SM5M2

4-Bit Single-Chip Microcomputer (LCD Driver)

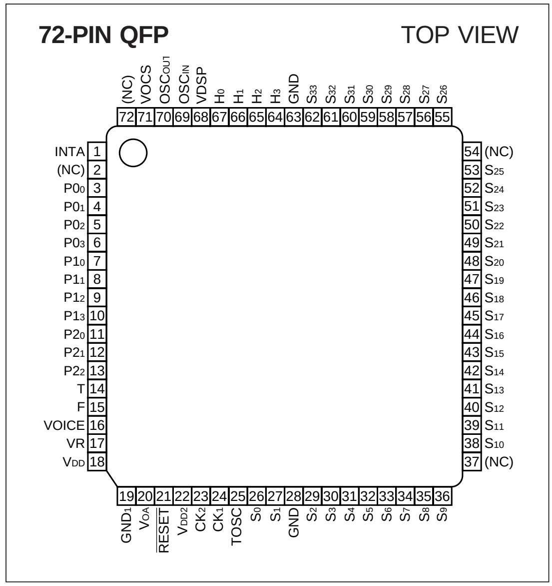
DESCRIPTION

The SM5M2 is a CMOS 4-bit single-chip microcomputer operated on 3.0 V single power supply. This microcomputer integrates 4-bit parallel processing function, ROM, RAM, display RAM, 15-stage divider, 2-kind of interrupt and 4-level of subroutine stack. With a built-in LCD drive circuit for a maximum of 136 elements, a 2-mode standby function, voice synthesizer and a melody generator circuit in a single chip, the SM5M2 permits the design of system configuration with a minimum of peripheral components. It can be used in a variety of products from handheld equipment to electrical appliances, such as hand held games with voice, and also achieves low power consumption.

FEATURES

- ROM capacity :
  - 3 072 x 8 bits (For main program)
  - 64k x 5 bits (For voice)
  - 256 x 6 bits (For melody)
- RAM capacity : 130 x 4 bits (including 34 x 4 bits display RAM)
- Instruction sets : 51
- Subroutine nesting : 4 levels
- I/O port :
  - Input 1
  - Output 6
  - Input/output 7
- Interrupts :
  - Internal interrupt x 1 (divider overflow)
  - External interrupt x 1 (INTA)
- Built-in voice synthesizer circuit (APCM) :
  - Number of phrases : 256
  - Voice ROM : 64 k x 5 bits
  - Bit rate : 25/35 kbps
  - Number of coded bit : 5 bits
  - Sampling frequency : 5/7 kHz
  - Generation period : 9.1 to 12.8 s
- Built-in main clock oscillator for system clock
- Built-in sub clock oscillator for real time clock

PIN CONNECTIONS

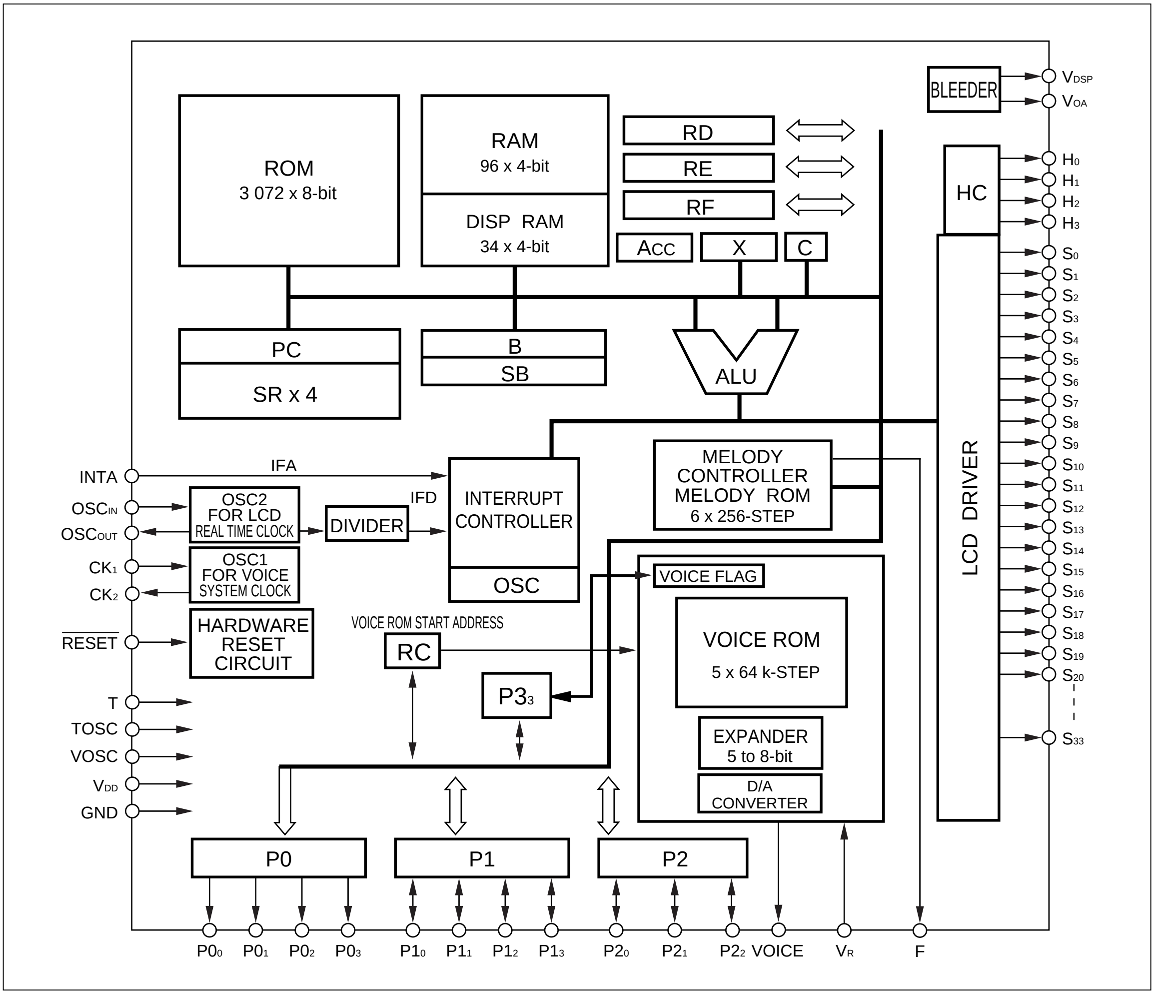


- Built-in 15 stages divider for real time clock
- Built-in LCD driver :
  - 136 segments, 1/2 bias, 1/4 duty cycle
- Built-in melody generator circuit :
  - Melody ROM : 256 steps
  - Generating time (at 32.768 kHz) : 32 s (MAX.)
- Instruction cycle time :
  - 25.9 µs (MIN.) (at 70 kHz ± 10%)
  - 61 µs* (TYP.) (at 32.768 kHz)

*When using the clock with the system clock.

- Standby function
- Supply voltage : 2.4 to 3.3 V
- Package : 72-pin QFP (QFP072-P-1010)

BLOCK DIAGRAM



Nomenclature

A_{CC}	: Accumulator
ALU	: Arithmetic logic unit
B	: RAM address register
C	: Carry flag
HC	: Common signal generator circuit
IFA	: External interrupt flag
IFD	: Divider overflow flag
RC	: Voice starting address
OSC_{IN},OSC_{OUT}	: Oscillator for LCD and real time clock
P0-P2	: Port registers
$P3_3$	: Voice flag port
PC	: Program counter
RAM	: Data memory
RD, RE, RF	: Mode registers
ROM	: Program memory
SB	: Stack B register
SR	: PC stack register
X	: X register
CK_1,CK_2	: Oscillator for voice and system clock

PIN DESCRIPTION

PIN NAME	I/O	FUNCTION
GND, V_{DD}, V_{DSP}, V_R	I	Power supply pins. The V_{DD}, V_{DSP}, V_R pins apply a positive supply with respect to the GND.
T, TOSC, VOSC	I	LSI chip test pins. Cannot be used by the user. Connect T and TOSC to GND. Connect VOSC to V_{DD}.
$\overline{\text{RESET}}$	I	Input pin with built-in pull-up resistor. Hardware-reset the LSI chip when a Low level signal is input. Normally, a capacitor is connected between it and GND to form a power-on reset circuit.
OSC_{IN}, OSC_{OUT}	I/O	Crystal oscillator pins. Connect a crystal oscillator accross [OSC_{IN}-OSC_{OUT}] to form a clock generator circuit.
CK_1, CK_2	I	RC oscillator pins. Connect a resistor across [CK_1-V_{DD}] to form a clock generator circuit. CK_2 is used to test its clock out.
Voice	O	Voice output pin. Output the contents of a voice ROM.
F	O	Melody output pin. Outputs the contents of a melody ROM with standard 12 musical scales (555 to 2 114 Hz) in two octaves.
H_0-H_3	O	Pins for the LCD's common signals.
S_0-S_{33}	O	Pins for the LCD's segment signals.
INTA	I	Input pin for external interrupt. The IFA flag is set at the rising edge of INTA.
$P0_0$-$P0_3$	O	Output ports. The P0 ports are an output port. The accumulator A_{CC} can be transferred to this port by instruction.
$P1_0$-$P1_3$, $P2_0$-$P2_2$	I/O	P1 and P2 are I/O pins which can switch to input or output pins in 4/3-bit units by instruction. They can be used as output pins when configured for a key matrix. The SM5M2 is forced to hardware-reset when all of $P1_0$-$P1_3$ pins are High level. (By mask option)

