Sitronix

ST2012

PRELIMINARY

12K 8-bit Single Chip Microcontroller

Notice: This is not a final specification. Some parameters are subject to change.

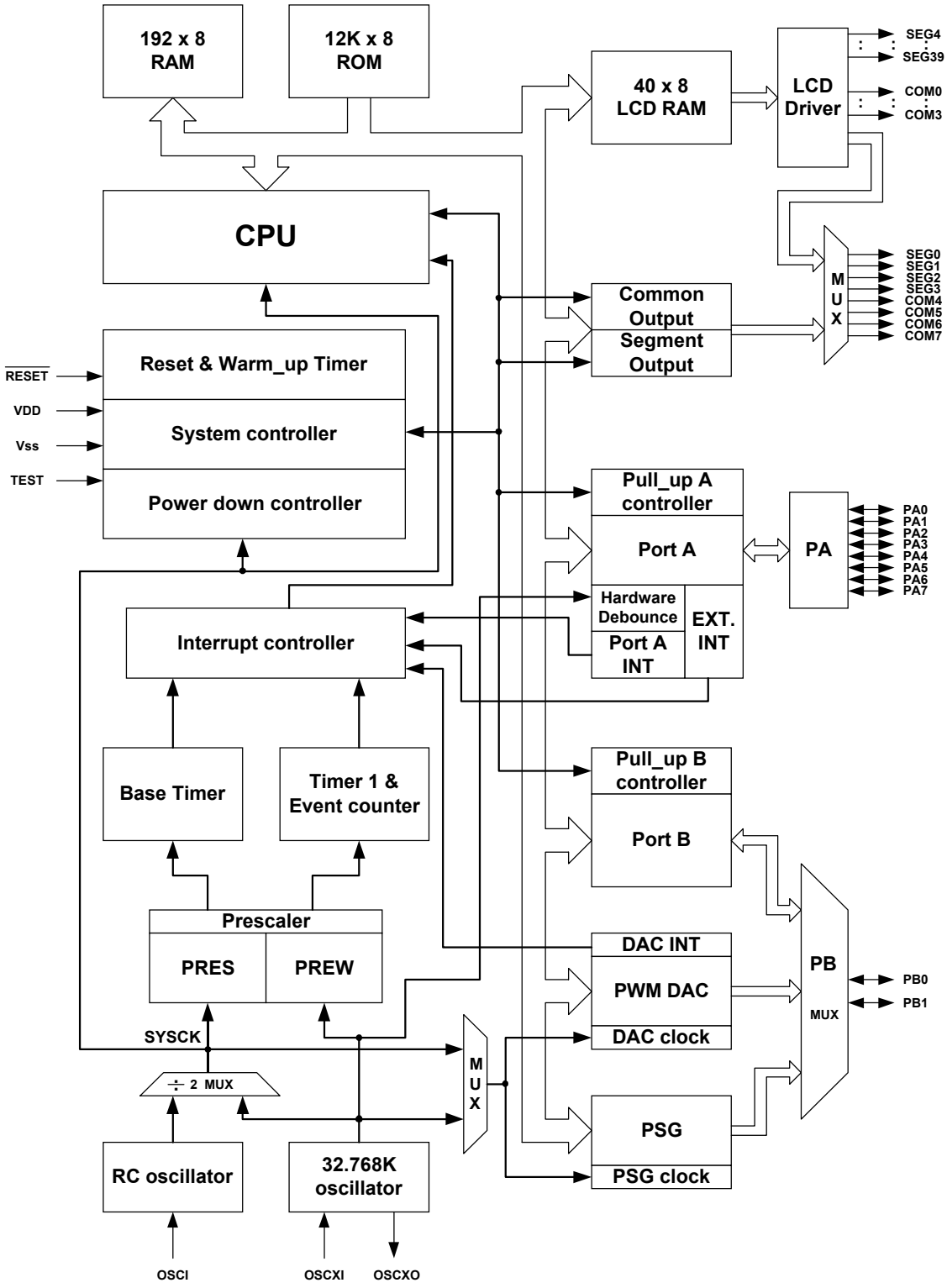
1. FEATURES

- 8-bit static pipeline CPU
- ROM: 12K x 8 bits
- RAM: 192 x 8 bits
- Operation voltage : 2.4V ~ 5V
- 10 CMOS Bi-directional bit programmable I/O pins
- 8 Output pins (Shared with LCD common/segment)
- Hardware debounce option for input port
- Bit programmable PULL-UP for input port
- Timer/Counter :
  - One 8-bit timer / 16-bit event counter
  - One 8-bit BASE timer
- Five powerful interrupt sources :
  - External interrupt (edge trigger)
  - TIMER1 interrupt
  - BASE timer interrupt
  - PORTA[7~0] interrupt (transition trigger)
  - DAC reload interrupt
- 32-level deep stack
- Dual clock source :
  - OSCX: Crystal oscillator: 32.768K Hz
  - OSC: RC oscillator 500K ~ 4M Hz
- Build-in oscillator with warm-up timer
- LCD driver programmable duty :
  - 320 (8x40) dots (1/8 duty, 1/4 bias)
  - 160 (4x40) dots (1/4 duty, 1/3 bias)
- Programmable Sound Generator (PSG) includes :
  - Tone generator
  - Sound effect generator
  - 4 level volume control
  - Digital DAC for speech / tone
- Three power down modes :
  - WAI0 mode
  - WAI1 mode
  - STP mode

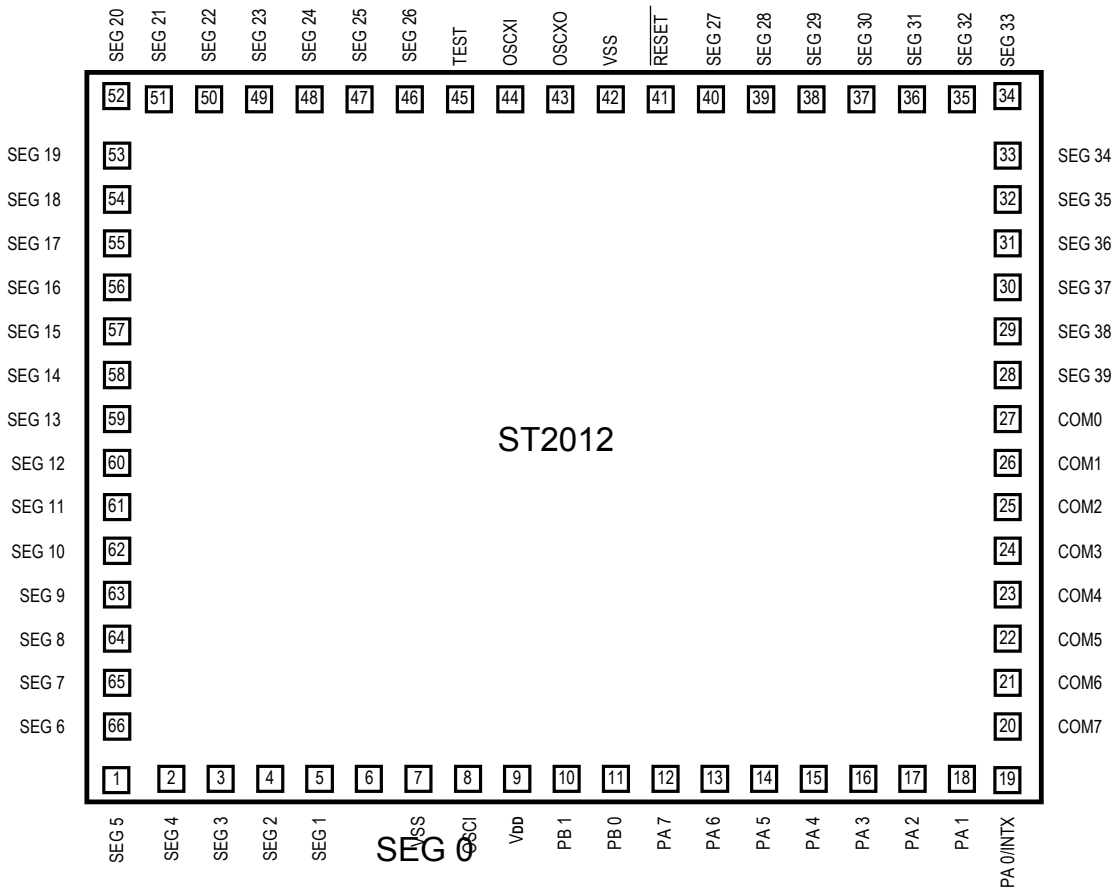
2. GENERAL DESCRIPTION

ST2012 is a low-cost, high-performance, fully static, 8-bit microcontroller designed with CMOS silicon gate technology. It comes with 8-bit pipeline CPU core, SRAM, timer, LCD driver, I/O port, PSG and mask program ROM. A build-in dual oscillator is specially integrated to enhance chip performance. For business equipment and consumer applications. Such as watch, calculator, LCD game and IR remote control, ST2012 is definitely a perfect solution for implementation.

