.

When 1 is read:	Interrupt generation; 0 after readout
When 0 is read:	Interrupt is not generated
Writing:	Invalid

4.12 Interrupt and HALT

The S1C62N81 Series provide the following interrupt settings, each of which is maskable.

External interrupt:	Input interrupt (two)
Internal interrupt:	Timer interrupt (one)
	Stopwatch interrupt (one)
	Melody interrupt (one)

To enable interrupts, the interrupt flag must be set to 1 (EI) and the necessary related interrupt mask registers must be set to 1 (enable). When an interrupt occurs, the interrupt flag is automatically reset to 0 (DI) and interrupts after that are inhibited.

When a HALT instruction is input, the CPU operating clock stops and the CPU enters the halt state. The CPU is reactivated from the halt state when an interrupt request occurs.

Figure 4.12.1 shows the configuration of the interrupt circuit.

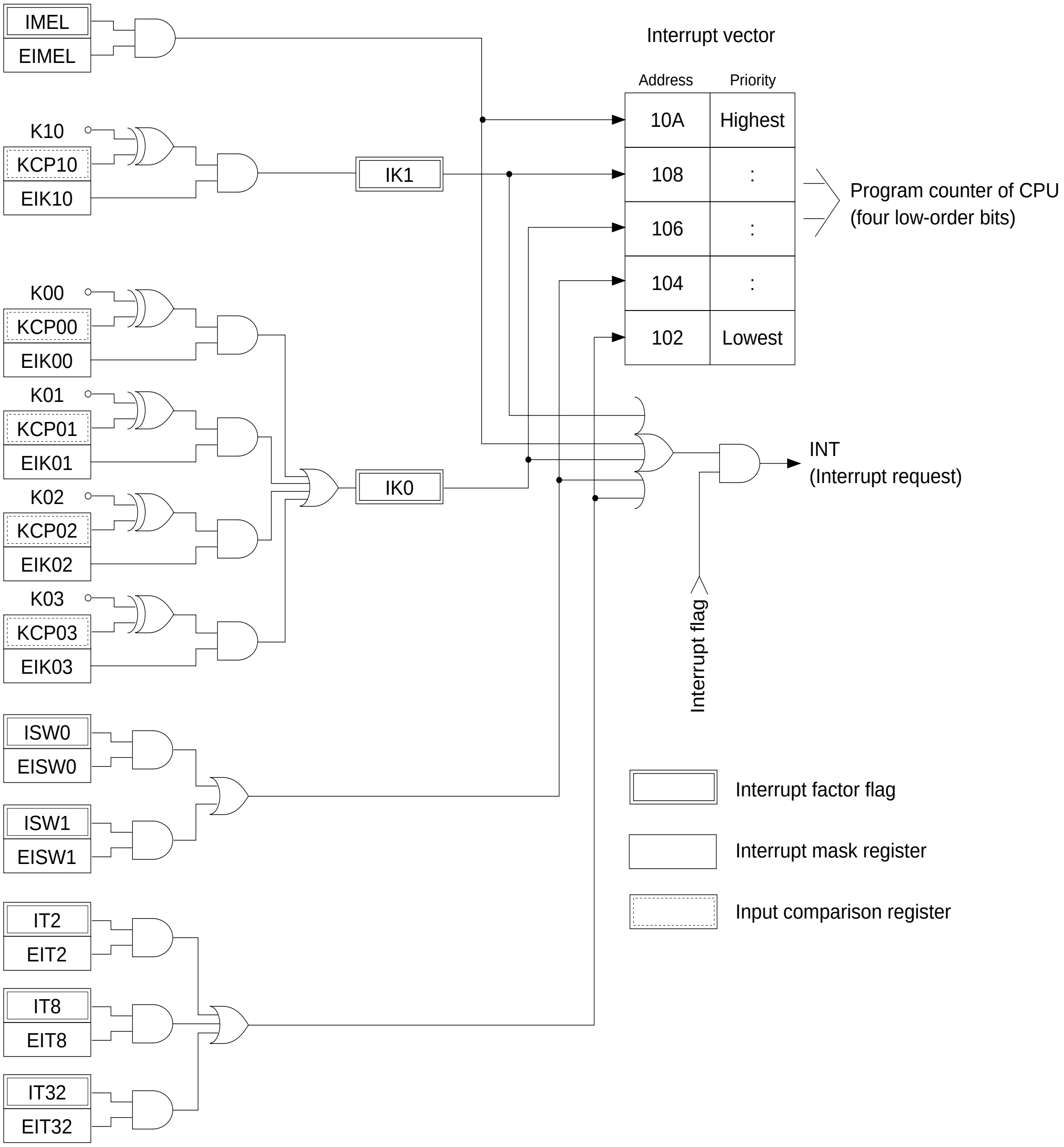


Fig. 4.12.1 Configuration of interrupt circuit

Interrupt factors

Table 4.12.1 shows the factors that generate interrupt requests.

The interrupt factor flags are set to 1 depending on the corresponding interrupt factors.

The CPU is interrupted when the following two conditions occur and an interrupt factor flag is set to 1.

- The corresponding mask register is 1 (enabled)
- The interrupt flag is 1 (EI)

The interrupt factor flag is a read-only register, but can be reset to 0 when the register data is read.
After an initial reset, the interrupt factor flags are reset to 0.

Note *Read the interrupt factor flags only in the DI status (interrupt flag = 0). A malfunction could result from a read during the EI status (interrupt flag = 1).*

Table 4.12.1
Interrupt factors

Interrupt Factor	Interrupt Factor Flag	
Clock timer 2 Hz falling edge	IT2	(0EFH D2)
Clock timer 8 Hz falling edge	IT8	(0EFH D1)
Clock timer 32 Hz falling edge	IT32	(0EFH D0)
Stopwatch timer 1 Hz falling edge	ISW1	(0EEH D1)
Stopwatch timer 10 Hz falling edge	ISW0	(0EEH D0)
Input data (K00–K03) Rising or falling edge	IK0	(0EDH D0)
Input data (K10) Rising or falling edge	IK1	(0EDH D1)
Melody generator End of melody	IMEL	(0ECH D0)

Specific masks and factor flags for interrupt

The interrupt factor flags can be masked by the corresponding interrupt mask registers. The interrupt mask registers are read/write registers. They are enabled (interrupt enabled) when 1 is written to them, and masked (interrupt disabled) when 0 is written to them. After an initial reset, the interrupt mask register is set to 0.

Table 4.12.2 shows the correspondence between interrupt mask registers and interrupt factor flags.

Table 4.12.2
Interrupt mask registers and interrupt factor flags

Interrupt Mask Register	Interrupt Factor Flag
EIT2 (0EBH D2)	IT2 (0EFH D2)
EIT8 (0EBH D1)	IT8 (0EFH D1)
EIT32 (0EBH D0)	IT32 (0EFH D0)
EISW1 (0EAH D1)	ISW1 (0EEH D1)
EISW0 (0EAH D0)	ISW0 (0EEH D0)
EIK03 * (0E8H D3)	IK0 (0EDH D0)
EIK02 * (0E8H D2)	
EIK01 * (0E8H D1)	
EIK00 * (0E8H D0)	
EIK10 * (0E9H D0)	IK1 (0EDH D1)
EIMEL (0E7H D0)	IMEL (0ECH D0)

* There is an interrupt mask register for each input port pin.

Note *Writing to the interrupt mask registers should be done only in the DI status (interrupt flag = 0). Otherwise it causes malfunction.*

Interrupt vectors and priorities

When an interrupt request is input to the CPU, the CPU begins interrupt processing. After the program being executed is suspended, interrupt processing is executed in the following order:

① The address data (value of the program counter) of the program step to be executed next is saved on the stack (RAM).

② The interrupt request causes the value of the interrupt vector (page 1, 02H–0BH) to be loaded into the program counter.

③ The program at the specified address is executed (execution of interrupt processing routine).

Table 4.12.3 shows the correspondence of interrupt vectors and priorities.

Note *The processing in steps 1 and 2, above, takes 12 cycles of the CPU system clock.*

Table 4.12.3
Interrupt vectors
and priorities

Vector	Priority	Interrupt Request
10AH	1	Melody interrupt
108H	2	Input (K10) interrupt
106H	3	Input (K00–K03) interrupt
104H	4	Stopwatch timer interrupt
102H	5	Clock timer interrupt

Note *When multiple interrupts occur simultaneously, the interrupt vectors with higher priority will be executed.*

Control of interrupt

Tables 4.12.4 (a)–(c) shows the interrupt control bits and their addresses.

Table 4.12.4 (a) Interrupt control bits (1)

Address	Register								Comment
	D3	D2	D1	D0	Name	SR	1	0	
0E5H	KCP03	KCP02	KCP01	KCP00	KCP03	0	Falling	Rising	Input comparison register (K03)
	R/W				KCP02	0	Falling	Rising	Input comparison register (K02)
					KCP01	0	Falling	Rising	Input comparison register (K01)
					KCP00	0	Falling	Rising	Input comparison register (K00)
0E6H	0	0	0	KCP10	0				
	R			R/W	0				
					0				
					KCP10	0	Falling	Rising	Input comparison register (K10)
0E7H	0	0	0	EIMEL	0				
	R			R/W	0				
					0				
					EIMEL	0	Enable	Mask	Interrupt mask register (melody)
0E8H	EIK03	EIK02	EIK01	EIK00	EIK03	0	Enable	Mask	Interrupt mask register (K03)
	R/W				EIK02	0	Enable	Mask	Interrupt mask register (K02)
					EIK01	0	Enable	Mask	Interrupt mask register (K01)
					EIK00	0	Enable	Mask	Interrupt mask register (K00)

Table 4.12.4 (b) Interrupt control bits (2)

Address	Register								Comment
	D3	D2	D1	D0	Name	SR	1	0	
0E9H	0	0	0	EIK10	0				
	R			R/W	0				
					0				
					EIK10	0	Enable	Mask	Interrupt mask register (K10)
0EAH	0	0	EISW1	EISW0	0				
	R		R/W		0				
					EISW1	0	Enable	Mask	Interrupt mask register (stopwatch 1 Hz)
					EISW0	0	Enable	Mask	Interrupt mask register (stopwatch 10 Hz)
0EBH	0	EIT2	EIT8	EIT32	0				
	R	R/W			EIT2	0	Enable	Mask	Interrupt mask register (clock timer 2 Hz)
					EIT8	0	Enable	Mask	Interrupt mask register (clock timer 8 Hz)
					EIT32	0	Enable	Mask	Interrupt mask register (clock timer 32 Hz)
0ECH	0	0	0	IMEL	0				
	R				0				
					0				
					IMEL	0	Yes	No	Interrupt factor flag (melody)

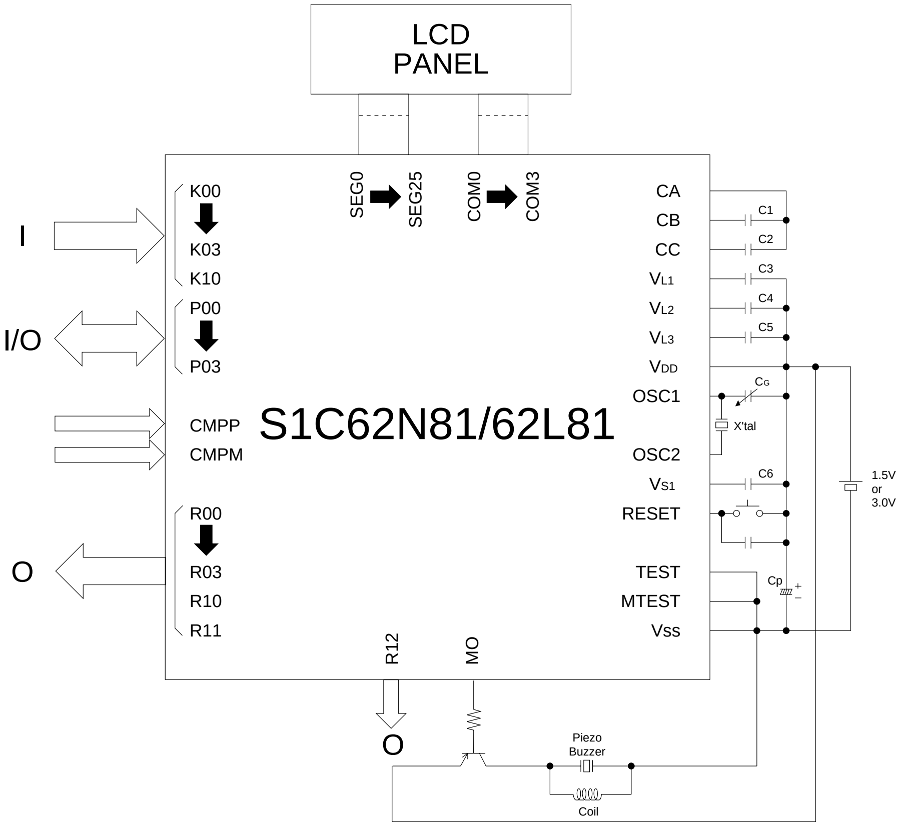
Table 4.12.4 (c) Interrupt control bits (3)

Address	Register								Comment
	D3	D2	D1	D0	Name	SR	1	0	
0EDH	0	0	IK1	IK0	0				
	R				0				
					IK1	0	Yes	No	Interrupt factor flag (K10)
					IK0	0	Yes	No	Interrupt factor flag (K00–K03)
0EEH	0	0	ISW1	ISW0	0				
	R				0				
					ISW1	0	Yes	No	Interrupt factor flag (stopwatch 1 Hz)
					ISW0	0	Yes	No	Interrupt factor flag (stopwatch 10 Hz)
0EFH	0	IT2	IT8	IT32	0				
	R				IT2	0	Yes	No	Interrupt factor flag (clock timer 2 Hz)
					IT8	0	Yes	No	Interrupt factor flag (clock timer 8 Hz)
					IT32	0	Yes	No	Interrupt factor flag (clock timer 32 Hz)

EIT32, EIT8, EIT2	Interrupt mask registers (0EBH D0–D2)
IT32, IT8, IT2	Interrupt factor flags (0EFH D0–D2)
	See 4.7, "Clock Timer".
EISW0, EISW1	Interrupt mask registers (0EAH D0–D1)
ISW0, ISW1	Interrupt factor flags (0EEH D0–D1)
	See 4.8, "Stopwatch Timer".
KCP00–KCP03	Input comparison registers (0E5H)
EIK00–EIK03	Interrupt mask registers (0E8H)
IK0	Interrupt factor flag (0EDH D0)
	See 4.3, "Input Ports".
KCP10	Input comparison register (0E6H D0)
EIK10	Interrupt mask register (0E9H D0)
IK1	Interrupt factor flag (0EDH D1)
	See 4.3, "Input Ports".
EIMEL	Interrupt mask register (0E7H D0)
IMEL	Interrupt factor flag (0ECH D0)
	See 4.11, "Melody Generator".

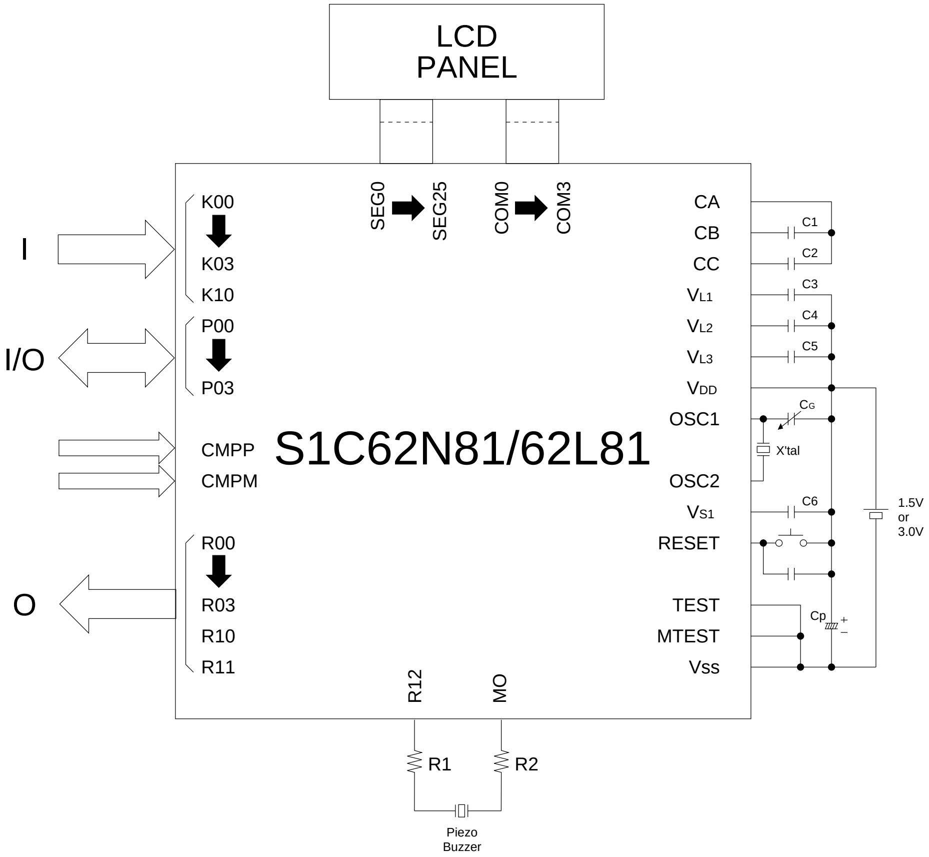
CHAPTER 5 BASIC EXTERNAL WIRING DIAGRAM

(1) Piezo Buzzer Single Terminal Driving



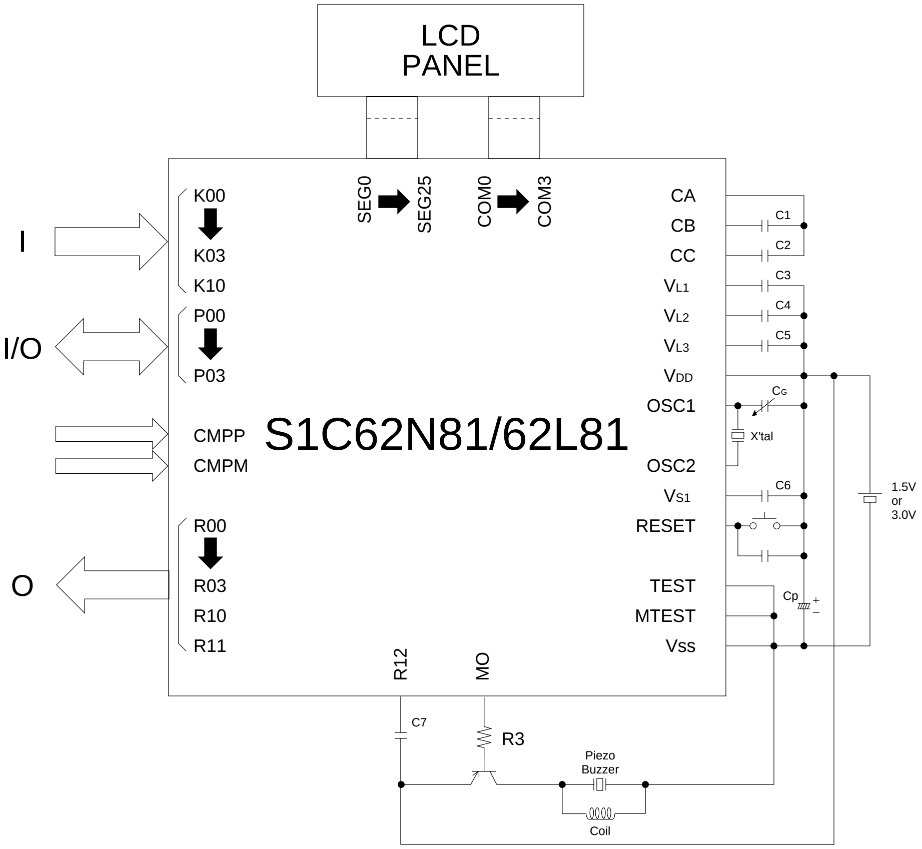
X'tal	Crystal oscillator	32.768kHz CI(MAX)=35kΩ
C_G	Trimmer capacitor	5–25pF
C1–C6	Capacitor	0.1 µF
Cp	Capacitor	3.3 µF

(2) Piezo Buzzer Direct Driving



X'tal	Crystal oscillator	32.768kHz CI(MAX)=35kΩ
CG	Trimmer capacitor	5–25pF
C1–C6	Capacitor	0.1µF
Cp	Capacitor	3.3µF
R1, R2	Protection resistance	100Ω

(3) Envelope Driving



X'tal	Crystal oscillator	32.768kHz CI(MAX)=35kΩ
C_G	Trimmer capacitor	5–25pF
C1–C6	Capacitor	0.1 µF
C7	Capacitor	1 µF–10 µF
Cp	Capacitor	3.3 µF
R3	Resistor	1kΩ or more

CHAPTER 6 ELECTRICAL CHARACTERISTICS

6.1 Absolute Maximum Rating

S1C62N81/62A81

(V_{DD} =0V)

Item	Symbol	Rated Value	Unit
Power voltage	Vss	-5.0 to 0.5	V
Input voltage (1)	V_I	Vss-0.3 to 0.5	V
Input voltage (2)	V_{IOSC}	Vss-0.3 to 0.5	V
Permissible total output current *1	ΣIvss	10	mA
Operating temperature	Topr	-20 to 70	°C
Storage temperature	Tstg	-65 to 150	°C
Soldering temperature / Time	Tsol	260°C, 10sec (lead section)	–
Allowable dissipation *2	P_D	250	mW

*1 The permissible total output current is the sum total of the current (average current) that simultaneously flows from the output pins (or is drawn in).

*2 In case of 64-pin plastic package.

S1C62L81/62B81

(V_{DD} =0V)

Item	Symbol	Rated Value	Unit
Power voltage	Vss	-5.0 to 0.5	V
Input voltage (1)	V_I	Vss-0.3 to 0.5	V
Input voltage (2)	V_{IOSC}	Vss-0.3 to 0.5	V
Permissible total output current *1	ΣIvss	10	mA
Operating temperature	Topr	-20 to 70	°C
Storage temperature	Tstg	-65 to 150	°C
Soldering temperature / Time	Tsol	260°C, 10sec (lead section)	–
Allowable dissipation *2	P_D	250	mW

*1 The permissible total output current is the sum total of the current (average current) that simultaneously flows from the output pins (or is drawn in).

*2 In case of 64-pin plastic package.

6.2 Recommended Operating Conditions

S1C62N81/62A81

(Ta=-20 to 70°C)

Item	Symbol	Condition	Min	Typ	Max	Unit
Power voltage	Vss	V_{DD}=0V	-3.5	-3.0	-1.8	V
Oscillation frequency	fosc			32.768		kHz

S1C62L81/62B81

(Ta=-20 to 70°C)

Item	Symbol	Condition	Min	Typ	Max	Unit
Power voltage	Vss	V_{DD}=0V	-3.5	-1.5	-1.1	V
		V_{DD}=0V, With software correspondence [*1]	-3.5	-1.5	-0.9 [*2]	V
		V_{DD}=0V, When analog comparator is used	-3.5	-1.5	-1.3	V
Oscillation frequency	fosc			32.768		kHz

*1 When switching to the heavy load protection mode. The BLD circuit and analog voltage comparator are turned OFF.
(For details, refer to Section 4.9).

*2 The voltage which can be displayed on the LCD panel will differ according to the characteristics of the LCD panel.

6.3 DC Characteristics

S1C62N81/62A81

Unless otherwise specified
V_{DD}=0 V, V_{SS}=-3.0 V, fosc=32.768 kHz, Ta=25°C, V_{S1},
V_{L1}, V_{L2} and V_{L3} are internal voltages, and
C1=C2=C3=C4=C5=C6=0.1 µF

Item	Symbol	Condition		Min	Typ	Max	Unit
High level input voltage (1)	V_{IH1}		K00–K03, K10, P00–P03, MTEST	0.2•Vss		0	V
High level input voltage (2)	V_{IH2}		RESET, TEST	0.10•Vss		0	V
Low level input voltage (1)	V_{IL1}		K00–K03, K10, P00–P03, MTEST	Vss		0.8•Vss	V
Low level input voltage (2)	V_{IL2}		RESET, TEST	Vss		0.90•Vss	V
High level input current (1)	I_{IH1}	V_{IH} =0V Without pull down resistor	K00–K03, K10, P00–P03 CMPP, CMPM	0		0.5	µA
High level input current (2)	I_{IH2}	V_{IH} =0V With pull down resistor	K00–K03, K10	5		16	µA
High level input current (3)	I_{IH3}	V_{IH} =0V With pull down resistor	P00–P03 RESET, TEST, MTEST	30		100	µA
Low level input current	I_{IL}	V_{IL} =Vss	K00–K03, K10 P00–P03 CMPP, CMPM RESET, TEST, MTEST	-0.5		0	µA
High level output current (1)	I_{OH1}	V_{OH1}=0.1•Vss	R11			-1.0	mA
High level output current (2)	I_{OH2}	V_{OH2}=0.1•Vss	R00–R03, R10 P00–P03			-1.0	mA
High level output current (3)	I_{OH3}	V_{OH3}=0.1•Vss	MO, R12			-2.0	mA
Low level output current (1)	I_{OL1}	V_{OL1} =0.9•Vss	R11	3.0			mA
Low level output current (2)	I_{OL2}	V_{OL2} =0.9•Vss	R00–R03, R10 P00–P03	3.0			mA
Low level output current (3)	I_{OL3}	V_{OL3} =0.9•Vss	MO, R12	4.5			mA
Common output current	I_{OH4}	V_{OH4}=-0.05V	COM0–COM3			-3	µA
	I_{OL4}	V_{OL4} =V_{L3} +0.05V		3			µA
Segment output current (during LCD output)	I_{OH5}	V_{OH5}=-0.05V	SEG0–SEG25			-3	µA
	I_{OL5}	V_{OL5} =V_{L3} +0.05V		3			µA
Segment output current (during DC output)	I_{OH6}	V_{OH6}=0.1•Vss	SEG0–SEG25			-300	µA
	I_{OL6}	V_{OL6} =0.9•Vss		300			µA

S1C62L81/62B81

Unless otherwise specified
V_{DD}=0 V, V_{SS}=-1.5 V, fosc=32.768 kHz, Ta=25°C, V_{S1}, V_{L1}, V_{L2} and V_{L3} are internal voltages, and C1=C2=C3=C4=C5=C6=0.1 µF

Item	Symbol	Condition		Min	Typ	Max	Unit
High level input voltage (1)	V_{IH1}		K00–K03, K10, P00–P03, MTEST	0.2•Vss		0	V
High level input voltage (2)	V_{IH2}		RESET, TEST	0.10•Vss		0	V
Low level input voltage (1)	V_{IL1}		K00–K03, K10, P00–P03, MTEST	Vss		0.8•Vss	V
Low level input voltage (2)	V_{IL2}		RESET, TEST	Vss		0.90•Vss	V
High level input current (1)	I_{IH1}	V_{IH} =0V Without pull down resistor	K00–K03, K10, P00–P03 CMPP, CMPM	0		0.5	µA
High level input current (2)	I_{IH2}	V_{IH} =0V With pull down resistor	K00–K03, K10	2.0		10	µA
High level input current (3)	I_{IH3}	V_{IH} =0V With pull down resistor	P00–P03 RESET, TEST, MTEST	9.0		60	µA
Low level input current	I_{IL}	V_{IL} =Vss	K00–K03, K10 P00–P03 CMPP, CMPM RESET, TEST, MTEST	-0.5		0	µA
High level output current (1)	I_{OH1}	V_{OH1}=0.1•Vss	R11			-450	µA
High level output current (2)	I_{OH2}	V_{OH2}=0.1•Vss	R00–R03, R10 P00–P03			-200	µA
High level output current (3)	I_{OH3}	V_{OH3}=0.1•Vss	MO, R12			-0.8	mA
High level output current (4)	I_{OH4}	V_{OH4}=0.1•Vss When envelope is used	MO			-0.4	mA
Low level output current (1)	I_{OL1}	V_{OL1} =0.9•Vss	R11	1300			µA
Low level output current (2)	I_{OL2}	V_{OL2} =0.9•Vss	R00–R03, R10 P00–P03	700			µA
Low level output current (3)	I_{OL3}	V_{OL3} =0.9•Vss	MO, R12	1.5			mA
Common output current	I_{OH5}	V_{OH5}=-0.05V	COM0–COM3			-3	µA
	I_{OL5}	V_{OL5} =V_{L3} +0.05V		3			µA
Segment output current (during LCD output)	I_{OH6}	V_{OH6}=-0.05V	SEG0–SEG25			-3	µA
	I_{OL6}	V_{OL6} =V_{L3} +0.05V		3			µA
Segment output current (during DC output)	I_{OH7}	V_{OH7}=0.1•Vss	SEG0–SEG25			-100	µA
	I_{OL7}	V_{OL7} =0.9•Vss		130			µA

6.4 Analog Circuit Characteristics and Power Current Consumption

S1C62N81 (Normal Operating Mode)

Unless otherwise specified

V_{DD}=0 V, V_{SS}=-3.0 V, fosc=32.768 kHz, Ta=25°C, C_G=25 pF, V_{S1}, V_{L1}, V_{L2} and V_{L3} are internal voltages, and C1=C2=C3=C4=C5=C6=0.1 µF

Item	Symbol	Condition		Min	Typ	Max	Unit
Internal voltage	V_{L1}	Connect 1MΩ load resistor between V_{DD} and V_{L1} (without panel load)		-1.15	-1.05	-0.95	V
	V_{L2}	Connect 1MΩ load resistor between V_{DD} and V_{L2} (without panel load)		2•V_{L1} -0.1		2•V_{L1} × 0.9	V
	V_{L3}	Connect 1MΩ load resistor between V_{DD} and V_{L3} (without panel load)		3•V_{L1} -0.1		3•V_{L1} × 0.9	V
BLD voltage	V_{BLD}			-2.55	-2.40	-2.25	V
BLD circuit response time	t_{BLD}					100	µs
Analog comparator input voltage	V_{IP} V_{IM}	Non-inverted input (CMPP) Inverted input (CMPM)		Vss+0.3		V_{DD} -0.9	V
Analog comparator offset voltage	V_{OF}					10	mV
Analog comparator response time	t_{CMP}	V_{IP} =-1.5V V_{IM}=V_{IP}±15mV				3	ms
Power current consumption	I_{OP}	During HALT	Without panel load		1.0	2.5	µA
		During execution [*1]			2.5	5.0	µA

*1 The BLD circuit and analog voltage comparator are turned OFF.

