SINO WEALTH

# SH6613D

## 4K 4-Bit Microcontroller with LCD Driver

### Features

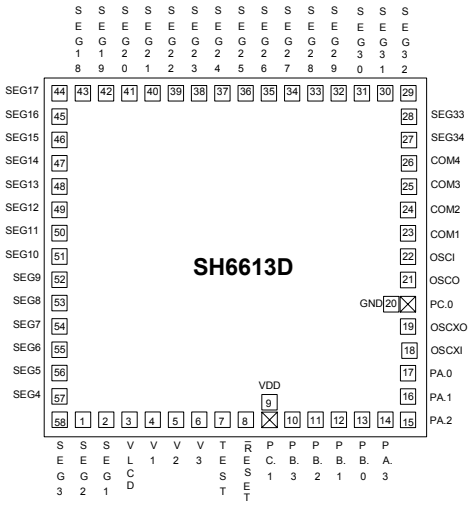
- SH6610C-based single-chip 4-bit micro-controller
- ROM: 4096×16 bits
- RAM: 512×4 bits
- Operation voltage: 2.4V – 6.0V
- 8 CMOS bi-directional I/O pins
- 4-Level subroutine nesting (include interrupts)
- One 8-bit auto re-load timer/counter
- Warm-up timer for power-on reset
- Powerful interrupt sources:
  - External interrupts ( $\overline{INT0}$ ).
  - Internal interrupt (Timer0).
  - Internal interrupt (Base Timer).
  - Port's falling edge interrupt: PORTB ( $\overline{INT1}$ )
- 8-bit Base timer
- LCD driver: 136 dots(1/4 duty 1/3 bias)
- LCD used as scan output
- Built-in dual tone PSG with one noise generator
- 2 Clock source
  OSC: (code option select crystal or RC type)
  - Crystal oscillator 32.768K
  - RC oscillator: 262K

  OSCX: (system register select ceramic or RC type)
  - Ceramic oscillator 455K
  - RC oscillator 1.8M or 2M
- Instruction cycle time:
  - 122.07μs for 32.768 kHz crystal
  - 15.27μs for 262 kHz RC
  - 8.79μs for 455KHz ceramic
  - 2.22μs for 1.8 MHz RC
  - 2μs for 2.0 MHz RC
- Two low power operation mode: HALT and STOP
- Low power consumption

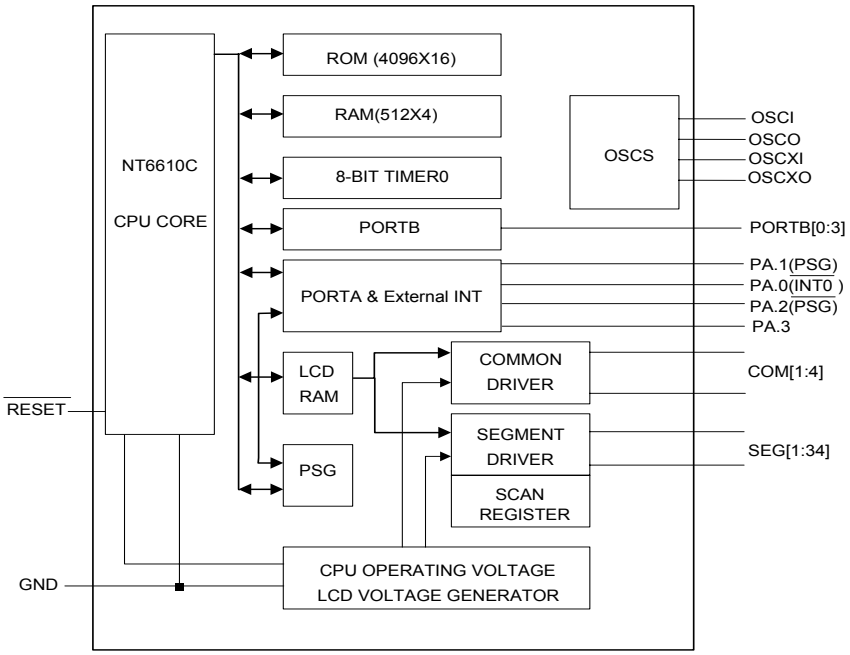
### General Description

SH6613D is a single chip microcontroller integrated with SRAM, timer and dual-tone PSG, LCD driver and I/O port. This chip builds in a dual-oscillator to enhance the total chip performance.

### Pad Diagram



## Block Diagram




## Pad Description

| Pad No. | Designation | I/O | Description |
|---|---|---|---|
| 1,2,27~58 | SEG1~SEG34 | O | Segment signal output for LCD display;<br>Share with scans output. |
| 3~6 | $V_{LCD}$,$V_1$,$V_2$,$V_3$ | I | Connect with external LCD divided resistance |
| 7 | TEST | I | Test pin (Internal pull-low). No connect for user. |
| 8 | $\overline{RESET}$ | I | Reset input(No internal pull-up) |
| 9 | $V_{DD}$ | P | Power supply. |
| 10~13 | Port B.3~Port B.0 | I/O | Bit programmable I/O ,Vector interrupt( $\overline{INT1}$ ) |
| 14~17 | Port A.3~Port A.0 | I/O | Bit programmable I/O,PA.0 shared with $\overline{INT0}$<br>PA.1 , PA.2shared with PSG output |
| 18 | OSCXI | I | Oscillator X input pin |
| 19 | OSCXO | O | Oscillator X output pin |
| 20 | GND | P | Ground pin |
| 21 | OSCO | O | Oscillator output pin |
| 22 | OSCI | I | Oscillator input pin |
| 23~26 | COM1~COM4 | O | Common signal output for LCD display |

## Functional Description

### 1. CPU

The CPU core contains the following function blocks: Program Counter, ALU, Carry Flag, Accumulator, Table Branch Register (TBR), Data Pointer (INX, DPH, DPM and DPL), and Stack.

**1.1** PC (Program Counter)

The PC is used for ROM addressing consisting of 12-bits:

Page Register (PC11), and Ripple Carry Counter (PC10 -- PC0).

The program counter normally increases by one (+1) with each execution of an instruction except in the following cases:

1) When executing a jump instruction (such as JMP, BA0, BAC),
2) When executing a subroutine call instruction (CALL),
3) When an interrupt occurs,
4) When the chip is at INITIAL RESET.

The program counter is loaded with data corresponding to each instruction. The unconditional jump instruction (JMP) can be set at 1-bit page register for higher than 2K. Program Counter can only address a 4K program ROM.

**1.2** ALU and CY

ALU performs arithmetic and logic operations.

The ALU provides the following functions:

Binary addition/subtraction

(ADC, SBC, ADD, SUB, ADI, SBI)

Decimal adjustment for addition/subtraction (DAA, DAS), Logic operations (AND, EOR, OR, ANDI, EORI, ORI) Decision (BA0, BA1, BA2, BA3, BAZ, BAC)

The Carry Flag (CY) holds the arithmetic operation ALU overflow.

During interrupt or call instruction, carry is pushed into stack and restored from stack by RTNI. It is unaffected by an RTNW instruction.

**1.3** Accumulator

Accumulator is a 4-bit register holding the results of the arithmetic logic unit. In conjunction with ALU, data transfers between the accumulator and system register, LCD RAM, or data memory can be performed.

**1.4** Stack

A group of registers used to save the contents of CY & PC (11-0) sequentially with each subroutine call or interrupt. It is organized 13 bits X 4 levels. The MSB is saved for CY. 4 levels are the maximum allowed for subroutine calls and interrupts.

The contents of Stack are returned sequentially to the PC with the return instructions (RTNI/RTNW). Stack is operated on a first-in, last-out basis. This 4-level nesting includes both subroutine calls and interrupts requests. Note that program execution may enter an abnormal state if the number of calls and interrupt requests exceeds 4, and the bottom of stack will be shifted out.

### 2. ROM

SH6613D can address 4096×16 bit of program area $000 to $FFF. There is an area from addresses $000 through $004 that is reserved for special interrupts service routines as starting vector address.

| Address | Instruction | Function |
|---|---|---|
| 000H | JMP Instruction | Jump to RESET service routine |
| 001H | JMP Instruction | Jump to $\overline{\text{INT0}}$ service routine |
| 002H | JMP Instruction | Jump to Timer0 service routine |
| 003H | JMP Instruction | Jump to Base Timer service routine |
| 004H | JMP Instruction | Jump to $\overline{\text{INT1}}$ service routine |

**3. RAM**

Built-in SRAM contains general-purpose data memory, LCD RAM, and system registers. They can be accessed by direct addressing in one instruction.

The following is the memory allocation map:

$000~$01F: System register and I/O; $020~$1FF: Data memory (480×4bits,partitioned into 4 banks).

$300~$321, $350~$36D: LCD RAM space (34×4 bits).

The configuration of system register

| Address | Bit3 | Bit2 | Bit1 | Bit0 | Function | Initial Value | R/W |
|---|---|---|---|---|---|---|---|
| $00 | IEX | IET0 | IEBT | IEP | Interrupt enable flags | 0000 | R/W |
| $01 | IRQX | IRQT0 | IRQBT | IRQP | Interrupt request flags | 0000 | R/W |
| $02 | TM0.3 | TM0.2 | TM0.1 | TM0.0 | Timer0 mode register | 0000 | R/W |
| $03 | BTM.3 | BTM.2 | BTM.1 | BTM.0 | Base timer mode register | 0000 | R/W |
| $04 | T0L.3 | T0L.2 | T0L.1 | T0L.0 | Timer0 load/counter low nibble | 0000 | R/W |
| $05 | T0H.3 | T0H.2 | T0H.1 | T0H.0 | Timer0 load/counter high nibble | 0000 | R/W |
| $06~$07 | - | - | - | - | Reserved | - | - |
| $08 | PA.3 | PA.2 | PA.1 | PA.0 | PORTA | 0000 | R/W |
| $09 | PB.3 | PB.2 | PB.1 | PB.0 | PORTB | 0000 | R/W |
| $0A | - | - | PC.1 | PC.0 | Bonding option | 01(default) | R |
| $0B | PACR.3 | PACR.2 | PACR.1 | PACR.0 | Set PORTA to be output port | 0000 | W |
| $0C | PBCR.3 | PBCR.2 | PBCR.1 | PBCR.0 | Set PORTB to be output port | 0000 | W |
| $0D | - | - | - | - | Reserved | - | - |
| $0E | TBR.3 | TBR.2 | TBR.1 | TBR.0 | Table branch register | 0000 | R/W |
| $0F | INX.3 | INX.2 | INX1 | INX.0 | Index register(INX) | 0000 | R/W |
| $10 | DPL3 | DPL2 | DPL1 | DPL0 | Data pointer for INX low nibble | 0000 | R/W |
| $11 | - | DPM.2 | DPM.1 | DPM.0 | Data pointer for INX middle nibble | 0000 | R/W |
| $12 | - | DPH.2 | DPH.1 | DPH.0 | Data pointer for INX high nibble | 0000 | R/W |
| $13 | PPULL | PAM2 | PAM1 | HLM | Bit1,2:PA.1 & PA.2 as PSG output or I/O PORT<br>Bit0:Heavy load mode<br>Bit3:Port pull-up control | 0000 | R/W |
| $14 | OXS | - | OXM | OXON | Bit0:Turn on OSCX oscillator<br>Bit1:CPU clocks select (1:OSCX/0:OSC)<br>Bit3:OSCX type selection | 0000 | R/W |
| $15 | LPS1 | LPS0 | LCDOFF | **Should be set "1"** | Bit0:**Programmer should be set "1"**<br>Bit1:LCD off<br>Bit2,3:LCD frequency control | 0000 | R/W |
| $16 | LPD | O/S | - | - | Bit2:Set LCD segment as output<br>Bit3:LCD Power degrade | 0000 | R/W |
| $17 | C1.3 | C1.2 | C1.1 | C1.0 | PSG channel 1 low nibble | 0000 | W |
| $18 | OCT1 | C1.6 | C1.5 | C1.4 | PSG channel 1high nibble<br>Bit3:channel 1 octave shift control | 0000 | W |
| $19 | C2.3 | C2.2 | C2.1 | C2.0 | PSG channel 2 nibble 1 or alarm output | 0000 | W |
| $1A | C2.7 | C2.6 | C2.5 | C2.4 | PSG channel 2 nibble 2 | 0000 | W |
| $1B | C2.11 | C2.10 | C2.9 | C2.8 | PSG channel 2 nibble 3 | 0000 | W |
| $1C | OCT2 | C2.14 | C2.13 | C2.12 | PSG channel 2 nibble 4<br>Bit3:channel 2 octave shift control | 0000 | W |
| $1D | VOL1 | VOL0 | CH2EN | CH1EN | Bit0,Bit1:Channel 1,2 enable<br>Bit2,Bit3:volume control | 0000 | W |
| $1E | SEL1 | SEL0 | C2M | C1M | Bit0,1:PSG1,PSG2 mode control<br>Bit2,3:PSG1,PSG2 clock source selection | 0000 | W |
| $1F | - | - | - | - | Reserved | - | - |

*System Register $00~$12 refer to "SH6610C User manual".