ABSOLUTE MAXIMUM RATINGS

PARAMETER	SYMBOL	RATING	UNIT	NOTE
Supply voltage	V_{DD}	–0.3 to 4.0	V	
Input voltage	V_I	–0.3 to V_{DD} + 0.3	V	
Output voltage	V_O	–0.3 to V_{DD} + 0.3	V	
Source output current for each pin	I_{O1}	2	mA	1
	I_{O2}	2	mA	2
	I_{O3}	2	mA	3
	I_{O4}	2	mA	4
Sink output current for each pin	I_{O5}	2	mA	1
	I_{O6}	100	µA	2
	I_{O7}	2	mA	3
	I_{O8}	2	mA	4
Total source output current	I_{OH}	10	mA	
Total sink output current	I_{OL}	10	mA	
Operating temperature	T_{OPR}	0 to 50	°C	
Storage temperature	T_{STG}	–55 to 150	°C	

NOTES :

1. Applicable pins : $P0_0$-$P0_3$
2. Applicable pins : $P1_0$-$P1_3$, $P2_0$-$P2_2$
3. Applicable pin : F
4. Applicable pins : H_0-H_3, S_0-S_{33}

RECOMMENDED OPERATING CONDITIONS

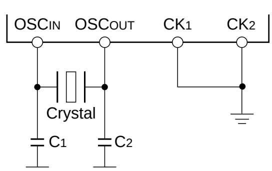
PARAMETER	SYMBOL	RATING	UNIT	NOTE
Supply voltage	V_{DD}	2.4 to 3.3	V	
Instruction cycle	T_{SYS}	Crystal+CR 25.9 to 31.7	µs	
		Crystal 61.0		
Oscillation starting voltage	V_{OSC}	2.0	V	1

NOTE :

1. Use the crystal oscillation circuit

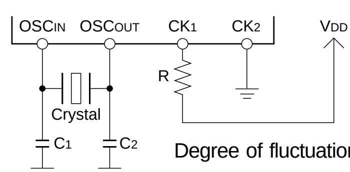
Oscillation Circuit

- Crystal oscillation (frequency = 32.768 kHz)



Crystal : 32.768 kHz
C_1 = 15 pF
C_2 = 15 pF

- CR oscillation (frequency = 70 kHz)



Degree of fluctuation frequency : ± 10%
(V_{DD} = 3 V, T_{OPR} = 25°C)
C_1 = 15 pF, C_2 = 15 pF, R = 1.0 MΩ

NOTE : In case of using RC resonator, crystal is also required.

NOTE : Mount the R, C and crystal as close to the LSI chip as possible to minimize the effects of stray capacitance.

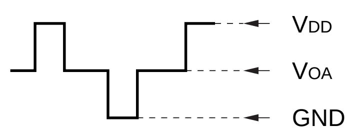
DC CHARACTERISTICS

(V_{DD} = 2.4 to 3.3 V, T_{OPR} = 0 to +50°C)

PARAMETER	SYMBOL	CONDITIONS	MIN.	TYP.	MAX.	UNIT	NOTE
Input voltage	V_{IH1}		0.8 x V_{DD}		V_{DD}	V	1
	V_{IL1}		0		0.2 x V_{DD}		
	V_{IH2}		V_{DD} –0.25		V_{DD}	V	2
	V_{IL2}		0		0.25		
Input current	I_{IH1}	V_{IH} = V_{DD}			30.0	µA	3
	I_{IH2}	V_{IH} = V_{DD}			30.0		4
	I_{IL1}	V_{IL} = 0 V			25.0		5
Output current	$-I_{OH1}$	V_{OH} = V_{DD} – 0.5 V	500	1 300		µA	6
	I_{OL1}	V_{OL} = 0.5 V	1 000	2 000			
	$-I_{OH2}$	V_{OH} = V_{DD} – 0.5 V	500	1 300			7
	I_{OL2}	V_{OL} = 0.5 V	25	90.0			
	I_O1	D = $1F_H$		980			8
	I_O2	D = $0F_H$		740			
	I_O3	D = 01_H		200			
Supply current	$I_{OP}11$	CRRUN1		120	150	µA	9
	$I_{OP}12$	CRRUN2		110	130		
	$I_{St}11$	CRSTOP1		15.0	40.0		
	$I_{St}12$	CRSTOP2		4.00	15.0		
	$I_{St}13$	CRSTOP3		3.00	13.0		
	$I_{OP}21$	XTALRUN1		50.0	100.0		
	$I_{OP}22$	XTALRUN2		40.0	80.0		
	I_{St} 21	XTALHALT1		30.0	60.0		
	I_{St} 22	XTALHALT2		26.0	52.0		
	I_{St} 23	XTALHALT3		26.0	52.0		
	I_{St} 24	XTALSTOP		4.0	15.0		
Output impedance	D_{COM}	V_{DD}=3.0 V		15		kΩ	10
	D_S	V_{DD}=3.0 V		30			11

NOTES :

1. Applicable pins : $P1_0$-$P1_3$, $P2_0$-$P2_2$
2. Applicable pins : OSC_{IN}, $\overline{RESET}$, T, INTA
3. Applicable pins : $P2_0$-$P2_2$
4. Applicable pins : $P1_0$-$P1_3$
5. Applicable pin : $\overline{RESET}$
6. Applicable pins : $P0_0$-$P0_3$, F
7. Applicable pins : $P1_0$-$P1_3$, $P2_0$-$P2_2$
8. Applicable pins : VOICE, value of external resistor = 2 kΩ
9. Measurement conditions in detail are mentioned in the tables next page.
10. Applicable pins : H_0-H_3
11. Applicable pins : S_0-S_{33}



LCD wave form (EXAMPLE)

CR + X'TAL Standby Mode

STATUS	STOP	$P3_3$	RF1	RD2	CR	X'TAL	CPU	Voice	LCD	Divider
CRRUN1	0	1	1	0	ON	ON	ON	ON	ON	ON
CRRUN2	0	0	1	0	ON	ON	ON	OFF	ON	ON
CRSTOP1	1	0	1	0	OFF	ON	OFF	OFF	ON	ON
CRSTOP2	1	0	0	0	OFF	ON	OFF	OFF	OFF	ON
CRSTOP3	1	0	0	1	OFF	ON	OFF	OFF	OFF	OFF

NOTES :
- When CR = OFF, CPU and Voice are OFF.
- When Divider = OFF, neither LCD nor Melody is in operation (undefined).
- STOP = 1 stands for executing STOP instruction.

Only X'TAL Standby Mode

STATUS	STOP	HALT	$P3_3$	RF1	RD2	CR	X'TAL	CPU	Voice	LCD	Divider
XTALRUN1	0	0	1	1	0	OFF	ON	ON	ON	ON	ON
XTALRUN2	0	0	0	1	0	OFF	ON	ON	OFF	ON	ON
XTALHALT1	0	1	0	1	0	OFF	ON	OFF	OFF	ON	ON
XTALHALT2	0	1	0	0	0	OFF	ON	OFF	OFF	OFF	ON
XTALHALT3	0	1	0	0	1	OFF	ON	OFF	OFF	OFF	OFF
XTALSTOP	1	0	0	0	0	OFF	OFF	OFF	OFF	OFF	OFF

NOTES :
- When CR = OFF, CPU and Voice are OFF.
- When Divider = OFF, neither LCD nor Melody is in operation (undefined).
- STOP = 1 stands for executing STOP instruction.
- HALT = 1 stands for executing HALT instruction.

SYSTEM CONFIGURATION

A Resister and X Register

The A register (or accumulator : A_{CC}) is a 4-bit general purpose register. The register is mainly used in conjunction with the ALU, C flag and RAM to transfer numerical value and data to perform various operations. The A register is also used to transfer data between input and output pins.

The X register (or auxiliary accumulator) is a 4-bit register and can be used as a temporary register. It loads contents of the A register or its content is transferred to the A register.

When the table reference instruction PAT is used, the X and A registers load ROM data.

A pair of A and X registers can accommodate 8-bit data.

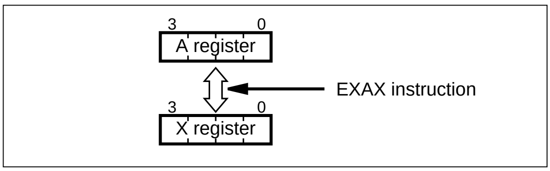


Fig. 1 Data Transfer Example Between A Register and X Register

Arithmetic and Logic Unit (ALU) and Carry Signal Cy

The ALU performs 4-bit parallel operation.

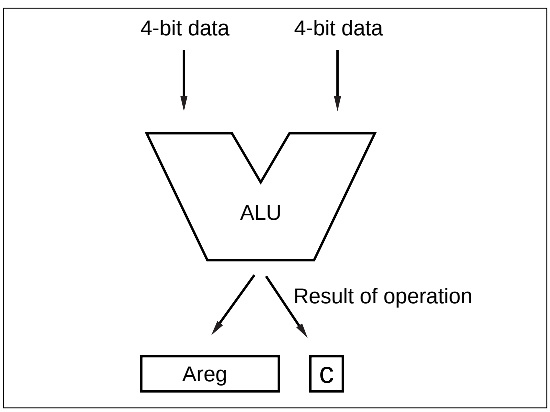


Fig. 2 ALU

The ALU operates binary addition in conjunction with RAM, C flag and A register. The carry signal Cy is generated if a carry occurs during ALU operation. Some instructions use Cy : ADC instruction sets/clears the content of the C flag; ADX instruction causes the program to skip the next instruction. Note that Cy is the symbol for carry signal and not for C flag.

B Register and SB Register

• B register (B_M, B_L)

The B register is an 8-bit register that is used to specify the RAM address.

The upper 4-bit section is called B_M register and lower 4-bit B_L.

• SB register

The SB register is an 8-bit register used as the save register for the B register. The contents of B register and SB register can be exchanged through EX instruction.