3. BLOCK DIAGRAM



4. PAD DIAGRAM

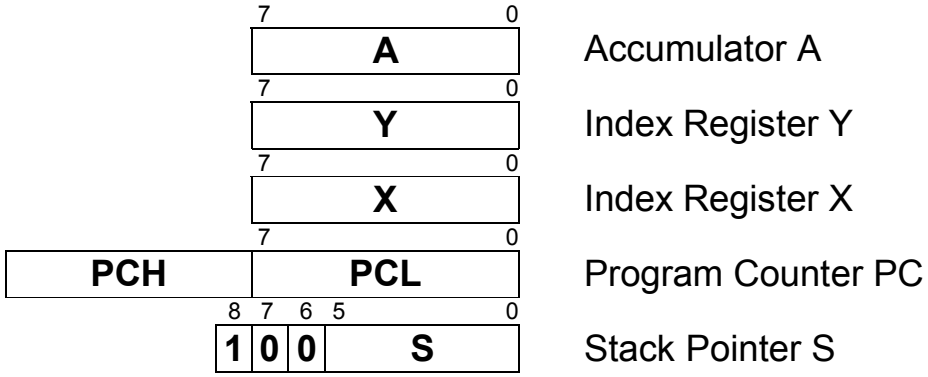


5. PAD DESCRIPTION

Designation	Pad #	Type	Description
SEG 0 - 3	3~6	O O	LCD Segment output Output port
SEG 4 - 39	1,2, 28~40, 46~66	O	LCD Segment output
COM 0 - 3	24~27	O	LCD Common output
COM 4 - 7	20~23	O O	LCD Common output Output port
$\overline{\text{RESET}}$	41	I	Pad reset input (LOW Active)
VSS	7, 42	P	Ground Input and chip substrate
PA0/INTX	19	I/O I I I	Port-A bit programmable I/O Edge-trigger Interrupt. Transition-trigger Interrupt Programmable Timer1 clock source
PA 1-7	12~18	I/O I	Port-A bit programmable I/O Transition-trigger Interrupt
PB 0-1	10, 11	I/O O	Port-B bit programmable I/O PSG/DAC Output
V_{DD}	9	P	Power supply
OSCXI	44	I	OSC input pin. For 32768Hz crystal
OSCXO	43	O	OSC output pin. For 32768Hz crystal
OSCI	8	I	OSC input pin. toward to external resistor
TEST	45	I	Test pin

Legend: I = input, O = output, I/O = input/output, P = power.

6. CPU



CPU REGISTER MODEL

6.1 Accumulator (A)

The accumulator is a general purpose 8-bit register which stores the results of most arithmetic and logic operations. In addition, the accumulator usually contains one of the two data words used in these operations.

6.2 Index Registers (X,Y)

There are two 8-bit Index Registers (X and Y) which may be used to count program steps or to provide and index value to be used in generating an effective address. When executing an instruction which specifies indexed addressing, the CPU fetches the OP code and the base address, and modifies the address by adding the index register to it prior to performing the desired operation. Pre or post-indexing of indirect addresses is possible.

6.3 Stack Pointer (S)

The stack Pointer is an 8-bit register which is used to control the addressing of the variable-length stack. It's range from 100H to 13FH total for 64 bytes (32-level deep). The stack pointer is automatically incremented and decrement under control of the microprocessor to perform stack manipulations under direction of either the program or interrupts (IRQ). The stack allows simple implementation of nested subroutines and multiple level interrupts. The stack pointer is initialized by the user's software.

6.4 Program Counter (PC)

The 16-bit Program Counter register provides the address which step the microprocessor through sequential program instructions. Each time the microprocessor fetches and instruction from program memory, the lower byte of the program counter (PCL) is placed on the low-order bits of the address bus and the higher byte of the program counter (PCH) is placed on the high-order 8 bits. The counter is incremented each time an instruction or data is fetched from program memory.

6.5 Status Register (P)

The 8-bit Processor Status Register contains seven status flags. Some of the flags are controlled by the program, others may be controlled both by the program and the CPU. The instruction set contains a member of conditional branch instructions which are designed to allow testing of these flags.

TABLE 6-1: STATUS REGISTER (P)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
N	V	1	B	D	I	Z	C

Bit 7: **N :** Signed flag by arithmetic
1 = Negative
0 = Positive

Bit 6: **V :** Overflow of signed Arithmetic flag
1 = Negative
0 = Positive

Bit 4: **B : BRK** interrupt flag *
1 = **BRK** interrupt occur
0 = Non **BRK** interrupt occur

Bit 3: **D :** Decimal mode flag
1 = Decimal mode
0 = Binary mode

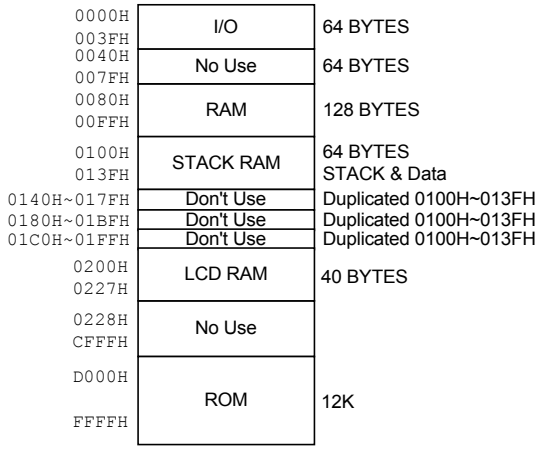
Bit 2: **I :** Interrupt disable flag
1 = Interrupt disable
0 = Interrupt enable

Bit 1: **Z :** Zero flag
1 = Zero
0 = Non zero

Bit 0: **C :** Carry flag
1 = Carry
0 = Non carry

*** Don't use "BRK" instruction.**

7. MEMORY CONFIGURATION



7.1 ROM ($D000~$FFFF)

The ST2012 has 12K bytes ROM used for program, data and vector address.

Vector address mapping :

$FFFE	Reserved.
$FFFC	RESET vector.
$FFFA	Reserved.
$FFF8	INTX (PA0) edge interrupter.
$FFF6	Reload DAC data interrupter.
$FFF4	Reserved.
$FFF2	Timer1 interrupter.
$FFF0	PORTA transition interrupter.
$FFEE	Base Timer interrupter.

7.2 RAM

The RAM mapping includes Control Registers, Data RAM, Stack RAM and LCD RAM.

TABLE 7-2: CONTROL REGISTERS ($0000~$003E)

Address	Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default
$000	**PA**	R/W	PA[7]	PA[6]	PA[5]	PA[4]	PA[3]	PA[2]	PA[1]	PA[0]	1111 1111
$001	**PB**	R/W	-	-	-	-	-	-	PB[1]	PB[0]	- - - - - -11
$008	**PCA**	R/W	PCA[7]	PCA[6]	PCA[5]	PCA[4]	PCA[3]	PCA[2]	PCA[1]	PCA[0]	0000 0000
$009	**PCB**	R/W	-	-	-	-	-	-	PCB[1]	PCB[0]	- - - - - -00
$00F	**PMCR**	R/W	PULL	PDBN	INTEG	-	-	-	PSGO	PSGB	100 - - -00
$012	**PSGL**	R/W	PSG[7]	PSG[6]	PSG[5]	PSG[4]	PSG[3]	PSG[2]	PSG[1]	PSG[0]	0000 0000
$013	**PSGH**	R/W	-	-	-	-	PSG[11]	PSG[10]	PSG[9]	PSG[8]	- - - - 0000
$014	**DAC**	R/W	DAC[7]	DAC[6]	DAC[5]	DAC[4]	DAC[3]	DAC[2]	DAC[1]	DAC[0]	0000 0000
$016	**PSGC**	R/W	-	PCK[2]	PCK[1]	PCK[0]	PRBS	C1EN	-	DACE=0	-000 00-0
		R/W	-	PCK[2]	PCK[1]	PCK[0]	DMD[1]	DMD[0]	INH	DACE=1	-000 0000
$017	**VOL**	R/W	VOL[1]	VOL[0]	-	-	-	-	-	-	00- - - - - -
$020	**LCK**	R/W	-	-	-	-	-	LCK[2]	LCK[1]	LCK[0]	- - - - - 100
$021	**BTM**	R/W	-	-	-	-	BTM[3]	BTM[2]	BTM[1]	BTM[0]	- - - - 0000
$023	**PRS**	R	PRS[7]	PRS[6]	PRS[5]	PRS[4]	PRS[3]	PRS[2]	PRS[1]	PRS[0]	0000 0000
		W	SRES	SENA	SENT	-	-	-	-	-	000 - - - - -
$026	**T1M**	R/W	-	-	-	T1M[4]	T1M[3]	T1M[2]	T1M[1]	T1M[0]	- - -0 0000
$027	**T1C**	R/W	T1C[7]	T1C[6]	T1C[5]	T1C[4]	T1C[3]	T1C[2]	T1C[1]	T1C[0]	0000 0000
$030	**SYS**	R/W	XSEL	OSTP	XSTP	XBAK	WSKP	WAIT	-	-	0000 00- -
$03A	**LCTL**	W	LPWR	BLANK	COMO	LENH	SEGO	-	-	DUTY	0000 0- -0
$03B	**SCAN**	R/W	SCAN[7]	SCAN[6]	SCAN[5]	SCAN[4]	-	-	-	-	0000 - - - -
$03C	**IREQ**	R/W	-	-	IRBT	IRPT	IRT1	-	IRDAC	IRX	- - 11 1-11
$03E	**IENA**	R/W	-	-	IEBT	IEPT	IET1	-	IEDAC	IEX	- - 00 0-00