S1C62N81 (Heavy Load Protection Mode)

Unless otherwise specified

V_{DD}=0 V, V_{SS}=-3.0 V, fosc=32.768 kHz, Ta=25°C,
C_G=25 pF V_{S1}, V_{L1}, V_{L2} and V_{L3} are internal voltages, and
C1=C2=C3=C4=C5=C6=0.1 µF

Item	Symbol	Condition	Min	Typ	Max	Unit
Internal voltage	V_{L1}	Connect 1MΩ load resistor between V_{DD} and V_{L1} (without panel load)	-1.15	-1.05	-0.95	V
	V_{L2}	Connect 1MΩ load resistor between V_{DD} and V_{L2} (without panel load)	2•V_{L1} -0.1		2•V_{L1} ×0.85	V
	V_{L3}	Connect 1MΩ load resistor between V_{DD} and V_{L3} (without panel load)	3•V_{L1} -0.1		3•V_{L1} ×0.85	V
BLD voltage	V_{BLD}		-2.55	-2.40	-2.25	V
BLD circuit response time	t_{BLD}				100	µs
Analog comparator input voltage	V_{IP} V_{IM}	Non-inverted input (CMPP) Inverted input (CMPM)	Vss+0.3		V_{DD} -0.9	V
Analog comparator offset voltage	V_{OF}				10	mV
Analog comparator response time	t_{CMP}	V_{IP} =-1.5V V_{IM} =V_{IP} ±15mV			3	ms
Power current consumption	I_{OP}	During HALT Without panel load		2.0	5.5	µA
		During execution [*1]		5.5	10.0	µA

*1 The BLD circuit and analog voltage comparator are turned OFF.

S1C62L81 (Normal Operating Mode)

Unless otherwise specified
V_{DD}=0 V, V_{SS}=-1.5 V, fosc=32.768 kHz, Ta=25°C, C_G=25 pF V_{S1}, V_{L1}, V_{L2} and V_{L3} are internal voltages, and C1=C2=C3=C4=C5=C6=0.1 µF

Item	Symbol	Condition		Min	Typ	Max	Unit
Internal voltage	V_{L1}	Connect 1MΩ load resistor between V_{DD} and V_{L1} (without panel load)		-1.15	-1.05	-0.95	V
	V_{L2}	Connect 1MΩ load resistor between V_{DD} and V_{L2} (without panel load)		2•V_{L1} -0.1		2•V_{L1} × 0.9	V
	V_{L3}	Connect 1MΩ load resistor between V_{DD} and V_{L3} (without panel load)		3•V_{L1} -0.1		3•V_{L1} × 0.9	V
BLD voltage	V_{BLD}			-1.30	-1.20	-1.10	V
BLD circuit response time	t_{BLD}					100	µs
Analog comparator input voltage	V_{IP} V_{IM}	Non-inverted input (CMPP) Inverted input (CMPM)		Vss+0.3		V_{DD} -0.9	V
Analog comparator offset voltage	V_{OF}					20	mV
Analog comparator response time	t_{CMP}	V_{IP} =-1.1V V_{IM} =V_{IP}±30mV				3	ms
Power current consumption	I_{OP}	During HALT	Without panel load		1.0	2.5	µA
		During execution [*1]			2.5	5.0	µA

*1 The BLD circuit and analog voltage comparator are turned OFF.

S1C62L81 (Heavy Load Protection Mode)

Unless otherwise specified

V_{DD}=0 V, V_{SS}=-1.5 V, fosc=32.768 kHz, Ta=25°C, C_G=25 pF V_{S1}, V_{L1}, V_{L2} and V_{L3} are internal voltages, and C1=C2=C3=C4=C5=C6=0.1 µF

Item	Symbol	Condition	Min	Typ	Max	Unit
Internal voltage	V_{L1}	Connect 1MΩ load resistor between V_{DD} and V_{L1} (without panel load)	-1.15	-1.05	-0.95	V
	V_{L2}	Connect 1MΩ load resistor between V_{DD} and V_{L2} (without panel load)	2•V_{L1} -0.1		2•V_{L1} × 0.85	V
	V_{L3}	Connect 1MΩ load resistor between V_{DD} and V_{L3} (without panel load)	3•V_{L1} -0.1		3•V_{L1} × 0.85	V
BLD voltage	V_{BLD}		-1.30	-1.20	-1.10	V
BLD circuit response time	t_{BLD}				100	µs
Analog comparator input voltage	V_{IP} V_{IM}	Non-inverted input (CMPP) Inverted input (CMPM)	Vss+0.3		V_{DD} -0.9	V
Analog comparator offset voltage	V_{OF}				20	mV
Analog comparator response time	t_{CMP}	V_{IP} =-1.1V V_{IM}=V_{IP}±30mV			3	ms
Power current consumption	I_{OP}	During HALT Without panel load		2.0	5.5	µA
		During execution [*1] Without panel load		5.5	10.0	µA

*1 The BLD circuit and analog voltage comparator are turned OFF.

S1C62A81 (Normal Operating Mode)

Unless otherwise specified
V_{DD}=0 V, V_{SS}=-3.0 V, fosc=32.768 kHz, Ta=25°C,
C_G=25pF V_{S1}, V_{L1}, V_{L2} and V_{L3} are internal voltages, and
C1=C2=C3=C4=C5=C6=0.1 µF

Item	Symbol	Condition	Min	Typ	Max	Unit
Internal voltage	V_{L1}	Connect 1MΩ load resistor between V_{DD} and V_{L1} (without panel load)	-1.15	-1.05	-0.95	V
	V_{L2}	Connect 1MΩ load resistor between V_{DD} and V_{L2} (without panel load)	2•V_{L1} -0.1		2•V_{L1} × 0.9	V
	V_{L3}	Connect 1MΩ load resistor between V_{DD} and V_{L3} (without panel load)	3•V_{L1} -0.1		3•V_{L1} × 0.9	V
BLD voltage	V_{BLD}		-2.55	-2.40	-2.25	V
BLD circuit response time	t_{BLD}				100	µs
Analog comparator input voltage	V_{IP} V_{IM}	Non-inverted input (CMPP) Inverted input (CMPM)	Vss+0.3		V_{DD} -0.9	V
Analog comparator offset voltage	V_{OF}				10	mV
Analog comparator response time	t_{CMP}	V_{IP} =-1.5V V_{IM} =V_{IP} ±15mV			3	ms
Power current consumption	I_{OP}	During HALT / Without panel load		5.5	10.0	µA
		During execution [*1] / Without panel load		7.2	12.0	µA

*1 The BLD circuit and analog voltage comparator are turned OFF.

S1C62A81 (Heavy Load Protection Mode)

Unless otherwise specified
V_{DD}=0 V, V_{SS}=-3.0 V, fosc=32.768 kHz, Ta=25°C,
C_G=25pF V_{S1}, V_{L1}, V_{L2} and V_{L3} are internal voltages, and
C1=C2=C3=C4=C5=C6=0.1 μF

Item	Symbol	Condition	Min	Typ	Max	Unit
Internal voltage	V_{L1}	Connect 1MΩ load resistor between V_{DD} and V_{L1} (without panel load)	-1.15	-1.05	-0.95	V
	V_{L2}	Connect 1MΩ load resistor between V_{DD} and V_{L2} (without panel load)	2•V_{L1} -0.1		2•V_{L1} × 0.85	V
	V_{L3}	Connect 1MΩ load resistor between V_{DD} and V_{L3} (without panel load)	3•V_{L1} -0.1		3•V_{L1} × 0.85	V
BLD voltage	V_{BLD}		-2.55	-2.40	-2.25	V
BLD circuit response time	t_{BLD}				100	µs
Analog comparator input voltage	V_{IP} V_{IM}	Non-inverted input (CMPP) Inverted input (CMPM)	Vss+0.3		V_{DD} -0.9	V
Analog comparator offset voltage	V_{OF}				10	mV
Analog comparator response time	t_{CMP}	V_{IP} =-1.5V V_{IM} =V_{IP}±15mV			3	ms
Power current consumption	I_{OP}	During HALT Without panel load		11.0	20.0	µA
		During execution [*1]		15.0	25.0	µA

[*1] The BLD circuit and analog voltage comparator are turned OFF.

S1C62B81 (Normal Operating Mode)

Unless otherwise specified
V_{DD}=0 V, V_{SS}=-1.5 V, fosc=32.768 kHz, Ta=25°C,
C_G=25pF V_{S1}, V_{L1}, V_{L2} and V_{L3} are internal voltages, and
C1=C2=C3=C4=C5=C6=0.1 µF

Item	Symbol	Condition	Min	Typ	Max	Unit
Internal voltage	V_{L1}	Connect 1MΩ load resistor between V_{DD} and V_{L1} (without panel load)	-1.15	-1.05	-0.95	V
	V_{L2}	Connect 1MΩ load resistor between V_{DD} and V_{L2} (without panel load)	2•V_{L1} -0.1		2•V_{L1} × 0.9	V
	V_{L3}	Connect 1MΩ load resistor between V_{DD} and V_{L3} (without panel load)	3•V_{L1} -0.1		3•V_{L1} × 0.9	V
BLD voltage	V_{BLD}		-1.30	-1.20	-1.10	V
BLD circuit response time	t_{BLD}				100	µs
Analog comparator input voltage	V_{IP} V_{IM}	Non-inverted input (CMPP) Inverted input (CMPM)	Vss+0.3		V_{DD}-0.9	V
Analog comparator offset voltage	V_{OF}				20	mV
Analog comparator response time	t_{CMP}	V_{IP} =-1.1V V_{IM} =V_{IP}±30mV			3	ms
Power current consumption	I_{OP}	During HALT Without panel load		5.5	10.0	µA
		During execution *1		7.2	12.0	µA

*1 The BLD circuit and analog voltage comparator are turned OFF.

S1C62B81 (Heavy Load Protection Mode)

Unless otherwise specified
V_{DD}=0 V, V_{SS}=-1.5 V, fosc=32.768 kHz, Ta=25°C, V_{S1}, V_{L1}, V_{L2} and V_{L3} are internal voltages, and C1=C2=C3=C4=C5=C6=0.1 µF

Item	Symbol	Condition		Min	Typ	Max	Unit
Internal voltage	V_{L1}	Connect 1MΩ load resistor between V_{DD} and V_{L1} (without panel load)		-1.15	-1.05	-0.95	V
	V_{L2}	Connect 1MΩ load resistor between V_{DD} and V_{L2} (without panel load)		2•V_{L1} -0.1		2•V_{L1} × 0.85	V
	V_{L3}	Connect 1MΩ load resistor between V_{DD} and V_{L3} (without panel load)		3•V_{L1} -0.1		3•V_{L1} × 0.85	V
BLD voltage	V_{BLD}			-1.30	-1.20	-1.10	V
BLD circuit response time	t_{BLD}					100	µs
Analog comparator input voltage	V_{IP} V_{IM}	Non-inverted input (CMPP) Inverted input (CMPM)		Vss+0.3		V_{DD} -0.9	V
Analog comparator offset voltage	V_{OF}					20	mV
Analog comparator response time	t_{CMP}	V_{IP} =-1.1V V_{IM} =V_{IP}±30mV				3	ms
Power current consumption	I_{OP}	During HALT	Without panel load		11.0	20.0	µA
		During execution [*1]			15.0	25.0	µA

*1 The BLD circuit and analog voltage comparator are turned OFF.

6.5 Oscillation Characteristics

Oscillation characteristics will vary according to different conditions. Use the following characteristics are as reference values.

S1C62N81

Unless otherwise specified,
V_{DD}=0 V, V_{SS}=-3.0 V, Crystal: Q13MC146, C_G=25 pF,
C_D=built-in, Ta=25°C

Item	Symbol	Condition	Min	Typ	Max	Unit
Oscillation start voltage	Vsta (Vss)	tsta ≤ 3 sec	-1.8			V
Oscillation stop voltage	Vstp (Vss)	tstp ≤ 10 sec	-1.8			V
Built-in capacity (drain)	C_D	Including the parasitic capacity inside the IC		20		pF
Frequency voltage deviation	f/V	Vss=-1.8 to -3.5 V			5	ppm
Frequency IC deviation	f/I_C		-10		10	ppm
Frequency adjustment range	f/C_G	C_G =5–25pF	40			ppm
Higher harmonic oscillation start voltage	V_{hho} (Vss)				-3.5	V
Allowable leak resistor	R_{leak}	Between OSC1 and V_{DD} and Vss	200			MΩ

S1C62L81

Unless otherwise specified,
V_{DD}=0 V, V_{SS}=-1.5 V, Crystal: Q13MC146, C_G=25 pF,
C_D=built-in, Ta=25°C

Item	Symbol	Condition	Min	Typ	Max	Unit
Oscillation start voltage	Vsta (Vss)	tsta ≤ 3 sec	-1.1			V
Oscillation stop voltage	Vstp (Vss)	tstp ≤ 10 sec	-1.1 (-0.9) *1			V
Built-in capacity (drain)	C_D	Including the parasitic capacity inside the IC		20		pF
Frequency voltage deviation	f/V	Vss=-1.1 to -3.5 V (-0.9) *1			5	ppm
Frequency IC deviation	f/I_C		-10		10	ppm
Frequency adjustment range	f/C_G	C_G =5–25pF	40			ppm
Higher harmonic oscillation start voltage	V_{hho} (Vss)				-3.5	V
Allowable leak resistor	R_{leak}	Between OSC1 and V_{DD} and Vss	200			MΩ

*1 Items enclosed in parentheses ( ) are those used when operating at heavy load protection mode.

S1C62A81

Unless otherwise specified,

V_{DD}=0 V, V_{SS}=-3.0 V, R_{CR}=850 kΩ, Ta=25°C

Item	Symbol	Condition	Min	Typ	Max	Unit
Oscillation frequency dispersion	fosc		-20	32.768 kHz	20	%
Oscillation start voltage	Vsta		-1.8			V
Oscillation start time	tsta	Vss=-1.8 to -3.5V		3		ms
Oscillation stop voltage	Vstp		-1.8			V

S1C62B81

Unless otherwise specified,

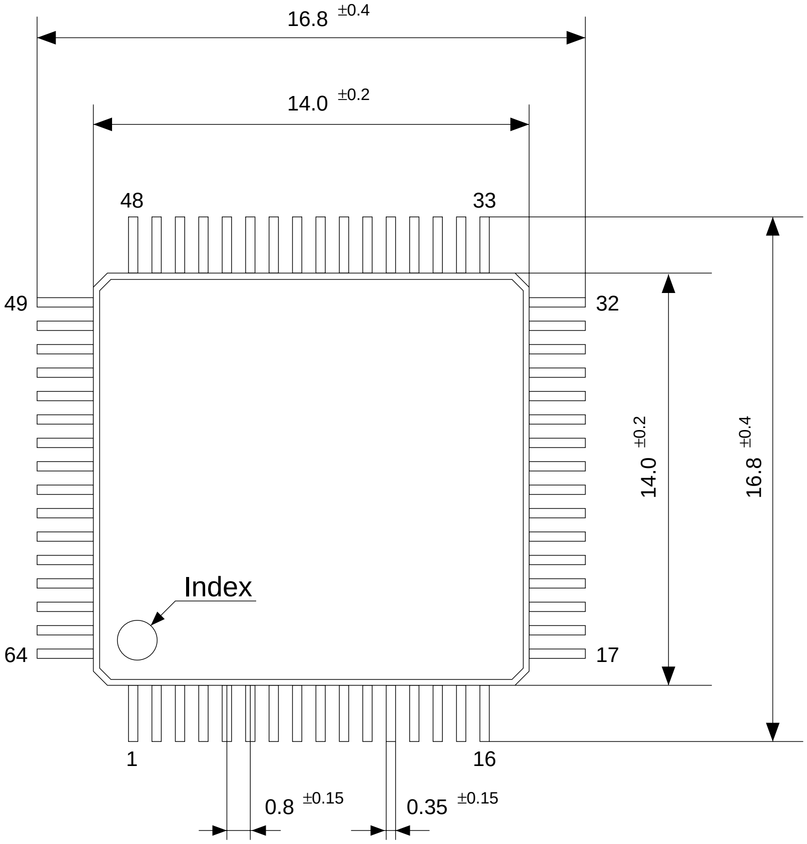
V_{DD}=0 V, V_{SS}=-1.5 V, R_{CR}=850 kΩ, Ta=25°C

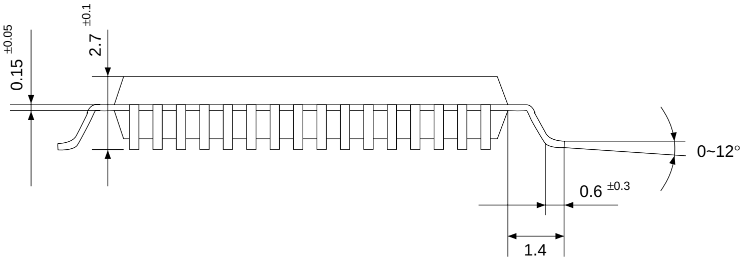
Item	Symbol	Condition	Min	Typ	Max	Unit
Oscillation frequency dispersion	fosc		-20	32.768 kHz	20	%
Oscillation start voltage	Vsta		-0.9			V
Oscillation start time	tsta	Vss=-0.9 to -3.5V		3		ms
Oscillation stop voltage	Vstp		-0.9			V

CHAPTER 7 PACKAGE

7.1 Plastic Package (1)

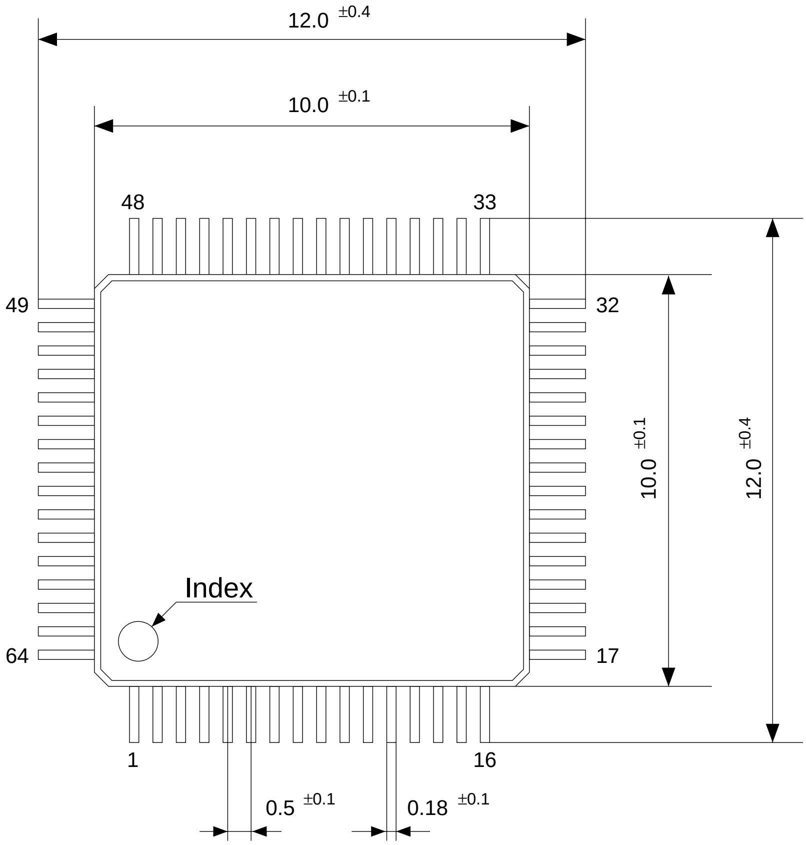
QFP6-64pin

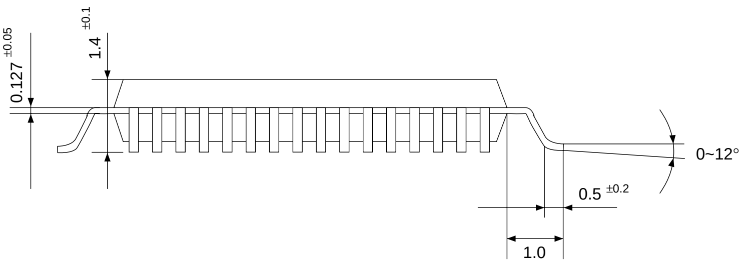




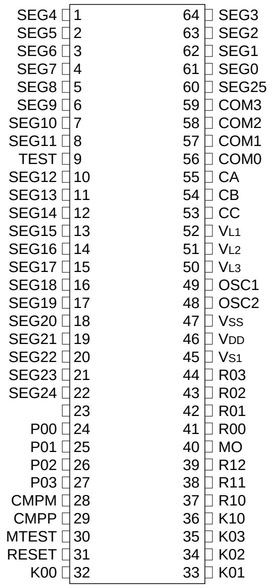
7.2 Plastic Package (2)

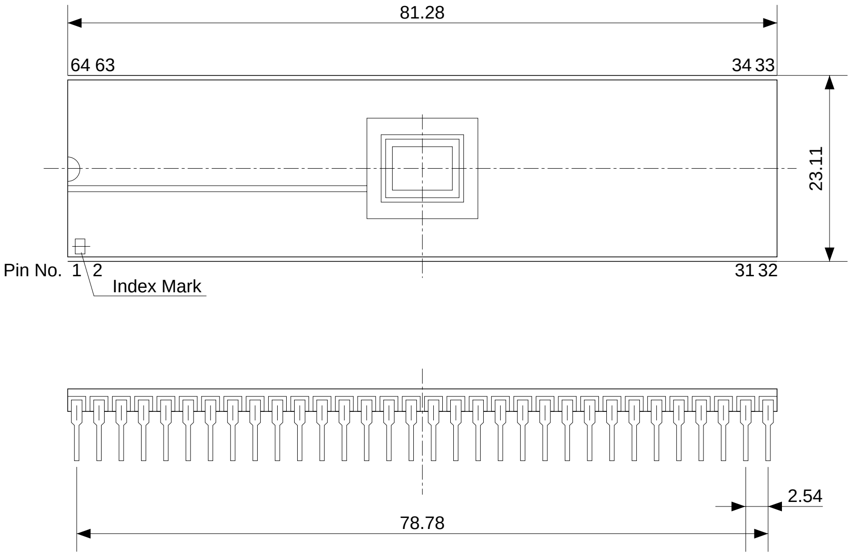
QFP13-64pin





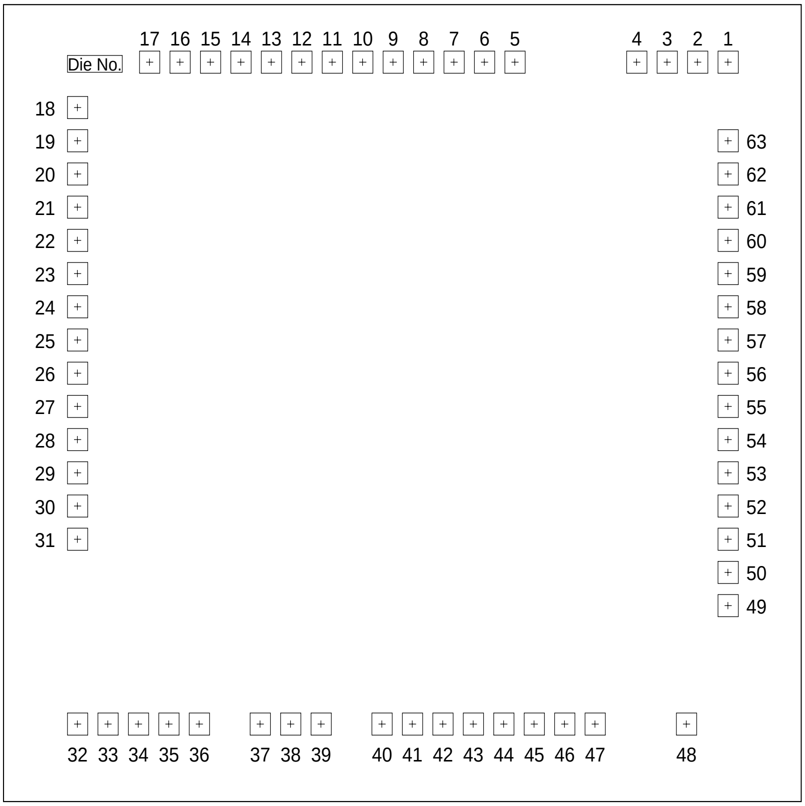
7.3 Ceramic Package for Test Sample





CHAPTER 8 PAD LAYOUT

8.1 Diagram of Pad Layout



Chip size:	3.84 mm × 3.84 mm
Chip thickness:	400 µm
Pad opening:	95 µm

8.2 S1C62N81 List of Pad Names

No	PAD Name	No	PAD Name	No	PAD Name
1	COM0	22	SEG15	43	K03
2	COM1	23	SEG16	44	K10
3	COM2	24	SEG17	45	R10
4	COM3	25	SEG18	46	R11
5	SEG25	26	SEG19	47	R12
6	SEG0	27	SEG20	48	MO
7	SEG1	28	SEG21	49	R00
8	SEG2	29	SEG22	50	R01
9	SEG3	30	SEG23	51	R02
10	SEG4	31	SEG24	52	R03
11	SEG5	32	P00	53	V_{S1}
12	SEG6	33	P01	54	V_{DD}
13	SEG7	34	P02	55	V_{SS}
14	SEG8	35	P03	56	OSC2
15	SEG9	36	CMPM	57	OSC1
16	SEG10	37	CMPP	58	V_{L3}
17	SEG11	38	MTEST	59	V_{L2}
18	TEST	39	RESET	60	V_{L1}
19	SEG12	40	K00	61	CC
20	SEG13	41	K01	62	CB
21	SEG14	42	K02	63	CA