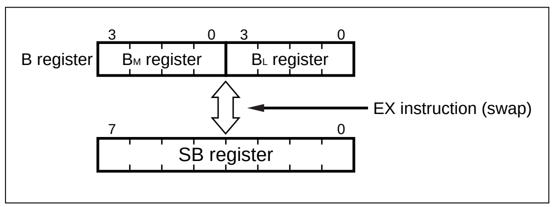


Fig. 3 B Register and SB Register

Data Memory (RAM)

The data memory (RAM) is used for data storage. The RAM capacity consists of 130 x 4-bit (include 34 x 4-bit display RAM).

Display RAM, which outputs data to an external pin for driving the segments of the LCD. Therefore, by writing data to the display RAM, the LCD can be driven at 1/4 duty (1/2 bias) to enable automatic display of the LCD.

As shown in Fig. 5 the display RAM is connected to segment outputs port from S_0 to S_{33} which correspond to the LCD common outputs H_0 to H_3. Data M_0 to M_3 for one column of the display RAM is output pins as a LCD drive waveform which corresponds to outputs H_0 to H_3. As a RAM, the display RAM operates exactly the same as other RAMs.

B_M \ B_L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0																
1																
2																
3																
4																
5																
8	S_0	S_2	S_4	S_6	S_8	S_{10}	S_{12}	S_{14}	S_{16}	S_{18}	S_{20}	S_{22}	S_{24}	S_{26}	S_{28}	S_{30}
9	S_1	S_3	S_5	S_7	S_9	S_{11}	S_{13}	S_{15}	S_{17}	S_{19}	S_{21}	S_{23}	S_{25}	S_{27}	S_{29}	S_{31}
A	S_{32}															
B	S_{33}															

* The area surrounded by the thick line represents the display RAM where S_0 to S_{33} corresponds to the segment output.

Fig. 4 RAM Organization

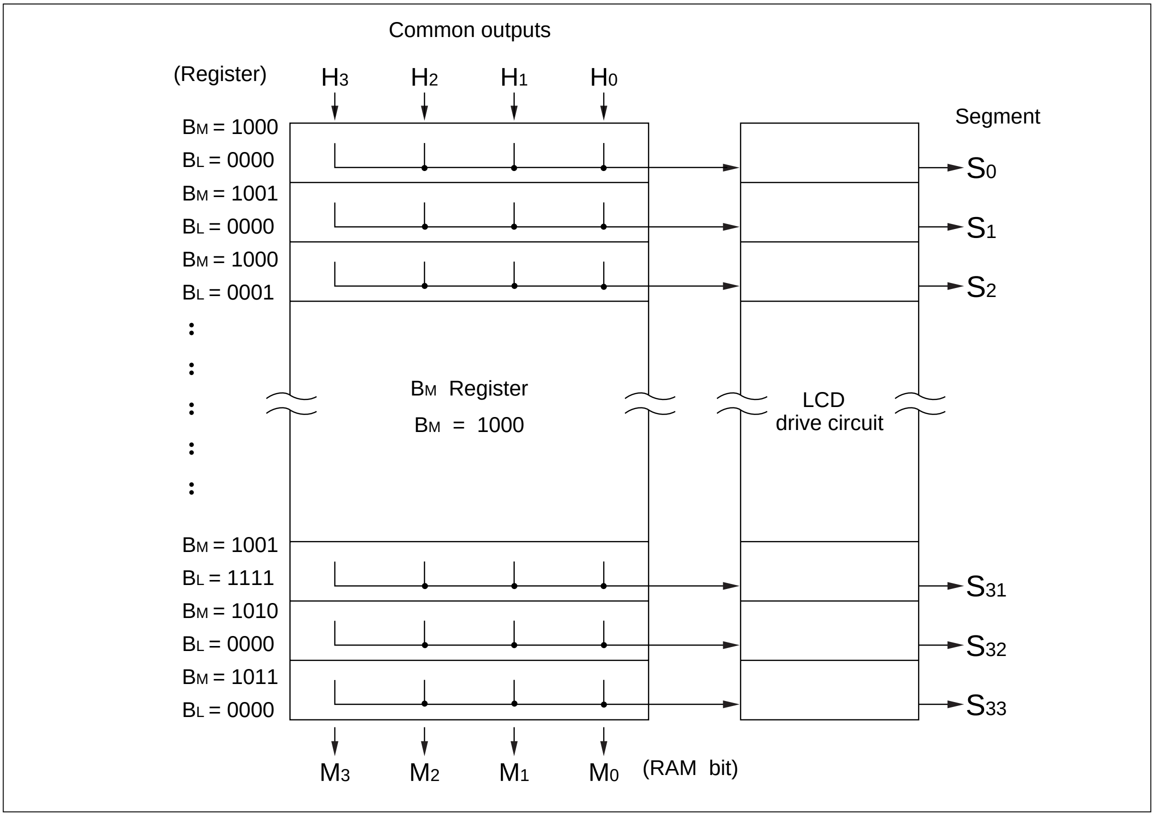


Fig. 5 Relationship between The Display RAM and LCD Segment Outputs/Common Outputs

Program Counter PC and Stack Register SR

A ROM address is specified by the program counter (PC). The PC comprises 12-bit where 6-bit (P_U) are used to specify the page (see Fig. 6) and 6-bit (P_L) are used to specify the step. P_U is a register and P_L is a binary counter.
The table reference instruction PAT executes a similar operation to that of the subroutine jump and uses one level of the stack register.

Program Memory (ROM)

The ROM is used for program storage. The ROM capacity of the SM5M2 is 3 072-step . The ROM is organized into 48-page where one page is organized into 64-step.

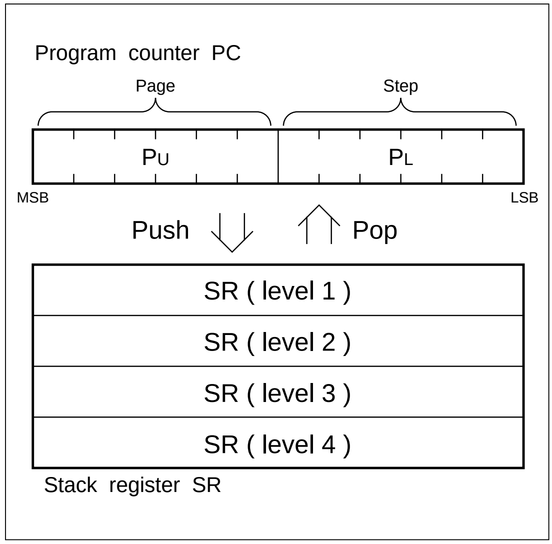


Fig. 6 Program Counter PC and Stack Register SR

Page	00H	01H	02H	03H	04H	05H	06H	07H	08H	09H	0AH	0BH	0CH	0DH	0EH	0FH
P_U	000000	000001	000010	000011	000100	000101	000110	000111	001000	001001	001010	001011	001100	001101	001110	001111
	Program start	First page of subroutine TRS	Interrupt	Standby release	Table reference page PAT											

Page	10H	11H	12H	13H	14H	15H	16H	17H	18H	19H	1AH	1BH	1CH	1DH	1EH	1FH
P_U	010000	010001	010010	010011	010100	010101	010110	010111	011000	011001	011010	011011	011100	011101	011110	011111

Page	20H	21H	22H	23H	24H	25H	26H	27H	28H	29H	2AH	2BH	2CH	2DH	2EH	2FH
P_U	100000	100001	100010	100011	100100	100101	100110	100111	101000	101001	101010	101011	101100	101101	101110	101111
																Last page

Fig. 7 ROM Organization

Flags

The SM5M2 provides 4-flag (C flag and interrupt request flag <IFA, IFD, $P3_3$>) which can be used to set or determine conditions.

Output Latch Registers and Mode Registers

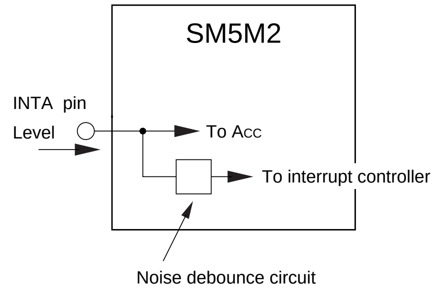
The output latch registers are connected to the P0, P1 and P2 pins. By instruction, the contents of the Acc can be transferred to the output latch registers. The SM5M2 also contains mode registers RC, RD, RE and RF. Setting the value of each register enables the voice start address, divider, LCD, melody or interrupt to be controlled. Setting a register is performed in the same way as the other output pins. The functions of the mode registers are shown in Table 1.

• INTA pin

INTA level can be loaded to Acc (bit 0), as follows.

```
LBLX    4
IN
```

INTA level does not through the noise debounce circuit.



CAUTION :

Connecting considerations of I/O port

When using an I/O port as bidirectional bus such as data bus, avoid setting the I/O port to output when the target pin is also set output.

Whenever the both output data conflict each other, system failure will be caused due to damaged circuits or instantaneous supply voltage drop.