Note: 1. Some addresses of I/O area, $2~$7, $A~$E, $10~$11, $15, $18~$1F, $22, $24~$25, $28~$2F, $31~$39, $3F, are no used.
2. User should never use undefined addresses and bits.
3. Do not use Bit instructions for write-only registers, such as RMBx, SMBx….
4. E.V.B 's RAM Power On Initial Value are Same as Real Chip.
5. **Only ST2012B LCD Enhance Version Can Use XLCD($038)，ST2012 Normal Version Inhibit Use.**

7.2.2 DATA RAM ($0080~$00FF)

DATA RAM are organized in 128 bytes.

7.2.3 STACK RAM ($0100~$013F)

STACK RAM are organized in 64 bytes. It provides for a maximum of 32-level subroutine stacks And can be used as data memory.

7.2.4 LCD RAM ($0200~$0227)

Resident LCD-RAM, accessible through write and read instructions, are organized in 40 bytes for 40x8 LCD display. Note that this area can also be used as data memory.

8. INTERRUPTS

TABLE 8-3: PREDEFINED VECTORS FOR INTERRUPT

Name	Signal	Vector address	Priority	Comment
-	-	$FFFF,$FFFE	-	Reserved
RESET	External	$FFFD,$FFFC	1	RESET vector
-	-	$FFFB,$FFFA	-	Reserved
INTX	External	$FFF9,$FFF8	2	PA0 edge interrupt
DAC	Internal	$FFF7,$FFF6	3	Reload DAC data interrupt
-	-	$FFF5,$FFF4	-	Reserved
T1	INT/EXT	$FFF3,$FFF2	4	Timer1 interrupt
PT	External	$FFF1,$FFF0	5	Port-A transition interrupt
BT	Internal	$FFEF,$FFEE	6	Base Timer interrupt

8.2 Interrupt description

RESET
A positive transition of RESET pin will then cause an initialization sequence to begin. After the system has been operating, a low on this line of a least two clock cycles will cease ST2012 activity. When a positive edge is detected, there is an initialization sequence lasting six clock cycles. Then the interrupt mask flag is set, the decimal mode is cleared and the program counter will loaded with the restart vector from locations $FFFC (low byte) and $FFFD (high byte). This is the start location for program control. This input should be high in normal operation.

INTX interrupt
The IRX (INTX interrupt request) flag will be set while INTX edge signal occurs. The INTX interrupt will be active once IEX (INTX interrupt enable) is set, and interrupt mask flag is cleared. Hardware will push 'PC', 'P' Register to stack and set interrupt mask flag (I). Program counter will be loaded with the INTX vector from locations $FFF8 and $FFF9.

DAC interrupt
The IRDAC (DAC interrupt request) flag will be set while reload signal of DAC occurs. Then the DAC interrupt will be executed when IEDAC (DAC interrupt enable) is set, and interrupt mask flag is cleared. Hardware will push 'PC', 'P' Register to stack and set interrupt mask flag (I). Program counter will be loaded with the DAC vector from locations $FFF6 and $FFF7.

T1 interrupt
The IRT1 (TIMER1 interrupt request) flag will be set while T1 overflows. With IET1 (TIMER1 interrupt enable) being set, the T1 interrupt will executed, and interrupt mask flag will be cleared. Hardware will push 'PC', 'P' Register to stack and set interrupt mask flag (I). Program counter will be loaded with the T1 vector from locations $FFF2 and $FFF3.

PT interrupt
The IRPT (Port-A interrupt request) flag will be set while Port-A transition signal occurs. With IEPT (PT interrupt enable)being set, the PT interrupt will be execute, and interrupt mask flag will be cleared. Hardware will push 'PC', 'P' Register to stack and set interrupt mask flag (I). program counter will be loaded with the PT vector from locations $FFF0 and $FFF1.

BT interrupt
The IRBT (Base timer interrupt request) flag will be set when Base Timer overflows. The BT interrupt will be executed once the IEBT (BT interrupt enable) is set and the interrupt mask flag is cleared. Hardware will push 'PC', 'P' Register to stack and set interrupt mask flag (I). Program counter will be loaded with the BT vector from locations $FFEE and $FFEF.

8.3 Interrupt request clear

Interrupt request flag can be cleared by two methods. One is to write “0” to IENA, the other is to initiate the interrupt service routine when interrupt occurs. Hardware will automatically clear the Interrupt flag.

TABLE 8-4: INTERRUPT REQUEST REGISTER (IREQ)

Address	Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default
$03C	**IREQ**	R/W	-	-	IRBT	IRPT	IRT1	-	IRDAC	IRX	- - 11 1-11

Bit 5: **IRBT:** Base Timer Interrupt Request bit
1 = Time base interrupt occurs
0 = Time base interrupt doesn’t occur

Bit 4: **IRPT:** Port-A Interrupt Request bit
1 = Port-A transition interrupt occurs
0 = Port-A transition interrupt doesn’t occur

Bit 3: **IRT1:** Timer1 Interrupt Request bit
1 = Timer1 overflow interrupt occurs
0 = Timer1 overflow interrupt doesn’t occur

Bit 1: **IRDAC:** DAC reload Interrupt Request bit
1 = DAC time out interrupt occurs
0 = DAC time out interrupt doesn’t occur

Bit 0: **IRX:** INTX Interrupt Request bit
1 = INTX edge interrupt occurs
0 = INTX edge interrupt doesn’t occur

TABLE 8-5: INTERRUPT ENABLE REGISTER (IENA)

Address	Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default
$03E	**IENA**	R/W	-	-	IEBT	IEPT	IET1	-	IEDAC	IEX	- - 00 0-00

Bit 5: **IEBT:** Base Timer Interrupt Enable bit
1 = Time base interrupt enable
0 = Time base interrupt disable

Bit 4: **IEPT:** Port-A Interrupt Enable bit
1 = Port-A transition interrupt enable
0 = Port-A transition interrupt disable

Bit 3: **IET1:** Timer1 Interrupt Enable bit
1 = Timer1 overflow interrupt enable
0 = Timer1 overflow interrupt disable

Bit 1: **IEDAC:** DAC reload Interrupt Enable bit
1 = DAC time out interrupt enable
0 = DAC time out interrupt disable

Bit 0: **IEX:** INTX Interrupt Enable bit
1 = INTX edge interrupt enable
0 = INTX edge interrupt disable

9. I/O PORTS

ST2012 has four I/O ports, PORT-A, PORT-B, SEGMENT-PORT and COMMON-PORT. In total, ST2012 provides for a maximum of 18 I/O pins with both SEGMENT-PORT and COMMON-PORT being programmed as output ports. For detail pin assignment, please refer to Table 9-6 :

TABLE 9-6: I/O DESCRIPTION

PORT NAME	PAD NAME	PAD NUMBER	PIN TYPE	FEATURE
PORTA	PA0/INTX	19	I/O	Programmable input/output pin
	PA1	18	I/O	
	PA2	17	I/O	
	PA3	16	I/O	
	PA4	15	I/O	
	PA5	14	I/O	
	PA6	13	I/O	
	PA7	12	I/O	
PORTB	PB0	11	I/O	Programmable input/output pin
	PB1	10	I/O	
SEGMENT PORT	SEG0	6	O	These 4 segment pins can be programmed as output ports.
	SEG1	5	O	
	SEG2	4	O	
	SEG3	3	O	
COMMON PORT	COM4	23	O	These 4 common pins can be programmed as output ports.
	COM5	22	O	
	COM6	21	O	
	COM7	20	O	

9.2 PORT-A

Port- A is a bit-programmable bi-direction I/O port, which is controlled by PCA register. It provides user with bit programmable pull-up MOS, interrupt debounce and interrupt edge selection(PA0 only).