8.3 Pad Coordinates

PAD No	X	Y	PAD No	X	Y	PAD No	X	Y
1	1722.0	1753.2	22	-1753.2	837.2	43	571.6	-1752.4
2	1560.4	1753.2	23	-1753.2	677.2	44	732.4	-1752.4
3	1400.4	1753.2	24	-1753.2	517.2	45	914.0	-1752.4
4	1238.8	1753.2	25	-1753.2	357.2	46	1090.0	-1752.4
5	616.0	1753.2	26	-1753.2	197.2	47	1249.6	-1752.4
6	456.0	1753.2	27	-1753.2	37.2	48	1679.2	-1752.4
7	296.0	1753.2	28	-1753.2	-122.8	49	1753.2	-856.0
8	136.0	1753.2	29	-1753.2	-282.8	50	1753.2	-696.0
9	-24.0	1753.2	30	-1753.2	-442.8	51	1753.2	-536.0
10	-184.0	1753.2	31	-1753.2	-602.8	52	1753.2	-376.0
11	-344.0	1753.2	32	-1714.0	-1752.4	53	1753.2	-210.8
12	-504.0	1753.2	33	-1554.0	-1752.4	54	1753.2	-50.8
13	-664.0	1753.2	34	-1393.2	-1752.4	55	1753.2	109.2
14	-824.0	1753.2	35	-1233.2	-1752.4	56	1753.2	269.2
15	-984.0	1753.2	36	-1025.6	-1752.4	57	1753.2	430.8
16	-1144.0	1753.2	37	-685.2	-1752.4	58	1753.2	613.2
17	-1304.0	1753.2	38	-524.8	-1752.4	59	1753.2	773.2
18	-1753.2	1538.8	39	-349.6	-1752.4	60	1753.2	937.2
19	-1753.2	1317.2	40	92.4	-1752.4	61	1753.2	1097.2
20	-1753.2	1157.2	41	251.6	-1752.4	62	1753.2	1261.2
21	-1753.2	997.2	42	412.4	-1752.4	63	1753.2	1421.2

II. S1C62N81 Technical Software

CONTENTS

Software

CHAPTER 1 CONFIGURATION

1.1 S1C62N81 Block Diagram

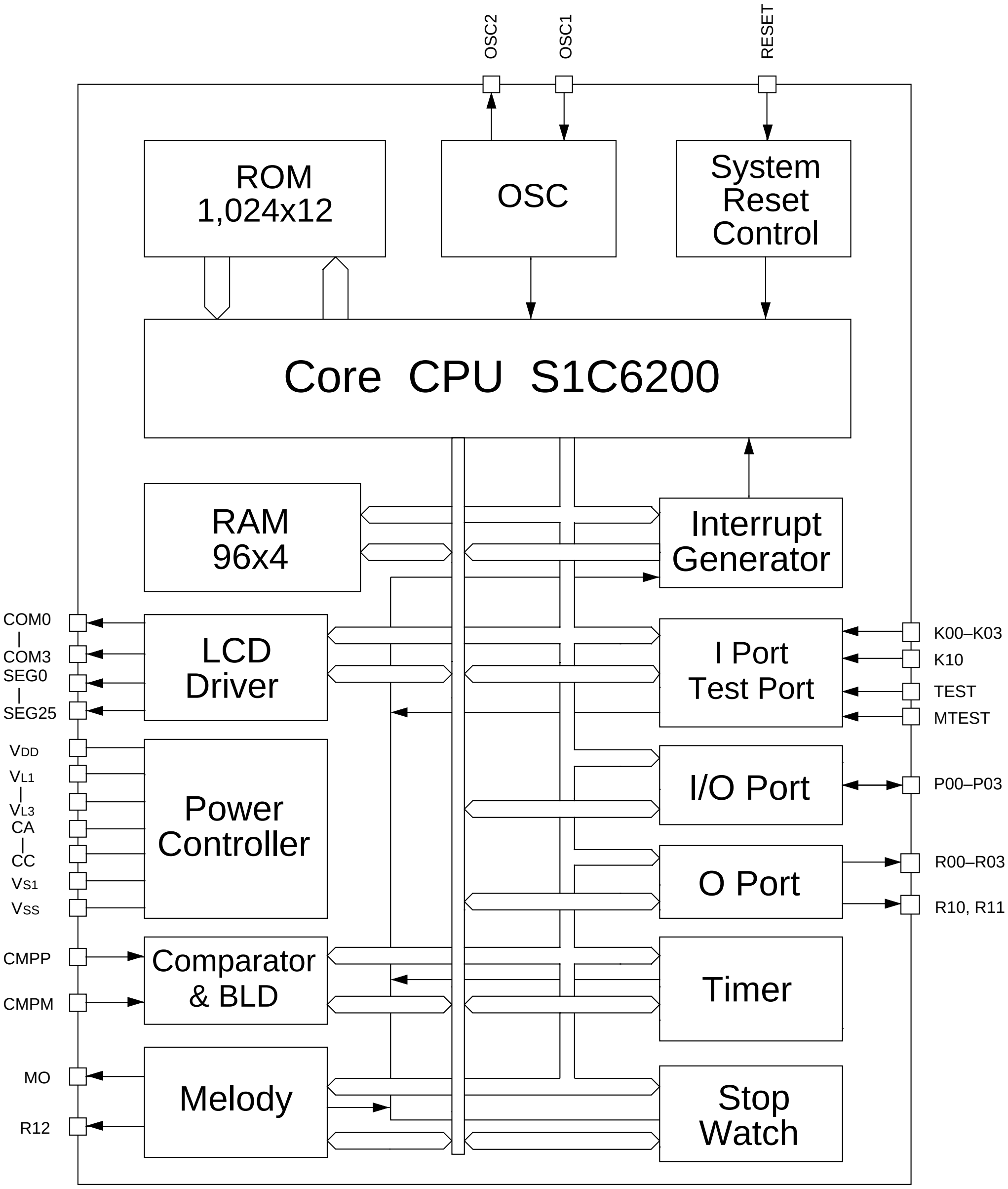


Fig. 1.1.1
S1C62N81 block diagram

1.2 ROM Map

The S1C62N81 has a built-in mask ROM with a capacity of 1,024 steps × 12 bits for program storage. The configuration of the ROM is shown in Figure 1.2.1.

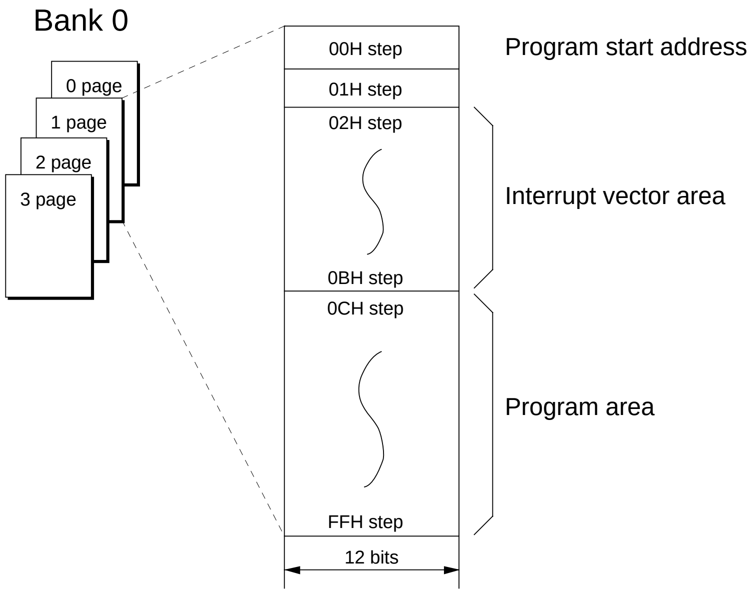


Fig. 1.2.1
Configuration of built-in ROM

1.3 Interrupt Vectors

When an interrupt request is received by the CPU, the CPU initiates the following interrupt processing after completing the instruction being executed.

(1) The address of the next instruction to be executed (the value of the program counter) is saved on the stack (RAM).

(2) The interrupt vector address corresponding to the interrupt request is loaded into the program counter.

(3) The branch instruction written in the vector is executed to branch to the software interrupt processing routine.

Note *Steps 1 and 2 require 12 cycles of the CPU system clock.*

The correspondence between interrupt requests and vectors are shown in Table 1.3.1.

Table 1.3.1
Interrupt requests and vectors

Vector	Priority	Interrupt Request
10AH	1	Melody interrupt
108H	2	Input (K10) interrupt
106H	3	Input (K00–K03) interrupt
104H	4	Stopwatch timer interrupt
102H	5	Clock timer interrupt

When multiple interrupts occur simultaneously, they are executed in order of priority.

1.4 Data Memory Map

The S1C62N81 built-in RAM has 96 words of data memory, 32 words of display memory for the LCD, and I/O memory for controlling the peripheral circuit. When writing programs, note the following:

(1) Since the stack area is in the data memory area, take care not to overwrite the stack with data. Subroutine calls or interrupts use 3 words on the stack.

(2) Data memory addresses 000H–00FH are memory register areas that are addressed with register pointer RP.

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 (96 words x 4 bits) R/W															
	2																
	3																
	4																
	5																
	6	Unused area															
	7																
	8																
	9	Display memory															
	A																
	B	Unused area															
	C																
	D																
	E	I/O memory															
	F																

Fig. 1.4.1
Data 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 1.4.1 (a) I/O memory map (0E0H–0E3H)

Address	Register								Comment
	D3	D2	D1	D0	Name	SR [1]	1	0	
0E0H	K03	K02	K01	K00	K03	– [2]	High	Low	Input port (K00–K03)
	R				K02	– [2]	High	Low	
					K01	– [2]	High	Low	
					K00	– [2]	High	Low	
0E1H	0	0	0	K10	0 [5]				
	R				0 [5]				
					0 [5]				
					K10	– [2]	High	Low	Input port (K10)
0E2H	SWL3	SWL2	SWL1	SWL0	SWL3	0			MSB
	R				SWL2	0			Stopwatch timer 1/100 sec (BCD)
					SWL1	0			
					SWL0	0			LSB
0E3H	SWH3	SWH2	SWH1	SWH0	SWH3	0			MSB
	R				SWH2	0			Stopwatch timer 1/10 sec (BCD)
					SWH1	0			
					SWH0	0			LSB

[1] Initial value following initial reset

[2] Not set in the circuit

[3] Undefined

[4] Reset (0) immediately after being read

[5] Always 0 when being read

[6] Refer to main manual

Table 1.4.1 (b) I/O memory map (0E4H–0E7H)

Address	Register								Comment
	D3	D2	D1	D0	Name	SR *1	1	0	
0E4H	TM3	TM2	TM1	TM0	TM3	–	High	Low	Timer data (clock timer 2 Hz)
	R				TM2	–	High	Low	Timer data (clock timer 4 Hz)
					TM1	–	High	Low	Timer data (clock timer 8 Hz)
					TM0	–	High	Low	Timer data (clock timer 16 Hz)
0E5H	KCP03	KCP02	KCP01	KCP00	KCP03	0	Falling	Rising	Input comparison register (K03)
	R/W				KCP02	0	Falling	Rising	Input comparison register (K02)
					KCP01	0	Falling	Rising	Input comparison register (K01)
					KCP00	0	Falling	Rising	Input comparison register (K00)
0E6H	0	0	0	KCP10	0 *5				
	R			R/W	0 *5				
					0 *5				
					KCP10	0	Falling	Rising	Input comparison register (K10)
0E7H	0	0	0	EIMEL	0 *5				
	R			R/W	0 *5				
					0 *5				
					EIMEL	0	Enable	Mask	Interrupt mask register (melody)

*1 Initial value following initial reset

*2 Not set in the circuit

*3 Undefined

*4 Reset (0) immediately after being read

*5 Always 0 when being read

*6 Refer to main manual

Table 1.4.1 (c) I/O memory map (0E8H–0EBH)

Address	Register								Comment
	D3	D2	D1	D0	Name	SR *1	1	0	
0E8H	EIK03	EIK02	EIK01	EIK00	EIK03	0	Enable	Mask	Interrupt mask register (K03)
	R/W				EIK02	0	Enable	Mask	Interrupt mask register (K02)
					EIK01	0	Enable	Mask	Interrupt mask register (K01)
					EIK00	0	Enable	Mask	Interrupt mask register (K00)
0E9H	0	0	0	EIK10	0 *5				
	R			R/W	0 *5				
					0 *5				
					EIK10	0	Enable	Mask	Interrupt mask register (K10)
0EAH	0	0	EISW1	EISW0	0 *5				
	R		R/W		0 *5				
					EISW1	0	Enable	Mask	Interrupt mask register (stopwatch 1 Hz)
					EISW0	0	Enable	Mask	Interrupt mask register (stopwatch 10 Hz)
0EBH	0	EIT2	EIT8	EIT32	0 *5				
	R	R/W			EIT2	0	Enable	Mask	Interrupt mask register (clock timer 2 Hz)
					EIT8	0	Enable	Mask	Interrupt mask register (clock timer 8 Hz)
					EIT32	0	Enable	Mask	Interrupt mask register (clock timer 32 Hz)

*1 Initial value following initial reset
*2 Not set in the circuit
*3 Undefined
*4 Reset (0) immediately after being read
*5 Always 0 when being read
*6 Refer to main manual

Table 1.4.1 (d) I/O memory map (0ECH–0EFH)

Address	Register								Comment
	D3	D2	D1	D0	Name	SR *1	1	0	
0ECH	0	0	0	IMEL	0 *5				
	R				0 *5				
					0 *4				
					IMEL *4	0	Yes	No	Interrupt factor flag (melody)
0EDH	0	0	IK1	IK0	0 *5				
	R				0 *5				
					IK1 *4	0	Yes	No	Interrupt factor flag (K10)
					IK0 *4	0	Yes	No	Interrupt factor flag (K00–K03)
0EEH	0	0	ISW1	ISW0	0 *5				
	R				0 *5				
					ISW1 *4	0	Yes	No	Interrupt factor flag (stopwatch 1 Hz)
					ISW0 *4	0	Yes	No	Interrupt factor flag (stopwatch 10 Hz)
0EFH	0	IT2	IT8	IT32	0 *5				
	R				IT2 *4	0	Yes	No	Interrupt factor flag (clock timer 2 Hz)
					IT8 *4	0	Yes	No	Interrupt factor flag (clock timer 8 Hz)
					IT32 *4	0	Yes	No	Interrupt factor flag (clock timer 32 Hz)

*1 Initial value following initial reset
*2 Not set in the circuit
*3 Undefined
*4 Reset (0) immediately after being read
*5 Always 0 when being read
*6 Refer to main manual

Table 1.4.1 (e) I/O memory map (0F0H–0F3H)

Address	Register								Comment
	D3	D2	D1	D0	Name	SR *1	1	0	
0F0H	MAD3	MAD2	MAD1	MAD0	MAD3	0	High	Low	Melody ROM address (AD3)
	R/W				MAD2	0	High	Low	Melody ROM address (AD2)
					MAD1	0	High	Low	Melody ROM address (AD1)
					MAD0	0	High	Low	Melody ROM address (AD0, LSB)
0F1H	0	MAD6	MAD5	MAD4	0 *5				
	R	R/W			MAD6	0	High	Low	Melody ROM address (AD6, MSB)
					MAD5	0	High	Low	Melody ROM address (AD5)
					MAD4	0	High	Low	Melody ROM address (AD4)
0F2H	CLKC1	CLKC0	TEMPC	MELC	CLKC1	0	High	Low	CLKC1(0)&CLKC0(0) : melody speed × 1 CLKC1(0)&CLKC0(1) : melody speed × 8
	R/W				CLKC0	0	High	Low	CLKC1(1)&CLKC0(0) : melody speed × 16 CLKC1(1)&CLKC0(1) : melody speed × 32
					TEMPC	0	High	Low	Tempo change control
					MELC	0	ON	OFF	Melody control ON/OFF
0F3H	R03	R02	R01	R00	R03	0	High	Low	Output port data (R00–R03)
	R/W				R02	0	High	Low	
					R01	0	High	Low	
					R00	0	High	Low	

*1 Initial value following initial reset
*2 Not set in the circuit
*3 Undefined
*4 Reset (0) immediately after being read
*5 Always 0 when being read
*6 Refer to main manual

Table 1.4.1 (f) I/O memory map (0F4H, 0F6H, 0F9H–0FAH)

Address	Register								Comment
	D3	D2	D1	D0	Name	SR *1	1	0	
0F4H	*3	R12 $\overline{\text{MO}}$ ENV	R11	R10 FOUT	R12	0	High	Low	Output port data (R12)
	R/W				$\overline{\text{MO}}$				Inverting melody output
					ENV				Melody envelope control
					R11	0	High	Low	Output port data (R11)
					R10	0	High	Low	Output port data (R10)
					FOUT		ON	OFF	Frequency output
0F6H	P03	P02	P01	P00	P03	– *2	High	Low	I/O port (P00–P03)
	R/W				P02	– *2	High	Low	
					P01	– *2	High	Low	
					P00	– *2	High	Low	
0F9H	0	TMRST	SWRUN	SWRST	0 *5				
	R	W	R/W	W	TMRST *5	Reset	Reset	–	Clock timer reset
					SWRUN	0	Run	Stop	Stopwatch timer RUN/STOP
					SWRST *5	Reset	Reset	–	Stopwatch timer reset
0FAH	HLMOD	0	BLDDT	BLDON	HLMOD	0	Heavy load	Normal load	Heavy load protection mode register
	R/W	R		R/W	0 *5				
					BLDDT	0	Battery voltage low	Battery voltage normal	Battery voltage low detector data
					BLDON	0	ON	OFF	Battery voltage low detector ON/OFF

*1 Initial value following initial reset

*2 Not set in the circuit

*3 Undefined

*4 Reset (0) immediately after being read

*5 Always 0 when being read

*6 Refer to main manual

Table 1.4.1 (g) I/O memory map (0FBH–0FCH)

Address	Register								Comment
	D3	D2	D1	D0	Name	SR *1	1	0	
0FBH	CSDC	0	CMPDT	CMPON	CSDC	0	Static	Dynamic	LCD drive switch
	R/W	R		R/W	0 *5				
					CMPDT	1	+ > -	- > +	Comparator's voltage condition: 1 = CMPP(+)input > CMPM(-)input, 0 = CMPM(-)input > CMPP(+)input
					CMPON	0	On	Off	Analog comparator ON/OFF
0FCH	0	0	0	IOC	0 *5				
	R			R/W	0 *5				
					0 *5				
					IOC	0	Output	Input	I/O port P00–P03 Input/Output

*1 Initial value following initial reset

*2 Not set in the circuit

*3 Undefined

*4 Reset (0) immediately after being read

*5 Always 0 when being read

*6 Refer to main manual

CHAPTER 2 INITIAL RESET

2.1 Internal Register Status on Initial Reset

Following an initial reset, the internal registers and internal data memory area are initialized to the values shown in Tables 2.1.1 and 2.1.2.

Table 2.1.1
Initial values of internal registers

Internal Register		Bit Length	Initial Value Following Reset
Program counter step	PCS	8	00H
Program counter page	PCP	4	1H
New page pointer	NPP	4	1H
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	Undefined
Zero flag	Z	1	Undefined
Carry flag	C	1	Undefined

Table 2.1.2
Initial values of internal data memory area

Internal Data Memory Area	Bit Length	Initial Value Following Reset	Address
RAM data	4 × 96	Undefined	000H–05FH
Display memory	4 × 26	Undefined	090H–0AFH
Internal I/O register	See Tables 1.4.1 (a)–1.4.1 (g)		0E0H–0FCH

After an initial reset, the program counter page (PCP) is initialized to 1H, and the program counter step (PCS), to 00H. This is why the program is executed from step 00H of the first page.

The initial values of some internal registers and internal data memory area locations are undefined after a reset. Set them as necessary to the proper initial values in the program.

The peripheral I/O functions (memory-mapped I/O) are assigned to internal data memory area addresses 0E0H to 0FCH. Each address represents a 4-bit internal I/O register, allowing access to the peripheral functions in 1-word (4-bit) read/write units.

2.2 Initialize Program Example

The following is a program that clears the RAM and LCD, resets the flags, registers, timer, and stopwatch timer, and sets the stack pointer immediately after resetting the system.

```
Label    Mnemonic/operand        Comment
         ORG     100H
         JP      INIT            ;Jump to "INIT"
;
         ORG     110H
INIT     RST     F,0011B         ;Interrupt mask, decimal
                                 ;adjustment off
;
         LD      X,0             ;
RAMCLR   LDPX    MX,0            ;
         CP      XH,6H           ;  Clear RAM (00H–5FH)
         JP      NZ,RAMCLR       ;
         LD      X,90H           ;
LCPCLR   LDPX    MX,0            ;
         CP      X,0BH           ;  Clear LCD (90H–AFH)
         JP      NZ,LCDCLR       ;
;
         LD      A,0             ;
         LD      B,5             ;  Set stack pointer to 50H
         LD      SPL,A           ;
         LD      SPH,B           ;
;
         LD      X,0F9H          ;  Reset timer and stopwatch
         OR      MX,0101B        ;  timer
;
         LD      X,0EBH          ;  Enable timer interrupt
         OR      MX,0111B        ;
;
         LD      X,0E8H          ;  Enable input interrupt
         OR      MX,1111B        ;  (K03–K00)
;
         LD      X,0             ;
         LD      Y,0             ;
         LD      A,0             ;  Reset register flags
         LD      B,0             ;
         RST     F,0             ;
         EI                      ;Enable interrupt
```

The above program is a basic initialization program for the S1C62N81. The setting data are all initialized as shown in Table 2.1.1 by executing this program. When using this program, add setting items necessary for each specific application. (Figure 2.2.1 is the flow chart for this program.)

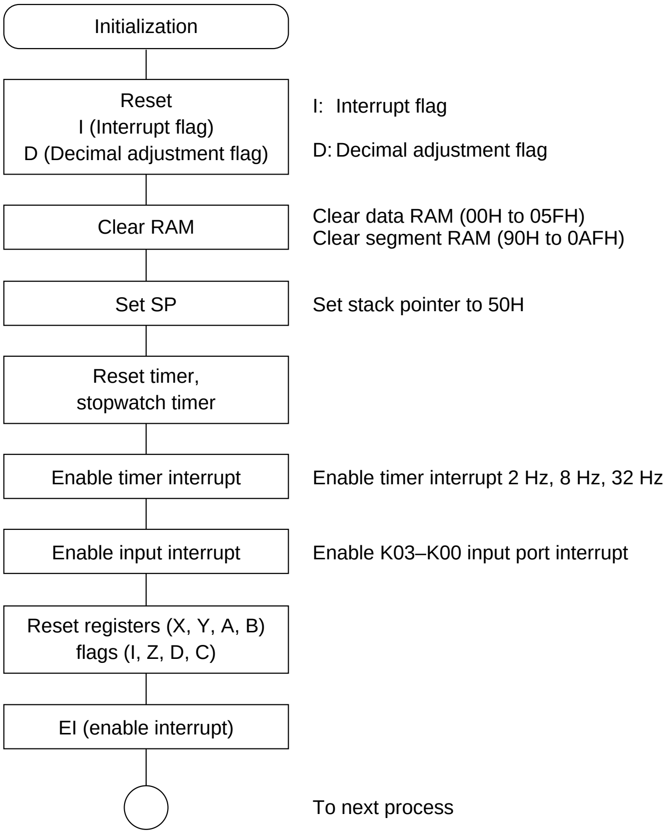

Initialization
Reset
I (Interrupt flag)
D (Decimal adjustment flag)
I: Interrupt flag
D: Decimal adjustment flag
Clear RAM
Clear data RAM (00H to 05FH)
Clear segment RAM (90H to 0AFH)
Set SP
Set stack pointer to 50H
Reset timer, stopwatch timer
Enable timer interrupt
Enable timer interrupt 2 Hz, 8 Hz, 32 Hz
Enable input interrupt
Enable K03–K00 input port interrupt
Reset registers (X, Y, A, B) flags (I, Z, D, C)
EI (enable interrupt)
To next process

Fig. 2.2.1
Flow chart of the initialization program

Table 2.2.2
Execution result of the initialization program

Internal circuit	Setting value
General register A	0H
General register B	0H
Index register X	00H
Index register Y	00H
Stack pointer SP	50H
Interrupt flag I	1
Decimal flag D	0
Zero flag Z	0
Carry flag C	0
RAM data (00H~5FH)	0H
Segment data (90H~0AFH)	0H
Clock timer: reset, Clock timer interrupt: valid	
Stopwatch timer: reset	
K00–K03 interrupt: valid	

CHAPTER 3 PERIPHERAL CIRCUITS

Details on how to control the S1C62N81 peripheral circuit is given in this chapter.

3.1 Input Ports

Input port memory map

Table 3.1.1 (a) I/O memory map

Address	Register								Comment
	D3	D2	D1	D0	Name	SR *1	1	0	
0E0H	K03	K02	K01	K00	K03	– *2	High	Low	Input port (K00–K03)
	R				K02	– *2	High	Low	
					K01	– *2	High	Low	
					K00	– *2	High	Low	
0E1H	0	0	0	K10	0 *5				
	R				0 *5				
					0 *5				
					K10	– *2	High	Low	Input port (K10)
0E5H	KCP03	KCP02	KCP01	KCP00	KCP03	0	Falling	Rising	Input comparison register (K03)
	R/W				KCP02	0	Falling	Rising	Input comparison register (K02)
					KCP01	0	Falling	Rising	Input comparison register (K01)
					KCP00	0	Falling	Rising	Input comparison register (K00)
0E6H	0	0	0	KCP10	0 *5				
	R			R/W	0 *5				
					0 *5				
					KCP10	0	Falling	Rising	Input comparison register (K10)

*1 Initial value following initial reset
*2 Not set in the circuit
*3 Undefined
*4 Reset (0) immediately after being read
*5 Always 0 when being read
*6 Refer to main manual

Table 3.1.1 (b) I/O memory map

Address	Register								Comment
	D3	D2	D1	D0	Name	SR *1	1	0	
0E8H	EIK03	EIK02	EIK01	EIK00	EIK03	0	Enable	Mask	Interrupt mask register (K03)
	R/W				EIK02	0	Enable	Mask	Interrupt mask register (K02)
					EIK01	0	Enable	Mask	Interrupt mask register (K01)
					EIK00	0	Enable	Mask	Interrupt mask register (K00)
0E9H	0	0	0	EIK10	0 *5				
	R			R/W	0 *5				
					0 *5				
					EIK10	0	Enable	Mask	Interrupt mask register (K10)
0EDH	0	0	IK1	IK0	0 *5				
	R				0 *5				
					IK1 *4	0	Yes	No	Interrupt factor flag (K10)
					IK0 *4	0	Yes	No	Interrupt factor flag (K00–K03)

*1 Initial value following initial reset
*2 Not set in the circuit
*3 Undefined
*4 Reset (0) immediately after being read
*5 Always 0 when being read
*6 Refer to main manual

Control of the input port

The S1C62N81 Series have a 4-bit input port (K00–K03) and a 1-bit input port (K10).