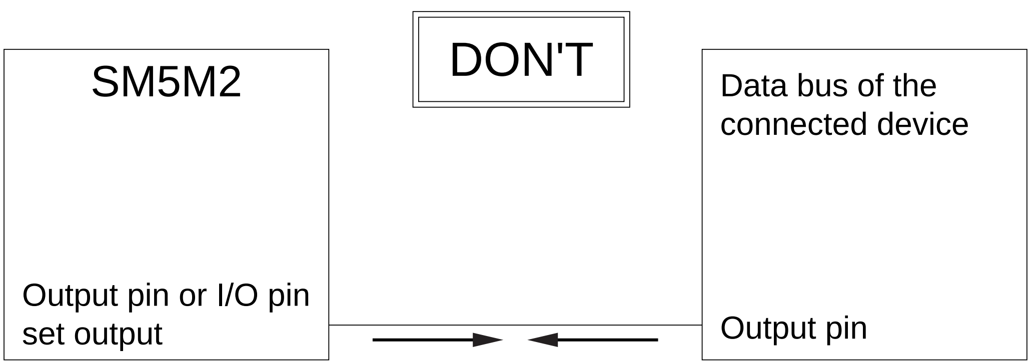


Table 1 Mode Register Setting

REGISTER TYPE	REGISTER BIT	SET VALUE	MODE DESCRIPTION
RC	RC0 – RC7	–	Sets voice synthesizer starting address.
RD	RD0	0	Clears the ME F/F to stop a melody.
		1	Sets the ME F/F to start a melody from a ROM pointer address.
	RD1		Sets by stop instruction (of melody code) and reset by TPB instruction.
	RD2	0	Accepts divider clock-in.
		1	Masks divider clock-in.
	RD3	0	Sets voice synthesizer to 7 kHz sampling rate.
		1	Sets voice synthesizer to 5 kHz sampling rate.
RE	RE0	0	Masks the interrupt based on the IFA flag.
		1	Accepts the interrupt based on the IFA flag.
	RE1	–	Sets "0" only.
	RE2	0	Masks the interrupt based on the IFD flag.
		1	Accepts the interrupt based on the IFD flag.
	RE3	–	No setting.
RF	RF0	0	Turns off the LCD.
		1	Turns on the LCD.
	RF1	0	Stops the function of a bleeder circuit.
		1	Operates the function of a bleeder circuit.
	RF2	0	Creates the system clock frequency by dividing two the main oscillation frequency.
		1	Creates the system clock frequency by dividing four the main oscillation frequency.
	RF3	–	Sets "0" only.

System Clock Generator and Dividers

The main oscillation frequency (CR oscillator) which is input through CK_1 is divided into 2 or 4 to generate the system clock f_{SYS} (Fig. 8).

System clock f_{SYS} determines the execution instruction cycle so that the system clock period is the same as the instruction cycle.

However, the instruction execution cycle of two-word instruction is twice that of one-word instructions.

Use of a CR oscillating element or a crystal oscillating element for the oscillator circuit is determined by the mask option. The crystal oscillator which is input through "OSC_{IN}-OSC_{OUT}" can be used as both real time clock and display signal of LCD. On the final stage of the divider, fc can be set 1 Hz or 2 Hz (in case of 32 kHz crystal oscillation) depending on the mask option.

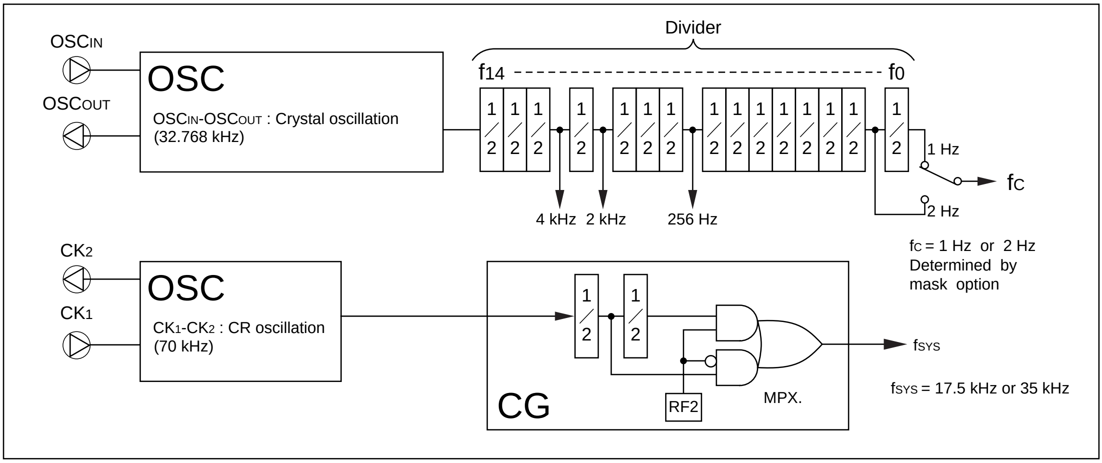


Fig. 8 System Clock Generator and Divider

Either of the system clock frequencies 35 kHz or 17.5 kHz (in case of CR oscillation) can be selected by the RF2 flag (See Table 2). The 17.5 kHz clock has slower command execution speed, but uses less power for the same function.

The system clock is initialized to 35 kHz after hardware reset operation.

The Table 2 shows the relationship between the contents of RF2 flag for OSC resonator and the generated frequency, f_{SYS}.

Table 2 OSC Resonator and Frequency f_{SYS}

FOR OSC RESONATOR	CONTENTS OF RF2 FLAG	GENERATED FREQUENCY f_{SYS}
70 kHz CR oscillation	0	35 kHz
	1	17.5 kHz
32.768 kHz crystal	0	16.384 kHz
	1	8.192 kHz

FUNCTIONAL DESCRIPTION
Voice Synthesizer

• How to select a voice start address

There is a voice start address and RC, composed of 8-bit to select a voice start address. Voice start address is 16 bits. However, RC register can points only upper 8 bits in voice start address in partial. Lower 8-bit is always fixed "0". Refer to "RC register".

Minimum unit (shortest block) is equal voice ROM capacity. Each minimum unit is composed of 256 steps. Refer to Fig.9.

Core CPU detects the status whether voice synthesizer run or not, by reading the content of $P3_3$ flag. ($P3_3$ flag is "1" during voice generation.)

Terminator (11111B) can be set as a voice data in the voice ROM.

When controller found a terminator, immediately stops voice and reset a flag.

When reached the bottom of the voice data address, voice data address automatically becomes 0000_H and voice continuously generates until come across a terminator.

CPU can reset the $P3_3$ flag and stop voice generation by force.

NOTE :

Voice ROM data "11111" means terminator of voice data. That is, an encoder must encode voice data except "11111".

• Voice sampling frequency (5 kHz / 7 kHz)

In case of sampling frequency is 5 kHz, total generation period becomes 12.8 s. In case of 7 kHz, it's 9.1 s.

Voice sampling frequency (5 kHz or 7 kHz) is selected by RD3 register. In case of RD3 is "0", voice sampling frequency becomes 7 kHz. In case of RD3 is "1", it's 5 kHz.

• RC register

The RC register is composed of 8-bit. It can points only upper 8 bits in the voice start address as shown below. The data is filled with both A and X registers.

NOTE :

A voice start address is corresponding to the RC register (8 bits). Maximum 256 (SM5M2) pieces of voice start address can be selected. Each voice start address is based on multiple number of 100H. When voice generates, P33 flag becomes "1".

• Voice start address

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0

NOTE : "x" stands for "0" or "1"

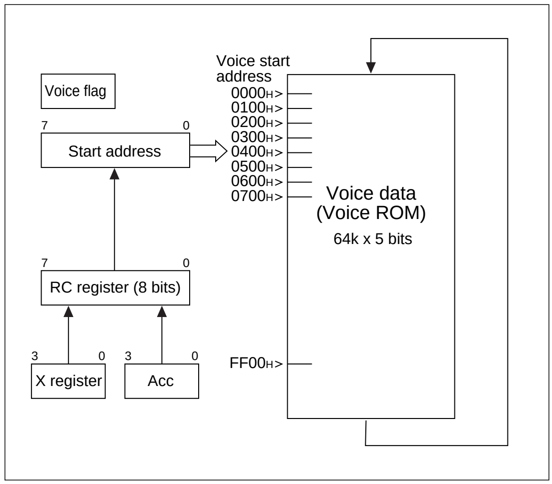


Fig. 9 Voice ROM Configuration

First set the sampling rate of the voice synthesizer. The voice synthesizer start address is corresponding to the RC register and the 8 bits in the RC register are obtained by A and X register. After setting the $P3_3$ voice flag High, the voice synthesizer would start playing. After detect $P3_3$ Low, the voice synthesizer could play the next section of voice.

Melody Output Function

The built-in melody generation circuit provides a variety of sound signals. Fig. 10 shows the block diagram of the melody generating circuit.

The melody ROM can store notes, rest and stop commands in 256-step (1 step consists of 6-bit), allowing the generation of 12-scale over two octaves (555 to 2 097 Hz) and the section of the time base for notes (125/62.5 ms).