**4. Data memory**

The general-purpose data memory is organized as 512*4 bits. Because of its static feature. The RAM can maintain its data after CPU enters the STOP or HALT mode.

**5. Oscillator circuit**

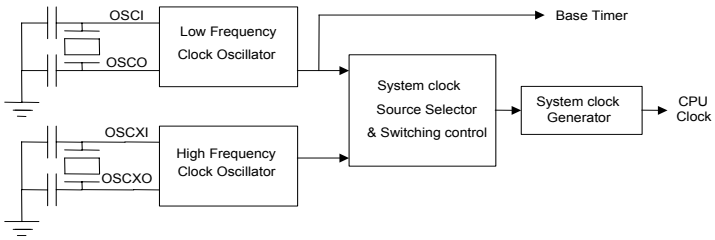
**5.1** Circuit Configuration

SH6613D has two on-chip oscillation circuits OSC and OSCX.

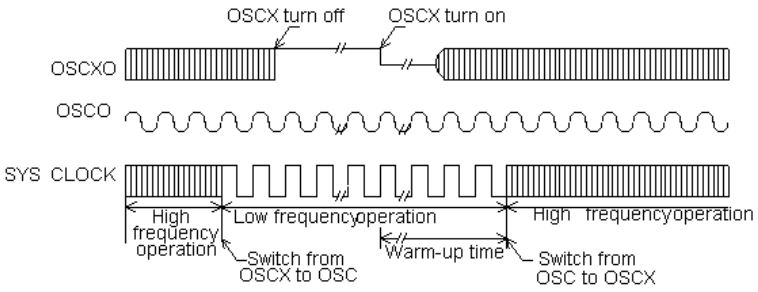
OSC is a low frequency crystal (Typ. 32.768KHz) or RC (Typ.262KHz) determined by the code option. This is designed for low frequency operation. OSCX also has two types: ceramic (Typ.455KHz) or RC (1.8M or 2MHz) to determine by software option. It is designed for high frequency operation.

It is possible to select the high speed CPU processing by a high frequency clock and select low power operation by low operation clock. At starting of reset initialization, OSC starts oscillation and OSCX is turned off. Immediate after reset initialization, the OSC clock is automatically selected as the system clock input source.

Oscillator Block Diagram



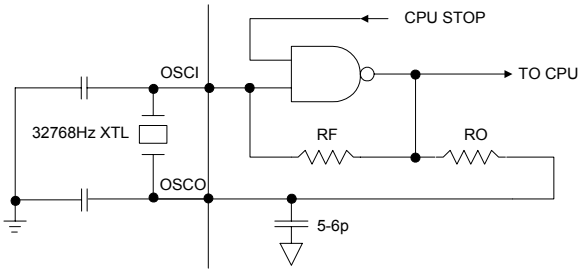

Timing of system Clock Switching



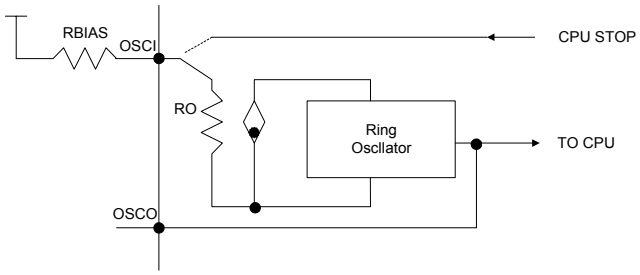
**5.2** OSC oscillation

The OSC generates the basic clock pulses that provide the CPU and peripherals (Timer0, LCD) with an operating clock.

OSC Crystal oscillator type



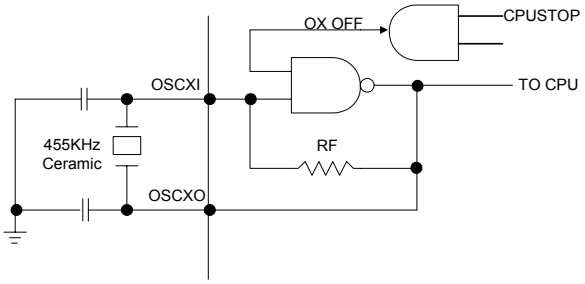

OSC RC oscillator type




**5.3** OSCX oscillation

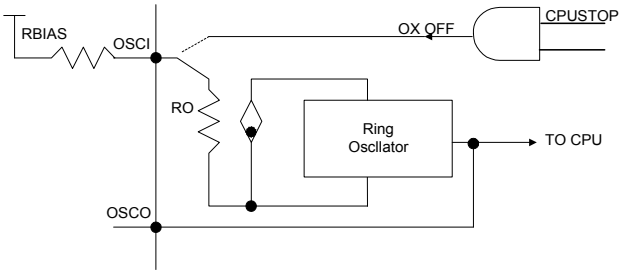
OSCX has two clock oscillators. The software options select the ceramic or RC as the CPU's subclock.

OSCX Ceramic oscillator type




OSCX RC oscillator type

If the OSCX is not used, it must be masked to be Ceramic resonator and the OSCXI must be connected to GND.



**5.4** Control of oscillator

The oscillator control register configuration is shown as blow.

| Add. | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|
| $14 | OXS | - | OXM | OXON |

OXON: OSCX oscillation on/off.

0:Turn off OSCX oscillation 1:Turn on OSCX oscillation

OXM: switching system clock.

0:select OSC as system clock 1:select OSCX as system clock

OXS: OSCX oscillator type selection

0:OSCX set as ceramic oscillator 1:OSCX set as RC oscillator

**5.5** Programming notes

It takes at least 5 ms for the OSCX oscillation circuit to go on until the oscillation stabilizes. When switching the CPU system clock from OSC to OSCX, you must wait a minimum of 5ms since the OSCX oscillation goes. However, the start time varies a lot with respect to oscillator characteristics and the condition of use. So the waiting time depends on applications. When switching from OSCX to OSC, and turn off OSCX in one instruction. The OSCX turn off control would be delayed for one instruction cycle automatically to prevent CPU operation error.

**6. System clock**

The system clock is varies as the clock source changes. The following table shows the instruction execution time according to each frequency of the system clock source.

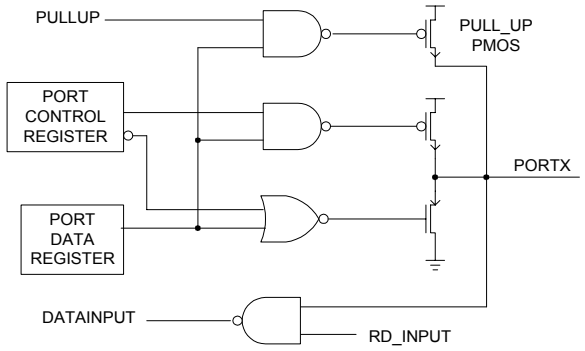
| | 32.768 Xtal(OSC ) | 262K RC(OSC ) | 455K ceramic Xtal(OSCX) | 1.8M RC(OSCX) | 2M RC(OSCX) |
|---|---|---|---|---|---|
| **Cycle time** | 122.07 μs | 17.778 μs | 8.79 μs | 2.22 μs | 2μs |

**7. I/O PORTs**

The MCU provides 8-bidirectional I/O pins. Each I/O pin contains pull-up MOS controllable by program. When every I/O is used as input, the PORT control register (PACR, PBCR) controls ON/OFF of the output buffer.

**7.1** PORTA~B

These ports contain 8-bidirectional I/O ports.
The circuit configuration of PORTA~B as below.




I/O ports of SH6613D can be accessed by read/write system register.
User can output any value to any I/O port bit at any time.

Memory map addresses are listed as follow:

| Address | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|
| $08 | PORTA.3 | PORTA.2 | PORTA.1 | PORTA.0 |
| $09 | PORTB.3 | PORTB.2 | PORTB.1 | PORTB.0 |

**7.2** Controlling the pull-up MOS

These ports contain pull-up MOS controlled by program. Bit3 of the PMOD register controls On/Off of all pull-up MOS simultaneously. Pull-up MOS also controlled by the port data registers (PA, PB) of each port also. So the pull-up MOS can be turned On/Off.

**7.3 Port Interrupt**

PORTB interrupt (falling edge) is not controlled by Port I/O register. It is means that if a interrupt request（IEx is set to 1 & one port bit high go low）is been touched and that the condition is the other port bits are high level whenever the port bit is output or input.

**7.4** External $\overline{INT}$

PortA.0 is shared by external interrupts (active low).

**7.5** Port I/O control register:

| Address | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|
| $0B | PACR.3 | PACR.2 | PACR.1 | PACR.0 |
| $0C | PBCR.3 | PBCR.2 | PBCR.1 | PBCR.0 |

I/O control register: PACR.X, PBCR.X (X=0,1,2,3)

1: Use as an output buffer.

0: Use as input buffer (Power on initial).

**7.6** Port mode register (PMOD)

| | Bit3 | Bit2 | Bit1 | Bit0 | Function | |
|---|---|---|---|---|---|---|
| $13 | PPULL | PAM2 | PAM1 | HLM | Bit1,2:Select PA.1,PA.2 as I/O port or PSG output<br>Bit0:Hevey load mode<br>Bit3:Port pull-up control | R/W |

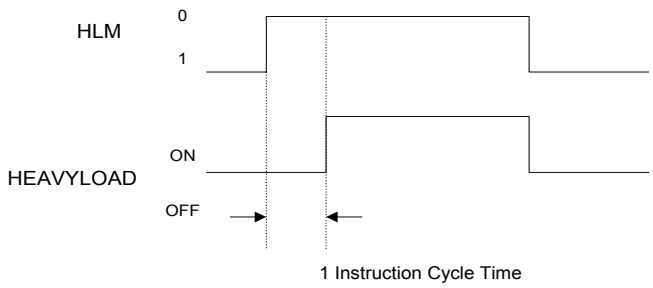
| | | | |
|---|---|---|---|
| PAM1, PAM2: | Please sees the PSG | | |
| HLM: | Enable heavy load mode | 0:Disable | 1:Enable |
| PPULL: | Port pull-up MOS | 0:Disable pull-up MOS | 1:Enable pull-up MOS |

**7.7** Heavy load mode (HLM)

The MCU has a heavy load protection circuit for when the battery load becomes heavy. For examples, when an external buzzer sound or an external speaker is turned on. In this mode, the crystal oscillator circuit has been backup for high gain. When setup this mode, more power would be provided to oscillator circuit. Unless it is necessary, be careful not to set this mode with software. Since the mode enter would delay for one instruction. Please activate heavy load driving only after setting HLM at least one instruction wait cycle through the software. Following shows the programming setting.