The data registers for the input ports K00–K03 and K10 are assigned to the addresses 0E0H (D0–D3) and 0E1H (D0), respectively. The status of the input port terminals can be read from the addresses in 4-bit units (K00–K03 and K10).

The input ports have an interrupt function that can be controlled using the interrupt factor flags and interrupt mask registers which have been set in each bit. See Section 3.11, "Interrupt and Halt", for details.

Examples of input port control program

• Loading K00–K03 into the A register

```
Label    Mnemonic/operand        Comment
         LD       Y,0E0H         ;Set address of port
         LD       A,MY           ;A register ← K00–K03
```

As shown in Figure 3.1.1, the two instruction steps above load the data of the input port into the A register.

A register	D3	D2	D1	D0
	K03	K02	K01	K00

Fig. 3.1.1
Loading the A register

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

• Bit-unit checking of input ports

```
Label      Mnemonic/operand          Comment
           DI                        ;Disable interrupt
           LD     Y,0E0H             ;Set address of port
INPUT1:    FAN    MY,0010B           ;
           JP     NZ,INPUT1          ;Loop until K01 becomes "0"
INPUT2:    FAN    MY,0010B           ;
           JP     Z,INPUT2           ;Loop until K01 becomes "1"
```

This program loopes until a rising edge is input to input port K01.

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

Note *When the input port is changed from high level to low level with a pull-down resistor, the signal falls following a certain delay caused by the time constants of the pull-down resistance and the input gate capacitance. It is therefore necessary to observe a proper wait time before the input port data is read.*

3.2 Output Ports

Output port memory map

Table 3.2.1 I/O memory map

Address	Register								Comment
	D3	D2	D1	D0	Name	SR *1	1	0	
0F3H	R03	R02	R01	R00	R03	0	High	Low	Output port data (R00–R03)
	R/W				R02	0	High	Low	
					R01	0	High	Low	
					R00	0	High	Low	
0F4H	*3	R12 $\overline{\text{MO}}$ ENV	R11	R10 FOUT	R12	0	High	Low	Output port data (R12)
	R/W				$\overline{\text{MO}}$				Inverting melody output
					ENV				Melody envelope control
					R11	0	High	Low	Output port data (R11)
					R10	0	High	Low	Output port data (R10)
					FOUT		ON	OFF	Frequency output

*1 Initial value following initial reset
*2 Not set in the circuit
*3 Undefined
*4 Reset (0) immediately after being read
*5 Always 0 when being read
*6 Refer to main manual

Control of the output port

The S1C62N81 Series have 7 bits of general output ports (R00–R03, R10–R12).

The output ports are assigned to the address 0F3H (D0–D3) for R00–R03 and the address 0F4H (D0–D2) for R10–R12 as the registers that can read and write. The output port terminals output the contents written to the registers. In addition, since the output status of the output port can be read via the registers, the output ports can be controlled in each bit using a logical operation instruction such as AND and OR.

At initial reset, the output ports go to a low level (the registers are 0).

Examples of output port control program

• Loading B register data into R00–R03

```
Label   Mnemonic/operand        Comment
        LD      Y,0F3H          ;Set address of port
        LD      MY,B            ;R00–R03 ← B register
```

As shown in Figure 3.2.1, the two instruction steps above load the data of the B register into the output ports.

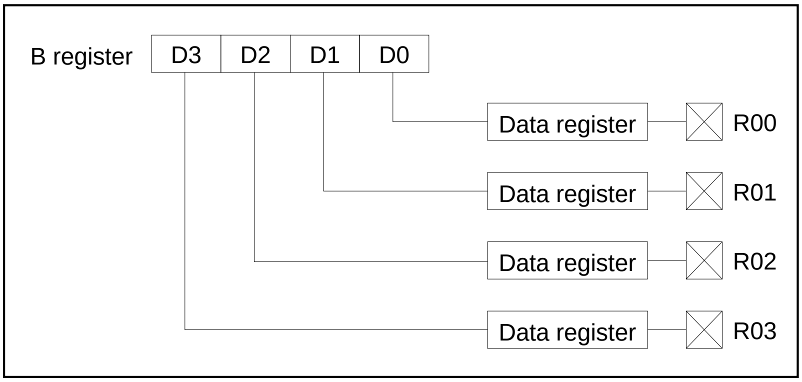


Fig. 3.2.1
Control of the output port

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

• Bit-unit operation of output ports

```
Label    Mnemonic/operand          Comment
         LD      Y,0F3H            ;Set address of port
         OR      MY,0010B          ;Set R01 to 1
         AND     MY,1011B          ;Set R02 to 0
```

The three instruction steps above cause the output port to be set, as shown in Figure 3.2.2.

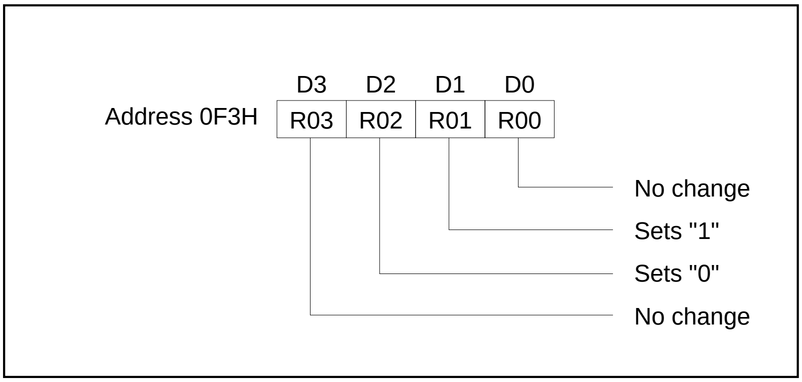


Fig. 3.2.2
Setting of the output port

3.3 Special Use Output Ports

Special use output port memory map

Table 3.3.1 I/O memory map

Address	Register								Comment
	D3	D2	D1	D0	Name	SR *1	1	0	
0F4H	*3	R12 $\overline{\text{MO}}$ ENV	R11	R10 FOUT	R12	0	High	Low	Output port data (R12)
	R/W				$\overline{\text{MO}}$				Inverting melody output
					ENV				Melody envelope control
					R11	0	High	Low	Output port data (R11)
					R10	0	High	Low	Output port data (R10)
					FOUT		ON	OFF	Frequency output

*1 Initial value following initial reset

*2 Not set in the circuit

*3 Undefined

*4 Reset (0) immediately after being read

*5 Always 0 when being read

*6 Refer to main manual

Control of the special use output port

In addition to the regular DC, special output can be selected for output ports R10–R12, as shown in Table 3.3.2. Figure 3.3.1 shows the structure of output ports R10–R12.

Table 3.3.2
Special output

Pin Name	When Special Output is Selected
R12	$\overline{\text{MO}}$ or ENV
R10	FOUT

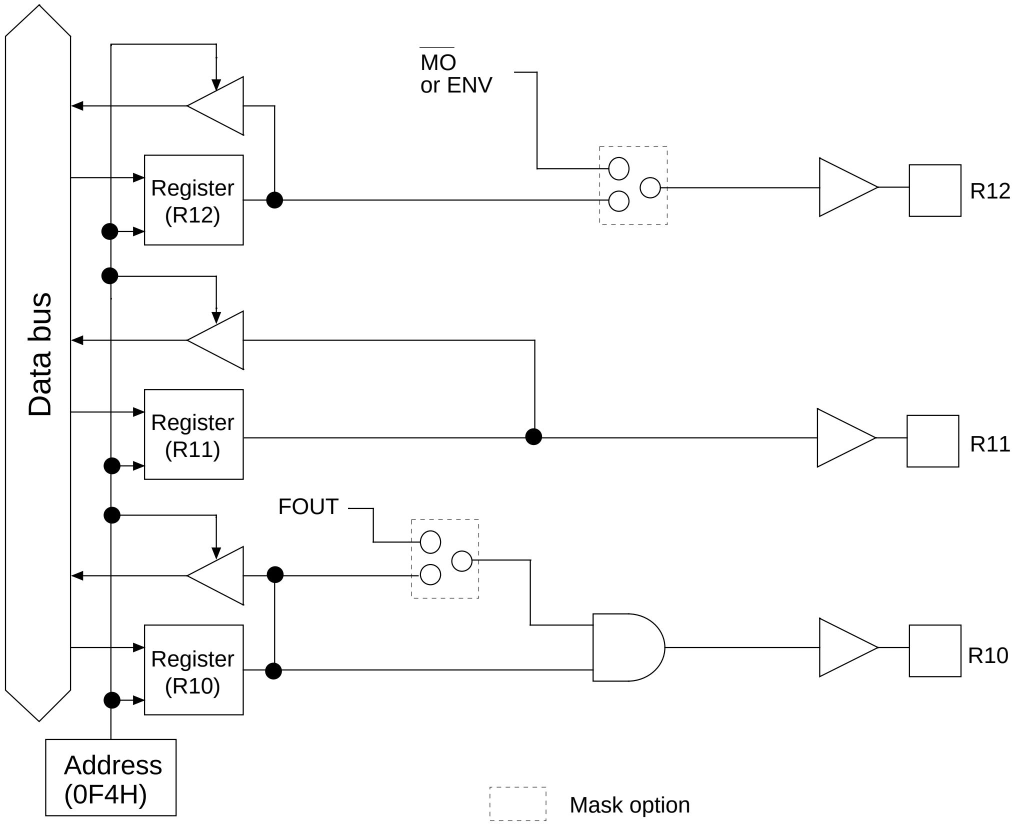


Fig. 3.3.1
Structure of output ports
R10–R12

Example of special use output port control program

• Melody output MO, $\overline{\text{MO}}$ or envelope output (R12)

MO and $\overline{\text{MO}}$ (or ENV) are the melody signal output pins for driving a piezo or speaker through an amplifying transistor. Refer to 3.10, "Melody Generator".

• FOUT (R10)

When output port R10 is set for FOUT, it outputs the fosc clock or the divided fosc. The clock frequencies listed in Table 3.3.3 selectable by mask option.

Table 3.3.3
Selectable by mask option

Setting Value	Clock Frequency (Hz)
	fosc = 32,768
fosc / 1	32,768
fosc / 2	16,384
fosc / 4	8,192
fosc / 8	4,096
fosc / 16	2,048
fosc / 32	1,024
fosc / 64	512
fosc / 128	256

```
Label   Mnemonic/operand          Comment
        LD      Y,0F4H            ;Set address of port
        OR      MY,0001B          ;Turn on FOUT
        AND     MY,1110B          ;Turn off FOUT
```

3.4 I/O Ports

I/O port memory map

Table 3.4.1 I/O memory map

Address	Register								Comment
	D3	D2	D1	D0	Name	SR *1	1	0	
0F6H	P03	P02	P01	P00	P03	– *2	High	Low	I/O port (P00–P03)
	R/W				P02	– *2	High	Low	
					P01	– *2	High	Low	
					P00	– *2	High	Low	
0FCH	0	0	0	IOC	0 *5				
	R			R/W	0 *5				
					0 *5				
					IOC	0	Output	Input	I/O port P00–P03 Input/Output

*1 Initial value following initial reset
*2 Not set in the circuit
*3 Undefined
*4 Reset (0) immediately after being read
*5 Always 0 when being read
*6 Refer to main manual

Control of the I/O port

The S1C62N81 contains a 4-bit general I/O port (4 bits × 1). This port can be used as an input port or an output port, according to I/O port control register IOC. When IOC is "0", the port is set for input, when it is "1", the port is set for output.

• How to set an input port

Set "0" in the I/O port control register (D0 of address 0FCH), and the I/O port is set as an input port. The state of the I/O port (P00–P03) is decided by the data of address 0F6H. (In the input mode, the port level is read directly.)

• How to set an output port

Set "1" in the I/O port control register, and the I/O port is set as an output port. The state of the I/O port is decided by the data of address 0F6H. This data is held by the register, and can be set regardless of the contents of the I/O control register. (The data can be set whether P00 to P03 ports are input ports or output ports.)

The I/O control registers are cleared to "0" (input/output ports are set as input ports), and the data registers are also cleared to "0" after an initial reset.

Examples of I/O port control program

• Loading P00–P03 input data into A register

Label	Mnemonic/operand	Comment
	`LD    Y,0FCH`	;Set address of I/O control port
	`AND   MY,1110B`	;Set port as input port
	`LD    Y,0F6H`	;Set address of port
	`LD    A,MY`	;A register ← P00–P03

As shown in Figure 3.4.1, the four instruction steps above load the data of the I/O ports into the A register.

Fig. 3.4.1
Loading into the A register

	D3	D2	D1	D0
A register	P03	P02	P01	P00

• Loading P00–P03 output data into A register

Label	Mnemonic/operand		Comment
	LD	Y,0FCH	;Set the address of input/output ;port control register
	OR	MY,0001B	;Set as output port
	LD	Y,0F6H	;Set the address of port
	LD	A,MY	;A register ← P00–P03

As shown in Figure 3.4.2, the four instruction steps above load the data of the I/O ports into the A register.

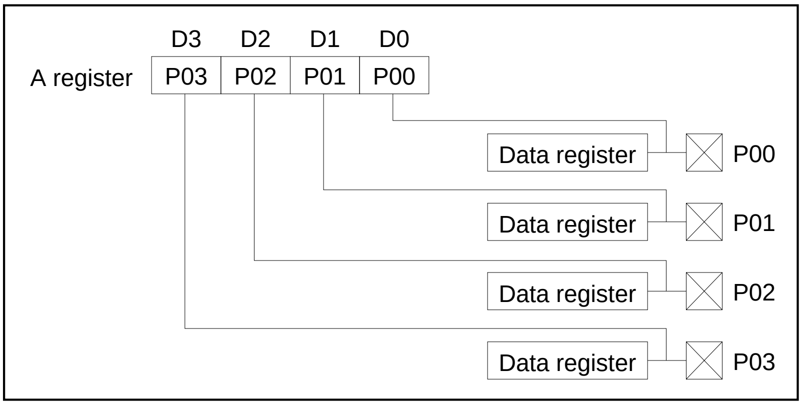


Fig. 3.4.2
Control of I/O port (input)

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

• Loading contents of B register into P00–P03

Label	Mnemonic/operand		Comment
	LD	Y,0FCH	;Set the address of input/output
			;port control register
	OR	MY,0001B	;Set port as output port
	LD	Y,0F6H	;Set the address of port
	LD	MY,B	;P00–P03 ← B register

As shown in Figure 3.4.3, the four instruction steps above load the data of the B register into the I/O ports.

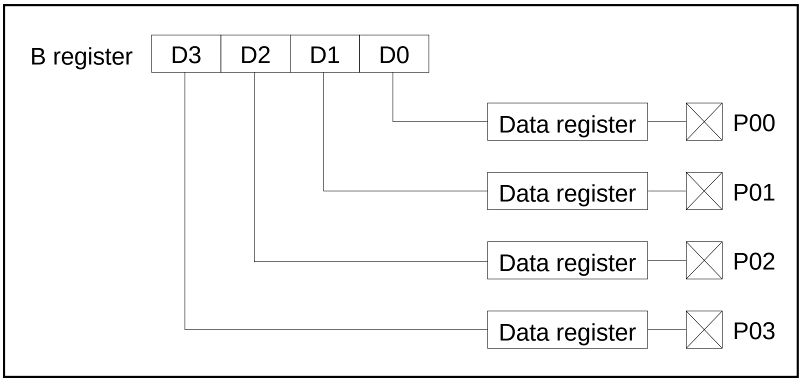


Fig. 3.4.3
Control of the I/O port (output)

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

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

3.5 LCD Driver

LCD driver memory map

Table 3.5.1 I/O memory map

Address	Register								Comment
	D3	D2	D1	D0	Name	SR *1	1	0	
0FBH	CSDC	0	CMPDT	CMPON	CSDC	0	Static	Dynamic	LCD drive switch
	R/W	R		R/W	0 *5				
					CMPDT	1	+ > -	- > +	Comparator's voltage condition: 1 = CMPP(+)input > CMPM(-)input, 0 = CMPM(-)input > CMPP(+)input
					CMPON	0	On	Off	Analog comparator ON/OFF

*1 Initial value following initial reset

*2 Not set in the circuit

*3 Undefined

*4 Reset (0) immediately after being read

*5 Always 0 when being read

*6 Refer to main manual

Address	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
090	Display memory (write only) 32 words x 4 bits															
0A0																

Fig. 3.5.1
Display memory map

Control of the LCD driver

The S1C62N81 contains 128 bits of display memory in addresses 090H to 0AFH of the data memory. Each display memory can be assigned to any 104 bits of the 128 bits for the LCD driver (26 SEG × 4 COM) or 78 bits of the 128 bits (26 SEG × 3 COM) by using a mask option. The remaining 24 bits or 50 bits of display memory are not connected to the LCD driver, and are not output even when data is written. An LCD segment is on with "1" set in the display memory, and off with "0" set in the display memory. Note that the display memory is a write-only.

• LCD drive control register (CSDC)

The LCD drive control register (CSDC: address 0FBH, D3) can be set either for dynamic drive or for static drive. Set "0" in CSDC for 1/3 duty or 1/4 duty (time-shared) dynamic drive. Set "1" in CSDC and the same value in the registers corresponding to COM0 to COM2 (1/3) or COM0 to COM3 (1/4) for static drive. Figure 3.5.2 is the static drive control of the LCD, and Figure 3.5.3 is an example of the 7-segment LCD assignment.

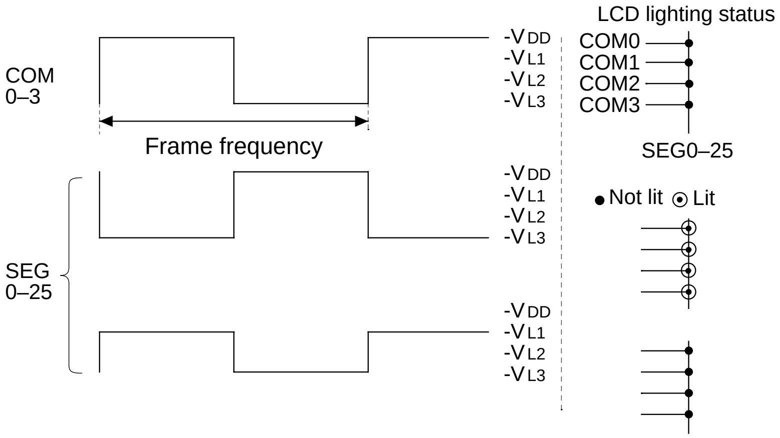


Fig. 3.5.2
LCD static drive control

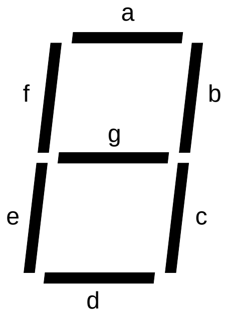


Fig. 3.5.3
7-segment LCD assignment

Address	Register			
	D3	D2	D1	D0
090H	d	c	b	a
091H		g	f	e

In the assignment shown in Figure 3.5.3, the 7-segment display pattern is controlled by writing data to display memory addresses 090H and 091H.

Examples of LCD driver control program

• Displaying 7-segment

The LCD display routine using the assignment of Figure 3.5.3 can be programmed as follows.

```
Label      Mnemonic/operand         Comment
           ORG     000H
           RETD    3FH              ;0 is displayed
           RETD    06H              ;1 is displayed
           RETD    5BH              ;2 is displayed
           RETD    4FH              ;3 is displayed
           RETD    66H              ;4 is displayed
           RETD    6DH              ;5 is displayed
           RETD    7DH              ;6 is displayed
           RETD    27H              ;7 is displayed
           RETD    7FH              ;8 is displayed
           RETD    6FH              ;9 is displayed

SEVENS:    LD      B,0              ;Set the address of jump
           LD      X,090H           ;Set address of display memory
           JPBA
```

When the above routine 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 3.5.4, seven segments are displayed according to the contents of the A register.

Fig. 3.5.4
Data set in A register and displayed patterns

A resister	Display	A resister	Display	A resister	Display	A resister	Display	A resister	Display
0	0	2	2	4	4	6	6	8	8
1	1	3	3	5	5	7	7	9	9

The RETD instruction can be used to write data to the display memory only if it is addressed using the X register. (Addressing 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 the display memory

Fig. 3.5.5
Example of segment assignment

Address	Data			
	D3	D2	D1	D0
090H			▲	●

▲ : SEG - A
● : SEG - B

```
Label   Mnemonic/operand         Comment
        LD      X,SEGBUF         ;Set address display
                                 ;memory buffer
        LD      Y,090H           ;Set address display memory
        LD      MX,3             ;Set buffer data
        LD      MY,MX            ;SEG-A, 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 display memory in bit-units for the assignment of Figure 3.5.5, a buffer must be provided in RAM to hold data. Note that, since the display memory is write-only, data cannot be changed directly using an ALU instruction (for example, AND or OR).

After manipulating the data in the buffer, write it into the corresponding display memory using the transfer command.

3.6 Timer

Timer memory map

Table 3.6.1 I/O memory map

Address	Register								Comment
	D3	D2	D1	D0	Name	SR *1	1	0	
0E4H	TM3	TM2	TM1	TM0	TM3	–	High	Low	Timer data (clock timer 2 Hz)
	R				TM2	–	High	Low	Timer data (clock timer 4 Hz)
					TM1	–	High	Low	Timer data (clock timer 8 Hz)
					TM0	–	High	Low	Timer data (clock timer 16 Hz)
0EBH	0	EIT2	EIT8	EIT32	0 *5				
	R	R/W			EIT2	0	Enable	Mask	Interrupt mask register (clock timer 2 Hz)
					EIT8	0	Enable	Mask	Interrupt mask register (clock timer 8 Hz)
					EIT32	0	Enable	Mask	Interrupt mask register (clock timer 32 Hz)
0EFH	0	IT2	IT8	IT32	0 *5				
	R				IT2 *4	0	Yes	No	Interrupt factor flag (clock timer 2 Hz)
					IT8 *4	0	Yes	No	Interrupt factor flag (clock timer 8 Hz)
					IT32 *4	0	Yes	No	Interrupt factor flag (clock timer 32 Hz)
0F9H	0	TMRST	SWRUN	SWRST	0 *5				
	R	W	R/W	W	TMRST *5	Reset	Reset	–	Clock timer reset
					SWRUN	0	Run	Stop	Stopwatch timer RUN/STOP
					SWRST *5	Reset	Reset	–	Stopwatch timer reset

*1 Initial value following initial reset
*2 Not set in the circuit
*3 Undefined
*4 Reset (0) immediately after being read
*5 Always 0 when being read
*6 Refer to main manual

Control of the timer

The S1C62N81 contains a timer with a basic oscillation of 32.768 kHz (typical). This timer is a 4-bit binary counter, and the counter data can be read as necessary. The counter data of the 16 Hz clock can be read by reading TM3 to TM0 (address 0E4H, D3 to D0). ("1" to "0" are set in TM3 to TM0, corresponding to the high-low levels of the 2 Hz, 4 Hz, 8 Hz, and 16 Hz 50 % duty waveform. See Figure 3.6.1.) The timer can also interrupt the CPU on the falling edges of the 32 Hz, 8 Hz, and 2 Hz signals. For details, see Section 3.11, "Interrupt and Halt".

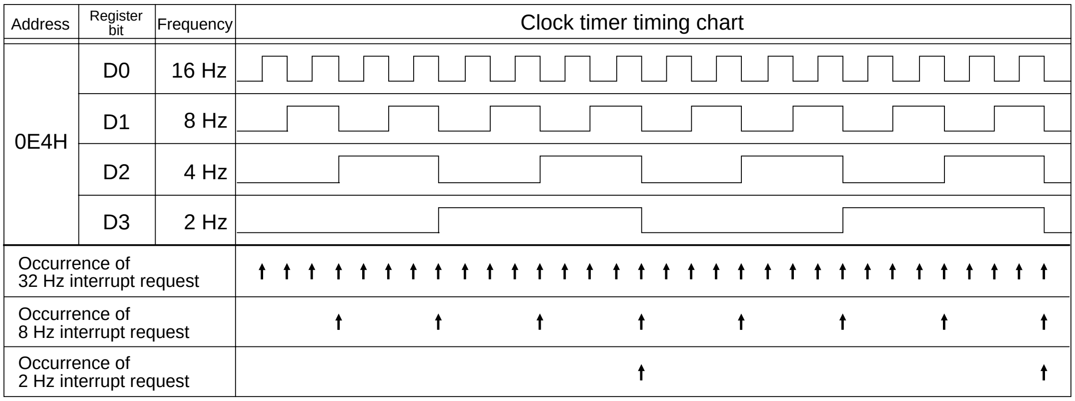

Address
Register bit
Frequency
Clock timer timing chart
0E4H
D0
16 Hz
D1
8 Hz
D2
4 Hz
D3
2 Hz
Occurrence of 32 Hz interrupt request
Occurrence of 8 Hz interrupt request
Occurrence of 2 Hz interrupt request

Fig. 3.6.1
Output waveform of timer and interrupt timing

The timer is reset by setting "1" in TMRST (address 0F9H, D2).

Note *The 128 Hz to 2 Hz of the internal divider is initialized by resetting the timer, and 128 Hz to 1 Hz of the internal divider is reset by resetting the stopwatch timer.*

The dividers of the timer and stopwatch timers are individual circuits, so resetting one circuit does not affect the other.

Examples of timer control program

• Initializing the timer

```
Label     Mnemonic/operand         Comment
          LD      Y,0F9H           ;Set address of the timer
                                   ;reset register
          OR      MY,0100B         ;Reset the timer
```

The two instruction steps above are used to reset (clear TM0–TM3 to 0) and restart the timer. The TMRST register is cleared to "0" by hardware 1 clock after it is set to "1".

• Loading the timer

```
Label     Mnemonic/operand         Comment
          LD      Y,0E4H           ;Set address of
                                   ;the timer data (TM0 to TM3)
          LD      A,MY             ;Load the data of
                                   ;TM0 to TM3 into A register
```

As shown in Table 3.6.2, the two instruction steps load the data of TM0 to TM3 into the A register.

Table 3.6.2
Loading the timer data

A register	D3	D2	D1	D0
	TM3 (2 Hz)	TM2 (4 Hz)	TM1 (8 Hz)	TM0 (16 Hz)

• Checking timer edge

```
Label        Mnemonic/operand       Comment
             LD    X,TMSTAT         ;Set address of the timer edge counter
             CP    MX,0             ;Check whether the timer edge
                                    ;counter is "0"
             JP    Z,RETURN         ;Jump if "0" (Z-flag is "1")
             LD    Y,0E4H           ;Set address of the timer
             LD    A,MY             ;Read the data of TM0 to TM3
                                    ;into A register
             LD    Y,TMDTBF         ;Set address of the timer data buffer
             XOR   MY,A             ;Did the count on the timer
                                    ;change?
             FAN   MY,0100B         ;Check bit D2 of the timer data buffer
             LD    MY,A             ;Set the data of A register into
                                    ;the timer data buffer
             JP    Z,RETURN         ;Jump, if the Z-flag is "1"
             ADD   MX,0FH           ;Decrement the 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 every 125 ms until the data reaches "0". The timing chart is shown in Figure 3.6.2. The timer can be addressed using the X register instead of the Y register.