TABLE 9-7: SUMMARY FOR PORT-A REGISTERS

Address	Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default
$000	**PA**	R/W	PA[7]	PA[6]	PA[5]	PA[4]	PA[3]	PA[2]	PA[1]	PA[0]	1111 1111
$008	**PCA**	R/W	PCA[7]	PCA[6]	PCA[5]	PCA[4]	PCA[3]	PCA[2]	PCA[1]	PCA[0]	0000 0000
$00F	**PMCR**	R/W	PULL	PDBN	INTEG	-	-	-	PSG0	PSGB	100 - - -00
$03C	**IREQ**	R/W	-	-	IRBT	IRPT	IRT1	-	IRDAC	IRX	- - 11 1-11
$03E	**IENA**	R/W	-	-	IEBT	IEPT	IET1	-	IEDAC	IEX	- - 00 0-00

9.2.2 PORT-A I/O control

Direction of Port-A is controlled by PCA. Every bit of PCA[7~0] is mapped to the I/O direction of PA[7~0] correspondingly, with “1” for output mode, and “0” for input mode.

TABLE 9-8: PORT-A CONTROL REGISTER (PCA)

Address	Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default
$008	**PCA**	R/W	PCA[7]	PCA[6]	PCA[5]	PCA[4]	PCA[3]	PCA[2]	PCA[1]	PCA[0]	0000 0000

Bit 7~0: **PCA[7~0] :** Port-A directional bits
1 = Output mode
0 = Input mode

9.2.3 PORT-A PULL-UP OPTION

PORT-A contains pull-up MOS transistors controlled by software. When an I/O is used as an input. The ON/OFF of the pull-up MOS transistor will be controlled by port data register (PA) and the pull-up MOS will be enabled with "1" for data bit and disable with "0" for data bit. The PULL control bit of PMCR controls the ON/OFF of all the pull-up MOS simultaneously. Please refer to the Figure 9-1.

FIGURE 9-1: Port-A Configuration Function Block Diagram

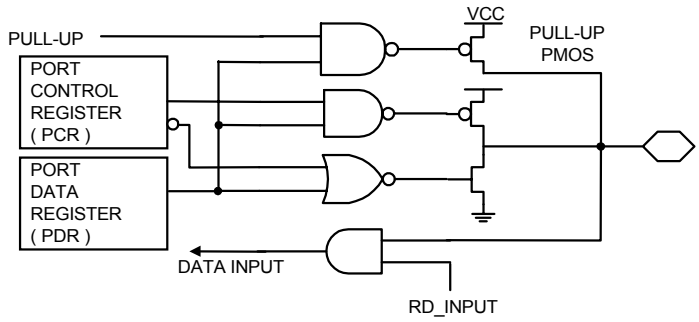


TABLE 9-9: PORT CONDITION CONTROL REGISTER (PMCR)

Address	Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default
$00F	**PMCR**	R/W	PULL	PDBN	INTEG	-	-	-	PSG0	PSGB	100 - - -00

Bit 7: **PULL :** Enable all pull-up function bit
1 = Enable pull-up function
0 = Disable pull-up function

Bit 6: **PDBN :** Enable Port-A interrupt debounce bit
1 = Debounce for Port-A interrupt
0 = No debounce for Port-A interrupt

Bit 5: **INTEG :** INTX interrupt edge select bit
1 = Rising edge
0 = Falling edge

9.2.4 Port-A interrupt

Port-A, a programmable I/O, can be used as a port interrupt when it is in the input mode. Any edge transition of the Port-A input pin will generate an interrupt request. The last state of Port-A must be kept before I/O transition and this can be accomplished by reading Port-A.

When programmer enables INTX and PT interrupts, PA0 trigger occur. INTX and PT interrupts will therefore happen sequentially. Please refer to the Figure 9-2.

Operating Port-A interrupt step by step :

1. Set input mode.
2. Read Port-A.
3. Clear interrupt request flag (IRPT).
4. Set interrupt enable flag (IEPT).
5. Clear CPU interrupt disable flag (I).
6. Read Port-A before 'RTI' instruction in INT-Subroutine.

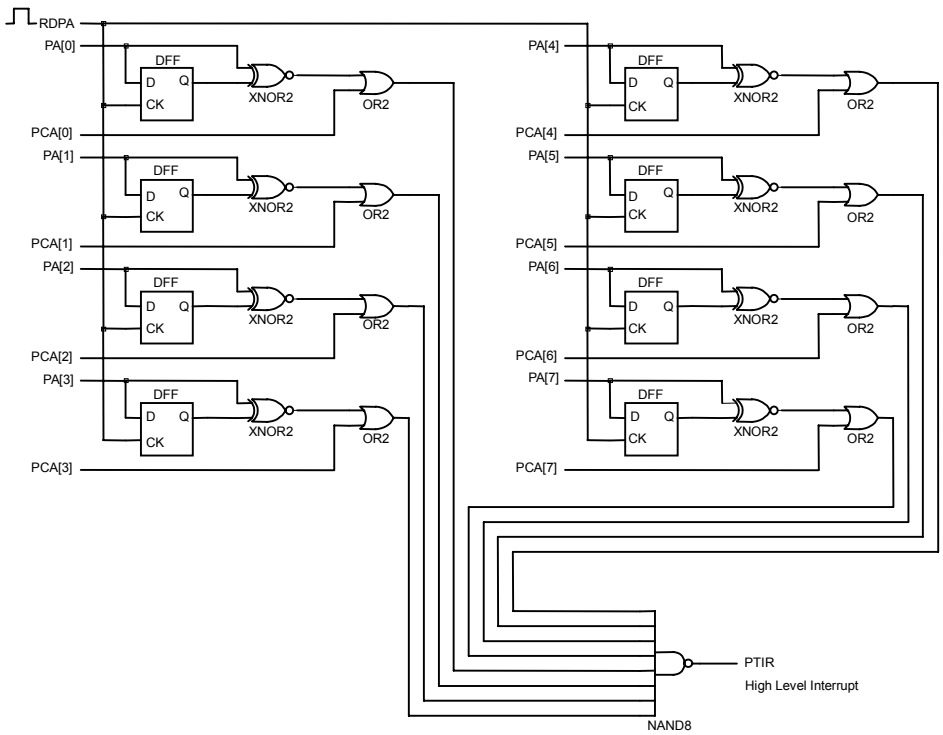
Example :

```
        .
        .
        .
STZ     PCA         ;Set input mode.
LDA     #$FF
STA     PA          ;PA be PULL-UP.
LDA     PA          ;Keep last state.
RMB4    <IREQ       ;Clear IRQ flag.
SMB4    <IENA       ;Enable INT.
CLI
        .
        .
```

INT-SUBROUTINE

```
        .
        .
LDA     PA          ;Keep last state.
RTI
```

FIGURE 9-2: Port Interrupt Logic Diagram



9.2.4.2 Port-A interrupt debounce

ST2012 has hardware debounce option for Port-A interrupt. The debounce will be enabled with "1" and disable with "0" for PDBN. The debounce will active when Port-A transition occurs, PDBN enable and **OSCX enable**.

The debounce time is **OSCX x 512 cycles(about 16 ms).** Refer to the TABLE 9-10.

TABLE 9-10: PORT CONDITION CONTROL REGISTER (PMCR)

Address	Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default
$00F	**PMCR**	R/W	PULL	PDBN	INTEG	-	-	-	PSG0	PSGB	100 - - -00

Bit 6: **PDBN :** Enable Port-A interrupt debounce bit
1 = Debounce for Port-A interrupt
0 = No debounce for Port-A interrupt

9.2.5 PA0/INTX

PA0 can be used as an external interrupt input(INTX). Falling or Rising edge is controlled by INTEG(PMCR[5]) and the external interrupt is set up with “0” for falling edge and “1” for rising edge. Please refer to the Figure 9-3.

When programmer enables INTX and PT interrupts, PA0 trigger will occur. Both INTX and PT interrupts will happen sequentially. Pelase refer to the operating steps.

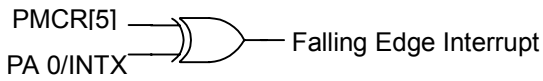
Operating INTX interrupt step by step :

1. Set PA0 pin into input mode. (PCA[0])
2. Select edge level. (INTEG)
3. Clear INTX interrupt request flag. (IRX)
4. Set INTX interrupt enable bits. (IEX)
5. Clear CPU interrupt mask flag (I).