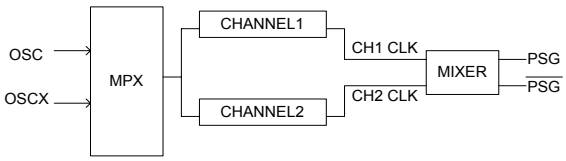
HLM: 0=Heavy load protection mode is released

1=Heavy load protection mode is set.



## 8. Programmable sound generator(PSG)

PSG has channel1 and channel2. The function block diagram is as follows.




The PSG function provides four subfunctions for wide application.

**Programmable sound**

Program sound is created by two channels. Every channel can be programmed as follows.
Enable/Disable every channel sounds.
Select every channel sound frequency.
Two channel sounds are mixed into one PSG output.
The PSG output can be controlled at 4 volume levels.

**Fine noise**

PSG can provide wide-band noise.
The wide-band noise volume can be controlled at 4 volume levels.

**Alarm**

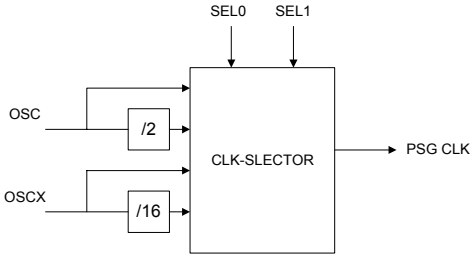
PSG can provide many alarm functions by software.
The alarm carrier frequency can be programmed individual.
The alarm volume can be controlled at 4 volume levels.

**Remote control**

The remote control is the only expandable application for PSG sound. Since the remote control frequency is 56.13KHz or 37.92KHz, the software could select the sound frequency.

**8.1** PSG subblock diagram

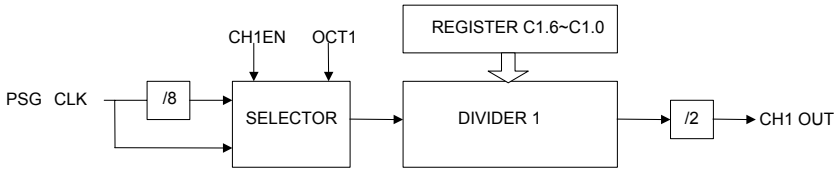
MPX block diagram




| SEL1 | SEL0 | Clk source | | PSG clk |
|---|---|---|---|---|
| 0 | 0 | OSC | OSC=32.768K | 32.768K |
| | | | OSC=262K | 262K |
| 0 | 1 | OSC/2 | OSC=32.768K | 16.384K |
| | | | OSC=262K | 131K |
| 1 | 0 | OSCX | OSCX=1.8M | 1.8M |
| | | | OSCX=455K | 455K |
| 1 | 1 | OSCX/16 | OSCX=1.8M | 112.5K |
| | | | OSCX=455K | 28.4K |

The MPX block selects 4 clock sources as PSG clk that provides for the two channel clk sources.

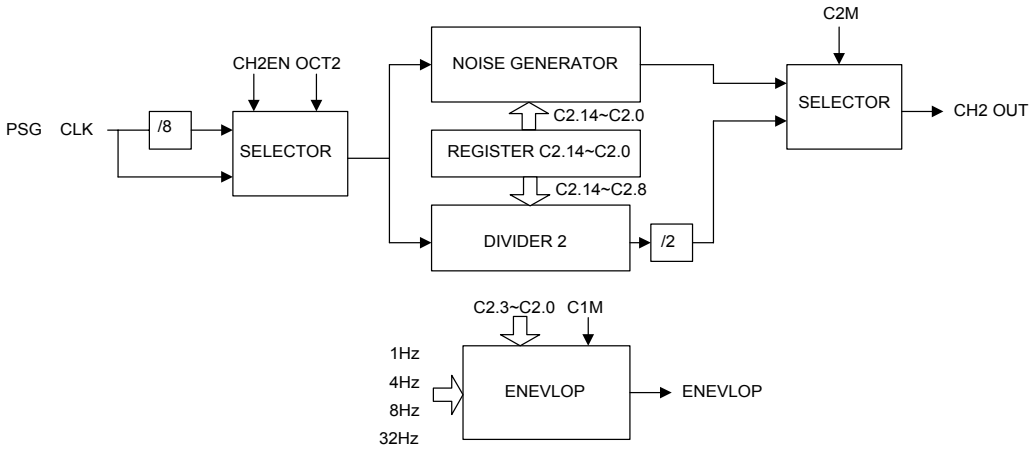
**Channel 1**




| OCT1 | Scaling ratio |
|---|---|
| 0 | 1 |
| 1 | 1/8 |

Channel 1 is constructed by a 7-bit pseudo random counter. Channel 1 is enabled/disabled by CH1EN. It creates either a sound frequency or an alarm carrier frequency or a remote carrier frequency.

**Channel 2**



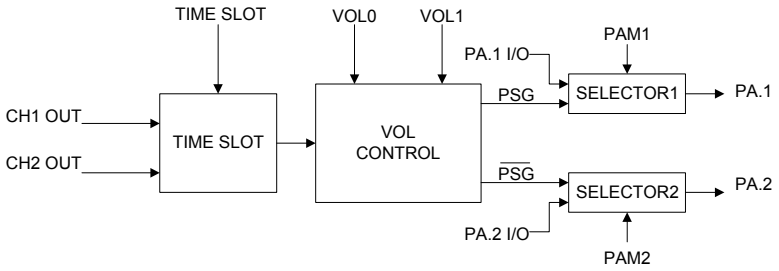

| OCT2 | Scaling ratio |
|---|---|
| 0 | 1 |
| 1 | 1/8 |

Channel 2 is constructed by a 15-bit pseudo random counter. Channel 2 is enabled/disabled by CH2EN

It can be a 15-bit wide-band noise generator or a 7-bit sound generator. It can also create an alarm envelope signal.

| C2M | C1M | Function |
|---|---|---|
| 0 | 0 | CH1 is Sound generator. CH2 is Sound generator. |
| 1 | 0 | CH1 is Sound generator. CH2 is Noise generator. |
| x | 1 | CH1 is Sound generator. CH2 is Alarm mode register. |

**Mixer**




The MIXER mixes CH1-OUT and CH2-OUT into one tone output to PA.1、PA.2, when PAM1=1、PAM2=1. Then the tone output is controlled by the volume control bit into 4 volume levels and in the end outputted by PSG.

PA.1 & PA.2 are controlled by PAM1 & PAM2

| PAM2 | PAM1 | Function |
|---|---|---|
| 0 | 0 | PA.1:I/O PORT PA.2:I/O PORT |
| 0 | 1 | PA.1:PSG output PA.2:I/O PORT |
| 1 | 0 | PA.1:I/O PORT PA.2: $\overline{\text{PSG}}$ output |
| 1 | 1 | PA.1:PSG output PA.2: $\overline{\text{PSG}}$ output |

| SEL1 | SEL0 | Vol. control |
|---|---|---|
| 0 | 0 | NO |
| 0 | 1 | YES |
| 1 | 0 | YES |
| 1 | 1 | YES |

| VOL1 | VOL0 | Vol. Level |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 2 |
| 1 | 0 | 3 |
| 1 | 1 | 4 |

**Note:** Don't enable two PSG channels together to produce one tone, or it will produce some unpredicted errors. If it is necessary to use 2 channels together (EX. To play two channel melody), don't let the score always be the same tones as we can do, then the unpredicted errors will not occur or it will be ignore through user hearing.

The value N of divider1 is corresponding to the REG C1.6~C1.0 or REG C2.14~C2.8 as shown in the following table:

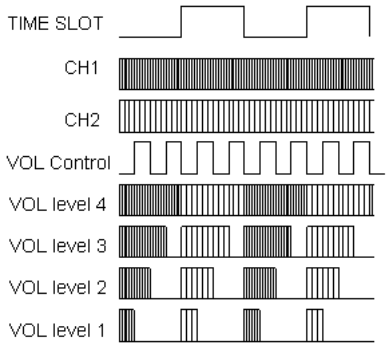
| LSFR (C1.6~C1.0) (C2.14~C2.8) | N | LSFR (C1.6~C1.0) (C2.14~C2.8) | N | LSFR (C1.6~C1.0) (C2.14~C2.8) | N | LSFR (C1.6~C1.0) (C2.14~C2.8) | N |
|---|---|---|---|---|---|---|---|
| 01 | 127 | 16 | 95 | 12 | 63 | 4B | 31 |
| 02 | 126 | 2C | 94 | 24 | 62 | 17 | 30 |
| 04 | 125 | 59 | 93 | 49 | 61 | 2E | 29 |
| 08 | 124 | 33 | 92 | 13 | 60 | 5D | 28 |
| 10 | 123 | 67 | 91 | 26 | 59 | 3B | 27 |
| 20 | 122 | 4E | 90 | 4D | 58 | 77 | 26 |
| 41 | 121 | 1D | 89 | 1B | 57 | 6E | 25 |
| 03 | 120 | 3A | 88 | 36 | 56 | 5C | 24 |
| 06 | 119 | 75 | 87 | 6D | 55 | 39 | 23 |
| 0C | 118 | 6A | 86 | 5A | 54 | 73 | 22 |
| 18 | 117 | 54 | 85 | 35 | 53 | 66 | 21 |
| 30 | 116 | 29 | 84 | 6B | 52 | 4C | 20 |
| 61 | 115 | 53 | 83 | 56 | 51 | 19 | 19 |
| 42 | 114 | 27 | 82 | 2D | 50 | 32 | 18 |
| 05 | 113 | 4F | 81 | 5B | 49 | 65 | 17 |
| 0A | 112 | 1F | 80 | 37 | 48 | 4A | 16 |
| 14 | 111 | 3E | 79 | 6F | 47 | 15 | 15 |
| 28 | 110 | 7D | 78 | 5E | 46 | 2A | 14 |
| 51 | 109 | 7A | 77 | 3D | 45 | 55 | 13 |
| 23 | 108 | 74 | 76 | 7B | 44 | 2B | 12 |
| 47 | 107 | 68 | 75 | 76 | 43 | 57 | 11 |
| 0F | 106 | 50 | 74 | 6C | 42 | 2F | 10 |
| 1E | 105 | 21 | 73 | 58 | 41 | 5F | 9 |
| 3C | 104 | 43 | 72 | 31 | 40 | 3F | 8 |
| 19 | 103 | 07 | 71 | 63 | 39 | 7F | 7 |
| 72 | 102 | 0E | 70 | 46 | 38 | 7E | 6 |
| 64 | 101 | 1C | 69 | 0D | 37 | 7C | 5 |
| 48 | 100 | 38 | 68 | 1A | 36 | 78 | 4 |
| 11 | 99 | 71 | 67 | 34 | 35 | 70 | 3 |
| 22 | 98 | 62 | 66 | 69 | 34 | 60 | 2 |
| 45 | 97 | 44 | 65 | 52 | 33 | 40 | 1 |
| 0B | 96 | 09 | 64 | 25 | 32 | | |

**8.2** Function description

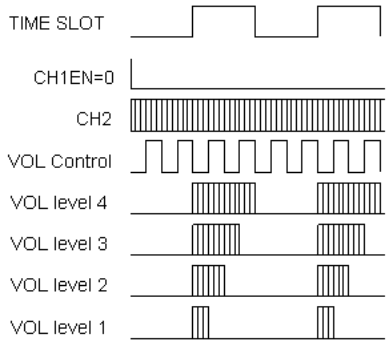
**PSG as sound generator**

The programmable sound is one of the 4 working modes. The software designer can select up to 16 clock sources as PSG clk. And then select the CH1 and CH2 frequency divided value that is controlled by the value of REG C1.6~C1.0 or C2.14~C2.8.In the end we can select the 4 volume level controlled by VOL0, VOL1. The music tone can output both PSG and $\overline{PSG}$. We also can control the OCT1, OCT2 bit that shifts the music tone 3 octaves.