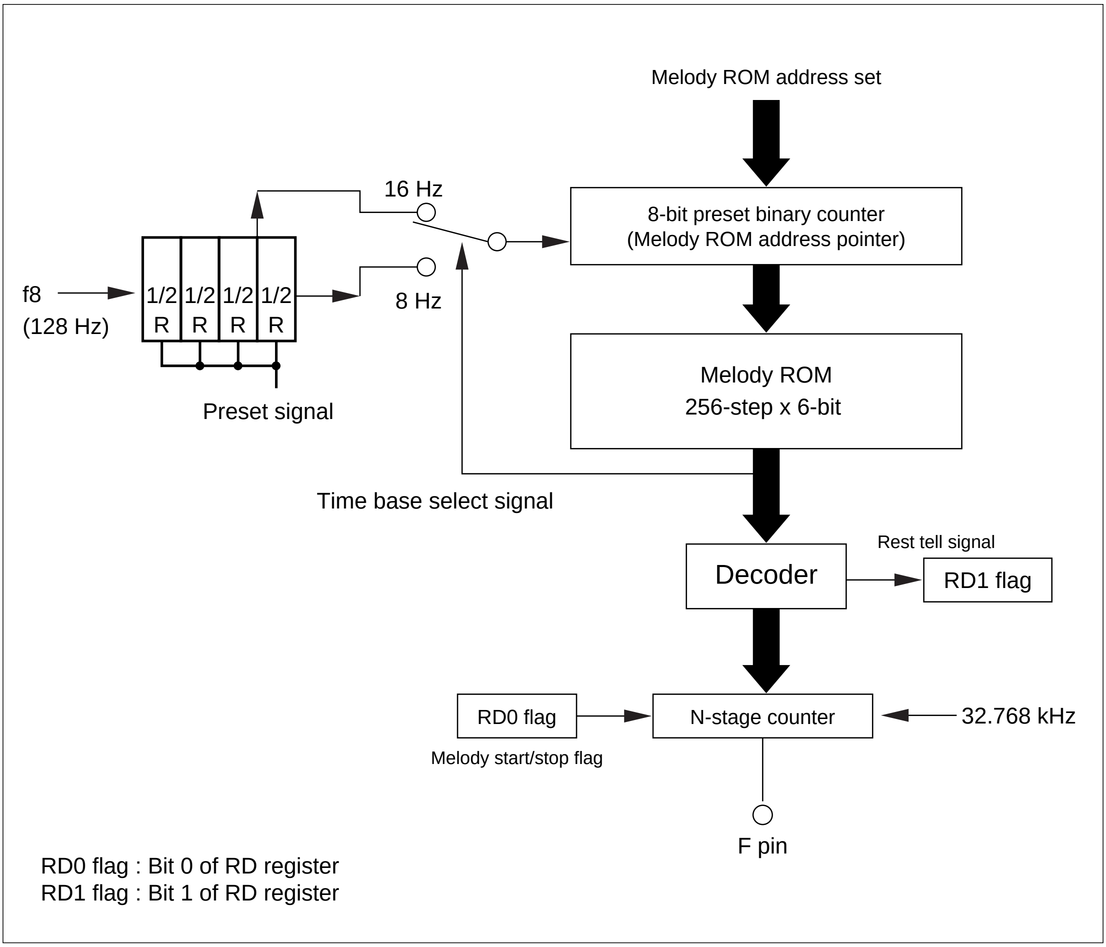


Fig. 10 Melody Generating Circuit

CONTROL PROCEDURE

The binary counter for designating the address of the melody ROM can be arbitrarily set using the PRE instruction. A performance is started and stopped by the RD0-flag to "1" and "0".

The stop code generates a "rest tell signal", and at the same time, sets the RD1 flag. The end of the melody can be found by testing the RD1 flag.

Accordingly, to stop a performance at the end of melody, the RD0 flag must be clear upon detection of RD1 flag = 1.

Next step of PRE instruction, put the NOP instruction.

The following is an example of a melody generating program.

```
MELO    LAX     2
        ATX
        LAX     1
        PRE                 ; Set the starting address of
                              the melody at the 21st.
                              Hexadecimal step.
        NOP                 ; Dummy command
        :
        :
        LBLX    0DH
        LAX     1
        OUT                 ; Start the melody
        TPB     1           ; Executed for clear the
                              RD1 flag
        NOP                 ; Dummy command
        :
        :
        LBLX    0DH
L1      TPB     1           ; Test the RD1 flag
        TR      L1          ; Loop for detect the stop
                              code
        LAX     0
        OUT                 ; Stop the melody
```

Using these functions, the user can generate music, sound effects, alarm signals, etc. as desired, and any portion of the music can be repeated. Table 3 lists the melody output frequencies. The output frequency can be halved by making bit 5 (OCT) of the melody ROM Low (0). In Table 3, m_0 to m_3 show data in bits 1 to 4 of the melody ROM.

Table 3 Melody Output Frequency

	m_3 m_2 m_1 m_0	OUTPUT FREQUENCY (Hz)	CLOCK NUMBER *1	*2
do	0 0 1 0	2114.1	15.5	7 8 8 8
si	0 0 1 1	1985.9	16.5	8 8 8 9
la#	0 1 0 0	1872.4	17.5	8 9 9 9
la	0 1 0 1	1771.2	18.5	9 9 9 10
sol#	0 1 1 0	1680.4	19.5	9 10 10 10
sol	0 1 1 1	1560.4	21.0	10 11 10 11
fa#	1 0 0 0	1489.5	22.0	11 11 11 11
fa	1 0 0 1	1394.4	23.5	11 12 12 12
mi	1 0 1 0	1310.7	25.0	12 13 12 13
re#	1 0 1 1	1236.5	26.5	13 13 13 14
re	1 1 0 0	1170.3	28.0	14 14 14 14
do#	1 1 0 1	1110.8	29.5	14 15 15 15

*1 Number of clocks for one cycle

*2 The number (n) in the waveforms represents the number of periods of the oscillation frequency (32.768 kHz) from the crystal oscillator for the duration in that particular part of the waveform.

MELODY ROM INSTRUCTION

The melody ROM instruction is composed of 6-bit. This 6-bit instruction (1 set), corresponding to a musical note, generates a sound signal.

I	OCT	m_3	m_2	m_1	m_0

I : Control the tone length. When “1”, 125 ms; when “0”, 62.5 ms.

OCT : When the octave is “1”, the frequency is determined by m_3 -m_0. When the octave is “0”, 1/2 of the frequency determined by m_3-m_0.

m_3 - m_0 : Frequency as shown in Table 3. Pause when $m_3 = m_2 = m_1 = m_0 = 0$, stop instruction when $m_3 = m_2 = m_1 = 0$, $m_0 = 1$.

EXAMPLE OF WRITING ON THE MELODY ROM

An example of writing a tone such as the following, on the melody ROM will be shown.

MUSICAL SCALE	TONE LENGTH (ms)	OCT	m_3	m_2	m_1	m_0
sol	375	0	0	1	1	1
la	125	0	0	1	0	1
sol	250	0	0	1	1	1
mi	250	0	1	0	1	0
do	375	1	0	0	1	0
re	125	1	1	1	0	0
do	250	1	0	0	1	0
la	250	0	0	1	0	1

ADDRESS	DATA	MUSICAL NOTE INSTRUCTION
00	00	pause
01	27	sol
02	27	sol
03	27	sol
04	25	la
05	27	sol
06	27	sol
07	2A	mi
08	2A	mi
09	22	do
0A	22	do
0B	22	do
0C	3C	re
0D	22	do
0E	22	do
0F	25	la
10	25	la
11	01	stop

The tone length of an initial musical note which is generated from ROM addressed data assigned by a PRE instruction has an error of maximum ±4 ms. Therefore, by applying a pause as an initial note, a melody performs with a precisely regulated tone length.

Standby Function

A standby function is available which temporarily stops program execution to conserve power consumption. The state during which a program is in execution is called the operation mode and the state during which the execution is temporarily stopped is called the standby mode.
Either CR or X'TAL oscillator can be selected to a system clock generator circuit of SM5M2. Each standby mode between CR + X'TAL and only X'TAL is entirely different as tables shown below.
In case of CR + X'TAL oscillator, HALT instruction can NOT be used, only STOP instruction is available.
On the other hand, in case of only X'TAL oscillator, both HALT and STOP instruction are available.

Table 4 CR + X'TAL Standby Mode

	Standby mode		Register status			Chip's status					
STATUS	**STOP**	**HALT**	**$P3_3$**	**RF1**	**RD2**	**CR**	**X'TAL**	**CPU**	**Voice**	**LCD**	**Divider**
CRSTOP1	1		0	1	0	OFF	ON	OFF	OFF	ON	ON
CRSTOP2	1		0	0	0	OFF	ON	OFF	OFF	OFF	ON
CRSTOP3	1		0	0	1	OFF	ON	OFF	OFF	OFF	OFF

NOTES :
- When CR = OFF, CPU and Voice are OFF.
- When Divider = OFF, neither LCD nor Melody is in operation (undefined).
- STOP = 1 stands for executing STOP instruction.

Table 5 Only X'TAL Standby Mode

	Standby mode		Register status			Chip's status					
STATUS	**STOP**	**HALT**	**$P3_3$**	**RF1**	**RD2**	**CR**	**X'TAL**	**CPU**	**Voice**	**LCD**	**Divider**
XTALHALT1	0	1	0	1	0	OFF	ON	OFF	OFF	ON	ON
XTALHALT2	0	1	0	0	0	OFF	ON	OFF	OFF	OFF	ON
XTALHALT3	0	1	0	0	1	OFF	ON	OFF	OFF	OFF	OFF
XTALSTOP	1	0	0	0	0	OFF	OFF	OFF	OFF	OFF	OFF

NOTES :
- When CR = OFF, CPU and Voice are OFF.
- When Divider = OFF, neither LCD nor Melody is in operation (undefined).
- STOP = 1 stands for executing STOP instruction.
- HALT = 1 stands for executing HALT instruction.

To get a condition mentioned in the first to the sixth boxes from right hand side, one of STOP or HALT instruction must be executed under the condition mentioned in the fourth to the sixth boxes from left hand side.
For instance, to get the status of XTALHALT1, of which contents are CR = OFF, X'TAL = ON, CPU = OFF, Voice = OFF, LCD = ON, and Divider = ON in the only X'tal standby mode, HALT instruction must be executed under the condition of $P3_3$ = 0, RF1 = 1 and RD2 = 0.

NOTE :
The halt mode stops only system clock generator circuit. This mode is used to activate the system immediately after a condition causes a release to the operation mode.

During the standby mode, the contents of the RAM and stack RAM are retained. The contents of the flags, registers and output latches shown below are also retained.