Note *TMSTAT and TMDTBF may be any address in RAM and not involve a hardware function.*

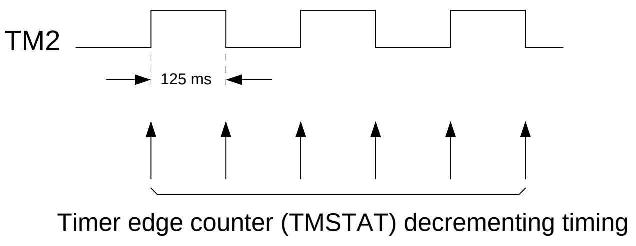


Fig. 3.6.2
Timing of the timer edge counter

3.7 Stopwatch Timer

Stopwatch timer memory map

Table 3.7.1 I/O memory map

Address	Register								Comment
	D3	D2	D1	D0	Name	SR *1	1	0	
0E2H	SWL3	SWL2	SWL1	SWL0	SWL3	0			MSB
	R				SWL2	0			Stopwatch timer 1/100 sec (BCD)
					SWL1	0			
					SWL0	0			LSB
0E3H	SWH3	SWH2	SWH1	SWH0	SWH3	0			MSB
	R				SWH2	0			Stopwatch timer 1/10 sec (BCD)
					SWH1	0			
					SWH0	0			LSB
0EAH	0	0	EISW1	EISW0	0 *5				
	R		R/W		0 *5				
					EISW1	0	Enable	Mask	Interrupt mask register (stopwatch 1 Hz)
					EISW0	0	Enable	Mask	Interrupt mask register (stopwatch 10 Hz)
0EEH	0	0	ISW1	ISW0	0 *5				
	R				0 *5				
					ISW1 *4	0	Yes	No	Interrupt factor flag (stopwatch 1 Hz)
					ISW0 *4	0	Yes	No	Interrupt factor flag (stopwatch 10 Hz)
0F9H	0	TMRST	SWRUN	SWRST	0 *5				
	R	W	R/W	W	TMRST *5	Reset	Reset	–	Clock timer reset
					SWRUN	0	Run	Stop	Stopwatch timer RUN/STOP
					SWRST *5	Reset	Reset	–	Stopwatch timer reset

*1 Initial value following initial reset
*2 Not set in the circuit
*3 Undefined
*4 Reset (0) immediately after being read
*5 Always 0 when being read
*6 Refer to main manual

Control of the stopwatch timer

The S1C62N81 contains 1/100 sec and 1/10 sec stopwatch timers.

This timer can be loaded in 4-bit units. Starting, stopping, and resetting the timer can be controlled by register.

Figure 3.7.1 shows the operation of the stopwatch timer.

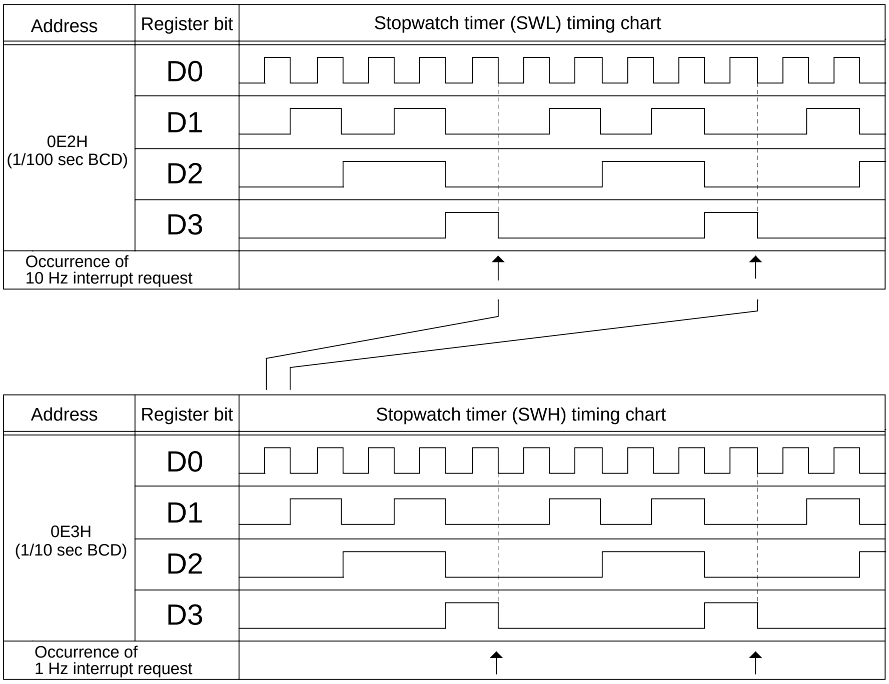


Fig. 3.7.1
Stopwatch timer
operating timing

Examples of stopwatch timer control program

• Initializing the stopwatch timer

Label	Mnemonic/operand		Comment
	LD	Y,0F9H	;Set address of the SWRST register
	OR	MY,0001B	;Reset the stopwatch timer

The two instruction steps above reset the stopwatch timer. (SWL3 to SWL0, SWH3 to SWH0 are all cleared to "0".)

Note *The stopwatch timer is reset by setting "1" in the SWRST register. However, the SWRST register is cleared to "0" by hardware 1 clock after it is set to "1".*

• Starting the stopwatch timer

Label	Mnemonic/operand		Comment
	LD	Y,0F9H	;Set address of SWRUN register
	OR	MY,0010B	;Start the stopwatch timer

The two instruction steps above run the stopwatch timer of SWL0 to SWL3, and SWH0 to SWH3 (addresses 0E2H and 0E3H, respectively).

• Stopping the stopwatch timer

Label	Mnemonic/operand		Comment
	LD	Y,0F9H	;Set address of SWRUN register
	AND	MY,1101B	;Stop the stopwatch timer

The two instruction steps above stop the stopwatch timer of SWL0 to SWL3, and SWH0 to SWH3 (addresses 0E2H and 0E3H, respectively).

• Loading the stopwatch timer

```
Label   Mnemonic/operand        Comment
        LD      Y,0E2H          ;Set address of the SWL of
                                ;the stopwatch
        LDPY    A,MY            ;Read the data of SWL0 to SWL3
                                ;into A register
        LD      B,MY            ;Read the data of SWH0 to SWH3
                                ;into B register
```

The three instruction steps above reads the contents of the stopwatch timer into A register and B register. (Also see Table 3.7.2.)

Table 3.7.2
Data load into A register and B register

	D3	D2	D1	D0
A register	SWL3	SWL2	SWL1	SWL0
B register	SWH3	SWH2	SWH1	SWH0

Note *A read-in error caused by a carry from the SWL is not taken into account in this program. You are recommended to add a handling routine in your application.*

3.8 Battery Voltage Low Detection (BLD) Circuit and Heavy Load Protection Function

The S1C62N81 Series has built-in battery voltage low detection circuit and drop in power battery voltage may be detected by controlling the register on the I/O memory. Criteria voltages are as follows:

Model	Criteria Voltage
S1C62N81/62A81	2.4 V ± 0.15 V
S1C62L81/62B81	1.2 V ± 0.10 V

Moreover, when the battery load becomes heavy, such as during external piezo buzzer driving or external lamp lighting, heavy load protection function is built-in in case the battery voltage drops. S1C62L81/62B81 operates at 0.9 V due to the BLD circuit and heavy load protection function.

BLD circuit and heavy load protection function memory map

Table 3.8.1 I/O memory map

Address	Register								Comment
	D3	D2	D1	D0	Name	SR *1	1	0	
0FAH	HLMOD	0	BLDDT	BLDON	HLMOD	0	Heavy load	Normal load	Heavy load protection mode register
	R/W	R		R/W	0 *5				
					BLDDT	0	Battery voltage low	Battery voltage normal	BLD data
					BLDON	0	ON	OFF	BLD ON/OFF

*1 Initial value following initial reset

*2 Not set in the circuit

*3 Undefined

*4 Reset (0) immediately after being read

*5 Always 0 when being read

*6 Refer to main manual

Control of the BLD circuit

The BLD circuit will turn ON by writing "1" on the BLDON register (address 0FAH, D0, R/W) and battery voltage low detection will be performed. By writing "0" on the BLDON register, the detection result is stored in the BLDDT register. However, in order to obtain a stable detection result, it is necessary to turn the BLD circuit ON for at least 100 µs. Accordingly, reading out the detection result from the BLDDT register is performed through the following procedures:

① Set the BLDON register to "1".
② Provide at least 100 µs waiting time.
③ Set the BLDON register to "0".
④ Read-out from the BLDDT register.

Note, however, that when S1C62N81 is to be used with the normal system clock at fosc = 32.768 kHz, there is no need for the waiting time stated in the above procedure ② since 1 instruction cycle will take longer than 100 µs.
Because the power current consumption of the IC becomes large when the BLD circuit is operated, turn the BLD circuit OFF when not in use. The operation timing chart is shown in Figure 3.8.1.

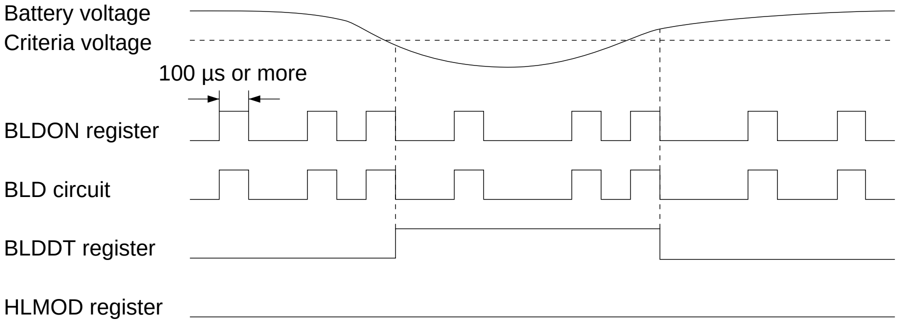


Fig. 3.8.1
Timing chart of battery voltage low detection operation through the BLDON register

Example of BLD circuit control program

```
Label   Mnemonic/Operand        Comment
        LD      X,0FAH          ;Sets the address of BLDON
        OR      MX,0001B        ;Sets BLDON to "1"
        AND     MX,1110B        ;Sets BLDON to "0"
        LD      A,MX            ;Loads the detection result
                                ;into the A register
```

Heavy load protection function

There are two ways to operate the heavy load protection function:

- **Operation through the HLMOD register**
  The heavy load protection function may be operated by writing "1" on the HLMOD register (address 0FAH, D3, R/W). Simultaneously, the BLD circuit will turn ON and battery voltage low detection by hardware every 2 Hz (0.5 sec) will automatically be performed.
  Operation through the HLMOD register is useful when heavy load can be anticipated such as when S1C62N81 drives the piezo buzzer. The operation timing chart is shown in Figure 3.8.2.

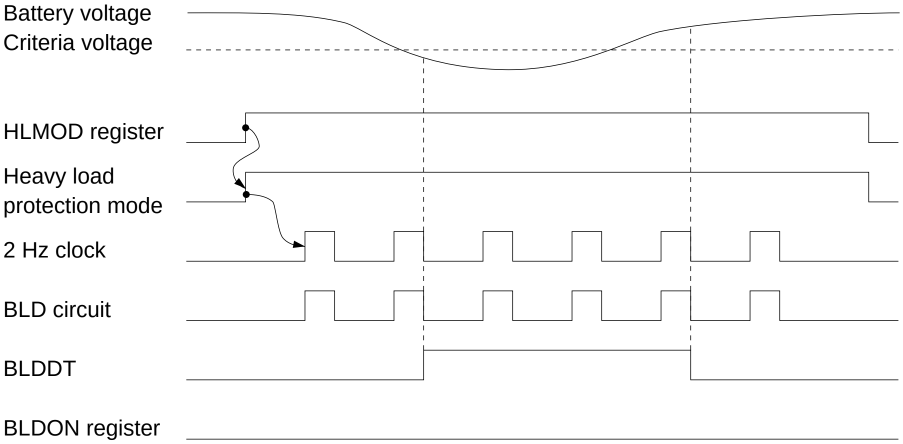


Fig. 3.8.2 Timing chart of battery voltage low detection operation through the HLMOD register

- **Operation through the BLDON register**
  The BLD circuit will turn ON by writing "1" on the BLDON register (address 0FAH, D0, R/W) and battery voltage low detection will be performed. By writing "0" on the BLDON register, the detection result is stored in the BLDDT register. If this results in the battery voltage being lower than the criteria voltage, the heavy load protection function will operate. In other words, the BLD circuit in this case serves as a sensor for detecting the operational state of the heavy load protection function.

Operation through the BLDON circuit is useful as a measure against unforeseen circumstances, such as drop in supply voltage due to expiring battery life, by way of promptly operating the heavy load protection function.
The following procedures for controlling the BLD circuit by the software are the same as those described in "Control of the BLDON circuit":

① Set the BLDON register to "1".
② Provide at least 100 µs waiting time.
③ Set the BLDON register to "0".
④ Read-out from the BLDDT register.

If the battery voltage is lower than the criteria voltage, the heavy load protection function will automatically start operating after the above procedure ③ has been performed.
Because battery voltage low detection by hardware every 2 Hz (0.5 sec) will automatically be performed when the heavy load protection function operates, refrain from operating the BLD circuit with the software in order to minimize power current consumption. The operation timing chart is shown in Figure 3.8.3.

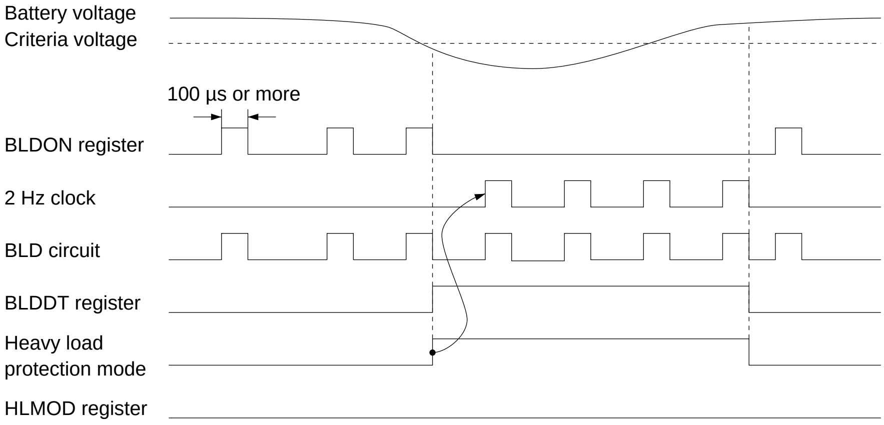


Fig. 3.8.3
Timing chart of heavy load protection function operation through the BLDON register

Examples of heavy load protection function control program

- **Operation through the HLMOD register**

  This is a sample program when lamp is driven with the R00 terminal during performance of heavy load protection.

```
Label   Mnemonic/Operand       Comment
        LD     X,0FAH          ;Sets the address of HLMOD
        OR     MX,1000B        ;Sets to the heavy protection mode
        LD     Y,0F3H          ;Sets the address of R0n port
        OR     MY,0001B        ;Turns lamp ON
           :
           :
        LD     Y,0F3H          ;Sets the R0n port address
        AND    MY,1110B        ;Turns the lamp on
        CALL   WT1S            ;1 second waiting time (software timer)
        AND    MX,0111B        ;Cancels the heavy load protection mode
```

In the above program, the heavy load protection mode is canceled after 1 second waiting time provided as the time for the battery voltage to stabilize after the lamp is turned off; however, since this time varies according to the nature of the battery, time setting must be done in accordance with the actual application.

- **Operation through the BLDON register**

```
Label   Mnemonic/Operand       Comment
        LD      X,0FAH         ;Sets the HLMOD/BLDDT address
        FAN     MX,1010B       ;Checks the HLMOD/BLDDT bits
        JP      NZ,HLMOD       ;Heavy load protection mode
        OR      MX,0001B       ;Sets the BLDON to "1"
        AND     MX,1110B       ;Sets the BLDON to "0"
        FAN     MX,0010B       ;Checks the BLDDT bit
        JP      Z,HLMOD        ;Shifts the mode to
                               ;the heavy load protection mode
        LD      Y,FLAG
        AND     MY,0           ;Resets the flag to "0"
        RET
;
HLMOD:  LD      Y,FLAG
        OR      MY,1           ;Sets the flag to "1"
        RET
```

The above program operates the heavy load protection function by using the BLDON register. In the normal operation mode, battery voltage low detection is done from the BLDON register and when the battery voltage drops below the criteria voltage, the mode shifts to the heavy load protection mode. In the heavy load protection mode, battery voltage low detection by the hardware is done every 2 Hz and the detection result is stored in the BLDDT register. Because of this, the BLDDT register will be "1" during the heavy load protection mode. Moreover, in the above program, battery voltage low detection by the BLDON is halted during the heavy load protection mode. If the battery voltage become grater than the criteria voltage, the BLDDT register value will become "0" and hence, battery voltage low detection through the BLDON register will resume after checking the BLDDT register value. When used as a sub-routine, the above program will enable the user to determine whether the present operation mode is the normal operation mode (flag = "0") or the heavy load protection mode (flag = "1").
The flow chart for the above program is shown in the next page.

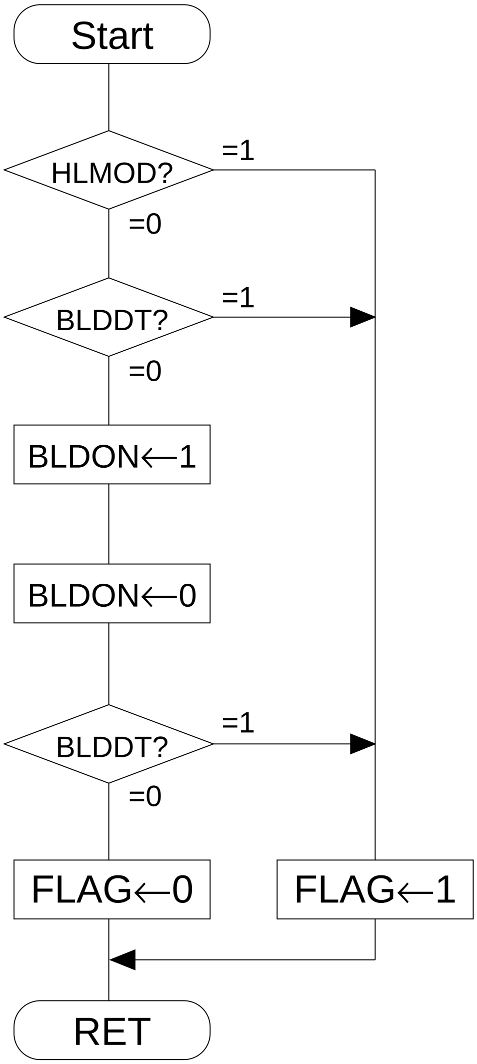


Fig. 3.8.4
Flow chart of operation
through the BLDON register

3.9 Analog Comparator

The S1C62N81 contains an analog comparator (CMP) the data of which can be read by software. This circuit can be turned on and off to save power. The CMPON bit controls analog comparator (CMP) power on/off. At initial reset, the CMP circuit is off. While the circuit is not in use, keep this bit set to "0" to save power.

The output data of the analog comparator appears in CMPDT, this bit is "1" when CMPP > CMPM, and "0" when CMPP < CMPM. If the CMPON bit is "0", the CMPDT bit is fixed at "1".

Analog comparator memory map

Table 3.9.1 I/O memory map

Address	Register								Comment
	D3	D2	D1	D0	Name	SR *1	1	0	
0FBH	CSDC	0	CMPDT	CMPON	CSDC	0	Static	Dynamic	LCD drive switch
	R/W	R		R/W	0 *5				
					CMPDT	1	+ > -	- > +	Comparator's voltage condition: 1 = CMPP(+)input > CMPM(-)input, 0 = CMPM(-)input > CMPP(+)input
					CMPON	0	On	Off	Analog comparator ON/OFF

*1 Initial value following initial reset
*2 Not set in the circuit
*3 Undefined
*4 Reset (0) immediately after being read
*5 Always 0 when being read
*6 Refer to main manual

Example of CMP control program (when fosc = 32.768 kHz)

```
Label   Mnemonic/operand        Comment
        LD      X,0FBH          ;Set CMP circuit address
        OR      MX,0001B        ;CMP circuit on
        LD      A,08H           ; ┐
LOOP:   ADD     A,01H           ; │ Wait about 3 ms
        JP      NZ,LOOP         ; ┘
        LD      A,MX            ;A register ← CMPDT
        AND     MX,1110B        ;CMP circuit off
```

Execution of the above program loads CMP output data CMPDT into D1 of the A register.

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

3.10 Melody Generator

Melody generator memory map

Table 3.10.1 I/O memory map

Address	Register								Comment
	D3	D2	D1	D0	Name	SR *1	1	0	
0ECH	0	0	0	IMEL	0 *5				
	R				0 *5				
					0 *4				
					IMEL *4	0	Yes	No	Interrupt factor flag (melody)
0F0H	MAD3	MAD2	MAD1	MAD0	MAD3	0	High	Low	Melody ROM address (AD3)
	R/W				MAD2	0	High	Low	Melody ROM address (AD2)
					MAD1	0	High	Low	Melody ROM address (AD1)
					MAD0	0	High	Low	Melody ROM address (AD0, LSB)
0F1H	0	MAD6	MAD5	MAD4	0 *5				
	R	R/W			MAD6	0	High	Low	Melody ROM address (AD6, MSB)
					MAD5	0	High	Low	Melody ROM address (AD5)
					MAD4	0	High	Low	Melody ROM address (AD4)
0F2H	CLKC1	CLKC0	TEMPC	MELC	CLKC1	0	High	Low	CLKC1(0)&CLKC0(0) : melody speed × 1 CLKC1(0)&CLKC0(1) : melody speed × 8
	R/W				CLKC0	0	High	Low	CLKC1(1)&CLKC0(0) : melody speed × 16 CLKC1(1)&CLKC0(1) : melody speed × 32
					TEMPC	0	High	Low	Tempo change control
					MELC	0	ON	OFF	Melody control ON/OFF

*1 Initial value following initial reset
*2 Not set in the circuit
*3 Undefined
*4 Reset (0) immediately after being read
*5 Always 0 when being read
*6 Refer to main manual

Address setting (Addresses 0F0H and 0F1H)

There are 7 bits for melody start address setting.

Fig. 3.10.1
Set of melody ROM address

0F1H				0F0H			
–	MSB	AD5	AD4	AD3	AD2	AD1	LSB
	↓	↓	↓	↓	↓	↓	↓
	MAD6	MAD5	MAD4	MAD3	MAD2	MAD1	MAD0

Note *The user programmable area is from 00H to 04FH (80 words).*

Play mode control

Address 0F2H (4 bits) is for melody control.

Description

MELC: (1) Melody start when this bit is set to "1".

(2) Melody stop when this bit is set to "0" and there is an end bit come from melody ROM.

TEMPC: Selection of tempo (TEMPC0 or TEMPC1); chosen by mask option. Two tempos (TEMPC0 and TEMPC1) can be chosen out of 16 tempos.

0: TEMPC0

1: TEMPC1

(See S1C62N81 Technical Hardware, 4.11, "Playing tempo".)

CLKC1, CLKC0: These two bits are combined to set the play speed.

Table 3.10.2
Set of play speed

CLKC1	CLKC0	Play Speed
0	0	Play as normal speed
0	1	Play as normal speed × 8
1	0	Play as normal speed × 16
1	1	Play as normal speed × 32

Play mode (1) One shot

In this mode, only one melody is played.
The control procedure is as follow:

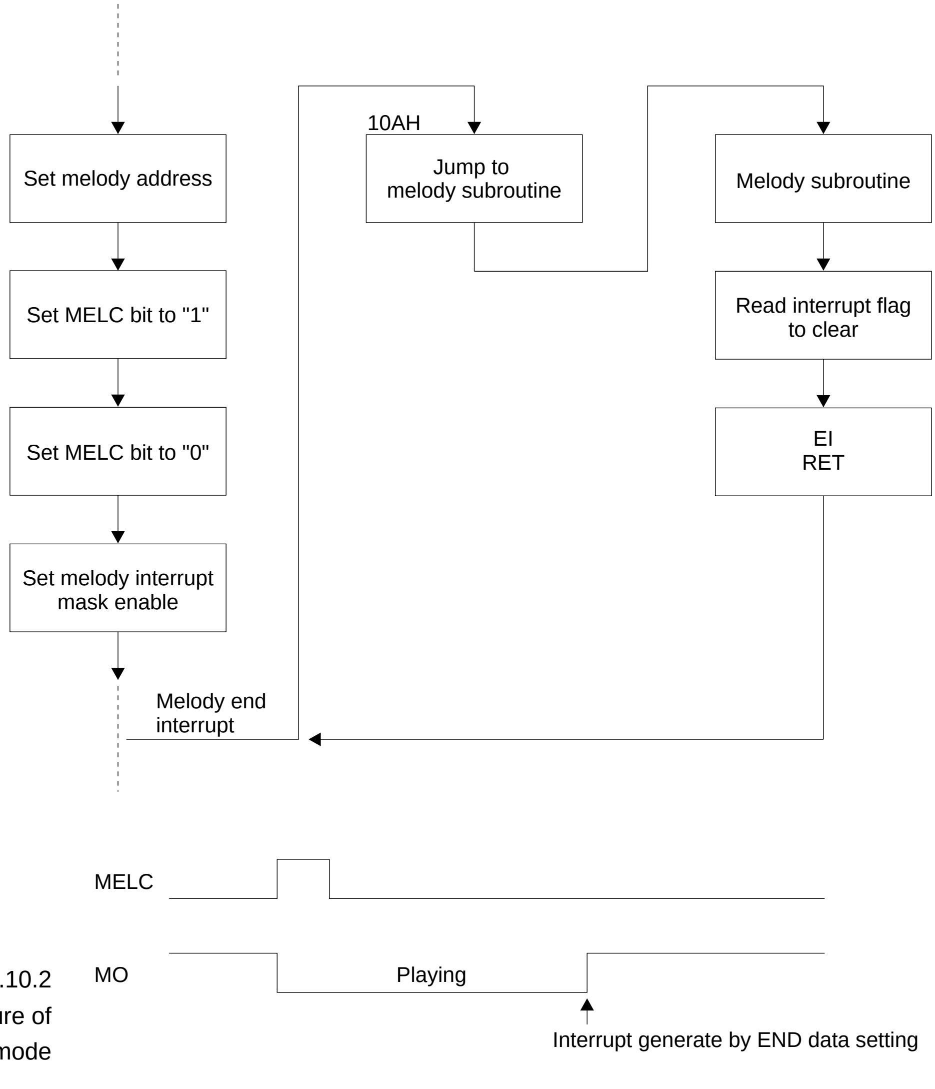

Set melody address
Set MELC bit to "1"
Set MELC bit to "0"
Set melody interrupt mask enable
Melody end interrupt
10AH
Jump to melody subroutine
Melody subroutine
Read interrupt flag to clear
EI
RET
MELC
MO
Playing
Interrupt generate by END data setting

Fig. 3.10.2
Control procedure of
one shot mode

When the MELC bit is set to "1", it makes the melody play. The user's program should set this bit to "0" before the end bit from the melody ROM. If not, the function will be like the level hold mode (see next function).