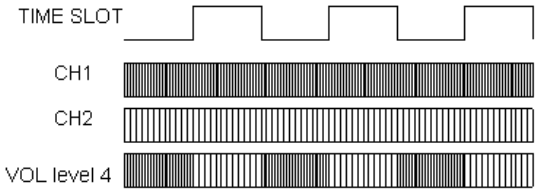
**Example1:**CH1EN=CH2EN=1
OSCX=1.8M,SEL0=SEL1=1
So PSG clk =112kHz;Switch clk=28kHz
Vol. clk=112kHz




**Example2:**CH1EN=0;CH2EN=1
OSCX=1.8M,SEL0=SEL1=1
So PSG clk =112kHz;Switch clk=28kHz;
Vol. clk=112kHz




**Example3:** CH1EN=CH2EN=1
OSC=32k,SEL0=SEL1=0
So PSG clk = 32kHz; Switch clk = 32kHz
No vol. control, the VOL level is set to 4 by hardware, so software should set VOL0=VOL1=1.




**Note:** For 32KHz operations, the volume control cannot be used, because the PWM multiplexing frequency is not high enough to switch sound! If a user want to turn off the PSG completely, the software must disable both channels. User should not turn off the PSG by zero wave from output. Both the CH1EN and CH2EN should be set to "0" for the low power operation mode.

**Example 4**

If software designer wants to create C2 (channel 1) mixed with F5 (channel 2) sound (the C2, F5 sound frequency please see the Music table1 and Music table2), VOL level=3.He can select the suggestion as follows.

(1) He first selects CH1EN=CH2EN=1,C1M=C2M=0.

(2) He can select OSCX=1.8M and SEL0=SEL1=1, so the PSG CLK=112.5KHz.

(3) Then he can select OCT1=1 and the value of channel 1 LSFR (C1.6~C1.0)=23,so the N =108. Please see the Music table1.
So the channel 1 sound frequency=112.5Khz/8/(2×108)=64.10Hz≒the C2 sound frequency.

(4) Then he can select OCT1=0 and the value of channel 1 LSFR (C1.6~C1.0)=4F,so the N =81. Please see the Music table1.
So the channel 1 sound frequency=112.5Khz/1/(2×81)=694.4Hz≒the F5 sound frequency .

(5) Lastly, he should select the VOL1=1 and VOL0=0, so the VOL level=3.

**Note:**

The designer provides two crossing tables as an appendix that is what the designer prefers PSG clk=32.768K or PSG clk=112.5K.

**PSG as noise generator**

The fine noise is created by CH2. If we want to create the single noise, we can make the CH1 music tone output. Otherwise we can mix the wide-band noise and the CH1 music tone into one output through the MIXER. Lastly we can select 4 volume levels controlled by VOL0, VOL1.

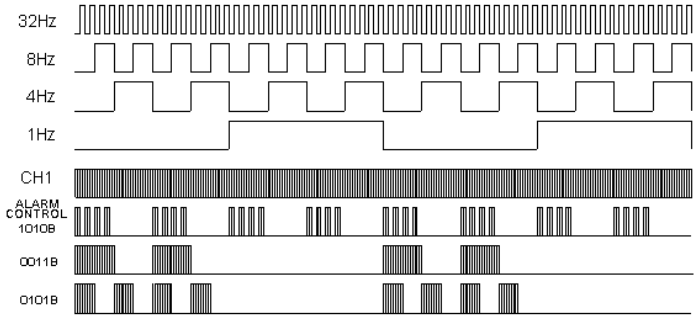
**PSG as an alarm generator**

When PSG is in the alarm mode, the CH1 provides the alarm carrier frequency and the CH2 provides the alarm envelope signal. Lastly we can select 4 volume levels controlled by VOL0, VOL1. The channel 2 low nibble C2.0~C2.3 will be the alarm control register. Channel 1 output would modulate with an ALARM envelope control for 32KHz or 262KHz. The carrier frequency can be programmed by PSG channel 1.In reading this alarm control register, we can read the corresponding output envelope frequency (the 1Hz, 4Hz, 8Hz, 32Hz).

**Alarm control register (OSC=32KHz or 262KHz)**

| $19 | C2.3 | C2.2 | C2.1 | C2.0 | Alarm output control |
|---|---|---|---|---|---|
| | 0 | 0 | 0 | 0 | DC envelop |
| | X | X | X | 1 | 1Hz output |
| | X | X | 1 | X | 4Hz output |
| | X | 1 | X | X | 8Hz output |
| | 1 | X | X | X | 32Hz output |

Figure: Alarm modulation output for OSC=32.768KHz or OSC=262KHz.




**PSG as remote control**

The remote control is only an expandable application for PSG sound. We can select the CH1 as tone output and the CH2 will create alarm frequency envelope signal.
When PSG channel is programmed in the ALARM mode. Programmer can set ALARM mode register to "0000B". Program the adequate frequency output to PSG output. Then use PAM1 or PAM2 control the envelope of code. In this way, remote control function can be implemented easily.

**The remote frequency =56.73KHz or 37.92KHz.**

The software should select OSCX=455KHz, SEL=1 and SEL0=0, so that the PSG CLK=455KHz.
Then select channel 1 alarm mode (C1M=1), and OCT1=0,C2.0~C2.3 are set to 00H. VOL1, VOL2=1,1.
Then select C1.6~C1.0=7E, so that N=6 and the PSG output frequency=455KHz/1/(2×6)=37.92KHz.
Or select C1.6~C1.0=78,so that N=4 and the PSG output frequency=455KHz/1/(2×4)=56.87KHz.

**9. Timer 0**

SH6613D has one 8-bit timer. The timer consists of an 8-bit up counter and an 8-bit preload register.

The timers provide the following functions:

- Programmable internal timer function
- Read the counter values

**9.1** Timer 0 configuration and operation

The timer 0 consists of an 8-bit write-only timer load register (TL0L, TL0H) and an 8-bit read-only timer counter (TC0L, TC0H). Each of them has low order digits and high order digits. The timer counter can be initialized by writing data into the timer load register (TL0L, TL0H). Write the low-order digit first and then the high-order digit. The timer counter is loaded with the content of the load register automatically when the high order digit is written or counts overflow happens. The timer overflow will generate an interrupt, if the interrupt enable flag is set.
The timer can be programmed in several different system clock sources by setting the Timer Mode register (TM0).

Timer 0 reads and writes operations follow these rules:

| **Write Operation:** | **Read Operation** |
|---|---|
| Low nibble first | High nibble first |
| High nibble to update the counter | Low nibble follows |

**9.2** Timer0 mode register (TM0)

The 8-bit counter counts prescaler overflow output pulses. TM0 are 4-bit registers used for timer control as shown in Table1. The register selects the input clock sources in the timer.

**Table1: Timer0 Mode registers ($02)**

| TM0.3 | TM0.2 | TM0.1 | TM0.0 | Prescaler | Clock Source |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | /2048 | System clock |
| 0 | 0 | 0 | 1 | /512 | System clock |
| 0 | 0 | 1 | 0 | /128 | System clock |
| 0 | 0 | 1 | 1 | /32 | System clock |
| 0 | 1 | 0 | 0 | /8 | System clock |
| 0 | 1 | 0 | 1 | /4 | System clock |
| 0 | 1 | 1 | 0 | /2 | System clock |
| 0 | 1 | 1 | 1 | External | $\overline{\text{INT0}}$ |

TM0.3 control function:

0: without Auto-Reload function 1: Auto-Reload function

**9.3** Warm-up counter

In RC mode, the warm-up counter prescaler is divided by $2^7$ (128).
In Crystal mode, the warm-up counter prescaler is divided by $2^{15}$ (65536).

**10. Base Timer**

The MCU has a base timer that is shared with the warm-up timer and the clock source is OSC (Low frequency oscillation: X'tal 32.768KHz or RC 262KHz). After MCU is reset, it counts at every clock-input signal. When it counts to $FF, right after next clock input, counter counts to $00 and generates an overflow This causes the interrupt of base timer interrupt request flag to 1.Therefor, base timer can function as an interval timer periodically, generating overflow output as every 256th clock signal output.

The timer accepts 4096Hz or 32KHz clock, and base timer generates an accurate timing interrupt. This base time prescaler can be reset by a program for accurate timing.

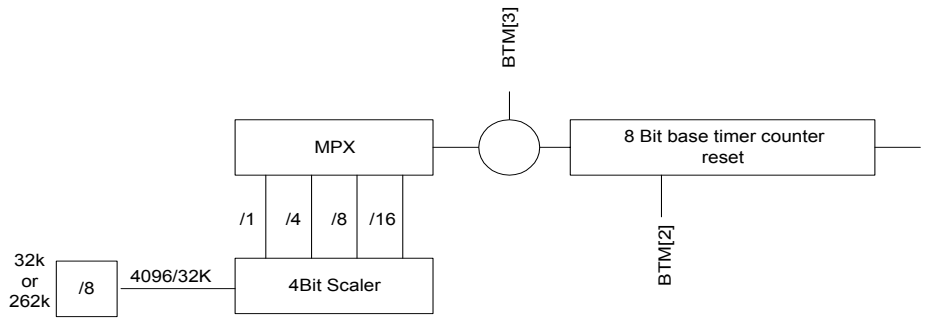
This clock-input source is sleeted by BTM register.

| Address | Bit3 | Bit2 | Bit1 | Bit0 | Function |
|---|---|---|---|---|---|
| **$03** | BTM.3 | BTM.2 | BTM.1 | BTM.0 | Base timer mode register |

BTM.3=0: Disable the base timer BTM.3=1: Enable the base timer

BTM.2=0: Non reset the base timer BTM.2=1: reset the base timer

| BTM.1 | BTM.0 | Prescaler Ratio | Clock source |
|---|---|---|---|
| 0 | 0 | /1 | 4096Hz or 32KHz |
| 0 | 1 | /4 | 4096Hz or 32KHz |
| 1 | 0 | /8 | 4096Hz or 32KHz |
| 1 | 1 | /16 | 4096Hz or 32KHz |



**11. LCD Driver**

The LCD driver contains a controller, a voltage generator, 4 common signal pins and 34 segment driver pins, 1/4 duty and 1/3 bias.

The LCD SEG1~30 can also be used as output port, it is selected by the bit 2 of the system register $16. When SEG1~30 is to be output port, you write data to bit 0 of the same addresses (350H-36DH). LCD RAM could be used as data memory if you need. When the "STOP" instruction is executed, the LCD will be turned off, but the data of LCD RAM keeps the same value before executing the "STOP" instruction.

**11.1** LCD Control Register

| Add. | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|
| $15 | LPS1 | LPS0 | LCDOFF | **Should be set "1"** |

LCDOFF: LCD on/off switch.