FLAG
IFA flag
IFD flag
IME flag
C flag
$P3_3$ flag

REGISTER
A_{CC}
X register
B_M, B_L register
SB register
SP
SR
RC, RD, RE, RF

OUTPUT LATCH REGISTER
P0 register
P1 register
P2 register

A release from the standby mode to the operation mode is performed by a reset port input, an interrupt from the nonmaskable INTA, any port High in Port 1, and divider. A maskable interrupt request cannot become a factor in releasing back to the operation mode. The mask setting is performed with RE register. (see Table 1)

CAUTION :

When all of $P1_0$ to $P1_3$ level are High, the SM5M2 is performed to release the standby mode and enter normally hardware reset operation. (Mask option)

TRANSITION FROM THE OPERATION MODE TO THE STANDBY MODE

The HALT instruction is executed to set the halt mode and the STOP instruction is executed to set the stop mode.

Since the interrupt is used to release from the standby mode, the mode does not transfer to the standby mode if any of the following conditions are satisfied during execution of the STOP or HALT instruction.

a) RE0 is set and the INTA level is High.

b) RE2 is set and the IFD flag is set.

If any of the conditions above is satisfied, the mode does not transfer to the standby mode even if the STOP or HALT instruction is executed and the instruction at the address following that of the STOP or HALT instruction is executed. Therefore, place the JUMP instruction which specifies step 0 on page 3 to the location at the address following that of the STOP or HALT instruction.

RELEASE FROM THE STANDBY MODE TO THE OPERATION MODE

Release based on an interrupt request from the INTA pin or divider overflow. However, the reset or any port High in Port 1 is limited to a nonmaskable interrupt request.

The program restarts from step 0 on page 3. However, if the IME flag is set, the instruction at step 0 on page 3 is executed and a subroutine jump is performed to the interrupt processing routine specified on page 2 according to the type of interrupt.

Even if Low level input on INTA pin is removed before 900 command cycles, the stop mode is released.

However, the program will not jump to 20_H page (interrupt process routine).

Interrupt request flag IFA is not set : the program continues at step 0 of page 3.

Interrupts

Interrupts originate from an INTA input or divider overflow. The IFA and IFD flags become interrupt request flags.

The interrupt block is composed of mask flags (RE0, RE2), the IME flag and interrupt processing circuit.

As shown in Fig. 11, resetting a mask flag enables the interrupt request flag to be independently masked. Thus, the mask flags can be used in a program to establish the interrupt priority. The priority for interrupts generated simultaneously is shown in Table 6.

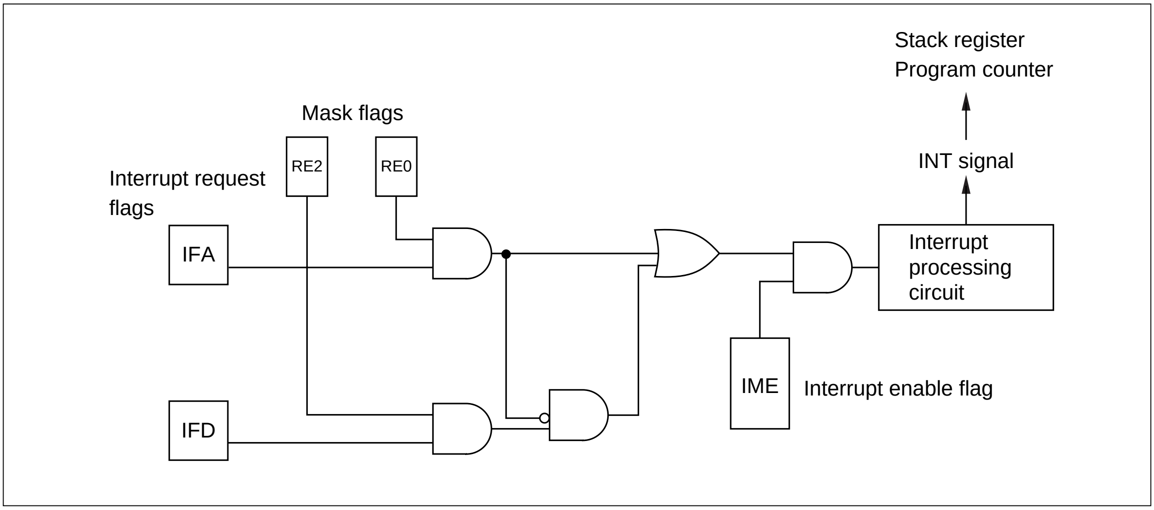


Fig. 11 Interrupt Block

Table 6 Interrupt Event Summary

INTERRUPT REQUEST (REQUEST FLAG)	JUMP DESTINATION		PRIORITY ORDER	INTERRUPT ENABLE FLAG
	PAGE	STEP		
INTA input (IFA)	2	0	1	RE0
Divider overflow (IFD)	2	4	2	RE2

When the IME flag is set, the interrupt circuit activates according to the interrupt request and a subroutine jump is performed to the specified address. The jump destinations according to interrupt origin are shown in Table 6. When the IME flag is cleared, an interrupt is not accepted even if an interrupt request is generated. The interrupt timing is shown in Fig. 12 and Fig. 13. The timing chart shown in Fig. 12 shows the interrupt enable state when an interrupt request has been generated. In this case, the interrupt processing signal INT goes High, one instruction cycle after the interrupt request flag is set. When INT goes High, the contents of the program counter are pushed into the stack register and execution jumps to the specified address. At this time, the INT signal and the IME flag are cleared to establish the interrupt disable mode. The IME flag is set again when the RTNI instruction is executed to establish the interrupt enable mode.

The timing chart shown in Fig. 13 shows the state when interrupts are enabled while multiple interrupts are generated. In this case, a subroutine jump is performed according to the interrupt having the highest priority. When returning from the subroutine by executing the RTNI instruction, the instruction (two words are executed for a two-word instruction) at the location of return is executed and the interrupt for the next highest priority is accepted.

If an interrupt request is generated during execution of a two-cycle instruction, the instruction is executed after which interrupt processing is performed. If consecutive LAX instructions are skipped or if the SKIP conditions are satisfied, the skip operation is terminated after which interrupt processing is performed.

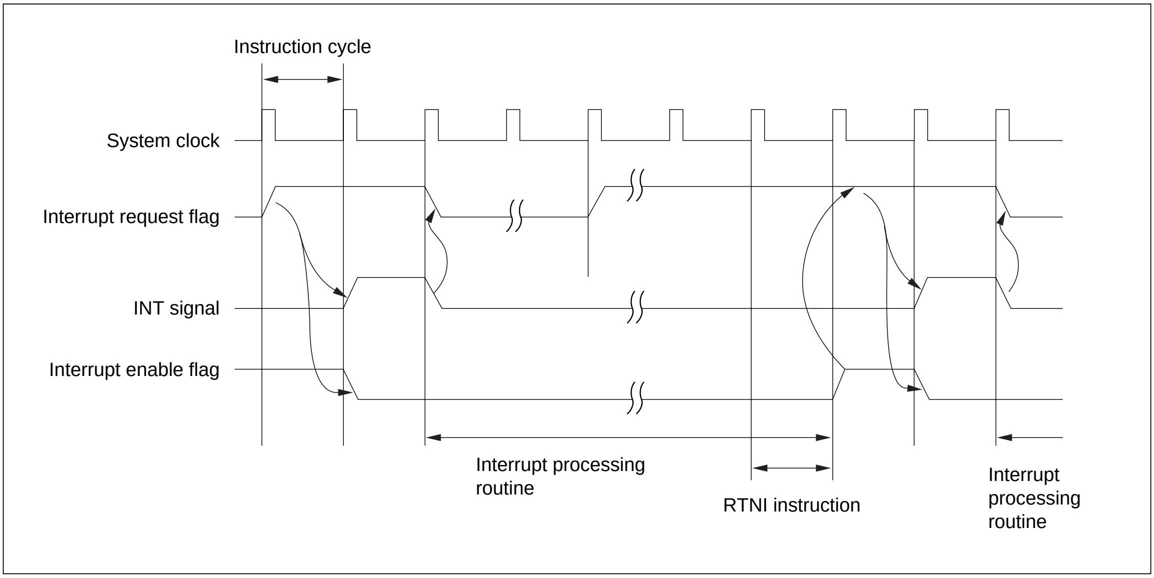


Fig. 12 Interrupt Timing Chart

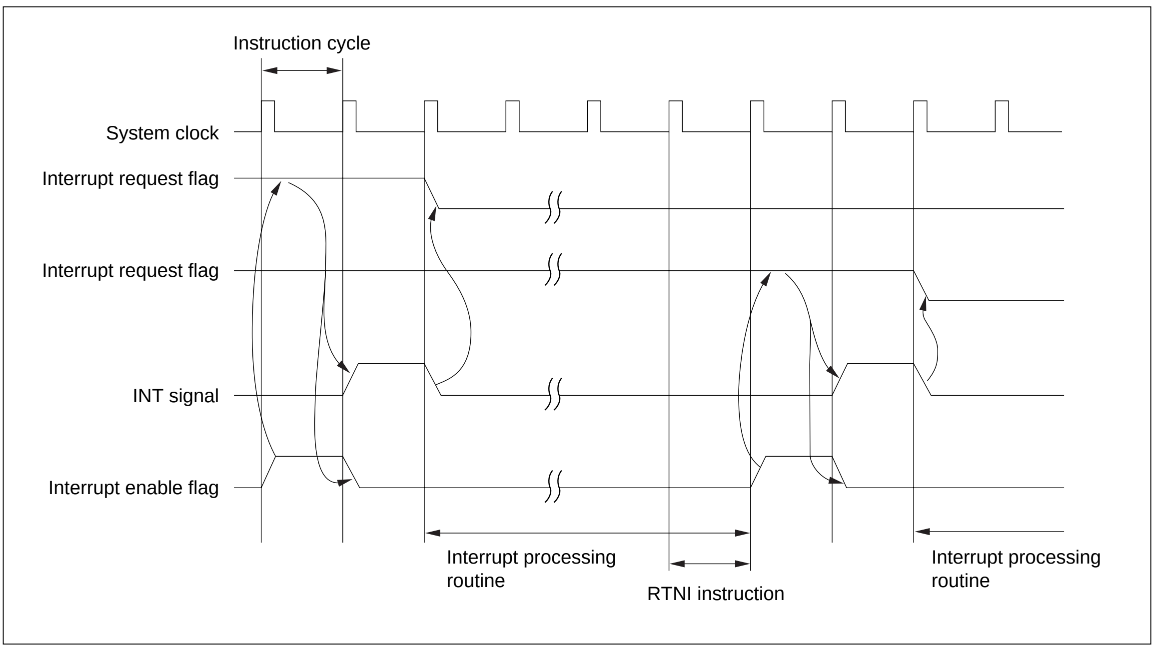


Fig. 13 Interrupt Timing Chart

NOTE :

Fig. 12 and Fig. 13 show the case where the interrupt request flags are not masked.