(2) Level hold

In this mode, after one melody has been played, the user can change the next play to any other melody. If there is no change, the melody is played repeatedly. The control procedure is as follows:

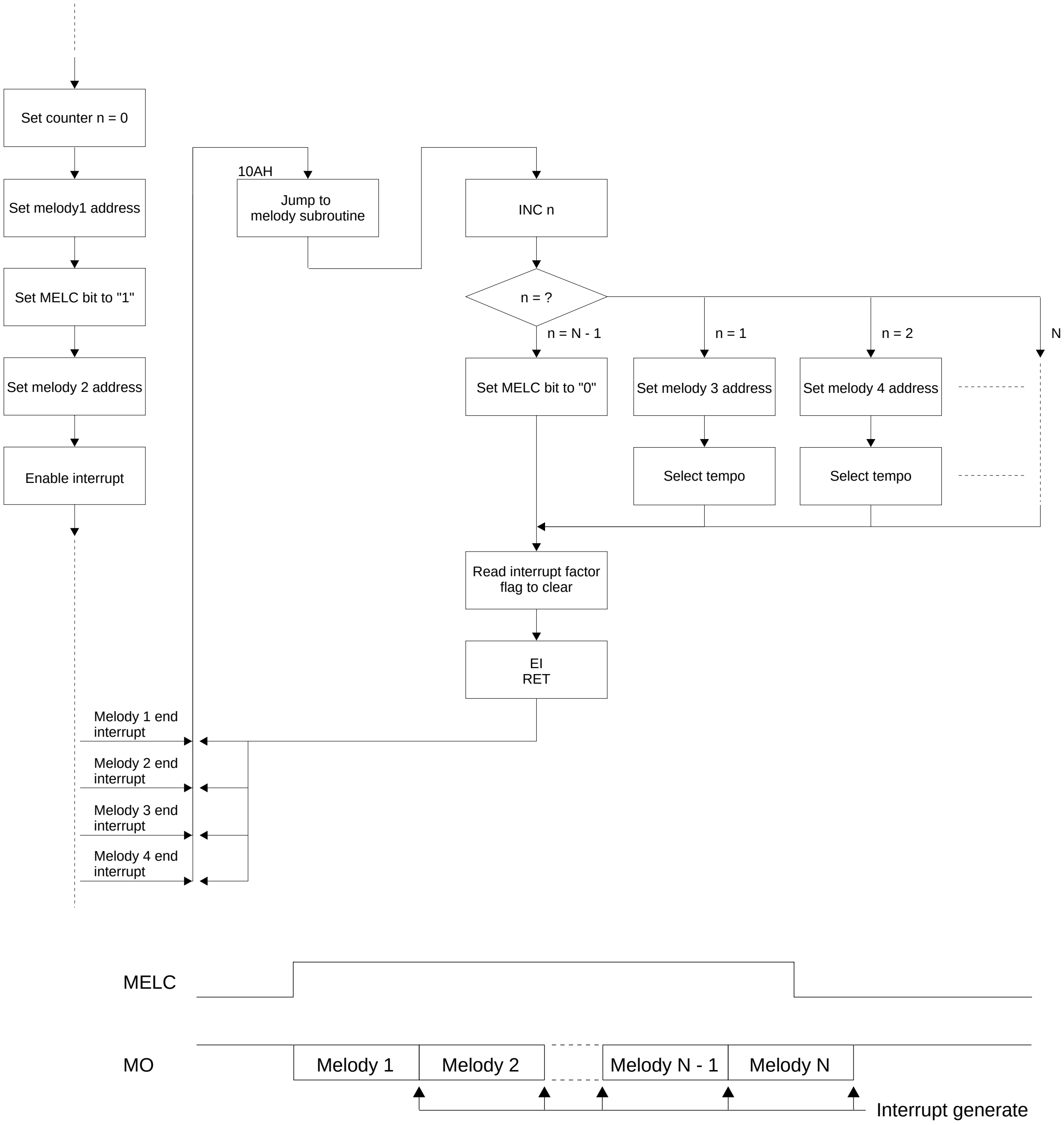


Fig. 3.10.3
Control procedure of level hold mode

(3) Retrigger play

In this mode, the melody can be stopped anywhere during playing, and it can be set to any another melody. The control procedure is as follows:

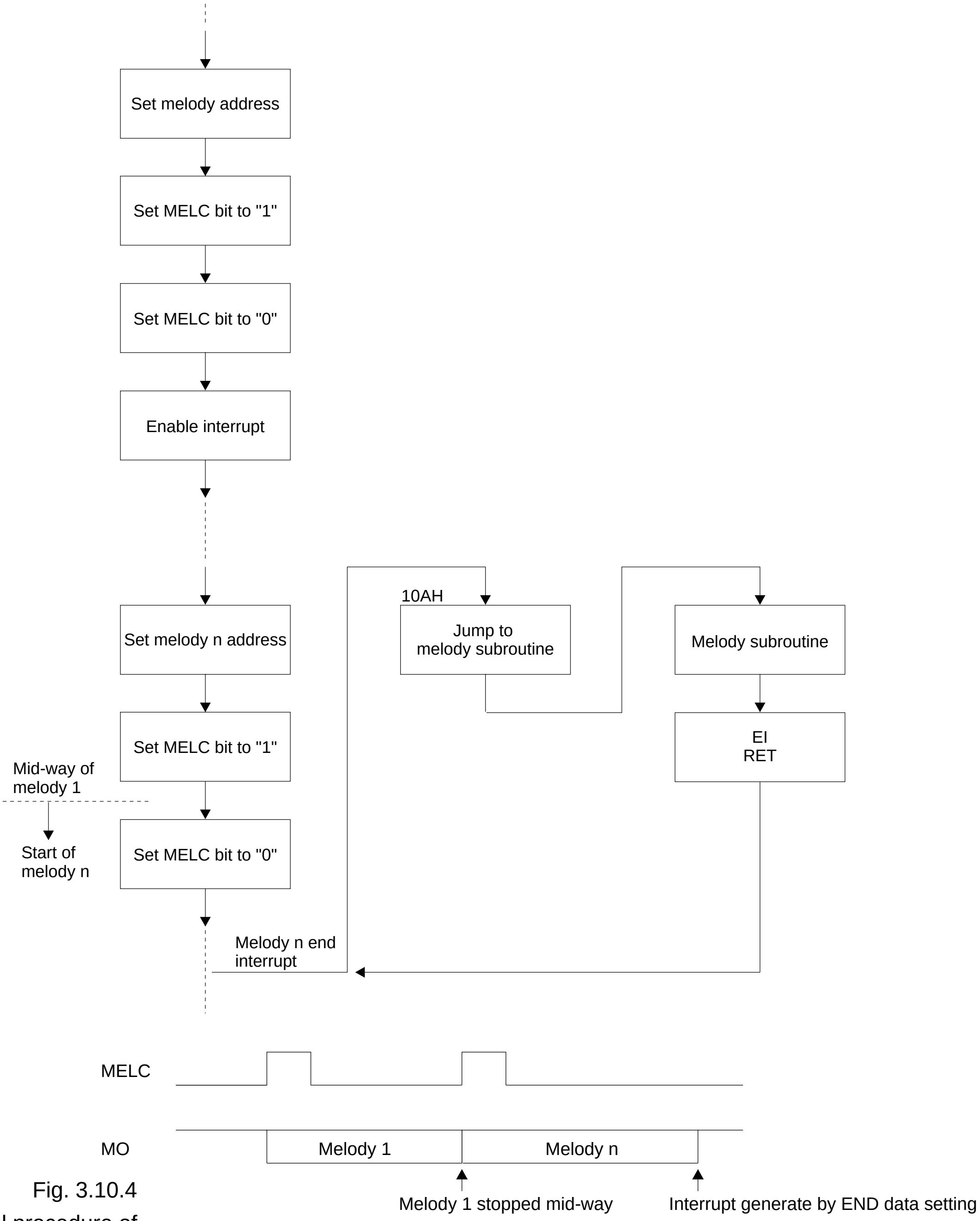


Fig. 3.10.4
Control procedure of retriggrer play mode

With this function, the user can force the melody to stop if there is a rest note with the End data = "1" in the melody ROM (See melody ROM data setting).

Tempo and speed control

(1) Tempo

Tempo selection is asigned to address 0F2H bit D1 (TEMPC).

This bit should be set at the same time that the MELC bit is set to "1". During playing, this bit will have no function for the melody playing. But in the level hold mode, when the next melody is loading, TEMPC will also be loaded. The tempo will then be changed. The control procedure is as follows:

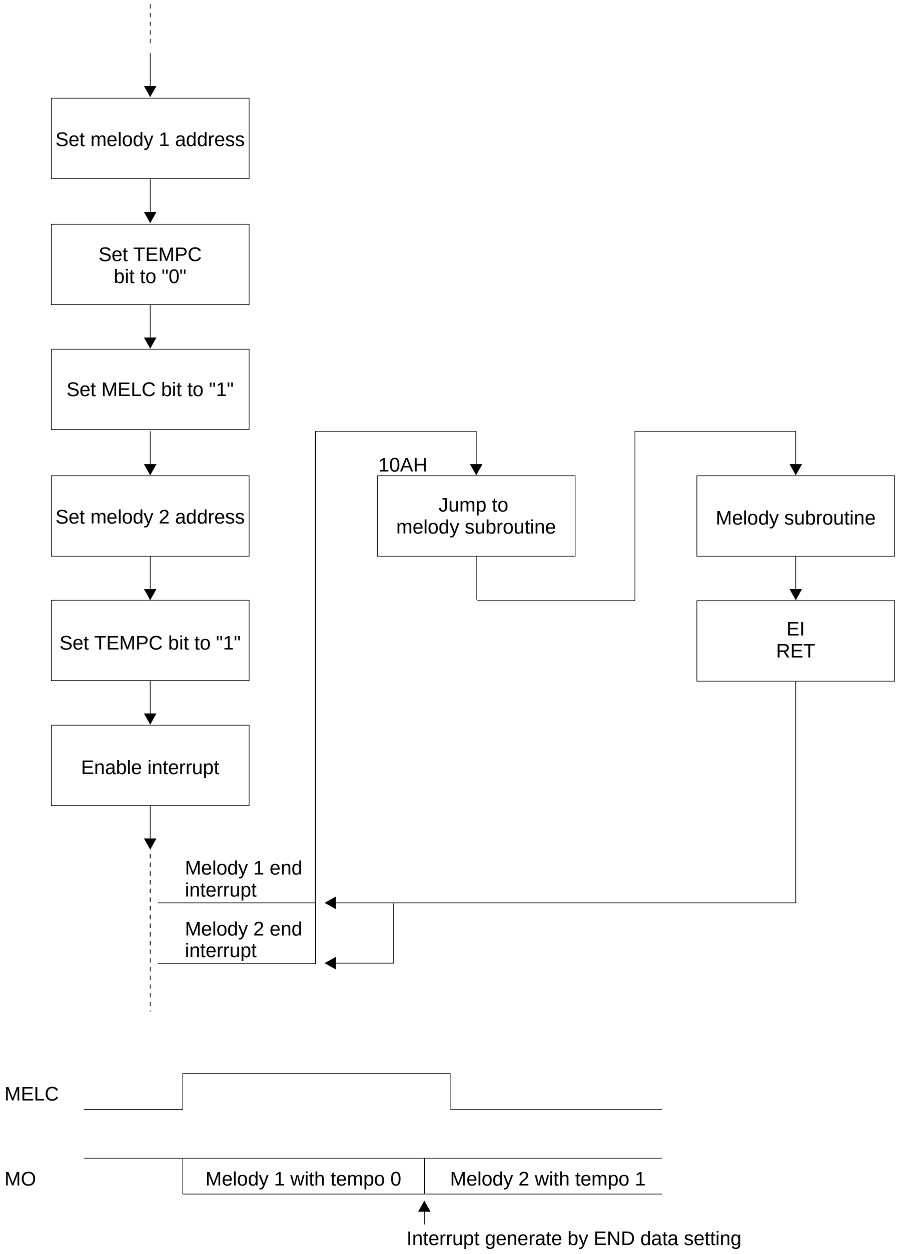


Fig. 3.10.5 Control procedure of tempo

(2) Speed

Speed control is asigned to address 0F2H, bits D2 and D3 (CLKC0 and CLKC1). These two bits are controlled independently. The user can change the speed during playing, or start with a different speed. The control procedure is as follows:

0F2H	D3	D2	D1	D0	
	0	0	0	1	→ Melody start with TEMPC0, speed normal
	0	1	0	1	→ Melody start with TEMPC0, speed × 8
	1	0	0	1	→ Melody start with TEMPC0, speed × 16
	1	1	0	1	→ Melody start with TEMPC0, speed × 32
	0	0	1	1	→ Melody start with TEMPC1, speed normal
	0	1	1	1	→ Melody start with TEMPC1, speed × 8
	1	0	1	1	→ Melody start with TEMPC1, speed × 16
	1	1	1	1	→ Melody start with TEMPC1, speed × 32

Example of changing speed during playing:

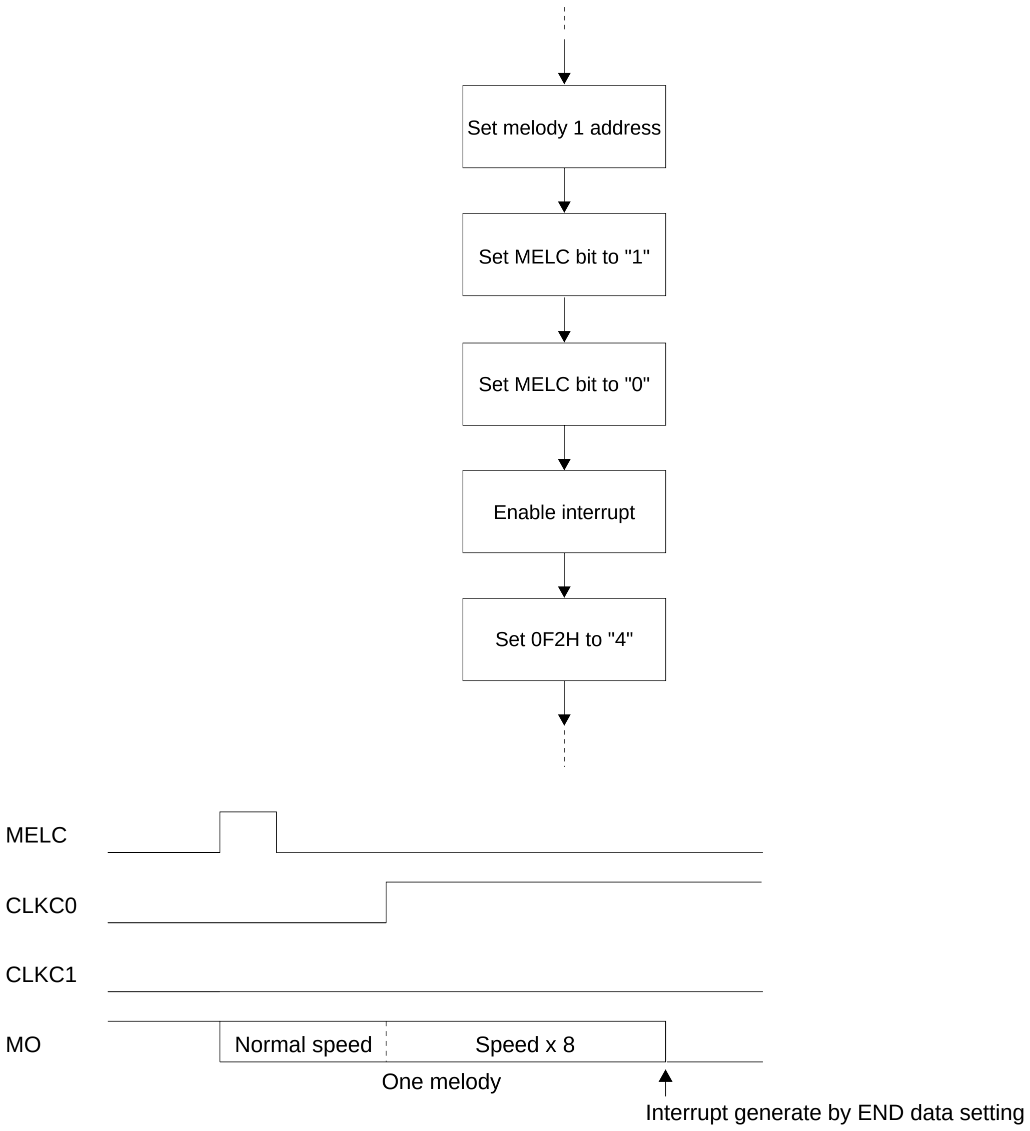


Fig. 3.10.6
Control procedure of
play speed

Melody interrupt

A melody interrupt occurs when the melody ROM data is read out with the end bit set to "1". This indicates the end of melody playing.

0E7H, D0: Interrupt mask bit

D0: 1 Enable interrupt at the end of melody play.

D0: 0 Interrupt cannot be generated even if play is ending.

0ECH, D0: Interrupt factor flag

This bit will be reset to "0" when the user reads it.

D0: 1 Interrupt has occured already, and program will jump to interrupt vector 10AH. Because the melody interrupt has the highest priority, the interrupt service will finish first, and this flag should be read to be cleared.

D0: 0 Interrupt has not been generated yet.

Melody ROM

Volume: 00H–4FH (80 words)
Word: 9 bits/word

Refer to data setting as below:

Table 3.10.3
Melody ROM data

D8	D7	D6	D5	D4	D3	D2	D1	D0
ATK data	Note data			Scale address data (Scale ROM address)				End data

D0: End Data

Melody play will stop after the note playing when this data is set to "1".

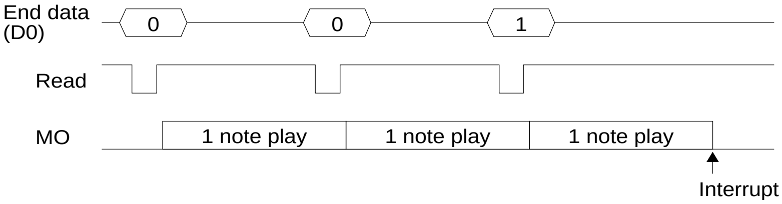


Fig. 3.10.7
End data

D1–D4: Scale Address Data (Scale ROM address)

What pitch is used depends on the address point of the scale ROM and the scale data contained. (See scale ROM data setting.)

D5–D7: Note Data

Note data table as below:

Table 3.10.4
Note data

D5	1	1	1	1	0	0	0	0
D6	1	1	0	0	1	1	0	0
D7	1	0	1	0	1	0	1	0
Note	𝅘𝅥𝅯	𝅘𝅥𝅮	𝅘𝅥𝅮.	𝅘𝅥	𝅘𝅥+𝅘𝅥𝅯	𝅘𝅥.	𝅘𝅥..	𝅗𝅥

D8: ATK Data

There will be a short break (≈12 ms) before the note playing if this data is set to "1". Usually, two notes of the same pitch are separated with this function, otherwise the two notes will play continuously without any break.

In each melody first word, set this data to "1". Otherwise, there will be no melody play even if the user starts play. Next, according to the user's definition it can set to "1" or "0". If the hardware mask option selects the R12 envelope function, this data also controls the note output by envelope.

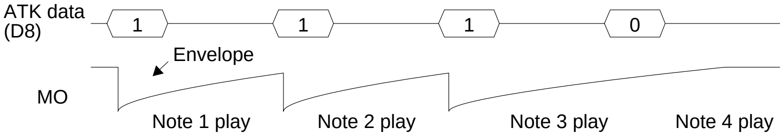


Fig. 3.10.8
Waveform of envelope

Scale ROM

Volume: 00H–0FH (16 words)
Word: 8 bits/word

Address 0FH is set to a rest note. The data contained is not connected with the scale. The scale may be selected according to the definition of the scale ROM address, which is defined by melody ROM data D4–D1. The scale data definition is as the table on the next page. The user has the choice of 15 types of scale from this table.

Melody ROM (D4–D1)	Scale ROM data		C major
00H	04H	→	C4 (Do)
01H	20H	→	D4 (Re)
02H	3BH	→	E4 (Mi)
03H	44H	→	F4 (Fa)
:	:		
0EH	C4H	→	C6 (Do)
0FH	C4H	→	Rest

Examples of melody control program

For level hold

```
Label   Mnemonic/operand        Comment
200     LD      A,00H           ;Set counter (melody point)
        LD      M0,A
        LD      X,0F0H          ;Set first melody address (00)
        LD      MX,00H
        INC     X
        LD      MX,00H
        LD      Y,0F2H          ;Start melody with TEMPC0
        LD      MY,01H
        LD      X,0F0H          ;Set second melody address (06)
        LD      MX06H
        INC     X
        LD      MX,00H
        LD      Y,0E7H          ;Enable melody interrupt mask
        LD      MY,01H
        EI                      ;Enable interrupt
        :
        :

10A     PSET    004H
        JP      000H

400     PUSH    XL
        PUSH    XH
        PUSH    YL
        PUSH    YH
        PUSH    A
        INC     M0              ;Melody pointer increment
        LD      A,M0            ;Decide which melody
        CP      A,01H
        JP      Z,MELDY3
        CP      A,02H
        JP      Z,MELDY4
        CP      A,03H
        JP      Z,MELDY5
        CP      A,04H
        JP      Z,MELDY6
        CP      A,05H
        JP      Z,MELSTP
        JP      MELEND
MELDY3  LD      X,0F0H        ;Set MEL3 address (0A)
        LD      MX,0AH
        INC     X
        LD      MX,00H
        JP      MELSTP
MELDY4  LD      X,0F0H        ;Set MEL4 address (12)
        LD      MX,02H
        INC     X
        LD      MX,01H
        JP      MELSTP
MELDY5  LD      X,0F0H        ;Set MEL5 address (28)
        LD      MX,08H
        INC     X
        LD      MX,02H
        LD      Y,0F2H        ;Set TEMPC1 for MEL6
        LD      MY,03H
        JP      MELSTP
MELDY6  LD      X,0F0H        ;Set MEL6 address (30)
        LD      MX,00H
        INC     X
        LD      MX,03H
MELSTP  LD      Y,0F2H        ;Melody stop after end
        LD      MY,00H
MELEND  LD      Y,0ECH        ;Read clear interrupt factor flag
        LD      A,MY
        POP     A
        POP     YH
        POP     YL
        POP     XH
        POP     XL
        EI
        RET
```

For one shot

```
Label    Mnemonic/operand         Comment
---------------------------------------------------------------------
         :
         LD     X,0F0H            ;Set melody address
         LD     MX,00H
         INC    X
         LD     MX,00H
         LD     Y,0F2H            ;Set melody start
         LD     MY,01H
         LD     MY,00H            ;Set MELC to "0"
         LD     X,0E7H            ;Enable melody interrupt mask
         LD     MX,01H
         EI                       ;Enable interrupt
         :
---------------------------------------------------------------------
```

For retrigger

```
Label    Mnemonic/operand         Comment
---------------------------------------------------------------------
         :
         LD     X,0F0H            ;Set melody 1 address
         LD     MX,00H
         INC    X
         LD     MX,00H
         LD     Y,0F2H            ;Set melody start
         LD     MY,01H
         LD     MY,00H            ;Set MELC to "0"     ← Start of melody 1
         LD     X,0E7H            ;Enable melody
         LD     MX,01H            ;Interrupt mask
         EI                       ;Enable interrupt
         :
         :
         LD     X,0F0H            ;Set melody n address
         LD     MX,04H
         INC    X
         LD     MX,02H
         LD     Y,0F2H            ;Retrigger melody with
         LD     MY,07H            ;TEMPC1, speed × 8 ← Mid-way through melody 1
         LD     MY,06H            ;Set MELC to "0"     ← Start of melody n
         :
         :
---------------------------------------------------------------------
```

3.11 Interrupt and Halt

Interrupt memory map

Table 3.11.1 (a) I/O memory map

Address	Register								Comment
	D3	D2	D1	D0	Name	SR *1	1	0	
0E5H	KCP03	KCP02	KCP01	KCP00	KCP03	0	Falling	Rising	Input comparison register (K03)
	R/W				KCP02	0	Falling	Rising	Input comparison register (K02)
					KCP01	0	Falling	Rising	Input comparison register (K01)
					KCP00	0	Falling	Rising	Input comparison register (K00)
0E6H	0	0	0	KCP10	0 *5				
	R			R/W	0 *5				
					0 *5				
					KCP10	0	Falling	Rising	Input comparison register (K10)
0E7H	0	0	0	EIMEL	0 *5				
	R			R/W	0 *5				
					0 *5				
					EIMEL	0	Enable	Mask	Interrupt mask register (melody)
0E8H	EIK03	EIK02	EIK01	EIK00	EIK03	0	Enable	Mask	Interrupt mask register (K03)
	R/W				EIK02	0	Enable	Mask	Interrupt mask register (K02)
					EIK01	0	Enable	Mask	Interrupt mask register (K01)
					EIK00	0	Enable	Mask	Interrupt mask register (K00)

*1 Initial value following initial reset
*2 Not set in the circuit
*3 Undefined
*4 Reset (0) immediately after being read
*5 Always 0 when being read
*6 Refer to main manual

Table 3.11.1 (b) I/O memory map

Address	Register								Comment
	D3	D2	D1	D0	Name	SR *1	1	0	
0E9H	0	0	0	EIK10	0 *5				
	R			R/W	0 *5				
					0 *5				
					EIK10	0	Enable	Mask	Interrupt mask register (K10)
0EAH	0	0	EISW1	EISW0	0 *5				
	R		R/W		0 *5				
					EISW1	0	Enable	Mask	Interrupt mask register (stopwatch 1 Hz)
					EISW0	0	Enable	Mask	Interrupt mask register (stopwatch 10 Hz)
0EBH	0	EIT2	EIT8	EIT32	0 *5				
	R	R/W			EIT2	0	Enable	Mask	Interrupt mask register (clock timer 2 Hz)
					EIT8	0	Enable	Mask	Interrupt mask register (clock timer 8 Hz)
					EIT32	0	Enable	Mask	Interrupt mask register (clock timer 32 Hz)
0ECH	0	0	0	IMEL	0 *5				
	R				0 *5				
					0 *4				
					IMEL *4	0	Yes	No	Interrupt factor flag (melody)

*1 Initial value following initial reset

*2 Not set in the circuit

*3 Undefined

*4 Reset (0) immediately after being read

*5 Always 0 when being read

*6 Refer to main manual

Table 3.11.1 (c) I/O memory map

Address	Register								Comment
	D3	D2	D1	D0	Name	SR *1	1	0	
0EDH	0	0	IK1	IK0	0 *5				
	R				0 *5				
					IK1 *4	0	Yes	No	Interrupt factor flag (K10)
					IK0 *4	0	Yes	No	Interrupt factor flag (K00–K03)
0EEH	0	0	ISW1	ISW0	0 *5				
	R				0 *5				
					ISW1*4	0	Yes	No	Interrupt factor flag (stopwatch 1 Hz)
					ISW0*4	0	Yes	No	Interrupt factor flag (stopwatch 10 Hz)
0EFH	0	IT2	IT8	IT32	0 *5				
	R				IT2 *4	0	Yes	No	Interrupt factor flag (clock timer 2 Hz)
					IT8 *4	0	Yes	No	Interrupt factor flag (clock timer 8 Hz)
					IT32 *4	0	Yes	No	Interrupt factor flag (clock timer 32 Hz)

*1 Initial value following initial reset
*2 Not set in the circuit
*3 Undefined
*4 Reset (0) immediately after being read
*5 Always 0 when being read
*6 Refer to main manual

Control of interrupts and halt

The S1C62N81 supports four types of a total of 11 interrupts. There are three timer interrupts (2 Hz, 8 Hz, 32 Hz), two stopwatch interrupts (1 Hz, 10 Hz), five input interrupts (K00–K03, K10) and one melody interrupt.

The 11 interrupts are individually enabled or masked (disabled) by interrupt mask registers. The EI and DI instructions can be used to set or reset the interrupt flag (I), which enables or disables all the interrupts at the same time.