Example :

```
.
.
.
RMB0    <PCA     ;Set input mode.
SMB5    <PMCR    ;Rising edge.
RMB0    <IREQ    ;Clear IRQ flag.
SMB0    <IENA    ;Enable INTX interrupt.
CLI
.
.
.
```

FIGURE 9-3: INTX Logic Diagram



9.3 PORT-B

Port -B is a bit programmable bi-direction I/O port, which is controlled by PCB register. It also provides user with bit-programmable pull-up MOS and sound output port separately.

TABLE 9-11: SUMMARY FOR PORT-B REGISTERS

Address	Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default
$001	**PB**	R/W	-	-	-	-	-	-	PB[1]	PB[0]	- - - - - -11
$009	**PCB**	R/W	-	-	-	-	-	-	PCB[1]	PCB[0]	- - - - - -00
$00F	**PMCR**	R/W	PULL	PDBN	INTEG	-	-	-	PSG0	PSGB	100 - - -00

9.3.2 PORT-B I/O control

Direction of Port-B is controlled by PCB. Every bit of PCB[1~0] is mapped into the I/O direction of PB[1~0] correspondingly, with “1” for output mode, and “0” for input mode.

TABLE 9-12: PORT-B CONTROL REGISTER (PCB)

Address	Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default
$009	**PCB**	R/W	-	-	-	-	-	-	PCB[1]	PCB[0]	- - - - - -00

Bit 1~0: **PCB[1~0] :** Port-B directional bits
1 = Output mode
0 = Input mode

9.3.3 PORT-B PULL-UP OPTION

This port contains pull-up MOS transistors which is controlled by software and can be enabled or disabled with "1" or with "0" accordingly in data bit of the port data register (PB) when an I/O is used as an input. The PULL control bit of PMCR also controls the ON/OFF of all pull-up MOS simultaneously. Please refer to the Figure 9-4.

FIGURE 9-4: Port-B Configuration Function Block Diagram

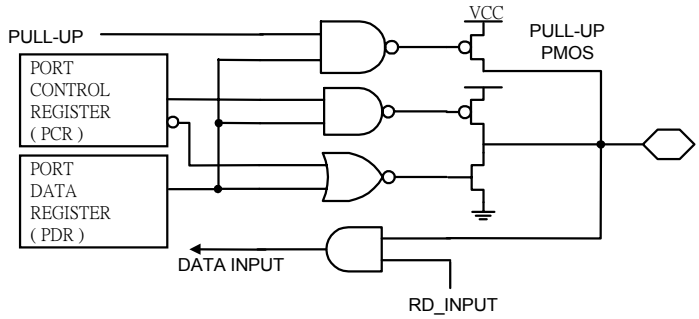


TABLE 9-13: PORT CONDITION CONTROL REGISTER (PMCR)

Address	Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default
$00F	**PMCR**	R/W	PULL	PDBN	INTEG	-	-	-	PSG0	PSGB	100 - - -00

Bit 7: **PULL :** Enable all pull-up functions bit
1 = Enable pull-up function
0 = Disable pull-up function

Bit 1: **PSGO :** PSG output enable bit
1 = PB1 is PSG data output pin if PB1 is set in output mode
0 = PB1 is normal I/O pin

Bit 0: **PSGB :** PSG inverse signal output enable bit
1 = PB0 is PSG inverse data output pin if PB0 is set in output mode
0 = PB0 is normal I/O pin

9.4 SEGMENT-PORT

The SEG0~SEG3 can be used as LCD drivers or output ports. In output port mode, programmer must write $FF($00) into LCD RAM in order to output HIGH(LOW). The assignments of SEGO will be decided by Bit 3 of LCTL[3]. Please refer to TABLE9-14.

TABLE 9-14: LCD CONTROL REGISTER (LCTL)

Address	Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default
$03A	**LCTL**	W	LPWR	BLANK	COMO	LENH	SEGO	-	-	DUTY	0000 0- -0

Bit 3: **SEGO :** Segment output selection bit
1 = SEG0-SEG3 used as output pins
0 = SEG0-SEG3 used as LCD segment pins

TABLE 9-15: SEGMENT OUT REGISTER

Address	Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default
$200	**SEGMENT OUT 0**	W	SEGMENT-0 OUTPUT BIT								???? ????
$201	**SEGMENT OUT 1**	W	SEGMENT-1 OUTPUT BIT								???? ????
$202	**SEGMENT OUT 2**	W	SEGMENT-2 OUTPUT BIT								???? ????
$203	**SEGMENT OUT 3**	W	SEGMENT-3 OUTPUT BIT								???? ????

In the output port mode, programmer must write $FF($00) into LCD RAM to output HIGH(LOW).

9.5 COMMON-PORT

The COM4~COM7 can be used as LCD drivers or output ports. In output port mode, SCAN[7~4] will be map to COM7~COM4 output ports, which pin assignment will be decided by Bit 5 of LCTL[5], Please refer to the following table.

TABLE 9-16: LCD CONTROL REGISTER (LCTL)

Address	Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default
$03A	**LCTL**	W	LPWR	BLANK	COMO	LENH	SEGO	-	-	DUTY	0000 0- -0

Bit 5: **COMO :** Common output selection bit
1 = COM4~COM7 used as output pins
0 = COM4~COM7 used as LCD Common pins

TABLE 9-17: SCAN OUTPUT REGISTER (SCAN)

Address	Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default
$03B	**SCAN**	R/W	SCAN[7]	SCAN[6]	SCAN[5]	SCAN[4]	-	-	-	-	0000 - - - -

Bit 4: **SCAN[4] :** COM4 scan output bit
1 = COM4 output =HIGH
0 = COM4 output =LOW

Bit 5: **SCAN[5] :** COM5 scan output bit
1 = COM5 output =HIGH
0 = COM5 output =LOW

Bit 6: **SCAN[6] :** COM6 scan output bit
1 = COM6 output =HIGH
0 = COM6 output =LOW

Bit 7: **SCAN[7] :** COM7 scan output bit
1 = COM7 output =HIGH
0 = COM7 output =LOW

10. Oscillator

ST2012 is with dual-clock system. Programmer can choose between OSC(RC) and OSCX(32.768k), or both as clock source through program. The system clock(SYSCK) also can be switched between OSC and OSCX. The OSC will be switch with "0" and OSCX will be switch with "1" for **XSEL**. Whenever system clock be switch, the warm-up cycles are occur at the same time. That is confirm SYSCK really switched when read **XSEL** bit. LCD driver, Timer1, Base Timer and PSG can utilize these two clock sources as well.

TABLE 10-18: SYSTEM CONTROL REGISTER (SYS)

Address	Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default
$030	**SYS**	R/W	XSEL	OSTP	XSTP	XBAK	WSKP	WAIT	-	-	0000 00- -

Bit 7: **XSEL :** System clock select(write) / confirm(read) bit
1 = OSCX
0 = OSC

Bit 6: **OSTP :** OSC stop control bit
1 = Disable OSC
0 = Enable OSC

Bit 5: **XSTP :** OSCX stop control bit
1 = Disable OSCX
0 = Enable OSCX

Bit 4: **XBAK :** OSCX driver heavy load bit
1 = OSCX normal load
0 = OSCX heavy load

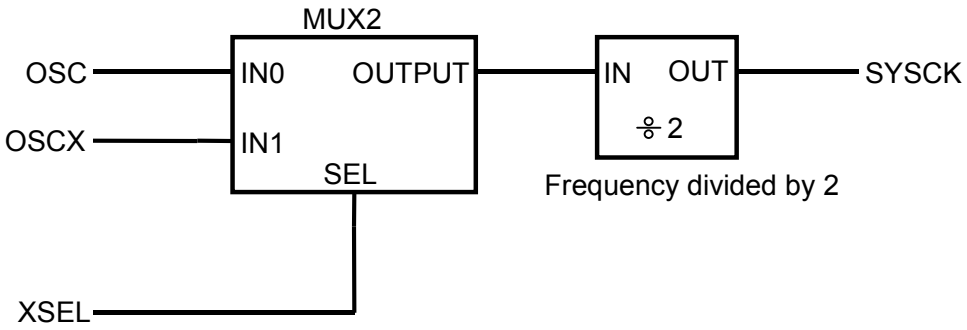
Bit 3: **WSKP :** System warm-up control bit
1 = Warm-up to 16 oscillation cycles
0 = Warm-up to 256 oscillation cycles

Bit 2: **WAIT :** WAI-0 / WAI-1mode select bit (Refer to POWER DOWN MODE)
1 = WAI instruction causes the chip to enter WAI-1 mode
0 = WAI instruction causes the chip to enter WAI-0 mode

Note:

1. The XSEL(SYS[7]) bit will show which real working mode is when it is read.

FIGURE 10-5: System Clock Diagram



11. TIMER/EVENT COUNTER

The ST2012 has two timers: Base timer/Timer1, and two prescalers (PRES and PREW). There are two clock sources for PRES and one clock source(OSCX) for PREW. Please refer to the following table:

TABLE 11-19: CLOCK SOURCE (TCLK) FOR PRES

SENT	Clock source(TCLK)	MODE
1	INTX	Event counter
0	SYSCK	Timer

TABLE 11-20: SUMMARY FOR TIMER REGISTERS

Address	Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default
$021	BTM	R/W	-	-	-	-	BTM[3]	BTM[2]	BTM[1]	BTM[0]	- - - - 0000
$023	PRS	R	PRS[7]	PRS[6]	PRS[5]	PRS[4]	PRS[3]	PRS[2]	PRS[1]	PRS[0]	0000 0000
		W	SRES	SENA	SENT	-	-	-	-	-	000 - - - - -
$026	T1M	R/W	-	-	-	T1M[4]	T1M[3]	T1M[2]	T1M[1]	T1M[0]	- - -0 0000
$027	T1C	R/W	T1C[7]	T1C[6]	T1C[5]	T1C[4]	T1C[3]	T1C[2]	T1C[1]	T1C[0]	0000 0000
$030	SYS	R/W	XSEL	OSTP	XSTP	XBAK	WSKP	WAIT	-	-	0000 00- -
$03C	IREQ	R/W	-	-	IRBT	IRPT	IRT1	-	IRDAC	IRX	- - 11 1-11
$03E	IENA	R/W	-	-	IEBT	IEPT	IET1	-	IEDAC	IEX	- - 00 0-00

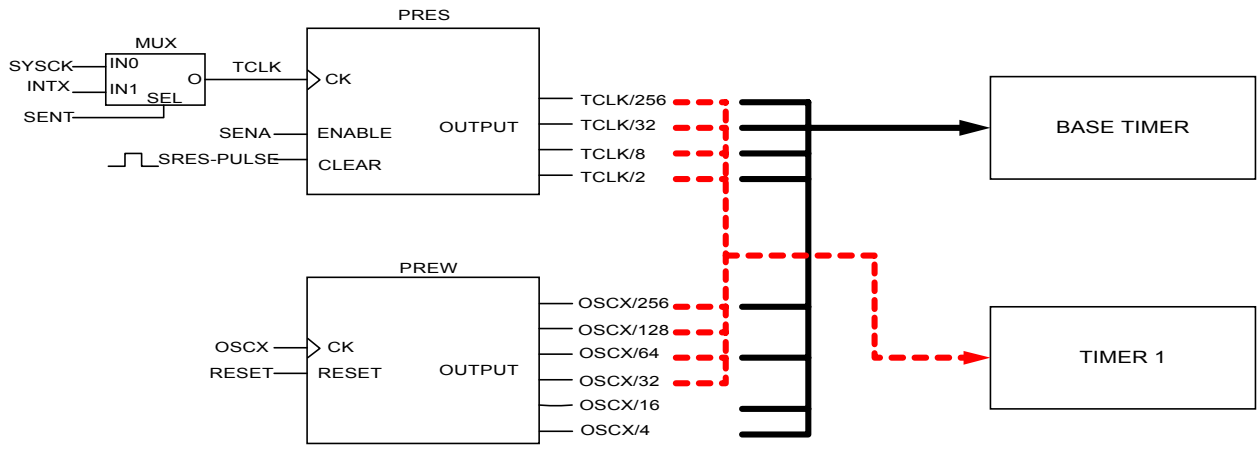


FIGURE 11-6: Prescaler for Timers

11.2 PRES

The prescaler PRES is an 8-bits counter as shown in Figure 11-6. Which provides four clock sources for base timer and timer1, and it is controlled by register PRS. The instruction read toward PRS will bring out the content of PRES and the instruction write toward PRS will reset, enable or select clock sources for PRES.
When user set external interrupt as the input of PRES for event counter, combining PRES and Timer1 will get a 16bit-event counter.

TABLE 11-21: PRESCALER CONTROL REGISTER (PRS)

Address	Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default
$023	PRS	R	PRS[7]	PRS[6]	PRS[5]	PRS[4]	PRS[3]	PRS[2]	PRS[1]	PRS[0]	0000 0000
		W	SRES	SENA	SENT	-	-	-	-	-	000 - - - -

READ
Bit 7~0: **PRS[7~0] :** 1's complement of PRES counter

WRITE
Bit 7: **SRES :** Prescaler Reset bit
Write "1" to reset the prescaler (PRS[7~0])

Bit 6: **SENA :** Prescaler enable bit
0 = Disable prescaler counting
1 = Enable prescaler counting

Bit 5: **SENT :** Clock source(TCLK) selection for prescaller PRES
0 = Clock source from system clock "SYSCK"
1 = Clock source from external events "INTX"

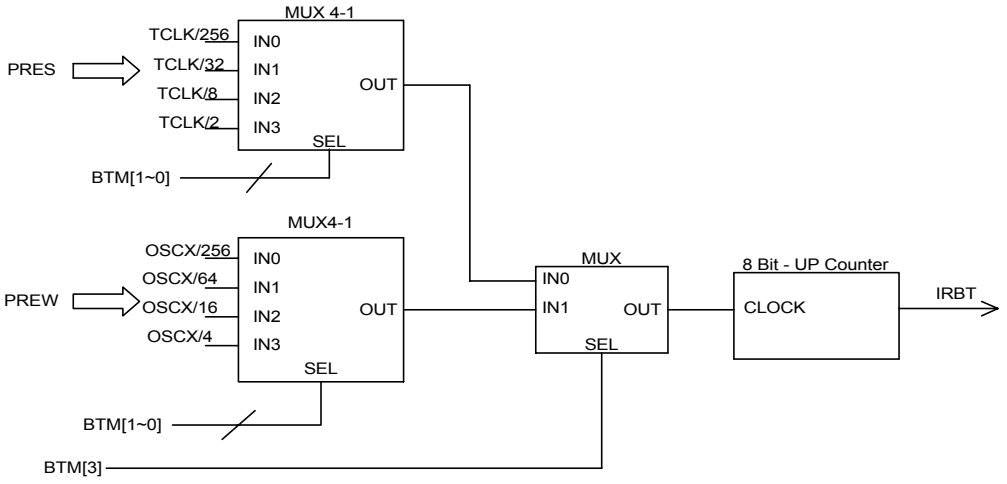
11.3 PREW

The prescaler PREW is an 8-bits counter as shown in Figure 11-6. PREW provides four clock source for base timer and timer1. It stops counting only if OSCX stops or hardware reset occurs.

11.4 Base timer

Base timer is an 8-bit up counting timer. When it overflows from $FF to $00, a timer interrupt request IRBT will be generated. Please refer to Figure 11-7. :

FIGURE 11-7: Structure of Base Timer



11.4.2 Clock source control for Base Timer

Several clock sources can be selected for Base Timer. Please refer to the following table:

TABLE 11-22: CLOCK SOURCE FOR BASE TIMER

* SENA	BTM[3]	BTM[2]	BTM[1]	BTM[0]	Base Timer source clock
0	0	X	X	X	STOP
1	0	X	0	0	TCLK / 256
1	0	X	0	1	TCLK / 32
1	0	X	1	0	TCLK / 8
1	0	X	1	1	TCLK / 2
X	1	X	0	0	OSCX / 256
X	1	X	0	1	OSCX / 64
X	1	X	1	0	OSCX / 16
X	1	X	1	1	OSCX / 4

* TCLK will stop when an ‘0’ is written to SENA(PRS[6]).

11.5 Timer 1

The Timer1 is an 8-bit up counter. It can be used as a timer or an event counter. T1C($27) is a real time read/write counter. When an overflow from $FF to $00, a timer interrupt request IRT1 will be generated. Timer1 will stop counting when system clock stops. Please refer to Figure 11-8.