Hardware Reset Function

The hardware reset function mode activated two instruction cycles after the falling edge from the $\overline{RESET}$ pin. When the $\overline{RESET}$ pin is changed from High to Low, the pulse which is input by the OSC_{IN} pin is counted 2^{15} times after which the reset mode clears and the program counter starts from address 0 on page 0.

The initialized status of the system after reset is shown in Table 7.

The following reset functions are available.

- The I/O port is set as an input port and the mode register RC, RD, RE and RF are cleared. The output only port (P0) is cleared and output Low.
- The interrupt request flags (IFA, IFD) and the interrupt enable flag (IME) are cleared and all interrupts become disabled.
- The program counter start from step 0 on page 0.

For activate reset function, when power is turned on, you must connect a capacitor (0.1 µF, TYP.) across the $\overline{RESET}$ pin and GND.

Table 7 Reset Status

FLAG OR REGISTER, X-REGISTER	STATUS (in reset mode and at program start)
PC	0
SP	Level 1
RAM	Undefined
A_{CC}	Undefined
X-register	Undefined
P0-P2 output latch registers	0
Divider	0
IFA flag	0
IFD flag	0
IME flag	0
$P3_3$ flag	0
C flag	Undefined
B_M, B_L registers	Undefined
Register RC (bit 7-0)	0
Register RD (bit 1)	Undefined
Register RD (bit 0)	0
Register RD (bit 3)	0
Register RD (bit 2)	0
Register RE (bit 2, 1, 0)	0
Register RF (bit 3, 2, 1, 0)	0

NOTES :

- Undefined flags and registers should be initialized by software.
- When all of P1 pins ($P1_0$ to $P1_3$) level goes to High, the SM5M2 is performed to reset operation. (Mask option)

LCD Function

• Display segment

The SM5M2 contains a built-in circuit which directly drive a 1/4 duty, 1/2 bias LCD.
A sample LCD pattern is shown in Fig. 14.

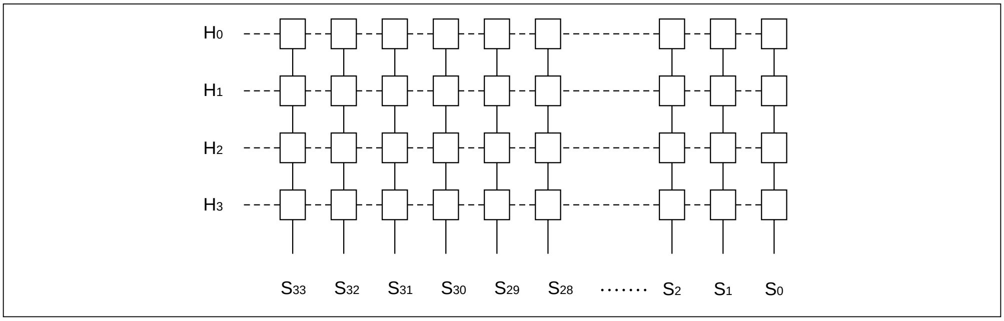


Fig. 14 LCD Pattern

A segment of the LCD can be turned on or off by setting the corresponding bit in the display RAM (see Fig. 5) to "1" or "0". The displayed segments can assume any configuration containing up to a maximum of 136 segments. An example of a 7-segment numeric display is shown in Fig. 15.

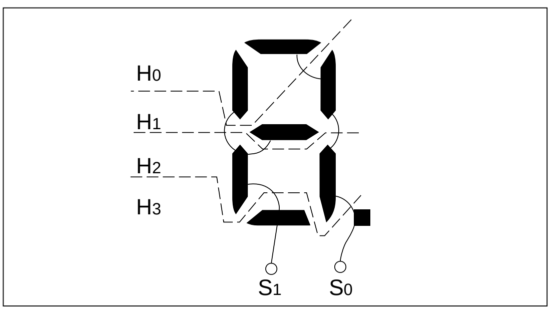


Fig. 15 Sample LCD Pattern for 7-Segment Numeric Display

• LCD drive waveforms

The LCD drive waveforms for the LCD pattern of Fig. 15 displaying a "5" are shown in Fig. 16 (the segment output uses S_0 and S_1). For Fig. 16, 3 V is applied to the V_{DD} pin, and 1.5 V is applied to the V_{OA} pin.

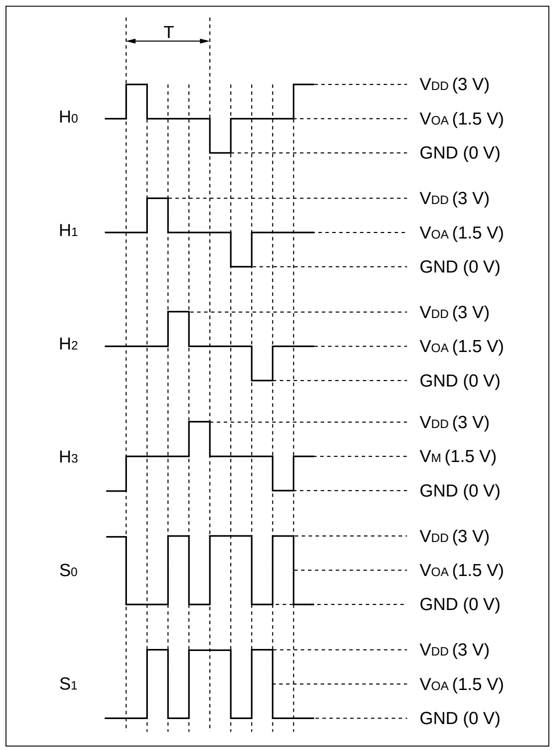


Fig. 16 LCD Drive Waveforms
(frame frequency = 1/T = 64 Hz or 128 Hz)

∗ Frame frequency is selectable by mask option.

• V_{OA} pin

Bleeder resistors are built-in to drive the LCD at 1/2 bias. The bieeder resistors have the configuration shown in Fig. 17. When bit 1 of the RF registor is set and V_{DD} is 3 V, V_{OA} output 1.5 V.

Normally, the V_{OA} pin is used in its open state. To drive an LCD with a large display area, the leading edge of the LCD drive waveform can be improved by connecting capacitor across the V_{OA} pin and V_{DD}. The same effect can be obtained by connecting capacitor across the V_{OA} pin and GND.

When bit 1 of the RF register is set "0", V_{OA} drop to GND level to reduce power consumption. At the same time, the H_0-H_3 and S_0-S_{33} pin are GND level.

• Booster circuit

It is necessary to apply external capacitors between V_{DD} pin and V_{OA} pin. (see Fig. 18)

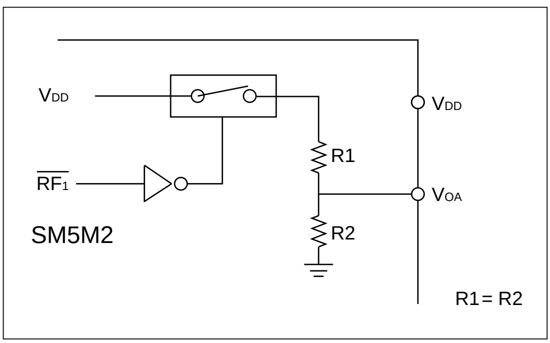


Fig. 17 Booster Circuit

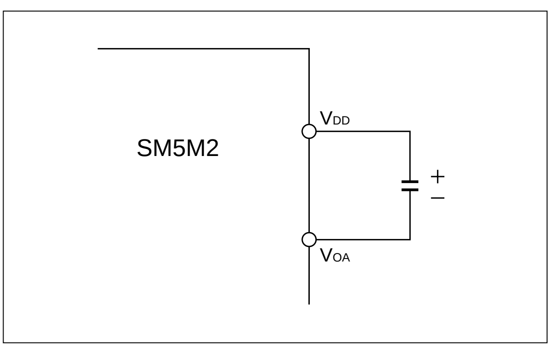


Fig. 18 Externally Connected Capacitor Circuit

• Blank display

There are two way to blank the entire display to match the purpose.

(a) Blanking the display for a short time.

Set bit 0 of the RF register to "1" : Display
Set bit 0 of the RF register to "0" : Blank state

(b) Blanking the display for a long period mainy to reduce supply current.

Set bit 0 and 1 of the RF register to "1" : Display
Set bit 0 and 1 of the RF register to "0" : Blank state

When bit 1 of the RF register is set "0", the voltage (V_{DD}) applied to the bleeder resistors is turned off and common outputs and segment outputs are dropped to GND level so that the display blanks. By cutting off the bleeder supply, the current consumption can be greatly reduced. However, when the display is blanked using method (b), the response speed of the LCD returning to the display state drops slightly. The RF register is in the blank state after initialization (reset state) from hardware reset.

INSTRUCTION SET

Definition of Symbols

The following symbols are used in descriptions for the instructions.