0: LCD on. 1: LCD off.

LPS1, LPS0: LCD clock frequency control

0,0: LCDCLK = OSC/64

0,1: LCDCLK = OSC/512

1,0: LCDCLK = System clock/512

1,1: LCDCLK = System clock/4096

*System clock= Instruction cycle time

Frame frequency=LCDCLK/16

Frame frequency for 1/8 duty; 1/4 duty has the same frequency cycle.

| CLK Frequency | LPS1, LPS0 | | | |
|---|---|---|---|---|
| | 0,0 | 0,1 | 1,0 | 1,1 |
| OSC=32kHz (#), OSCX=455kHz | 32Hz | 4Hz | 1Hz | 1/8Hz |
| OSC=32kHz, OSCX=455kHz (#) | 32Hz | 4Hz | 14Hz | 1.7Hz |
| OSC=32kHz (#), OSCX=1.8MHz | 32Hz | 4Hz | 1Hz | 1/8Hz |
| OSC=32kHz, OSCX=1.8MHz (#) | 32Hz | 4Hz | 55Hz | 6.9Hz |
| OSC=32kHz(#), OSCX=2MHz | 32Hz | 4Hz | 1Hz | 1/8Hz |
| OSC=32kHz, OSCX=2MHz (#) | 32Hz | 4Hz | 61Hz | 7.6Hz |
| OSC=262kHz (#), OSCX=455kHz | 256Hz | 32Hz | 8Hz | 1Hz |
| OSC=262kHz, OSCX=455kHz (#) | 256Hz | 32Hz | 14Hz | 1.7Hz |
| OSC=262kHz (#), OSCX=1.8MHz | 256Hz | 32Hz | 8Hz | 1Hz |
| OSC=262kHz , OSCX=1.8MHz (#) | 256Hz | 32Hz | 55Hz | 6.9Hz |
| OSC=262kHz (#), OSCX=2MHz | 256Hz | 32Hz | 8Hz | 1Hz |
| OSC=262kHz , OSCX=2MHz (#) | 256Hz | 32Hz | 61Hz | 7.6Hz |

The clk before (#) is sleeted as CPU system clk.

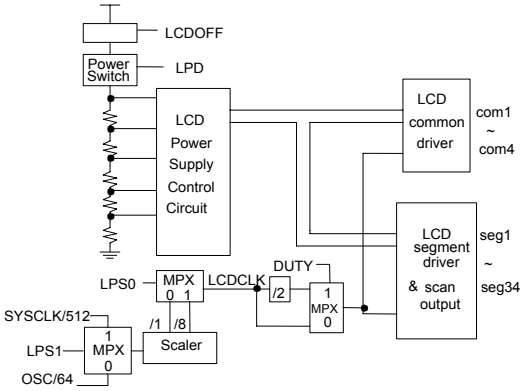
Frame frequency=LCDCLK/16 (32Hz)When LCD is set SCAN OUT, COMx are pulled high. It is easy to implement the keyboard scan.

When CPU is in STOP mode, the COMx and SEGx are pulled low. It can easily be waken up by keyboard scan (Port interrupt).

When CPU is in HALT mode, the COMx and SEGx are normal. It can easily be waken up by base timer、timer0 or port interrupt.

**LCD block diagram for reference only.**

**11.2** LCD power




Build-in special LCD power control for LCD power modulation.

| Add. | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|
| $16 | LPD | O/S | - | - |

O/S: Set LCD segment/common to be LCD segment output or output ports

0: LCD segment output 1: output ports.

When LPD is set to 1, the LCD voltage power would be degraded about 0.5V, depending on the $V_{DD}$ level. It is designed to reduce extra LCD contrast control output pins. Then the LCD can be fitted automatically for different voltage levels by the software.

**11.3** Configuration of LCD RAM

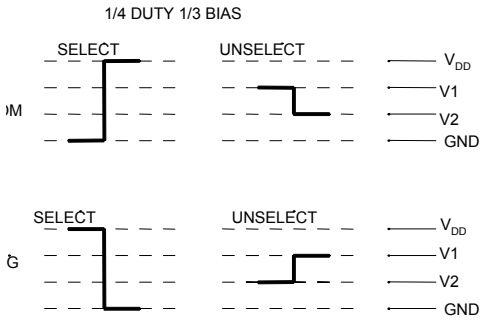
LCD 1/4 duty, 1/3 bias (COM1~4,SEG1~34)

| **Address** | **Bit3** | **Bit2** | **Bit1** | **Bit0** | **Address** | **Bit3** | **Bit2** | **Bit1** | **Bit0** |
|---|---|---|---|---|---|---|---|---|---|
| | **COM4** | **COM3** | **COM2** | **COM1** | | **COM4** | **COM3** | **COM2** | **COM1** |
| 300H | SEG1 | SEG1 | SEG1 | SEG1 | 311H | SEG18 | SEG18 | SEG18 | SEG18 |
| 301H | SEG2 | SEG2 | SEG2 | SEG2 | 312H | SEG19 | SEG19 | SEG19 | SEG19 |
| 302H | SEG3 | SEG3 | SEG3 | SEG3 | 313H | SEG20 | SEG20 | SEG20 | SEG20 |
| 303H | SEG4 | SEG4 | SEG4 | SEG4 | 314H | SEG21 | SEG21 | SEG21 | SEG21 |
| 304H | SEG5 | SEG5 | SEG5 | SEG5 | 315H | SEG22 | SEG22 | SEG22 | SEG22 |
| 305H | SEG6 | SEG6 | SEG6 | SEG6 | 316H | SEG23 | SEG23 | SEG23 | SEG23 |
| 306H | SEG7 | SEG7 | SEG7 | SEG7 | 317H | SEG24 | SEG24 | SEG24 | SEG24 |
| 307H | SEG8 | SEG8 | SEG8 | SEG8 | 318H | SEG25 | SEG25 | SEG25 | SEG25 |
| 308H | SEG9 | SEG9 | SEG9 | SEG9 | 319H | SEG26 | SEG26 | SEG26 | SEG26 |
| 309H | SEG10 | SEG10 | SEG10 | SEG10 | 31AH | SEG27 | SEG27 | SEG27 | SEG27 |
| 30AH | SEG11 | SEG11 | SEG11 | SEG11 | 31BH | SEG28 | SEG28 | SEG28 | SEG28 |
| 30BH | SEG12 | SEG12 | SEG12 | SEG12 | 31CH | SEG29 | SEG29 | SEG29 | SEG29 |
| 30CH | SEG13 | SEG13 | SEG13 | SEG13 | 31DH | SEG30 | SEG30 | SEG30 | SEG30 |
| 30DH | SEG14 | SEG14 | SEG14 | SEG14 | 31EH | SEG31 | SEG31 | SEG31 | SEG31 |
| 30EH | SEG15 | SEG15 | SEG15 | SEG15 | 31FH | SEG32 | SEG32 | SEG32 | SEG32 |
| 30FH | SEG16 | SEG16 | SEG16 | SEG16 | 320H | SEG33 | SEG33 | SEG33 | SEG33 |
| 310H | SEG17 | SEG17 | SEG17 | SEG17 | 321H | SEG34 | SEG34 | SEG34 | SEG34 |

SEG1-30 is used as scan output port.

| Address | Bit0 | Address | Bit0 | Address | Bit0 | Address | Bit0 |
|---|---|---|---|---|---|---|---|
| 350H | SEG1 | 358H | SEG9 | 360H | SEG17 | 368H | SEG25 |
| 351H | SEG2 | 359H | SEG10 | 361H | SEG18 | 369H | SEG26 |
| 352H | SEG3 | 35AH | SEG11 | 362H | SEG19 | 36AH | SEG27 |
| 353H | SEG4 | 35BH | SEG12 | 363H | SEG20 | 36BH | SEG28 |
| 354H | SEG5 | 35CH | SEG13 | 364H | SEG21 | 36CH | SEG29 |
| 355H | SEG6 | 35DH | SEG14 | 365H | SEG22 | 36DH | SEG30 |
| 356H | SEG7 | 35EH | SEG15 | 366H | SEG23 | | |
| 357H | SEG8 | 35FH | SEG16 | 367H | SEG24 | | |

**11.4** LCD waveform



## 12. Interrupt

4 interrupt sources are available on SH6613D:

- External interrupt ( $\overline{\text{INT0}}$ )
- Timer0 interrupt
- Base timer interrupt
- Port's falling edge detection interrupt( $\overline{\text{INT1}}$ )

The configuration of system register $0:

| | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Function |
|---|---|---|---|---|---|
| $00 | IEX | IET0 | IEBT | IEP | 1:Enable / 0:Disable |
| $01 | IRQX | IRQT0 | IRQBT | IRQP | 1:Request / 0:No request |

### 12.1 External Interrupt ( $\overline{\text{INT0}}$ )

External interrupt is shared with the PA.0, falling edge active. When the bit 3 of the register $0 (IEX) is set to 1, the external interrupt is enabled, writing a "0" to PA.0 will generate an external interrupt.

### 12.2 Timer 0 interrupt, Base timer interrupt, Port interrupt ( $\overline{\text{INT1}}$ )

If IEx=1 then all valid interrupt requests will cause interrupt. The overflow of timer 0 will create the interrupt of timer 0.The overflow of the Base timer will create the interrupt of the Base timer. The falling edge of every port in PORTB will create $\overline{\text{INT1}}$ interrupt (The condition is that the other port must be input/output high level).

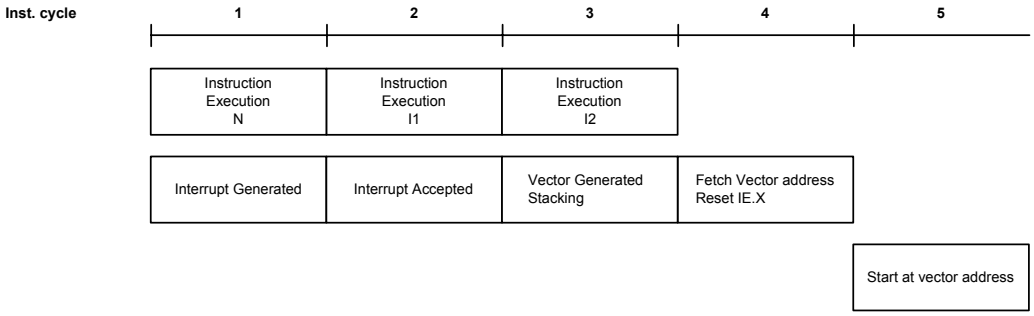
### 12.3 The Enable flags and Request flags

Both the Enable flags and Request flags can be read or written by the software.

But the Request flags will be set "1" by the hardware interrupt, the Enable flags will be reset by the hardware when the interrupt service routine is entered.

### 12.4 Interrupt Servicing Sequence Diagram:

In SH6610C CPU interrupt services routine, the user can enable any interrupt enable flag before returning from interrupt. The frequently asked question is when the next interrupt would be serviced? Will the nesting interrupt happen? From the servicing sequence timing diagram, if interrupt request is ready and instruction execution N is IE enable. Then the interrupt can start right after the next two instructions: I1 or instruction I2 disable the interrupt request or enable flag, then interrupt service sequence would be terminated.