FIGURE 11-8: Timer1 Structure Diagram

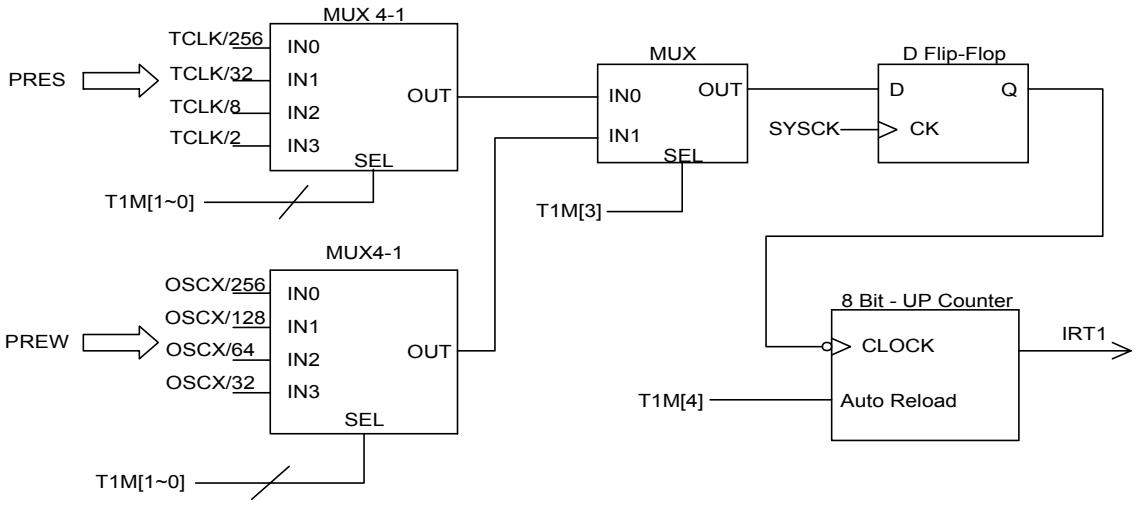


TABLE 11-23: TIMER1 COUNTING REGISTER (T1C)

Address	Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default
$027	**T1C**	R/W	T1C[7]	T1C[6]	T1C[5]	T1C[4]	T1C[3]	T1C[2]	T1C[1]	T1C[0]	0000 0000

11.5.1.2
Bit 7-0: **T1C[7-0] :** Timer1 up counter register

11.5.2 Clock source control for Timer1

Several clock source can be chosen from for Timer1. It's very important that Timer1 can keep counting as long as SYSCK stays active. Refer to the following table:

TABLE 11-24: CLOCK SOURCE FOR TIMER1

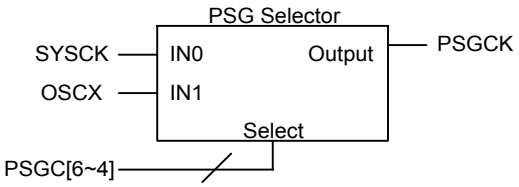
* SENA	T1M[4]	T1M[3]	T1M[2]	T1M[1]	T1M[0]	Clock source	Auto-Reload
0	X	0	X	X	X	STOP	-
1	0	0	X	0	0	TCLK / 256	No
1	0	0	X	0	1	TCLK / 32	No
1	0	0	X	1	0	TCLK / 8	No
1	0	0	X	1	1	TCLK / 2	No
X	0	1	X	0	0	OSCX / 256	No
X	0	1	X	0	1	OSCX / 128	No
X	0	1	X	1	0	OSCX / 64	No
X	0	1	X	1	1	OSCX / 32	No
1	1	0	X	0	0	TCLK / 256	Yes
1	1	0	X	0	1	TCLK / 32	Yes
1	1	0	X	1	0	TCLK / 8	Yes
1	1	0	X	1	1	TCLK / 2	Yes
X	1	1	X	0	0	OSCX / 256	Yes
X	1	1	X	0	1	OSCX / 128	Yes
X	1	1	X	1	0	OSCX / 64	Yes
X	1	1	X	1	1	OSCX / 32	Yes

* TCLK would stop when SENA is set to 0.

12. PSG

The built-in Programmable Sound Generator (PSG) is controlled by registers directly. Its flexibility through setting several parameters to registers makes it useful in many applications, such as music synthesis, sound effects generation, audible alarms and tone generation. PSG will finish the reset when user needs to create sound effect. The structure of PSG is shown in Figure 12-10 and its clock sources are shown in Figure 12-9. There are two sound types for PSG; tone and noise.

FIGURE 12-9: Clock Source for PSG



PSGC			PSGCK
B6	B5	B4	
0	0	0	SYSCK/2
X	0	1	SYSCK/4
X	1	0	SYSCK/8
0	1	1	SYSCK/16
1	0	0	SYSCK
1	1	1	OSCX

FIGURE 12-10: Program Sound Generator

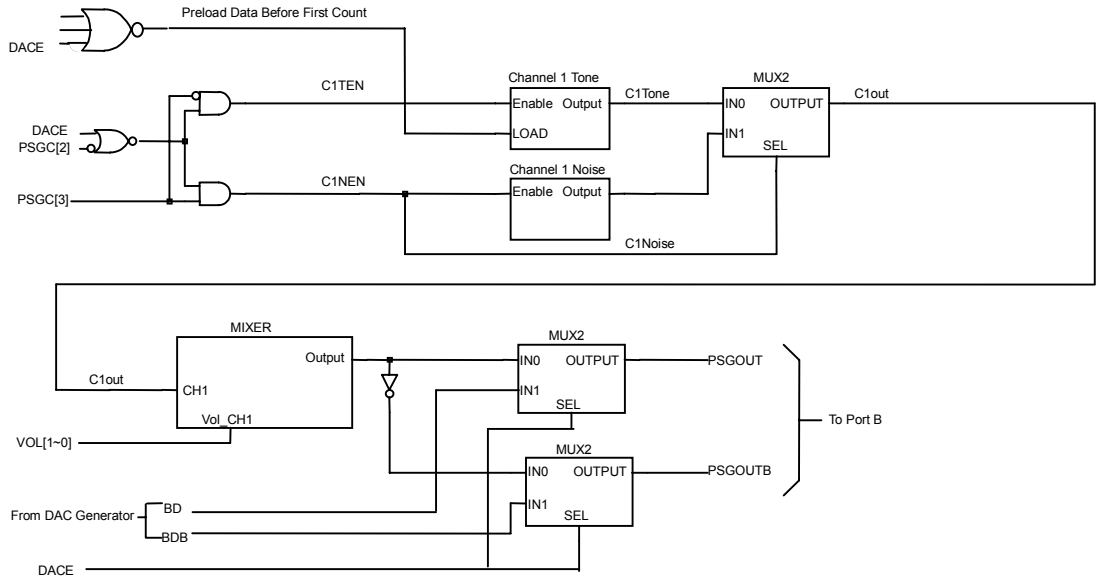


TABLE 12-25: SUMMARY FOR PSG REGISTERS

Address	Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default
$00F	**PMCR**	R/W	PULL	PDBN	INTEG	-	-	-	PSGO	PSGB	100 - - -00
$012	**PSGL**	R/W	PSG[7]	PSG[6]	PSG[5]	PSG[4]	PSG[3]	PSG[2]	PSG[1]	PSG[0]	0000 0000
$013	**PSGH**	R/W	-	-	-	-	PSG[11]	PSG[10]	PSG[9]	PSG[8]	- - - - 0000
$016	**PSGC**	R/W	-	PCK[2]	PCK[1]	PCK[0]	PRBS	C1EN	-	DACE=0	- 000 00-0
		R/W	-	PCK[2]	PCK[1]	PCK[0]	DMD[1]	DMD[0]	INH	DACE=1	- 000 0000
$017	**VOL**	R/W	VOL[1]	VOL[0]	-	-	-	-	-	-	00 - - - - - -

TABLE 12-26: CONTROL REGISTER FOR PSG OUTPUT (PMCR)

Address	Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default
$00F	**PMCR**	R/W	PULL	PDBN	INTEG	-	-	-	PSGO	PSGB	100 - - -00

Bit 1: **PSGO :** PSG output enable bit
1 = PSG data output pin if PB1 is set in output mode
0 = PB1 is normal I/O pin

Bit 0: **PSGB :** PSG inverse signal output enable bit
1 = PB0 is PSG inverse data output pin if PB0 is set in output mode
0 = PB0 is normal I/O pin

TABLE 12-27: CONTROL REGISTER FOR PSG VOLUME (VOL)

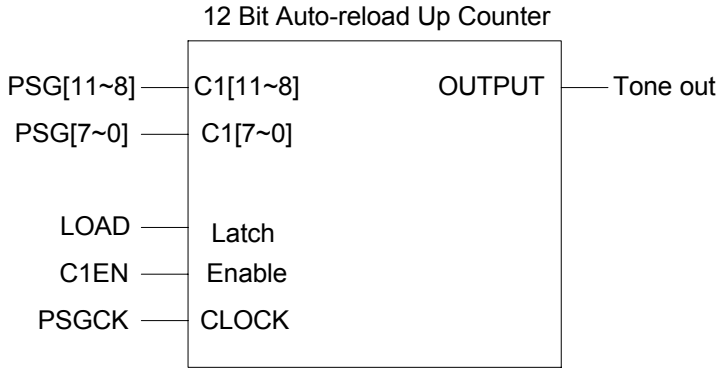
Address	Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default
$017	**VOL**	R/W	VOL[1]	VOL[0]	-	-	-	-	-	-	00 - - - - - -

Bit 7~6: **VOL[1~0] :** PSG volume control bit
00 = No sound output
01 = 1/4 volume (PSGCK must >= 128K Hz)
10 = 1/2 volume (PSGCK must >= 64K Hz)
11 = Maximum volume (PSGCK must >= 32K Hz)

12.2 Tone Generator

The tone frequency is decided by PSGCK and 12-bit programmable divider (PSG[11~0]) Please refer to Figure 12-11.