M	: Contents of RAM at the address specified by the B register
←	: Transfer direction
∪	: Logical OR
∩	: Logical AND
⊕	: Logical XOR
Ai	: ith bit of the A_{CC}
Push	: Content of the PC are decremented to the stack register.
Pop	: The decremented contents are transferred back to the PC.
Pj	: Pj register (j = 3, 2, 1, 0)
Rj	: Rj register (j = F, E, D)
ROM ( )	: ROM contents for address within ( )
Cy	: Carry of ALU (different from the C flag)

- Each bit of a register can be represented. For example, the ith bit of X register and R(0) register are represented as Xi and R(0) i. (i = 0, 1, 2, 3, ...)
- Increment and decrement denote the binary addition of 1_H and F_H, respectively.
- To skip a certain instruction means that the instruction is ignored and that no operation is performed until the execution transfers to the next instruction. In other words, the instruction is regarded as a NOP instruction. Therefore, one cycle is required to skip a one-word instruction and two cycles are required to skip a two-word instruction.

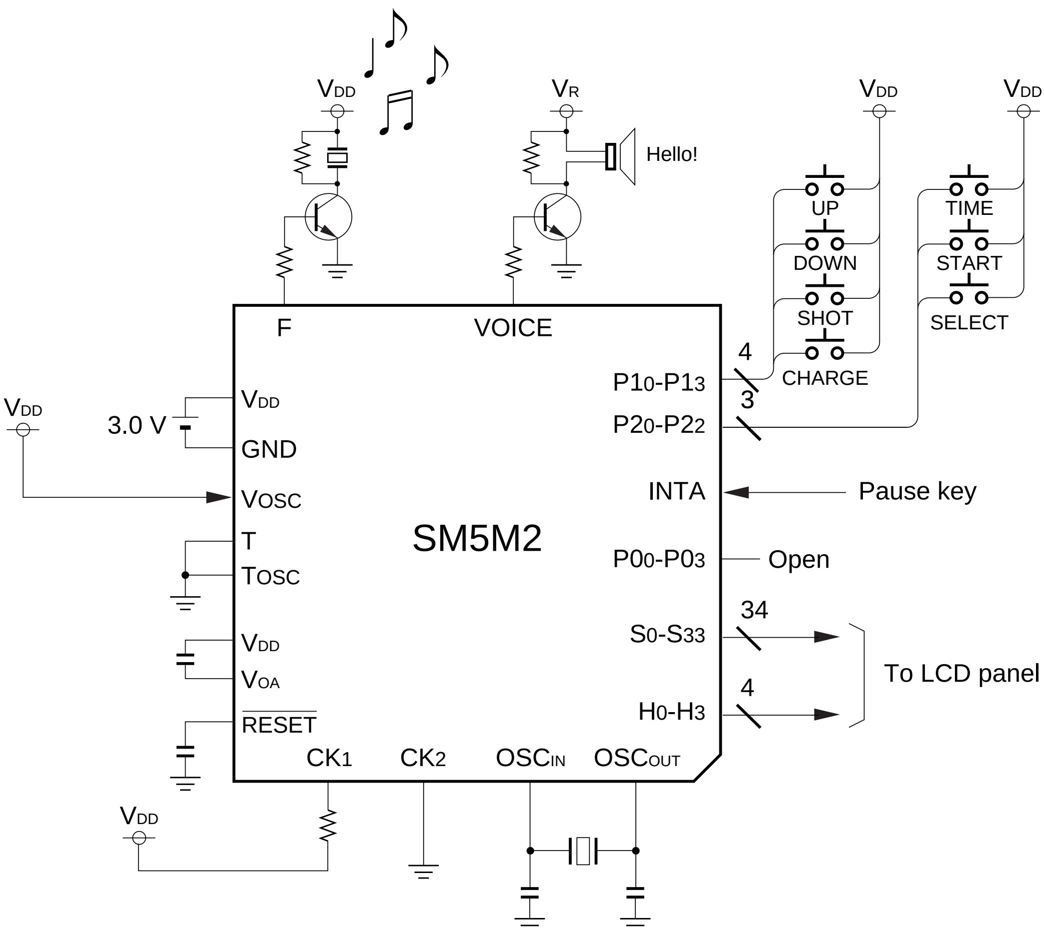
Instruction Summary

MNEMONIC	MACHINE CODE	OPERATION
ROM Address Control Instructions		
TR x	80 to BF	$P_L \leftarrow x\ (I_5\text{-}I_0)$
TL xy	E0 to EF 00 to FF	$P_U \leftarrow x\ (I_{11}\text{-}I_6)$ $P_L \leftarrow y\ (I_5\text{-}I_0)$
TRS x	C0 to DF	Push, $P_U \leftarrow 01_H$ $P_L \leftarrow x\ (I_4, I_3, I_2, I_1, I_0, 0)$
CALL xy	F0 to FF 00 to FF	Push, $P_U \leftarrow x\ (I_{11}\text{-}I_6)$ $P_L \leftarrow y\ (I_5\text{-}I_0)$
RTN	7D	Pop
RTNS	7E	Pop, Skip the next step
RTNI	7F	Pop, IME $\leftarrow$ 1
Data Transfer Instructions		
LAX x	10 to 1F	$A_{CC} \leftarrow x\ (I_3\text{-}I_0)$
LBMX x	30 to 2F	$B_M \leftarrow x\ (I_3\text{-}I_0)$
LBLX x	20 to 2F	$B_L \leftarrow x\ (I_3\text{-}I_0)$
LDA x	50 to 53	$A_{CC} \leftarrow M$ $B_{Mi} \leftarrow B_{Mi} \oplus x\ (I_1, I_0)\ (i = 1, 0)$
EXC x	54 to 57	$M \leftrightarrow A_{CC}$ $B_{Mi} \leftarrow B_{Mi} \oplus x\ (I_1, I_0)\ (i = 1, 0)$
EXCI x	58 to 5B	$M \leftrightarrow A_{CC}$, $B_L \leftarrow B_L+1$ $B_{Mi} \leftarrow B_{Mi} \oplus x\ (I_1, I_0)\ (i = 1, 0)$ Skip if Cy = 1 ($B_L = 0F_H \rightarrow 0$)
EXCD x	5C to 5F	$M \leftrightarrow A_{CC}$, $B_L \leftarrow B_L+0F_H$ $B_{Mi} \leftarrow B_{Mi} \oplus x\ (I_1, I_0)\ (i = 1, 0)$ Skip if Cy = 1 ($B_L = 0 \rightarrow 0F_H$)
EXAX	64	$A_{CC} \leftrightarrow X$
ATX	65	$x \leftarrow A_{CC}$
EXBM	66	$B_M \leftrightarrow A_{CC}$
EXBL	67	$B_L \leftrightarrow A_{CC}$
EX	68	B $\leftrightarrow$ SB
Arithmetic Instructions		
ADX x	00 to 0F	$A_{CC} \leftarrow A_{CC}+x\ (I_3\text{-}I_0)$, Skip if Cy = 1
ADD	7A	$A_{CC} \leftarrow A_{CC}+M$
ADC	7B	$A_{CC} \leftarrow A_{CC} + M+C$, $C \leftrightarrow Cy$ Skip if Cy = 1
COMA	79	$A_{CC} \leftarrow \overline{A_{CC}}$
INCB	78	$B_L \leftarrow B_L+1$, Skip if $B_L = 0F_H$
DECB	7C	$B_L \leftarrow B_L-1$, Skip if $B_L = 0$

MNEMONIC	MACHINE CODE	OPERATION
Test Instructions		
TAM	6F	Skip if A_{CC} = M
TC	6E	Skip if C = 1
TM x	48 to 4B	Skip if Mi = 1 (i = 3 to 0)
TABL	6B	Skip if A = B_L
TPB x	4C to 4F	Skip if P (R) i = 1 (i = I_1, I_0)
TA	6C	Skip if IFA = 1, and (IFA $\leftarrow$ 0)
TD	69 02	Skip if IFD = 1, and (IFD $\leftarrow$ 0)
Bit Manipulation Instructions		
SM x	44 to 47	Mi $\leftarrow$ 1 (i = 3 to 0)
RM x	40 to 43	Mi $\leftarrow$ 0 (i = 3 to 0)
SC	61	C $\leftarrow$ 1
RC	60	C $\leftarrow$ 0
IE	63	IME $\leftarrow$ 1
ID	62	IME $\leftarrow$ 0
I/O Control Instructions		
INL	70	$A_{CC} \leftarrow$ P1i (i = 3 to 0)
OUTL	71	P0i $\leftarrow A_{CC}$ (i = 3 to 0)
ANP	72	Pj $\leftarrow$ Pj $\cap$ A_{CC} (j = 3 to 0)
ORP	73	Pj $\leftarrow$ Pj $\cup$ A_{CC} (j = 3 to 0)
IN	74	$A_{CC} \leftarrow$ Pj (j = 3, 2, 1)
OUT	75	Pj $\leftarrow A_{CC}$ (j = 3 to 0) Rj $\leftarrow A_{CC}$ (j = F to D)
Table Reference Instruction		
PAT	6A 00 to FF	Push $P_U \leftarrow (0, 4)$, $P_L\ (X_1, X_0, A_{CC})$ $(X, A_{CC}) \leftarrow I_7\text{-}I_0$ Pop
Divider Instructions		
DR	69 03	DIV ($f_7\text{-}f_0$) Reset
DTA	69 04	$A_{CC} \leftarrow$ Divider (f_3 to f_0)
Melody Control Instruction		
PRE	6D	Melody ROM pointer preset Melody ROM pointer $\leftarrow$ X, A
Special Instructions		
STOP	76	Standby mode (STOP)
HALT	77	Standby mode (HALT)
NOP	00	No operation

SYSTEM CONFIGURATION EXAMPLE

- **Handheld LCD game**



72 QFP (QFP072-P-1010)