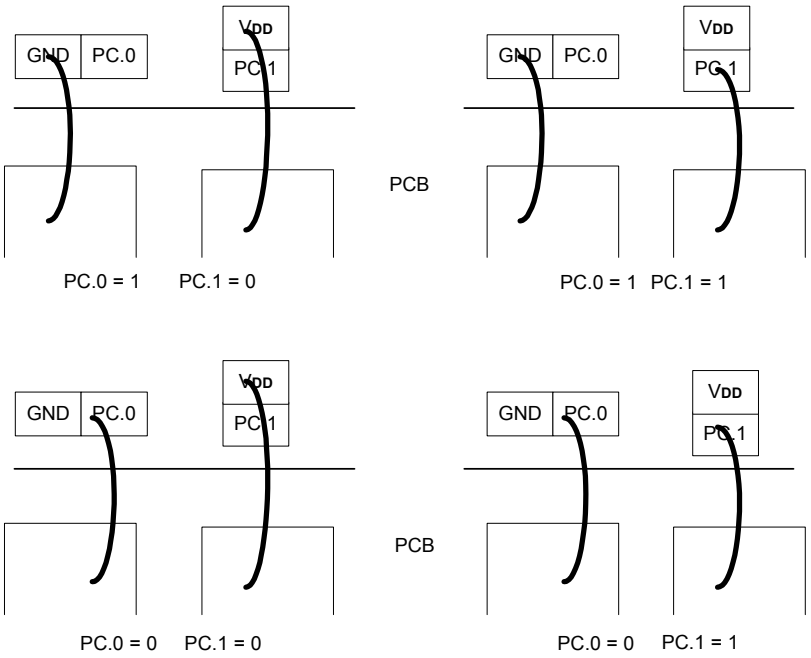


### 13. Options

Bonding options

System register $0A is reserved for the user .It is opened for system developer to select these 2 bonding options, selecting subprogram that is programmed by the user.

| $0A.1 (PC.1) | $0A.0 (PC.0) | |
|---|---|---|
| 0 | 0 | Go to subroutine 1 |
| 0 | 1 | Go to subroutine 2 ( Default ) |
| 1 | 0 | Go to subroutine 3 |
| 1 | 1 | Go to subroutine 4 |




**SH6613D Bonding Option**

### 14. STOP/HALT mode

| STOP/HALT mode | Oscillator | CPU core | Wake up | Executing after wake up |
|---|---|---|---|---|
| STOP<br>(STOP instruction) | OSC<br>OSCX<br>Stop | Hold | $\overline{RST}$ , $\overline{INT0}$ , $\overline{INT1}$ | a) If $\overline{RST}$ signal valid, system will be reset.<br>b) If $\overline{INT0}$ , $\overline{INT1}$ signal valid, system will enter interrupt subroutine, and then execute the main program to continue. |
| HALT<br>(HALT instruction) | OSC<br>OSCX<br>live | Hold | $\overline{RST}$ , $\overline{INT0}$ , $\overline{INT1}$ ,T0INT<br>BTINT | a) If $\overline{RST}$ signal valid, system will be reset.<br>b) If $\overline{INT0}$ , $\overline{INT1}$ , T0INT or BTINT signal valid, system will enter interrupt subroutine first, then execute the main program to continue. |

**15. Instruction set**

All instructions are one cycle and one word instruction. The characteristic is memory-oriented operation.

Arithmetic and Logical Instruction

Accumulator type

| Mnemonic | Instruction Code | Function | Flag Change |
|---|---|---|---|
| ADC X( , B) | 00000 0bbb xxx xxxx | AC ← Mx + Ac + CY | CY |
| ADCM X( , B) | 00000 1bbb xxx xxxx | AC, Mx ← Mx + Ac + CY | CY |
| ADD X( , B) | 00001 0bbb xxx xxxx | AC ← Mx + Ac | CY |
| ADDM X( , B) | 00001 1bbb xxx xxxx | AC, Mx ← Mx + Ac | CY |
| SBC X( , B) | 00010 0bbb xxx xxxx | AC ← Mx + -Ac + CY | CY |
| SBCM X( , B) | 00010 1bbb xxx xxxx | AC, Mx ← Mx + -Ac + CY | CY |
| SUB X( , B) | 00011 0bbb xxx xxxx | AC ← Mx + -Ac + 1 | CY |
| SUBM X( , B) | 00011 1bbb xxx xxxx | AC, Mx ← Mx + -Ac + 1 | CY |
| EOR X( , B) | 00100 0bbb xxx xxxx | AC ← Mx ⊕ Ac | |
| EORM X( , B) | 00100 1bbb xxx xxxx | AC, Mx ← Mx ⊕ Ac | |
| OR X( , B) | 00101 0bbb xxx xxxx | AC ← Mx \| Ac | |
| ORM X( , B) | 00101 1bbb xxx xxxx | AC, Mx ← Mx \| Ac | |
| AND X( , B) | 00110 0bbb xxx xxxx | AC ← Mx & Ac | |
| ANDM X( , B) | 00110 1bbb xxx xxxx | AC, Mx ← Mx & Ac | |
| SHR | 11110 0000 000 0000 | 0 → AC[3] ; AC[0] → CY ;<br>AC shift right one bit | CY |

Immediate Type

| Mnemonic | Instruction Code | Function | Flag Change |
|---|---|---|---|
| ADI X , I | 01000 iiii xxx xxxx | AC ← Mx + I | CY |
| ADIM X , I | 01001 iiii xxx xxxx | AC, Mx ← Mx + I | CY |
| SBI X , I | 01010 iiii xxx xxxx | AC ← Mx + -I + 1 | CY |
| SBIM X , I | 01011 iiii xxx xxxx | AC, Mx ← Mx + -I + 1 | CY |
| EORIM X , I | 01100 iiii xxx xxxx | AC, Mx ← Mx ⊕ I | |
| ORIM X , I | 01101 iiii xxx xxxx | AC, Mx ← Mx \| I | |
| ANDIM X , I | 01110 iiii xxx xxxx | AC, Mx ← Mx & I | |

* In the assembler ASM66 V1.0, EORIM mnemonic is EORI. However, EORI has the same operation identical with EORIM. Same for the ORIM with respect to ORI, and ANDIM with respect to ANDI.

Decimal Adjust

| Mnemonic | Instruction Code | Function | Flag Change |
|---|---|---|---|
| DAA X | 11001 0110 xxx xxxx | AC; Mx ←Decimal adjust for add. | CY |
| DAS X | 11001 1010 xxx xxxx | AC; Mx ←Decimal adjust for sub. | CY |

Transfer Instruction

| Mnemonic | Instruction Code | Function | Flag Change |
|---|---|---|---|
| LDA X( , B) | 00111 0bbb xxx xxxx | AC ← Mx | |
| STA X( , B) | 00111 1bbb xxx xxxx | Mx ← AC | |
| LDI X , I | 01111 iiii xxx xxxx | AC, Mx ← I | |

Control Instruction

| Mnemonic | Instruction Code | Function | Flag Change |
|---|---|---|---|
| BAZ X | 10010 xxxx xxx xxxx | PC ← X if AC=0 | |
| BNZ X | 10000 xxxx xxx xxxx | PC ← X if AC≠0 | |
| BC X | 10011 xxxx xxx xxxx | PC ← X if CY=1 | |
| BNC X | 10001 xxxx xxx xxxx | PC ← X if CY≠1 | |
| BA0 X | 10100 xxxx xxx xxxx | PC ← X if AC(0)=1 | |
| BA1 X | 10101 xxxx xxx xxxx | PC ← X if AC(1)=1 | |
| BA2 X | 10110 xxxx xxx xxxx | PC ← X if AC(2)=1 | |
| BA3 X | 10111 xxxx xxx xxxx | PC ← X if AC(3)=1 | |
| CALL X | 11000 xxxx xxx xxxx | ST ← CY ; PC+1<br>PC ← X (Not include p) | |
| RTNW H , L | 11010 000h hhh l 1 l l | PC← ST ; TBR← hhhh; AC←l l l l | |
| RTNI | 11010 1000 000 0000 | CY ; PC ← ST | CY |
| HALT | 11011 0000 000 0000 | | |
| STOP | 11011 1000 000 0000 | | |
| JMP X | 1110p xxxx xxx xxxx | PC ← X (Include p) | |
| TJMP | 11110 1111 111 1111 | PC ← (PC11-C8)(TBR)(AC) | |
| NOP | 11111 1111 111 1111 | No Operation | |

Where,

| | | | | | |
|---|---|---|---|---|---|
| PC | Program counter | I | Immediate data | p | ROM page = 0 |
| AC | Accumulator | ⊕ | Logical exclusive OR | ST | Stack |
| -AC | Complement of accumulator | \| | Logical OR | TBR | Table Branch Register |
| CY | Carry flag | & | Logical AND | | |
| Mx | Data memory | bbb | RAM bank=000 | | |

### Absolute Maximum Rating*

DC Supple Voltage . . . . . -0.3V to +7.0V

Input Voltage . . . . . -0.3V to $V_{DD}$+0.3V

Operating Ambient Temperature . . . . -10℃ to +60℃

Storage Temperature . . . . . -55℃ to +125℃

### *Comments

Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to this device. These are stress ratings only. Functional operation of this device at these or any other conditions above those indicated in the operational sections of this specification is not implied or intended. Exposure to the absolute maximum rating conditions for extended periods may affect device reliability.

### DC Electrical Characteristics

($V_{DD}$=3.0V, GND=0V, $T_A$=25℃, $F_{OSC}$=32.768KHz, $F_{OSCX}$ is not used, unless otherwise specified)

| Parameter | Symbol | Min. | Typ. | Max. | Unit | Conditions |
|---|---|---|---|---|---|---|
| Operating Voltage | $V_{DD}$ | 2.4 | 3 | 6.0 | V | |
| Operating Current | $I_{OP}$ | - | 10 | 20 | μA | All output pins unload execute NOP instruction exclude LCD bias current |
| LCD voltage divider resistor | $R_{LCD}$ | - | 275 | - | KΩ | |
| Standby Current | $I_{SB1}$ | - | 2 | 4 | μA | All output pins unload (HALT mode) exclude LCD bias current |
| Standby Current | $I_{SB2}$ | - | - | 1 | μA | All output pins unload (STOP mode), LCD off |
| Input High Voltage | $V_{IH}$ | 0.7×$V_{DD}$<br>0.85×$V_{DD}$ | - | $V_{DD}$+0.3<br>$V_{DD}$+0.3 | V | PORTA, PORTB<br>INT0, RESET |
| Input Low Voltage | $V_{IL}$ | -0.3 | - | 0.3×$V_{DD}$<br>0.15×$V_{DD}$ | V | PORTA, PORTB<br>INT0, RESET |
| Drive-high resistance | $R_{OH}$ | - | 300 | - | KΩ | PORTA, PORTB ($I_{OH}$ = -10μA, $V_{OH}$ = 0). |
| Output high voltage | $V_{OH1}$ | 0.7×$V_{DD}$ | - | - | V | PORTA.0, PORTA.3, PORTB ($I_{OH}$ = -2mA). |
| Output low voltage | $V_{OL1}$ | - | - | 0.8 | V | PORTA.0, PORTA.3, PORTB ($I_{OL}$ = 2mA). |
| Output high voltage | $V_{OH2}$ | 0.7×$V_{DD}$ | - | - | V | PORTA.1, PORTA.2 or PSG output, $I_{OH}$ = -5mA. |
| Output low voltage | $V_{OL2}$ | - | - | 0.8 | V | PORTA.1, PORTA.2 or PSG output, $I_{OL}$ = 5mA. |
| Output high voltage | $V_{OH3}$ | 2.8 | - | - | V | SEGx, C=50P, rise time<1000ns |
| Output low voltage | $V_{OL3}$ | - | - | 0.2 | V | SEGx |
| Output high voltage | $V_{OH4}$ | $V_{DD}$-0.6 | - | - | V | SEG1 - 30to be output port, $I_{OH}$ =-1mA |
| Output low voltage | $V_{OL4}$ | - | - | 0.8 | V | SEG1 - 30to be output port, $I_{OL}$ =1mA |
| Output high voltage | $V_{OH5}$ | $V_{DD}$-1.0 | - | - | V | COMx, $I_{OH}$ = -1mA. |
| Output low voltage | $V_{OL5}$ | - | - | 0.8 | V | COMx, $I_{OL}$ = 1mA. |
| LCD lighting | $I_{LCD}$ | - | 8 | 10 | μA | $V_{DD}$=3V, exclude CPU core operation current |

Operation frequency vs. $I_{SB1}$
$I_{SB1X}$ = (Frequency/32.768KHz)×$I_{SB1}$×0.8), ($V_{DD}$ = 3.0V).