Individual vector addresses are assigned to the four types of interrupt. The priority of the interrupts is determined by the hardware. The priority of the 2 Hz, 8 Hz, and 32 Hz timer interrupts where the vector address is the same is determined by the software. The priority of the stopwatch interrupts between 1 Hz and 10 Hz is also determined by software.

When an interrupt is accepted, the interrupt flag (I) is reset, and cannot accepts any other interrupts (DI state).

Restart from the halt state created by the HALT instruction, is done by interrupt.

• Interrupt factor flags

IK0 This flag is set when any of the K00 to K03 input interrupts occurs. The interrupt factor flag (IK0) is set to "1" when the contents of the input (K00–K03) and the input comparison register (KCP00–KCP03) do not match and the data of the corresponding interrupt mask register (EIK00–EIK03) is "1". The contents of the IK0 flag can be loaded by software to determine whether the K00–K03 input interrupts have occured.

The flag is reset when loaded by software. (See Figure 3.11.1.)

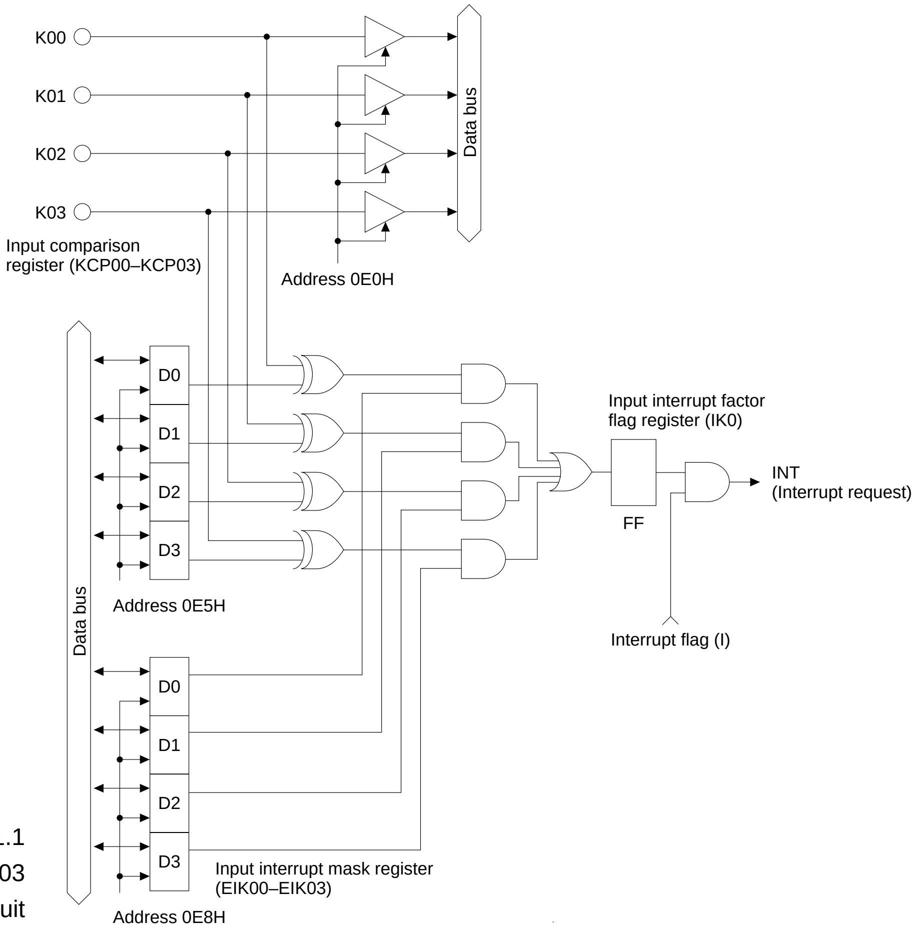

K00
K01
K02
K03
Data bus
Input comparison register (KCP00–KCP03)
Address 0E0H
D0
D1
D2
D3
Address 0E5H
Data bus
D0
D1
D2
D3
Input interrupt mask register (EIK00–EIK03)
Address 0E8H
Input interrupt factor flag register (IK0)
FF
INT (Interrupt request)
Interrupt flag (I)

Fig. 3.11.1
K00–K03
input interrupt circuit

IK1 This flag is set when the K10 input interrupt occurs.

The interrupt factor flag (IK1) is set to "1" when the contents of the input (K10) and the interrupt differential register (KCP10) do not match, and the corresponding interrupt mask register (EIK10) is "1".
The contents of the IK1 flag can be loaded by software to determine whether K10 input interrupt has occured.

The flag is reset when loaded by software. (See Figure 3.11.2.)

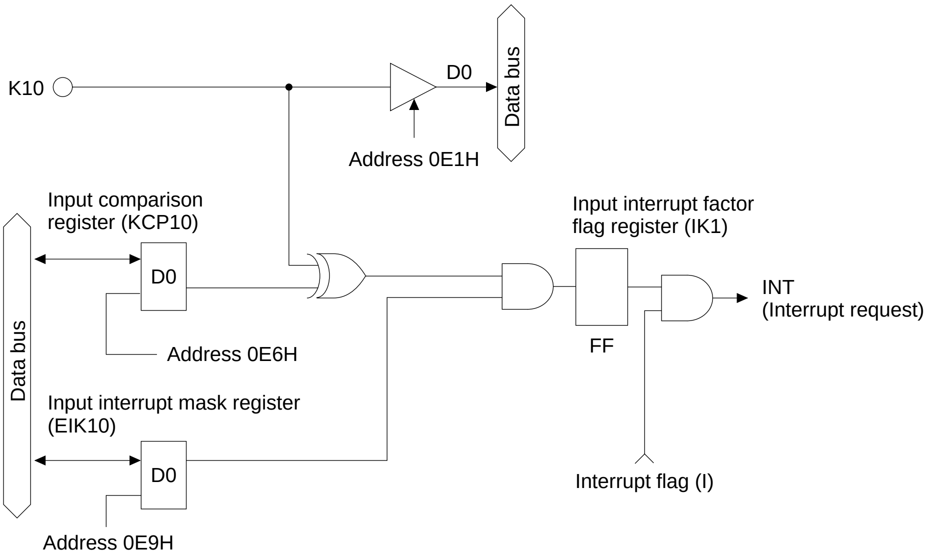

K10
D0
Data bus
Address 0E1H
Input comparison register (KCP10)
D0
Address 0E6H
Data bus
Input interrupt mask register (EIK10)
D0
Address 0E9H
Input interrupt factor flag register (IK1)
FF
INT
(Interrupt request)
Interrupt flag (I)

Fig. 3.11.2
K10 input interrupt circuit

IT32 This flag is set to "1" when a falling edge is detected in the timer TM1 (32 Hz) signal.
The contents of the IT32 flag can be loaded by software to determine whether a 32 Hz timer interrupt has occured.

The flag is reset, when it is loaded by software. (See Figure 3.11.3.)

IT8 This flag is set to "1" when a falling edge is detected in the timer TM1 (8 Hz) signal.
The contents of the IT8 flag can be loaded by software to determine whether an 8 Hz timer interrupt has occured.

The flag is reset, when it is loaded by software. (See Figure 3.11.3.)

IT2 This flag is set to "1" when a falling edge is detected in the timer TM1 (2 Hz) signal.
The contents of the IT2 flag can be loaded by software to determine whether a 2 Hz timer interrupt has occured.

The flag is reset, when it is loaded by software. (See Figure 3.11.3.)

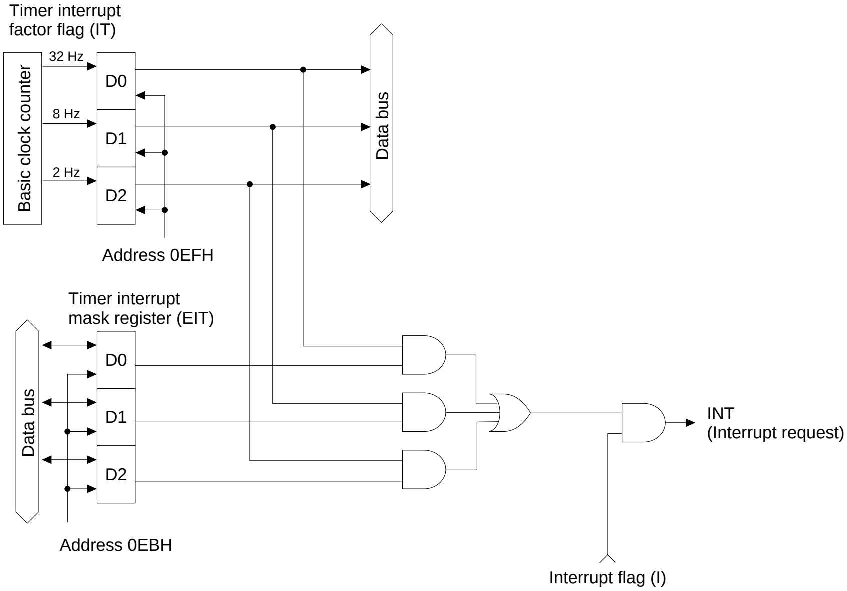


Fig. 3.11.3
Timer interrupt circuit

ISW1 This flag is set to "1" when a falling edge is detected in the stopwatch timer (SWH, 1 Hz).
The contents of the ISW1 flag can be loaded by software to determine whether a 1 Hz stopwatch interrupt has occured.

The flag is reset, when it is loaded by software. (See Figure 3.11.4.)

ISW0 This flag is set to "1" when a falling edge is detected in the stopwatch timer (SWH, 10 Hz).
The contents of the ISW0 flag can be loaded by software to determine whether a 10 Hz stopwatch interrupt has occured.

The flag is reset, when it is loaded by software. (See Figure 3.11.4.)

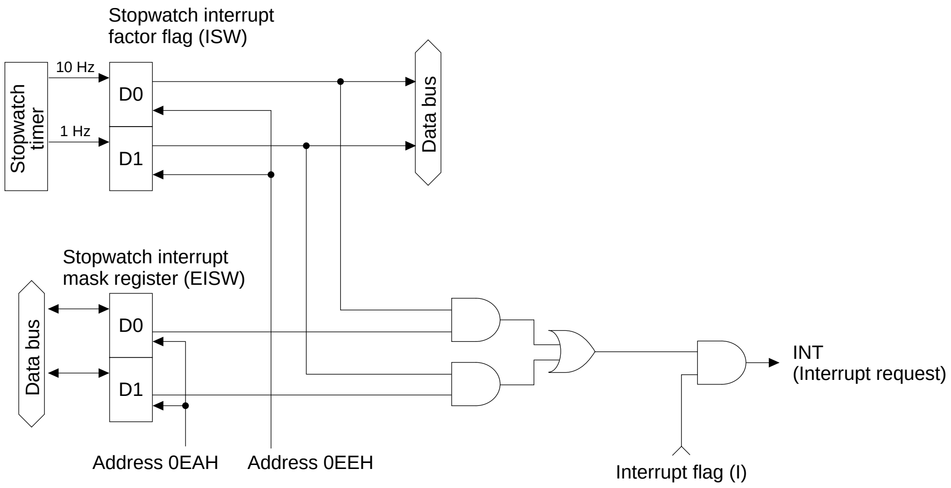


Fig. 3.11.4
Stopwatch interrupt circuit

Note The interrupt factor flags must always be loaded under the DI state (interrupt flag [I] = "0"). Reading under the EI state (interrupt flag [I] = "1") may cause an operation error.

• Interrupt mask registers

The interrupt mask registers are registers that individually specify whether to enable or mask the timer interrupt (2 Hz, 8 Hz, 32 Hz), stopwatch timer interrupt (1 Hz, 10 Hz), or input interrupt (K00–K03, K10).

The following are descriptions of the interrupt mask registers.

EIK00 to EIK03 This register enables or masks the K00–K03 input interrupt. The interrupt condition flag (IK0) is set to "1" when the contents of the input (K00–K03) and the interrupt differential register (KCP00–KCP03) do not match and the data of the corresponding interrupt mask register (EIK00–EIK03) is "1". The CPU is interrupted if it is in the EI state (interrupt flag [I] = "1"). (See Figure 3.11.1.)

EIK10 This register enables or masks the K10 input interrupt. The interrupt condition flag (IK1) is set to "1" when the contents of the input (K10) and the interrupt differential register (KCP10) do not match and the data of the corresponding interrupt mask register (EIK10) is "1". The CPU is interrupted if it is in the EI state (interrupt flag [I] = "0"). (See Figure 3.11.2.)

EIT32 This register enables or masks the 32 Hz timer interrupt. The CPU is interrupted if it is in the EI state when the interrupt mask register (EIT32) is set to "1" and the interrupt condition flag (IT32) is "1". (See Figure 3.11.3.)

EIT8 This register enables or masks the 8 Hz timer interrupt. The CPU is interrupted if it is in the EI state when the interrupt mask register (EIT8) is set to "1" and the interrupt condition flag (IT8) is "1". (See Figure 3.11.3.)

EIT2 This register enables or masks the 2 Hz timer interrupt. The CPU is intterrupted if it is in the EI state when the interrupt mask register (EIT2) is set to "1" and the interrupt condition flag (IT2) is "1". (See Figure 3.11.3.)

EISW1 This register enables or masks the 1 Hz stopwatch interrupt. The CPU is interrupted if it is in the EI state when the interrupt mask register (EISW1) is set to "1", and also the interrupt condition flag (ISW1) is "1". (See Figure 3.11.4.)

EISW0 This register enables or masks the 10 Hz stopwatch interrupt. The CPU is interrupted if it is in the EI state when the interrupt mask register (EISW0) is set to "1", and the interrupt condition flag (ISW0) is "1". (See Figure 3.11.4.)

Note Write to the interrupt mask registers (EIT32, EIT8, EIT2) and read the interrupt factor flags (IT32, IT8, IT2) in DI states only (interrupt flag [I] = "0").

• Interrupt control registers

KCP00 to KCP03 The data of the input comparison registers (KCP00–KCP03) is compared with the data of the corresponding input ports (K00–K03). If the data does not match and the corresponding input mask register (EIK00–EIK03) is "1", the interrupt factor flag (IK0) is set to "1".

These registers are used to determine the change in the input (K01–K03) level. (See Figure 3.11.1.)

KCP10 The data of the input comparison register (KCP10) is compared with the data of the corresponding input port (K10). If the data does not match and the corresponding input mask register (EIK10) is "1", the interrupt factor flag (IK1) is set to "1".

This register is used to determine the change in the input (K10) level. (See Figure 3.11.2.)

The input comparison register can effectively be used to determine the on/off state of the input.

However, as shown in Figure 3.11.1, the result of comparison of the input (K00–K03) is collected in the interrupt factor flag (IK0), so the input comparison register cannot be used to determine the on/off state of the key matrix.

• Interrupt vector address

The S1C62N81 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 Table 3.11.2.

Table 3.11.2 Assignment of the interrupt vector address

Interrupt Item	PCP3	PCP2	PCP1	PCP0	PCS7	PCS6	PCS5	PCS4	PCS3	PCS2	PCS1	PCS0	Interrupt Vector Address	Priority
Melody	0	0	0	1	0	0	0	0	1	0	1	0	10A	Highest
K10	0	0	0	1	0	0	0	0	1	0	0	0	108	
K03–K00	0	0	0	1	0	0	0	0	0	1	1	0	106	
Stopwatch	0	0	0	1	0	0	0	0	0	1	0	0	104	
Timer	0	0	0	1	0	0	0	0	0	0	1	0	102	Lowest

As shown in Table 3.11.2, the lower order 4 bits of the program counter are set according to which of the interrupts occurs. In other words, the interrupt vector address is set at page 1, steps 02H, 04H, 06H, 08H, 0AH.

Note that 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. For instance, when the 32 Hz timer interrupt and the 8 Hz timer interrupt are enabled at the same time, the accepted timer interrupt must be identified by software. (Similarly, the K00–K03 input interrupts and the 10 Hz/1 Hz stopwatch interrupts must be identified by software.)

When an interrupt is generated, the hardware resets the interrupt flag (I) to enter the DI state. Execute the EI instruction as necessary to recover the EI state after interrupt processing.

Set the EI state at the start of the interrupt processing routine to allow nesting of the interrupts. Then the priority of the interrupt or the nesting level is determined and set by hardware.

The interrupt factor flags must always be reset before setting the EI status in the corresponding interrupt processing routine. (The flag is reset when the interrupt condition flag is read by software.)

If the EI instruction is executed without resetting the interrupt factor flag after generating the timer interrupt or the stopwatch timer interrupt or melody, and if the corresponding interrupt mask register is still "1", the same interrupt is generated once more. (See Figure 3.11.5.)

If the EI state is set without resetting the interrupt condition flag after generating the input interrupt (K00–K03, K10), the same interrupt is generated once more. (See Figure 3.11.5.)

The interrupt factor flag must always be read (reset) in the DI state (interrupt flag [I] = "0"). There may be an operation error if read in the EI state.

The timer interrupt factor flags (IT32, IT8, IT2) and the stopwatch interrupt factor flags (ISW1, ISW0) are set whether the corresponding interrupt mask register is set or not.

The input interrupt factor flags (IK0, IK1) are allowed to be set in the condition when the corresponding interrupt mask register (EIK00–EIK03, EIK10) is set to "1" (interrupt is enabled). (See Figure 3.11.5.)

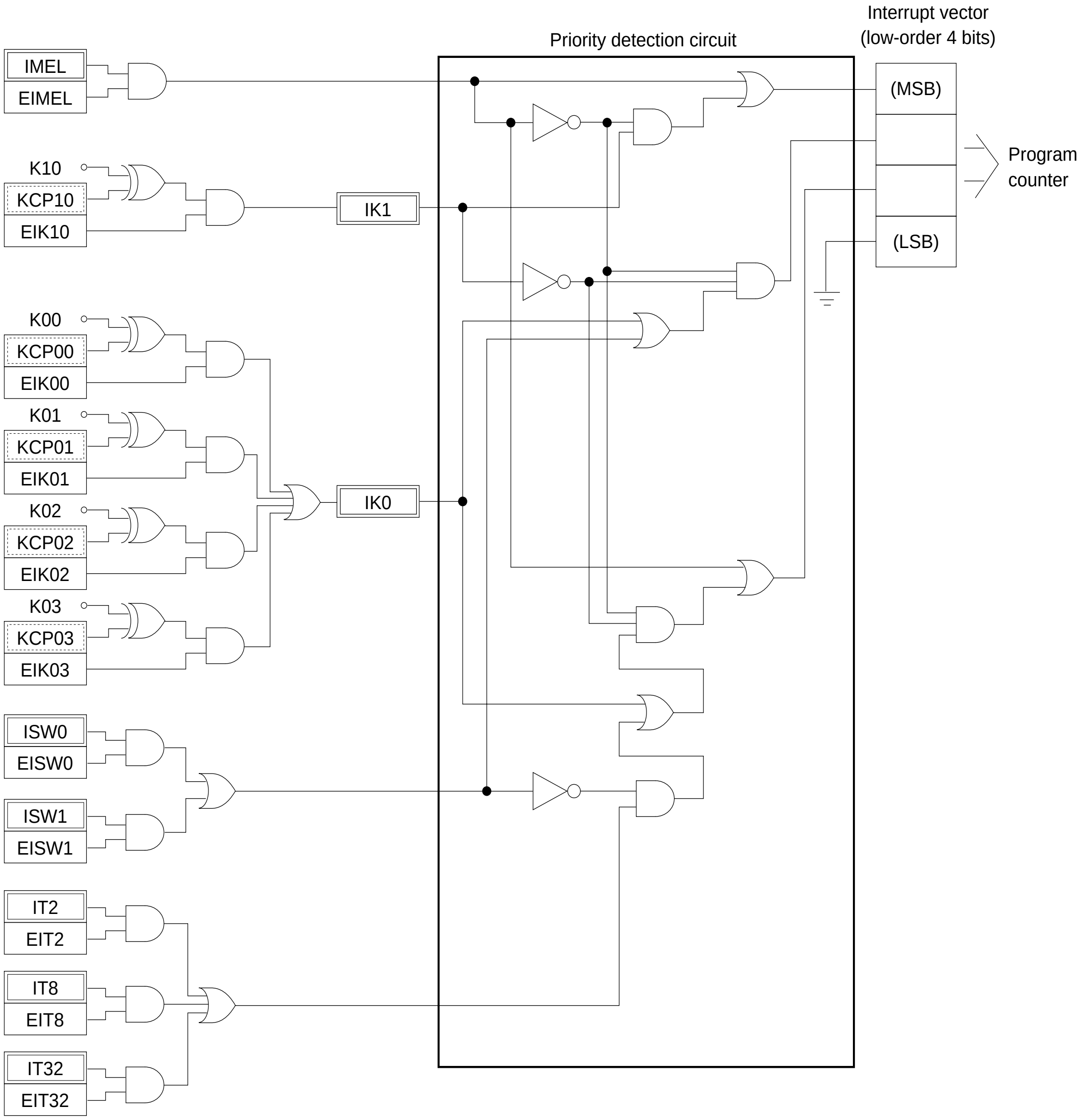


Fig. 3.11.5
Internal interrupt circuit

Examples of interrupt and halt control program

• Restart from halt state by interrupt

Main routine

```
Label   Mnemonic/operand        Comment
        LD      X,0E8H          ;Set address of K00 to K03
                                ;interrupt mask register
        OR      MX,1111B        ;Enable K00 to K03
                                ;input interrupt
;
        LD      X,0EAH          ;Set address of stopwatch
                                ;interrupt mask register
        OR      MX,0010B        ;Enable 1 Hz stopwatch interrupt
;
        LD      X,0EBH          ;Set address of timer interrupt
                                ;mask register
        OR      MX,0111B        ;Enable timer interrupt
                                ;(32 Hz, 8 Hz, 2 Hz)
        LD      X,E7H           ;Set address of melody interrupt
                                ;mask register
        OR      MX,0001B        ;Enable melody interrupt
MAIN:   EI                      ;Set interrupt flag (EI state is set)
        HALT                    ;Halt mode
        JP      MAIN            ;Jump to MAIN
```

Interruption vector routine

```
Label   Mnemonic/operand        Comment
        ORG     100H
        JP      INIT            ;Jump to initial routine
        HALT
        JP      TIINT           ;Jump to timer interrupt routine
        HALT
        JP      SWINT           ;Jump to stopwatch interrupt routine
        HALT
        JP      K0INT           ;Jump to K0 input interrupt routine
        HALT
        JP      K1INT           ;Jump to K1 input interrupt routine
        HALT
        JP      MELINT          ;Jump to melody interrupt routine
MELINT  LD      Y,0ECH          ;Address of melody interrupt
                                ;factor flag
        LD      A,MY            ;Reset melody interrupt
                                ;factor flag
RETURN  EI
        RET
K1INT   LD      Y,0EDH          ;Address of K10 input port interrupt
                                ;factor flag
        LD      A,MY            ;Reset K10 input port interrupt
                                ;factor flag
        JP      RETURN
K0INT   LD      Y,0EDH          ;Address of K0n input port interrupt
                                ;factor flag
        LD      A,MY            ;Reset K0n input port interrupt
                                ;factor flag
        JP      RETURN
SWINT   LD      Y,0EEH          ;Address of stopwatch interrupt
                                ;factor flag
        LD      X,SWFSTK        ;Address of stopwatch interrupt
                                ;factor flag buffer
        LD      MX,MY           ;Store stopwatch interrupt
                                ;factor flag in buffer
        FAN     MX,0010B        ;Check stopwatch 1 Hz
                                ;factor flag
        JP      Z,SW10RQ        ;Jump if not the 1 Hz request
                                ;interrupt
        CALL    SW1IN           ;Stopwatch 1 Hz interrupt
                                ;service routine
SW10RQ  LD      X,SWFSTK        ;Address of stopwatch interrupt
                                ;factor flag buffer
        FAN     MX,0001B        ;Check stopwatch 10 Hz
                                ;factor flag
        JP      Z,RETURN        ;Return
        CALL    SW10IN          ;Stopwatch 10 Hz interrupt
                                ;service routine
        JP      RETURN
TIINT   LD      Y,0EFH          ;Address of timer interrupt
                                ;factor flag
        LD      X,TMFSK         ;Address of timer interrupt
                                ;factor flag buffer
        LD      MX,MY           ;Store timer interrupt factor
                                ;flag in buffer
        FAN     MX,0100B        ;Check 2 Hz timer interrupt
                                ;factor flag
        CALL    TINT2           ;Call 2 Hz timer interrupt
                                ;service routine
        JP      RETURN          ;Return
TI8RQ   LD      X,TMFSK         ;Address of timer interrupt factor
                                ;flag buffer
        FAN     MX,0010B        ;Check 8 Hz timer interrupt
                                ;factor flag
        JP      Z,TI32RQ        ;Don't request interrupt
        CALL    TINT8           ;Call 8 Hz timer interrupt
                                ;service routine
TI32RQ  LD      X,TMFSK         ;Address of timer interrupt factor
                                ;flag buffer
        FAN     MX,0001B        ;Check 32 Hz timer interrupt
                                ;factor flag
        JP      Z,RETURN        ;Don't request interrupt
        CALL    TINT32          ;Call 32 Hz timer interrupt
                                ;service routine
        JP      RETURN
```

The above program is normally used to restart the CPU when in the halt state by interrupt and to return it to the halt state again after the interrupt processing is completed. The processing proceeds by repeating the → halt interrupt → halt → interrupt cycle.

All interrupts are enabled, and the priority when all interrupts are generated simultaneously is determined by hardware as follows:

(highest priority) Melody interrupt → K10 interrupt → K00–K03 interrupt → stopwatch interrupt → timer interrupt (lowest priority)

The two stopwatch interrupts (1 Hz, 10 Hz) have the same vector address (104H). The priority is decided by software; the stopwatch interrupt service routine first checks the 1 Hz interrupt factor flag, so the priority is (high priority) stopwatch 1 Hz interrupt → stopwatch 10 Hz interrupt (low priority).

The three timer interrupts (2 Hz, 8 Hz, 32 Hz) have the same vector address (102H). The priority is decided by software; the timer interrupt service routine first checks the 2 Hz interrupt factor flag, then 8 Hz, and finally 32 Hz, so the priority is (first priority) timer 2 Hz interrupt → (second priority) timer 8 Hz interrupt → (third priority) timer 32 Hz interrupt.

Always load (reset) the interrupt factor flags in the DI state (interrupt flag [I] = "0"). There may be an operation error if loaded in the EI state.

CHAPTER 4 SUMMARY OF PROGRAMMING POINTS

- **Core CPU**

After the system reset, only the program counter (PC), new page pointer (NPP) and interrupt flag (I) are initialized by the hardware. The other internal circuits whose settings are undefined must be initialized with the program.

- **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.

- **Input Port**

When modifying the input port from high level to low level with pull-down resistance, a delay will occur at the rise of the waveform due to time constant of the pull-down resistance and input gate capacities. Provide appropriate waiting time in the program when performing input port reading.

- **LCD Driver**

– Because the display memory is for writing only, re-writing the contents with computing instructions (e.G., AND, OR, etc.) which come with read-out operations is not possible. To perform bit operations, a buffer to hold the display data is required on the RAM.

– Even when 1/3 duty is selected, the display data corresponding to COM3 is valid for static drive. Hence, for static drive set the same value to all display memory corresponding COM0–COM3.

– For cadence adjustment, set the display data including display data corresponding to COM3.

– fosc indicates the oscillation frequency of the oscillation circuit.