Operation frequency vs. $I_{OP}$
$I_{OPX}$ = (Frequency/32.768KHz)×$I_{OP}$×0.8, ($V_{DD}$ = 3.0V).

HLM vs. $I_{OP}$, $I_{SB1}$ and $I_{SB2}$
If HLM = 1, $I_{OPX}$ = $I_{OP}$×2, $I_{SB2X}$ = $I_{SB2}$×2.

**AC Characteristics** ($V_{DD}$=3.0V, GND=0V, $T_A$=25℃, $F_{OSC}$=32.768KHz,unless otherwise specified)

| Parameter | Symbol | Min. | Typ. | Max. | Unit | Conditions |
|---|---|---|---|---|---|---|
| Oscillation Start Time | $t_{STT}$ | - | 2 | 5 | s | |
| Frequency Stability | \|ΔF\|/F | - | - | 1 | PPM | [F(3.0)-F(2.5)]/F(3.0), crystal oscillator |

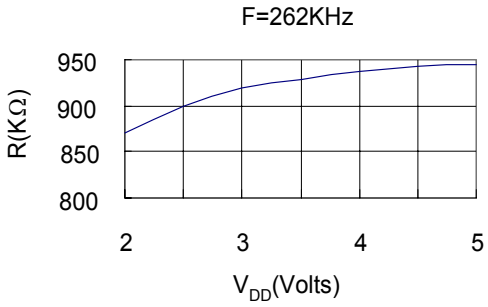
**AC Characteristics** ($V_{DD}$=3.0V, GND=0V, $T_A$=25℃, $F_{OSC}$=262KHz, $F_{OSCX}$ stop, unless otherwise specified)

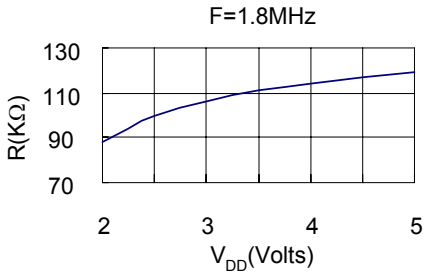
| Parameter | Symbol | Min. | Typ. | Max. | Unit | Conditions |
|---|---|---|---|---|---|---|
| Oscillation Start Time | $t_{STT}$ | - | - | 100 | μs | |
| Frequency Stability | \|ΔF\|/F | - | - | 10 | % | [F(3.0)-F(2.5)]/F(3.0), Bias resistance accuracy within 1% |

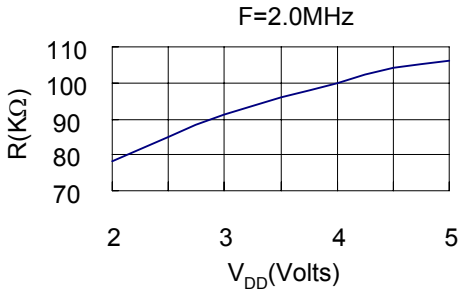
**AC Characteristics** ($V_{DD}$=4.5V,GND=0V, $T_A$=25℃, $F_{OSC}$=262KHz, $F_{OSCX}$ stop, unless otherwise specified)

| Parameter | Symbol | Min. | Typ. | Max. | Unit | Conditions |
|---|---|---|---|---|---|---|
| Frequency Stability | \|ΔF\|/F | - | - | 10 | % | [F(4.5)-F(3.6)]/F(4.5), Bias resistance accuracy within 1% |

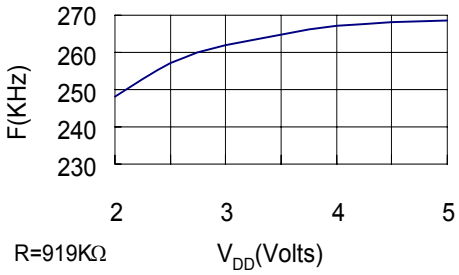
**Typical RC oscillator Resistor vs. $V_{DD}$:**
**(Reference only)**

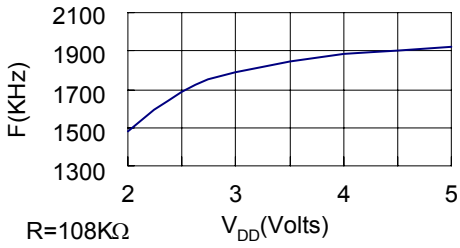


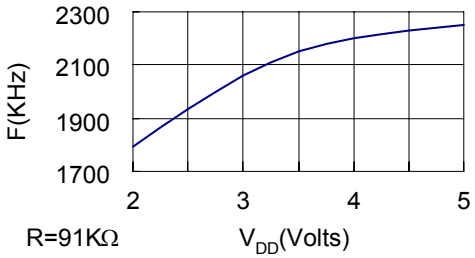




**Typical RC oscillator Frequency vs. $V_{DD}$:**
**(Reference only)**



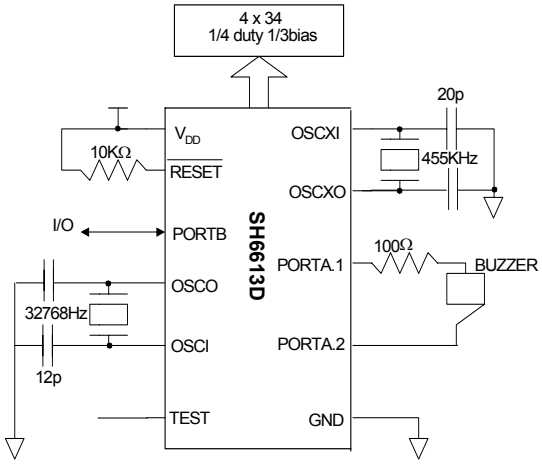



## Application Circuit (for reference only)

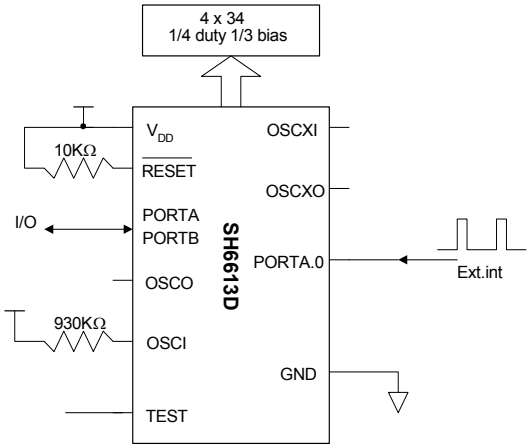
**AP1:**

OSC: Crystal oscillator 32.768KHz(mask option)
OSCX: Ceramic oscillator 455KHz
PORTB: I/O
PORTA.1, PORTA.2: ALARM output
LCD: Internal LCD 1/4 duty, 1/3 bias



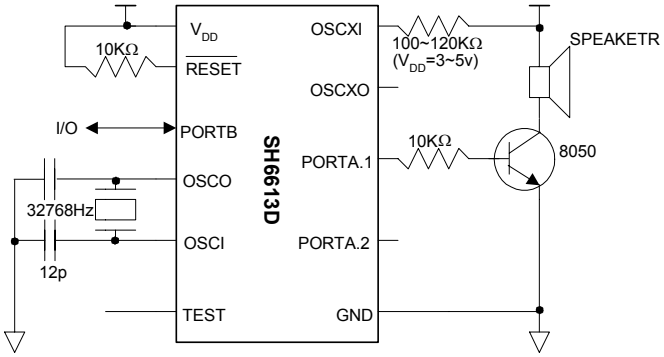

**AP2:**

OSC: RC oscillator 262KHz(mask option)
LCD: Internal LCD 1/4 duty, 1/3 bias
PORTA, PORTB: I/O
PORTA.0: External interrupt



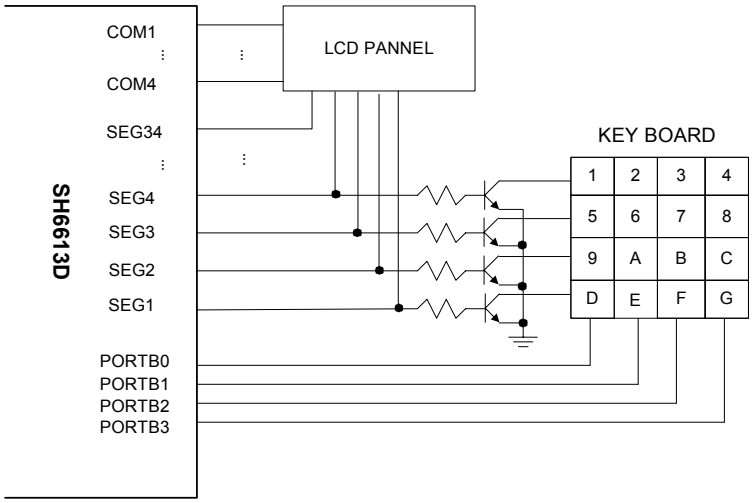
**AP3:**

OSC: Crystal oscillator 32.768KHz(mask option)
OSCX: RC oscillator 1.8MHz
PORTB: I/O
PORTA.1: PSG output
PORTA.2: $\overline{PSG}$ output




**AP4:**

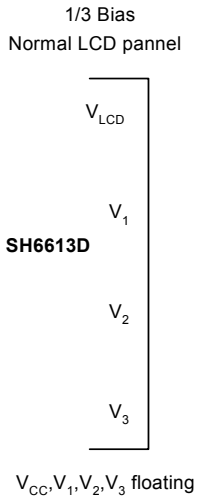
Internal LCD bias.1/4 DUTY 1/3bias

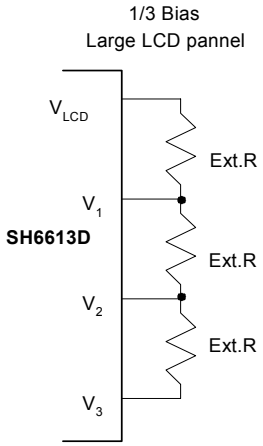



SEG1~SEG4 as SCAN outpost

**AP5:**

Large LCD panel: External LCD bias



### Music Table1

Following is the music scale reference table for channel 1(or channel 2) under OSCX=1.8MHz. Up to 6 octave is possible. Music scale data for 1.8M OSCX and SEL0=SEL1=1