- **Interrupt**

  - Even when the contents of the input data and input comparator register change from an unmatched state to another unmatched state or to a matched state, no interrupt will occur.
  - Re-start from the HALT state is performed by the interrupt. The return address after completion of the interrupt processing in this case will be the address following the HALT instruction.
  - When interrupt occurs, the interrupt flag will be reset by the hardware and it will become DI state. After completion of the interrupt processing, set to the EI state through the software as needed.
    Moreover, the nesting level may be set to be programmable by setting to the EI state at the beginning of the interrupt processing routine.
  - Be sure to reset the interrupt factor flag before setting to the EI state on the interrupt processing routine. The interrupt factor flag is reset by reading through the software. Not resetting the interrupt factor flag and interrupt mask register being "1", will cause the same interrupt to occur again.
  - The interrupt factor flag will be reset by reading through the software. Because of this, when multiple interrupt factor flags are to be assigned to the same address, perform the flag check after the contents of the address has been stored in the RAM. Direct checking with the FAN instruction will cause all the interrupt factor flag to be reset.
  - Be sure to perform the interrupt factor flag reading while in the DI (interrupt flag = "0") state. Performing the reading while in the EI (interrupt flag = "1") state may cause mis-operation.
  - Be sure to perform the interrupt mask register writing while in the DI (interrupt flag = "0") state. Writing while in the EI (interrupt flag = "1") state may cause mis-operation.
  - In case multiple interrupts occur simultaneously, interrupt processing will be done in the order of high priority first.

- **Power Supply**

  External load driving through the output voltage of constant voltage circuit or booster circuit is not permitted.

- **Initial Reset**

  When utilizing the simultaneous high input reset function of the input ports (K00–K03), take care not to make the ports specified during normal operation to go high simultaneously.

- **Data Memory**

  – Since some portions of the RAM are also used as stack area during sub-routine call or register saving, see to it that the data area and the stack area do not overlap.

  – The stack area consumes 3 words during a sub-routine call or interrupt.

  – Address 00H–0FH in the RAM is the memory register area addressed by the register pointer RP.

- **Output Port**

  The FOUT output signal may produce hazards when the output port R10 is turned on or off.

- **I/O Port**

  – When the I/O port is set to the output mode and a low-impedance load is connected to the port pin, the data written to the register may differ from the data read.

  – When the I/O port is set to the input mode and a low-level voltage (V_{SS}) is input by the built-in pull-down resistance, an erroneous input results if the time constant of the capacitive load of the input line and the built-in pull-down resistance load is greater than the read-out time. When the input data is being read, the time that the input line is pulled down is equivalent to 0.5 cycles of the CPU system clock.
  Hence, the electric potential of the pins must settle within 0.5 cycles. If this condition cannot be met, some measure must be devised, such as arranging a pull-down resistance externally, or performing multiple read-outs.

- **Analog Comparator**

  Data in the CMPDT register becomes "1" when CMPON is "0" (analog comparator circuit is off), and undefined when the CMPP and/or CMPM input is disconnected. Avoid reading operation under those conditions.

- **Vacant Register and Read/Write** Writing data into the addresses where read/write bits and read only bits are mixed in 1 word (4 bits) does not affect the read only bits.

- **Battery Voltage Low Detection (BLD) Circuit** Since battery voltage low detection is automatically performed by the hardware every 2 Hz (0.5 second) when the heavy load protection function operates, do not permit the operation of the BLD circuit by the software in order to minimize power current consumption.

- **Heavy Load Protection Function** In the heavy load protection function (heavy load protection mode flag = "1"), battery voltage low detection through the BLDON register is not permitted in order to minimize power current consumption.

APPENDIX A Table of Instructions

Classification	Mne-monic	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	l7	l6	l5	l4	l3	l2	l1	l0					12	PCSL←M(SP), PCSH←M(SP+1), PCP←M(SP+2) SP←SP+3, M(X)←i3~i0, M(X+1)←l7~l4, 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
		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
		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, 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, 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

Classification	Mne-monic	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	l7	l6	l5	l4	l3	l2	l1	l0					5	M(X)←l3~l0, M(X+1)←l7~l4, 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
		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
		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
		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

Classification	Mne-monic	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 B The S1C62N81 I/O Memory Map

AD- DRESS	DATA								COMMENT
	D3	D2	D1	D0	NAME	SR	1	0	
E0	K03	K02	K01	K00	K03	–	HIGH	LOW	INPORT DATA K03
	R	R	R	R	K02	–	HIGH	LOW	INPORT DATA K02
					K01	–	HIGH	LOW	INPORT DATA K01
					K00	–	HIGH	LOW	INPORT DATA K00
E1	0	0	0	K10	0	–	–	–	
	R	R	R	R	0	–	–	–	
					0	–	–	–	
					K10	–	HIGH	LOW	INPORT DATA K10
E2	SWL3	SWL2	SWL1	SWL0	SWL3	0	–	–	STOPWATCH TIMER DATA 3 (1/100) MSB
	R	R	R	R	SWL2	0	–	–	STOPWATCH TIMER DATA 2 (1/100)
					SWL1	0	–	–	STOPWATCH TIMER DATA 1 (1/100)
					SWL0	0	–	–	STOPWATCH TIMER DATA 0 (1/100) LSB
E3	SWH3	SWH2	SWH1	SWH0	SWH3	0	–	–	STOPWATCH TIMER DATA 3 (1/10) MSB
	R	R	R	R	SWH2	0	–	–	STOPWATCH TIMER DATA 2 (1/10)
					SWH1	0	–	–	STOPWATCH TIMER DATA 1 (1/10)
					SWH0	0	–	–	STOPWATCH TIMER DATA 0 (1/10) LSB
E4	TM3	TM2	TM1	TM0	TM3	0	HIGH	LOW	CLOCK TIMER DATA 2Hz
	R	R	R	R	TM2	0	HIGH	LOW	CLOCK TIMER DATA 4Hz
					TM1	0	HIGH	LOW	CLOCK TIMER DATA 8Hz
					TM0	0	HIGH	LOW	CLOCK TIMER DATA 16Hz
E5	KCP03	KCP02	KCP01	KCP00	KCP03	0	FALLING	RISING	K03 INPUT COMPARISON REGISTER
	R/W	R/W	R/W	R/W	KCP02	0	FALLING	RISING	K02 INPUT COMPARISON REGISTER
					KCP01	0	FALLING	RISING	K01 INPUT COMPARISON REGISTER
					KCP00	0	FALLING	RISING	K00 INPUT COMPARISON REGISTER
E6	0	0	0	KCP10	0	–	–	–	
	R	R	R	R/W	0	–	–	–	
					0	–	–	–	
					KCP10	0	FALLING	RISING	K10 INPUT COMPARISON REGISTER
E7	0	0	0	EIMEL	0	–	–	–	
	R	R	R	R/W	0	–	–	–	
					0	–	–	–	
					EIMEL	0	ENABLE	MASK	MELODY INTERRUPT MASK REGISTER
E8	EIK03	EIK02	EIK01	EIK00	EIK03	0	ENABLE	MASK	K03 INTERRUPT MASK REGISTER
	R/W	R/W	R/W	R/W	EIK02	0	ENABLE	MASK	K02 INTERRUPT MASK REGISTER
					EIK01	0	ENABLE	MASK	K01 INTERRUPT MASK REGISTER
					EIK00	0	ENABLE	MASK	K00 INTERRUPT MASK REGISTER
E9	0	0	0	EIK10	0	–	–		
	R	R	R	R/W	0	–	–		
					0	–	–		
					EIK10	0	ENABLE	MASK	K10 INTERRUPT MASK REGISTER
EA	0	0	EISW1	EISW0	0	–	–	–	
	R	R	R/W	R/W	0	–	–	–	
					EISW1	0	ENABLE	MASK	S/W INTERRUPT MASK REGISTER 1Hz
					EISW0	0	ENABLE	MASK	S/W INTERRUPT MASK REGISTER 10Hz
EB	0	EIT2	EIT8	EIT32	0	–	–	–	
	R	R/W	R/W	R/W	EIT2	0	ENABLE	MASK	TIMER INTERRUPT MASK REGISTER 2Hz
					EIT8	0	ENABLE	MASK	TIMER INTERRUPT MASK REGISTER 8Hz
					EIT32	0	ENABLE	MASK	TIMER INTERRUPT MASK REGISTER 32Hz
EC	0	0	0	IMEL	0	–	–	–	
	R	R	R	R	0	–	–	–	
					0	–	–	–	
					IMEL	0	YES	NO	MELODY INTERRUPT FACTOR FLAG
ED	0	0	IK1	IK0	0	–	–	–	
	R	R	R	R	0	–	–	–	
					IK1	0	YES	NO	K10 INTERRUPT FACTOR FLAG
					IK0	0	YES	NO	K00–K03 INTERRUPT FACTOR FLAG
EE	0	0	ISW1	ISW0	0	–	–	–	
	R	R	R	R	0	–	–	–	
					ISW1	0	YES	NO	S/W INTERRUPT FACTOR FLAG 1Hz
					ISW0	0	YES	NO	S/W INTERRUPT FACTOR FLAG 10Hz
EF	0	IT2	IT8	IT32	0	–	–	–	
	R	R	R	R	IT2	0	YES	NO	TIMER INTERRUPT FACTOR FLAG 2Hz
					IT8	0	YES	NO	TIMER INTERRUPT FACTOR FLAG 8Hz
					IT32	0	YES	NO	TIMER INTERRUPT FACTOR FLAG 32Hz

AD-DRESS	DATA								COMMENT
	D3	D2	D1	D0	NAME	SR	1	0	
F0	MAD3	MAD2	MAD1	MAD0	MAD3	0	HIGH	LOW	MEL. ROM ADDR. SETTING REG. AD3
	R/W	R/W	R/W	R/W	MAD2	0	HIGH	LOW	MEL. ROM ADDR. SETTING REG. AD2
					MAD1	0	HIGH	LOW	MEL. ROM ADDR. SETTING REG. AD1
					MAD0	0	HIGH	LOW	MEL. ROM ADDR. SETTING REG. LSB
F1	0	MAD6	MAD5	MAD4	0	–	–	–	
	R	R/W	R/W	R/W	MAD6	0	HIGH	LOW	MEL. ROM ADDR. SETTING REG. MSB
					MAD5	0	HIGH	LOW	MEL. ROM ADDR. SETTING REG. AD5
					MAD4	0	HIGH	LOW	MEL. ROM ADDR. SETTING REG. AD4
F2	CLKC1	CLKC0	TEMPC	MELC	CLK1	0	HIGH	LOW	REG. TO CHANGE MELODY CLOCK
	R/W	R/W	R/W	R/W	CLK0	0	HIGH	LOW	REG. TO CHANGE MELODY CLOCK
					TEMPC	0	HIGH	LOW	REG. TO CHANGE TWO KINDS OF TEMPO
					MELC	0	ON	OFF	MELODY ON/OFF CONTROL REGISTER
F3	R03	R02	R01	R00	R03	0	HIGH	LOW	R03 OUT PORT DATA
	R/W	R/W	R/W	R/W	R02	0	HIGH	LOW	R02 OUT PORT DATA
					R01	0	HIGH	LOW	R01 OUT PORT DATA
					R00	0	HIGH	LOW	R00 OUT PORT DATA
F4	–	R12	R11	R10	R12	0	HIGH	LOW	R12 OUT PORT DATA
		$\overline{\text{MO}}$		FOUT	$\overline{\text{MO}}$	1	–	–	MELODY INVERTED OUTPUT
		ENV			ENV	Hz	–	–	MELODY ENVELOPE CONTROL
	R/W	R/W	R/W	R/W	R11	0	HIGH	LOW	R11 OUT PORT DATA
					R10	0	HIGH	LOW	R10 OUT PORT DATA
					FOUT		ON	OFF	FREQUENCY OUTPUT
F5	0	0	0	0	0	–	–	–	
	R	R	R	R	0	–	–	–	
					0	–	–	–	
					0	–	–	–	
F6	P03	P02	P01	P00	P03	–	HIGH	LOW	P03 I/O PORT DATA
	R/W	R/W	R/W	R/W	P02	–	HIGH	LOW	P02 I/O PORT DATA
					P01	–	HIGH	LOW	P01 I/O PORT DATA
					P00	–	HIGH	LOW	P00 I/O PORT DATA
F7	0	0	0	0	0	–	–	–	
	R	R	R	R	0	–	–	–	
					0	–	–	–	
					0	–	–	–	
F8	0	0	0	0	0	–	–	–	
	R	R	R	R	0	–	–	–	
					0	–	–	–	
					0	–	–	–	
F9	0	TMRST	SWRUN	SWRST	0	–	–	–	
	R	W	R/W	W	TMRST	RESET	RESET	–	TIMER RESET
					SWRUN	0	RUN	STOP	STOPWATCH RUN/STOP CONTROL REG.
					SWRST	RESET	RESET	–	STOPWATCH RESET
FA	HLMOD	0	BLDDT	BLDON	HLMOD	0	HEAVY	NORMAL	HEAVY LOAD PROTECTION MODE
	R/W	R	R	R/W	0	–	–	–	
					BLDDT	0	LOW	NORMAL	BLD DATA
					BLDON	0	ON	OFF	BLD ON-OFF CONTROL REGISTER
FB	CSDC	0	CMPDT	CMPON	CSDC	0	STATIC	DYNAMIC	LCD DRIVER CONTROL REG.
	R/W	R	R	R/W	0	–	–	–	
					CMPDT	1	+>-	->+	CMP DATA
					CMPON	0	ON	OFF	COMPARATOR ON-OFF CONTROL REG.
FC	0	0	0	IOC	0	–	–	–	
	R	R	R	R/W	0	–	–	–	
					0	–	–	–	
					IOC	0	OUT	IN	I/O IN-OUT CONTROL REG.

APPENDIX C 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 ↵ #BAR,a ↵	Sets Break at program address "a" Breakpoint is canceled
		#BD ↵ #BDR ↵	Break condition is set for data RAM Breakpoint is canceled
		#BR ↵ #BRR ↵	Break condition is set for Evaluation Board CPU internal registers Breakpoint is canceled
		#BM ↵ #BMR ↵	Combined break conditions set for program data RAM address and registers 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
		#OPTLD,n,file ↵	Load HEXA data flom 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.

APPENDIX D 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`
			`LOCAL LOOP`
			`LOOP CP MX,DATA`
			`JP NZ,LOOP`
			`ENDM`
			
			`CHECK 1`

9	LOCAL (Local)	To make local specification of label during macro definition	
10	ENDM (End Macro)	To end macro definition	

APPENDIX E The Format of Melody Source File

Contents of the source file, created with an editor such as EDLIN, are configured from the S1C62N81 Series melody codes and the pseudo-instructions described later.

Source File Name

The source file can be named with a maximum of any seven characters. As a rule, keep to the following format.

C281YYY.MDT

Three alphanumerics are entered in the "YYY" part. Refer to the model name from Seiko Epson. The extension must be ".MDT".

Statement (line)

Write each of the source file statements (lines) as follows:

```
Basic format: <attack>   <note>   <scale>   <end bit>   <comment>
Example:         .TEMPC0=5
                 .TEMPC1=8
                 .OCTAVE=32
                 ;
                 1        1        C3
                 0        4        D4
                 0        4        E4#
                 0        2        F5
                 0        3        G5#
                 1        7        A4
                 1        5        B4
                 0        6        A4#        1      ;1st Melody
                 ;
                 ORG      10H
                 ;
                 1        2        C3#
                 0        3        $45
                 0        7        $E3
                 1        6        $97
                 0        5        C6
                 0        7        A5#
                 1        3        $42        1      ;2nd Melody

           Attack field  Note field  Scale field  End bit field  Comment field
```

The statement is made up of the five fields: attack field, note field, scale field, end bit field, and comment field. Up to 80 characters can be written in the statement. The fields are separated by one or more spaces or by inserting tabs.
The end bit fields and comment fields can be filled in on an as-needed basis.
A blank line is also permitted for the CR (carriage return) code only. However, it is not permitted on the last line. Each of the fields can be started from any column.

Attack field

Control of the attack output is written.
When "1" is written, attack output is performed. When "0" is written, attack output is not performed.

Note field

Eight notes can be specified with the melody ROM codes D5 through D7. Fill in the note field with numbers from 1 to 8.

No.	1	2	3	4	5	6	7	8
Note	𝅘𝅥𝅯	𝅘𝅥𝅮	𝅘𝅥𝅮.	𝅘𝅥	𝅘𝅥𝅘𝅥𝅯	𝅘𝅥.	𝅘𝅥..	𝅗𝅥

Scale field

The scale field can be filled in with any scale data (C3 through C6#).
When inputting the code directly, prefix the code with "$". In this case, the input code range is 00H through FDH.

End bit field

The instruction indicating the end of the melody is written in the end bit field. When "1" is written, the melody finishes with the melody ROM code of that address. Otherwise, write "0", or omit it altogether.

Comment field

Any comment, such as the program index or processing details, can be written in the comment field, with no affect on the object file created with the assembler.
The comment field is the area between the semicolon ";" and the CR code at the end of the line.
A line can be made up of a comment field alone. However, if the comment extends into two or more lines, each line must be headed with a semicolon.

APPENDIX F Dividing Table

Dividing table at no use of octave 32.768 kHz

Scale Data	Frequency (Hz)	Scale ROM Code									Dividing Ratio	Absolute Error (%)	Standard Frequency (Hz)
		S7	S6	S5	S4	S3	S2	S1	S0	Hex.			
C3	128	0	0	0	0	0	1	0	0	04	1/128 x 1/2	0	128
C3#	135.405	0	0	0	1	0	0	1	0	12	1/121 x 1/2	-0.152	135.611
D3	143.719	0	0	1	0	0	0	0	0	20	1/114 x 1/2	0.031	143.675
D3#	152.409	0	0	1	0	1	1	1	1	2F	1/107 + 103	0.024	152.218
E3	161.419	0	0	1	1	1	0	1	1	3B	1/101 + 102	0.092	161.270
F3	170.667	0	1	0	0	0	1	0	0	44	1/96 x 1/2	-0.113	170.860
F3#	181.039	0	1	0	1	0	0	0	1	51	1/90 + 91	0.010	181.019
G3	191.626	0	1	0	1	1	0	1	1	5B	1/85 + 86	-0.030	191.783
G3#	203.528	0	1	1	0	0	1	0	1	65	1/80 + 81	0.167	203.187
A3	215.579	0	1	1	0	1	1	0	0	6C	1/76 x 1/2	0.143	215.270
A3#	227.556	0	1	1	1	0	1	0	0	74	1/72 x 1/2	-0.226	228.070
B3	240.941	0	1	1	1	1	1	0	0	7C	1/68 x 1/2	-0.287	241.632
C4	256	1	0	0	0	0	1	0	0	84	1/64 x 1/2	0	256
C4#	270.810	1	0	0	0	1	1	0	1	8D	1/60 + 61	-0.153	271.222
D4	287.439	1	0	0	1	0	0	1	0	92	1/57 x 1/2	0.031	287.350
D4#	303.407	1	0	0	1	1	0	0	0	98	1/54 x 1/2	-0.339	304.436
E4	321.255	1	0	0	1	1	1	1	0	9E	1/51 x 1/2	-0.400	322.540
F4	341.333	1	0	1	0	0	1	0	0	A4	1/48 x 1/2	-0.113	341.720
F4#	360.088	1	0	1	0	1	0	1	1	AB	1/45 + 46	-0.542	362.038
G4	385.506	1	0	1	1	0	0	0	1	B1	1/42 + 43	0.503	383.566
G4#	404.543	1	0	1	1	0	1	0	1	B5	1/40 + 41	-0.453	406.374
A4	431.158	1	0	1	1	1	0	0	0	B8	1/38 x 1/2	0.144	430.540
A4#	455.111	1	0	1	1	1	1	0	0	BC	1/36 x 1/2	-0.226	456.140
B4	481.882	1	1	0	0	0	0	0	0	C0	1/34 x 1/2	-0.287	483.264
C5	512	1	1	0	0	0	1	0	0	C4	1/32 x 1/2	0	512
C5#	546.133	1	1	0	0	1	0	0	0	C8	1/30 x 1/2	0.675	542.444
D5	574.877	1	1	0	0	1	1	0	1	CD	1/28 + 29	0.031	574.700
D5#	606.815	1	1	0	0	1	1	1	0	CE	1/27 x 1/2	-0.339	608.872
E5	642.510	1	1	0	1	0	0	1	1	D3	1/25 + 26	-0.400	645.080
F5	682.667	1	1	0	1	0	1	0	0	D4	1/24 x 1/2	-0.113	683.440
F5#	728.178	1	1	0	1	1	0	0	1	D9	1/22 + 23	0.563	724.076
G5	762.047	1	1	0	1	1	0	1	1	DB	1/21 + 22	-0.668	767.132
G5#	819.200	1	1	0	1	1	1	0	0	DC	1/20 x 1/2	0.787	812.748
A5	862.316	1	1	0	1	1	1	1	0	DE	1/19 x 1/2	0.144	861.080
A5#	910.222	1	1	1	0	0	0	0	0	E0	1/18 x 1/2	-0.226	912.280
B5	963.765	1	1	1	0	0	0	1	0	E2	1/17 x 1/2	-0.287	966.528
C6	1024	1	1	1	0	0	1	0	0	E4	1/16 x 1/2	0	1024
C6#	1092.267	1	1	1	0	0	1	1	0	E6	1/15 x 1/2	0.675	1084.888

Dividing table at no use of octave 65.536 kHz

Scale Data	Frequency (Hz)	Scale ROM Code									Dividing Ratio	Absolute Error (%)	Standard Frequency (Hz)
		S7	S6	S5	S4	S3	S2	S1	S0	Hex.			
C4	256	0	0	0	0	0	1	0	0	04	1/128 x 1/2	0	256
C4#	270.810	0	0	0	1	0	0	1	0	12	1/121 x 1/2	-0.152	271.222
D4	287.439	0	0	1	0	0	0	0	0	20	1/114 x 1/2	0.031	287.350
D4#	304.819	0	0	1	0	1	1	1	1	2F	1/107 + 103	2.448	304.436
E4	322.837	0	0	1	1	1	0	1	1	3B	1/101 + 102	0.092	322.540
F4	341.333	0	1	0	0	0	1	0	0	44	1/96 x 1/2	-0.113	341.720
F4#	362.077	0	1	0	1	0	0	0	1	51	1/90 + 91	0.011	362.038
G4	383.251	0	1	0	1	1	0	1	1	5B	1/85 + 86	-0.082	383.566
G4#	407.056	0	1	1	0	0	1	0	1	65	1/80 + 81	0.168	406.374
A4	431.158	0	1	1	0	1	1	0	0	6C	1/76 x 1/2	0.143	430.540
A4#	455.111	0	1	1	1	0	1	0	0	74	1/72 x 1/2	-0.226	456.140
B4	481.882	0	1	1	1	1	1	0	0	7C	1/68 x 1/2	-0.287	483.264
C5	512	1	0	0	0	0	1	0	0	84	1/64 x 1/2	0	512
C5#	541.620	1	0	0	0	1	1	0	1	8D	1/60 + 61	-0.152	542.444
D5	574.877	1	0	0	1	0	0	1	0	92	1/57 x 1/2	0.031	574.700
D5#	606.815	1	0	0	1	1	0	0	0	98	1/54 x 1/2	-0.339	608.872
E5	642.510	1	0	0	1	1	1	1	0	9E	1/51 x 1/2	-0.400	645.080
F5	682.667	1	0	1	0	0	1	0	0	A4	1/48 x 1/2	-0.113	683.440
F5#	720.176	1	0	1	0	1	0	1	1	AB	1/45 + 46	-0.541	724.076
G5	771.012	1	0	1	1	0	0	0	1	B1	1/42 + 43	0.503	767.132
G5#	809.086	1	0	1	1	0	1	0	1	B5	1/40 + 41	-0.453	812.748
A5	862.316	1	0	1	1	1	0	0	0	B8	1/38 x 1/2	0.143	861.080
A5#	910.222	1	0	1	1	1	1	0	0	BC	1/36 x 1/2	-0.226	912.280
B5	963.765	1	1	0	0	0	0	0	0	C0	1/34 x 1/2	-0.287	966.528
C6	1024	1	1	0	0	0	1	0	0	C4	1/32 x 1/2	0	1024
C6#	1092.267	1	1	0	0	1	0	0	0	C8	1/30 x 1/2	0.676	1084.888
D6	1149.754	1	1	0	0	1	1	0	1	CD	1/28 + 29	0.031	1149.400
D6#	1213.630	1	1	0	0	1	1	1	0	CE	1/27 x 1/2	-0.339	1217.748
E6	1285.020	1	1	0	1	0	0	1	1	D3	1/25 + 26	-0.399	1290.160
F6	1365.333	1	1	0	1	0	1	0	0	D4	1/24 x 1/2	-0.113	1366.880
F6#	1456.356	1	1	0	1	1	0	0	1	D9	1/22 + 23	0.563	1448.152
G6	1524.093	1	1	0	1	1	0	1	1	DB	1/21 + 22	-0.667	1534.264
G6#	1638.400	1	1	0	1	1	1	0	0	DC	1/20 x 1/2	0.788	1625.496
A6	1724.632	1	1	0	1	1	1	1	0	DE	1/19 x 1/2	0.143	1722.160
A6#	1820.444	1	1	1	0	0	0	0	0	E0	1/18 x 1/2	-0.226	1824.560
B6	1927.529	1	1	1	0	0	0	1	0	E2	1/17 x 1/2	-0.287	1933.056
C7	2048	1	1	1	0	0	1	0	0	E4	1/16 x 1/2	0	2048
C7#	2194.533	1	1	1	0	0	1	1	0	E6	1/15 x 1/2	0.676	2169.776

APPENDIX G RAM Map

PROGRAM NAME: C281____ /

P	H \ L			0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	MAME	MSB																
			LSB																
	1	MAME	MSB																
			LSB																
	2	MAME	MSB																
			LSB																
	3	MAME	MSB																
			LSB																
	4	MAME	MSB																
			LSB																
	5	MAME	MSB																
			LSB																

PROGRAM NAME: C281_____ /

P	H	L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	9	MAME																
		MSB																
		LSB																
	A	MAME																
		MSB																
		LSB																
	E	MAME	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
		MSB	ZK00	ZK10	ZSWL0	ZSWH0	ZTM0	ZKCP00	ZKCP10	ZEIMEL	ZEIK00	ZEIK10	ZEISW0	ZEIT32	ZIMEL	ZIK0	ZISW0	ZIT32
			ZK01	—	ZSWL1	ZSWH1	ZTM1	ZKCP01	—	—	ZEIK01	—	ZEISW1	ZEIT8	—	ZIK1	ZISW1	ZIT8
			ZK02	—	ZSWL2	ZSWH2	ZTM2	ZKCP02	—	—	ZEIK02	—	—	ZEIT2	—	—	—	ZIT2
		LSB	ZK03	—	ZSWL3	ZSWH3	ZTM3	ZKCP03	—	—	ZEIK03	—	—	—	—	—	—	—
	F	MAME	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
		MSB	ZMAD0	ZMAD4	ZMELC	ZR00	ZFOUT	—	ZP00	—	—	ZSWRST	ZBLDON	ZCMPON	ZIOC	—	—	—
			ZMAD1	ZMAD5	ZTEMPC	ZR01	ZR11	—	ZP01	—	—	ZSWRUN	ZBLDDT	ZCMPDT	—	—	—	—
			ZMAD2	ZMAD6	ZCLKC0	ZR02	ZR12	—	ZP02	—	—	ZTMRST	—	—	—	—	—	—
		LSB	ZMAD3	—	ZCLKC1	ZR03	—	—	ZP03	—	—	—	ZHLMOD	ZCSDC	—	—	—	—

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.

SEIKO EPSON CORPORATION
ELECTRONIC DEVICES MARKETING DIVISION

■ **EPSON Electronic Devices Website**

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

First issue March, 1990
Printed March, 2001 in Japan ⓂB