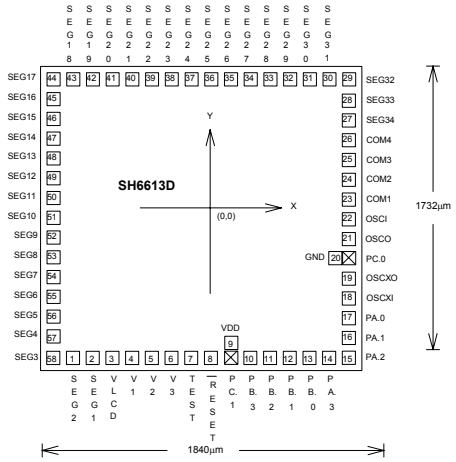
| Note | Ideal freq. | N | OCT1 /OCT2 | LSFR (C1.6~C1.0) (C2.14~C2.8) | Real freq. | Error% | Note | Ideal freq. | N | OCT1 /OCT2 | LSFR (C1.6~C1.0) (C2.14~C2.8) | Real freq. | Error% |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **B1** | 61.74 | 114 | 1 | 42 | 61.75 | 0.03% | **B4** | 493.88 | 114 | 0 | 42 | 494.04 | 0.03% |
| **C2** | 65.41 | 108 | 1 | 23 | 65.19 | -0.34% | **C5** | 523.25 | 108 | 0 | 23 | 521.48 | -0.34% |
| **D2** | 73.42 | 96 | 1 | 0B | 73.33 | -0.11% | **D5** | 587.33 | 96 | 0 | 0B | 586.67 | -0.11% |
| **E2** | 82.41 | 85 | 1 | 54 | 82.82 | 0.51% | **E5** | 659.26 | 85 | 0 | 54 | 662.59 | 0.51% |
| **F2** | 87.31 | 81 | 1 | 4F | 86.91 | -0.45% | **F5** | 698.46 | 81 | 0 | 4F | 695.31 | -0.45% |
| **G2** | 98.00 | 72 | 1 | 43 | 97.78 | -0.23% | **G5** | 783.99 | 72 | 0 | 43 | 782.22 | -0.23% |
| **A2** | 110.00 | 64 | 1 | 09 | 110.00 | 0.00% | **A5** | 880.00 | 64 | 0 | 09 | 880.00 | 0.00% |
| **B2** | 123.47 | 57 | 1 | 1B | 123.51 | 0.03% | **B5** | 987.77 | 57 | 0 | 1B | 988.07 | 0.03% |
| **C3** | 130.81 | 54 | 1 | 5A | 130.37 | -0.34% | **C6** | 1046.50 | 54 | 0 | 5A | 1042.96 | -0.34% |
| **D3** | 146.83 | 48 | 1 | 37 | 146.67 | -0.11% | **D6** | 1174.66 | 48 | 0 | 37 | 1173.33 | -0.11% |
| **E3** | 164.81 | 43 | 1 | 76 | 163.72 | -0.66% | **E6** | 1318.51 | 43 | 0 | 76 | 1309.77 | -0.66% |
| **F3** | 174.61 | 40 | 1 | 31 | 176.00 | 0.79% | **F6** | 1396.91 | 40 | 0 | 31 | 1408.00 | 0.79% |
| **G3** | 196.00 | 36 | 1 | 1A | 195.56 | -0.23% | **G6** | 1567.98 | 36 | 0 | 1A | 1564.44 | -0.23% |
| **A3** | 220.00 | 32 | 1 | 25 | 220.00 | 0.00% | **A6** | 1760.00 | 32 | 0 | 25 | 1760.00 | 0.00% |
| **B3** | 246.94 | 29 | 1 | 2E | 242.76 | -1.69% | **B6** | 1975.53 | 29 | 0 | 2E | 1942.07 | -1.69% |
| **C4** | 261.63 | 27 | 1 | 3B | 260.74 | -0.34% | **C7** | 2093.00 | 27 | 0 | 3B | 2085.93 | -0.34% |
| **D4** | 293.66 | 24 | 1 | 5C | 293.33 | -0.11% | **D7** | 2349.32 | 24 | 0 | 5C | 2346.67 | -0.11% |
| **E4** | 329.63 | 21 | 1 | 66 | 335.24 | 1.70% | **E7** | 2637.02 | 21 | 0 | 66 | 2681.90 | 1.70% |
| **F4** | 349.23 | 20 | 1 | 4C | 352.00 | 0.79% | **F7** | 2793.83 | 20 | 0 | 4C | 2816.00 | 0.79% |
| **G4** | 392.00 | 18 | 1 | 32 | 391.11 | -0.23% | **G7** | 3135.96 | 18 | 0 | 32 | 3128.89 | -0.23% |
| **A4** | 440.00 | 16 | 1 | 4A | 440.00 | 0.00% | **A7** | 3520.00 | 16 | 0 | 4A | 3520.00 | 0.00% |
| **B4** | 493.88 | 14 | 1 | 2A | 502.86 | 1.82% | **B7** | 3951.07 | 14 | 0 | 2A | 4022.86 | 1.82% |

**Music Table2**

Following is the music scale reference table for channel 1(or channel 2) under OSC=32.768KHz. Up to 4-octave is possible.
Music scale data for 32K OSC and SEL0=SEL1=0

| Note | Ideal freq. | N | OCT1 /OCT2 | LSFR (C1.6~C1.0) (C2.14~C2.8) | Real freq. | Error% | Note | Ideal freq. | N | OCT1 /OCT2 | LSFR (C1.6~C1.0) (C2.14~C2.8) | Real freq. | Error% |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **A1** | 55.00 | 37 | 1 | 0D | 55.35 | 0.64% | **C4** | 261.63 | 63 | 0 | 12 | 260.06 | -0.60% |
| **B1** | 61.74 | 33 | 1 | 52 | 62.06 | 0.53% | **D4** | 293.66 | 56 | 0 | 36 | 292.57 | -0.37% |
| **C2** | 65.41 | 31 | 1 | 4B | 66.06 | 1.01% | **E4** | 329.63 | 50 | 0 | 2D | 327.68 | -0.59% |
| **D2** | 73.42 | 28 | 1 | 5D | 73.14 | -0.37% | **F4** | 349.23 | 47 | 0 | 6F | 348.60 | -0.18% |
| **E2** | 82.41 | 25 | 1 | 6E | 81.92 | -0.59% | **G4** | 392.00 | 42 | 0 | 6C | 390.10 | -0.48% |
| **F2** | 87.31 | 23 | 1 | 39 | 89.04 | 1.99% | **A4** | 440.00 | 37 | 0 | 0D | 442.81 | 0.64% |
| **G2** | 98.00 | 21 | 1 | 66 | 97.52 | -0.48% | **B4** | 493.88 | 33 | 0 | 52 | 496.48 | 0.53% |
| **A2** | 110.00 | 19 | 1 | 19 | 107.79 | -2.01% | **C5** | 523.25 | 31 | 0 | 4B | 528.52 | 1.01% |
| **B2** | 123.47 | 17 | 1 | 65 | 120.47 | -2.43% | **D5** | 587.33 | 28 | 0 | 5D | 585.14 | -0.37% |
| **C3** | 130.81 | 125 | 0 | 04 | 131.07 | 0.20% | **E5** | 659.26 | 25 | 0 | 6E | 655.36 | -0.59% |
| **D3** | 146.83 | 112 | 0 | 0A | 146.29 | -0.37% | **F5** | 698.46 | 23 | 0 | 39 | 712.35 | 1.99% |
| **E3** | 164.81 | 99 | 0 | 11 | 165.49 | 0.41% | **G5** | 783.99 | 21 | 0 | 66 | 780.19 | -0.48% |
| **F3** | 174.61 | 94 | 0 | 2C | 174.30 | -0.18% | **A5** | 880.00 | 19 | 0 | 19 | 862.32 | -2.01% |
| **G3** | 196.00 | 84 | 0 | 29 | 195.05 | -0.48% | **B5** | 987.77 | 17 | 0 | 65 | 963.76 | -2.43% |
| **A3** | 220.00 | 74 | 0 | 50 | 221.41 | 0.64% | **C6** | 1046.50 | 16 | 0 | 4A | 1024.00 | -2.15% |
| **B3** | 246.94 | 66 | 0 | 62 | 248.24 | 0.53% | **D6** | 1174.66 | 14 | 0 | 2A | 1170.29 | -0.37% |

## Bonding diagram




**Substrate connect to GND**

| Pad No. | Designation | X(μm) | Y(μm) | Pad No. | Designation | X(μm) | Y(μm) |
|---|---|---|---|---|---|---|---|
| 1 | SEG[2] | -688.50 | -756.00 | 29 | SEG[32] | 814.50 | 756.00 |
| 2 | SEG[1] | -571.50 | -756.00 | 30 | SEG[31] | 697.50 | 756.00 |
| 3 | $V_{LCD}$ | -463.50 | -756.00 | 31 | SEG[30] | 580.50 | 756.00 |
| 4 | $V_1$ | -360.00 | -756.00 | 32 | SEG[29] | 472.50 | 756.00 |
| 5 | $V_2$ | -256.50 | -756.00 | 33 | SEG[28] | 369.00 | 756.00 |
| 6 | $V_3$ | -153.00 | -756.00 | 34 | SEG[27] | 265.50 | 756.00 |
| 7 | TEST | - 49.50 | -756.00 | 35 | SEG[26] | 162.00 | 756.00 |
| 8 | $\overline{\text{RESET}}$ | 54.00 | -756.00 | 36 | SEG[25] | 58.50 | 756.00 |
| 9 | $V_{DD}$ | 166.50 | -671.40 | 37 | SEG[24] | -49.50 | 756.00 |
| bonding option | PORTC[1] | 166.50 | -756.00 | 38 | SEG[23] | -153.00 | 756.00 |
| 10 | PORTB.3 | 270.00 | -756.00 | 39 | SEG[22] | -256.50 | 756.00 |
| 11 | PORTB.2 | 373.50 | -756.00 | 40 | SEG[21] | -360.00 | 756.00 |
| 12 | PORTB.1 | 477.00 | -756.00 | 41 | SEG[20] | -463.50 | 756.00 |
| 13 | PORTB.0 | 580.50 | -756.00 | 42 | SEG[19] | -571.50 | 756.00 |
| 14 | PORTA.3 | 697.50 | -756.00 | 43 | SEG[18] | -688.50 | 756.00 |
| 15 | PORTA.2 | 814.50 | -756.00 | 44 | SEG[17] | -814.50 | 756.00 |
| 16 | PORTA.1 | 814.50 | -639.00 | 45 | SEG[16] | -814.50 | 639.00 |
| 17 | PORTA.0 | 814.50 | -531.00 | 46 | SEG[15] | -814.50 | 531.00 |
| 18 | OSCXI | 814.50 | -423.00 | 47 | SEG[14] | -814.50 | 423.00 |
| 19 | OSCXO | 814.50 | -319.50 | 48 | SEG[13] | -814.50 | 319.50 |
| 20 | GND | 729.90 | -216.00 | 49 | SEG[12] | -814.50 | 216.00 |
| bonding option | PORTC[0] | 814.50 | -216.00 | 50 | SEG[11] | -814.50 | 108.00 |
| 21 | OSCO | 814.50 | -108.00 | 51 | SEG[10] | -814.50 | 0.00 |
| 22 | OSCI | 814.50 | 0.00 | 52 | SEG[9] | -814.50 | -108.00 |
| 23 | COM[1] | 814.50 | 108.00 | 53 | SEG[8] | -814.50 | -216.00 |
| 24 | COM[2] | 814.50 | 216.00 | 54 | SEG[7] | -814.50 | -319.50 |
| 25 | COM[3] | 814.50 | 319.50 | 55 | SEG[6] | -814.50 | -423.00 |
| 26 | COM[4] | 814.50 | 423.00 | 56 | SEG[5] | -814.50 | -531.00 |
| 27 | SEG[34] | 814.5 | 531.00 | 57 | SEG[4] | -814.50 | -639.00 |
| 28 | SEG[33] | 814.50 | 639.00 | 58 | SEG[3] | -814.50 | -756.00 |

**Ordering Information**

| Part No. | Package |
|---|---|
| SH6613DH | CHIP FORM |