FIGURE 12-11: PSG Tone Counter



Tone Frequency = PSGCK/(1000H-PSG[11~0])/2

12.3 PSG Tone programming

To program tone generator, PSGO (PMCR[1]) or PSGB (PMCR[0]) should be set to "1" for PB1 or PB0 in order to be in the PSG output mode. Tone or DAC function is defined by DACE, writing to C1EN will enable tone generator when PSG is in tone function. Noise or tone function is selected by PRBS.

TABLE 12-28: PSG CONTROL REGISTER (PSGC)

Address	Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default
$016	PSGC	R/W	-	PCK[2]	PCK[1]	PCK[0]	PRBS	C1EN	-	DACE=0	- 000 00-0
		R/W	-	PCK[2]	PCK[1]	PCK[0]	DMD[1]	DMD[0]	INH	DACE=1	- 000 0000

Bit 0: **DACE :** Tone(Noise) or DAC Generator selection bit
1 = PSG is used as the DAC generator
0 = PSG is used as the Tone(Noise) generator

Bit 2: **C1EN :** PSG (Tone or Noise) enable bit
1 = PSG (Tone or Noise) enable
0 = PSG (Tone or Noise) disable

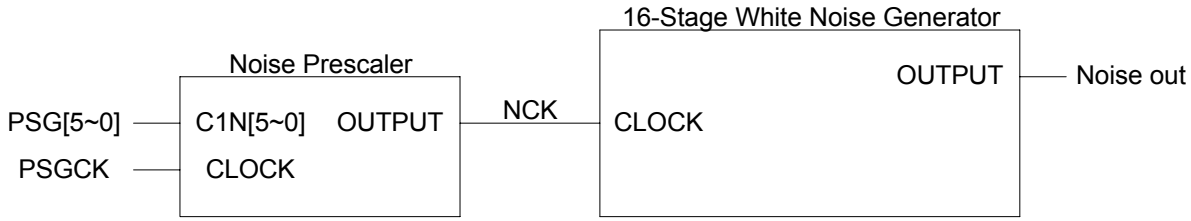
Bit 3: **PRBS :** Tone or Noise generator selection bit
1 = Noise generator
0 = Tone generator

Bit 6~4: **PCK[2~0] :** clock source(PSGCK) selection for PSG and DAC
000 = SYSCK / 2
X01 = SYSCK / 4
X10 = SYSCK / 8
011 = SYSCK / 16
100 = SYSCK
111 = OSCX

12.4 Noise Generator Control

Noise generator is shown in Figure 12-12., which base frequency is controlled by PSGL[5~0].

FIGURE 12-12: Noise Generator Diagram



NCK Frequency = PSGCK/(40H-PSG[5~0])

12.5 PSG Noise programming

To program noise generator, PSGO (PMCR[1]) or PSGB (PMCR[0]) should be set to “1” for PB1 or PB0 in order to be in PSG output. Noise or DAC function is defined by DACE.

Writing a “1” to C1EN will enable noise generator when PSG is in noise mode.

13. Digital DAC

A built-in digital DAC is for analog sampling data or voice signals. The structure of DAC is shown in Figure 13-13. There is an interrupt signal from DAC to CPU whenever DAC data update is needed and the same signal will decide the sampling rate of voice. In DAC mode, the OSC can't less 4 M Hz.

TABLE 13-29: SUMMARY FOR DAC REGISTERS

Address	Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default
$00F	PMCR	R/W	PULL	PDBN	INTEG	-	-	-	PSG0	PSGB	100 - - -00
$012	PSGL	R/W	PSG[7]	PSG[6]	PSG[5]	PSG[4]	PSG[3]	PSG[2]	PSG[1]	PSG[0]	0000 0000
$013	PSGH	R/W	-	-	-	-	PSG[11]	PSG[10]	PSG[9]	PSG[8]	- - - - 0000
$014	DAC	R/W	DAC[7]	DAC[6]	DAC[5]	DAC[4]	DAC[3]	DAC[2]	DAC[1]	DAC[0]	0000 0000
$016	PSGC	R/W	-	PCK[2]	PCK[1]	PCK[0]	PRBS	C1EN	-	DACE=0	- 000 00-0
		R/W	-	PCK[2]	PCK[1]	PCK[0]	DMD[1]	DMD[0]	INH	DACE=1	- 000 0000

TABLE 13-30: DAC DATA REGISTER (DAC)

Address	Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default
$014	DAC	R/W	DAC[7]	DAC[6]	DAC[5]	DAC[4]	DAC[3]	DAC[2]	DAC[1]	DAC[0]	0000 0000

Bit 7~0: **DAC[7~0] :** DAC output data

Note: For Single-Pin Single Ended mode, the effective output resolution is 7 bit.

TABLE 13-31: DAC CONTROL REGISTER (PSGC)

Address	Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default
$016	PSGC	R/W	-	PCK[2]	PCK[1]	PCK[0]	PRBS	C1EN	-	DACE=0	- 000 00-0
		R/W	-	PCK[2]	PCK[1]	PCK[0]	DMD[1]	DMD[0]	INH	DACE=1	- 000 0000

Bit 0: **DACE :** PSG play as Tone(Noise) or DAC Generator selection bit
1 = PSG is used as DAC Generator
0 = PSG is used as Tone(Noise) Generator

Bit 1: **INH :** DAC output inhibit control bit
1 = DAC output inhibit
0 = DAC output enable

Bit 3~2: **DMD[1~0] :** DAC output mode selection
00 = Single-Pin mode : 7 bit resolution
01 = Two-Pin Two Ended mode : 8 bit resolution
10 = Reserved
11 = Two-Pin Push Pull mode : 8 bit resolution

Bit 6~4: **PCK[2~0] :** PSGCK selection for PSG and DAC
000 = SYSCK / 2
X01 = SYSCK / 4
X10 = SYSCK / 8
011 = SYSCK / 16
100 = SYSCK *
111 = OSCX

*** In DAC mode, PSGCK must select SYSCK.**

13.2 Sampling Rate Control

The sample rate is controlled by PSGL and PSGH. PSG[11~7] controls sample rate/post scaling and PSG[6] must set '0' and PSG[5~0] must set '1'. The input clock source is controlled by PCK[2~0]. The block diagram is shown as the following:

FIGURE 13-13: DAC Generator Diagram

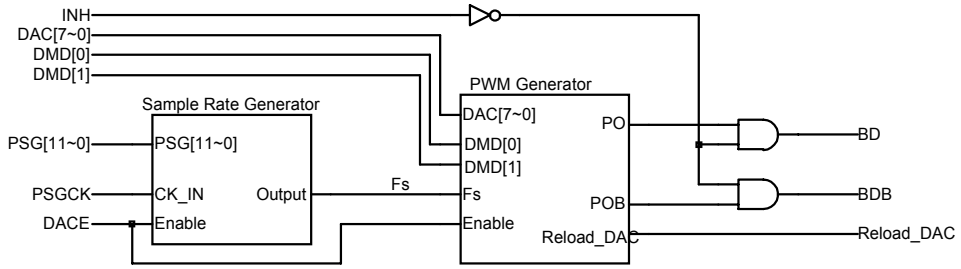
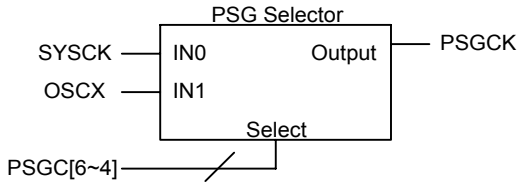


FIGURE 13-14: Clock Source for DAC



PSGC			PSGCK
B6	B5	B4	
0	0	0	SYSCK/2
X	0	1	SYSCK/4
X	1	0	SYSCK/8
0	1	1	SYSCK/16
1	0	0	SYSCK
1	1	1	OSCX

TABLE 13-32: Sample Rate description table

DAC SAMPLE RATE ALGORITHM DESCRIPTION
Sample-Rate = PSGCK / 128 / (20H-PSG[11~7])
Note: PSG[6] must set '0' and PSG[5~0] must set '1' by DAC mode.

13.3 PWM DAC Mode Select

The PWM DAC generator has three modes, Single-pin mode, Two-pin two ended mode and Two-pin push pull mode. They are depended on the application used. The DAC mode is controlled by DMD[1~0]. (TABLE 13-31)

13.3.1 Single-Pin Mode (Accurate to 7 bits)

Single-pin mode is designed for use with a single-transistor amplifier. It has 7 bits of resolution. The duty cycle of the **PB1** is proportional to the output value. If the output value is 0, the duty cycle is 50%. As the output value increases from 0 to 63, the duty cycle goes from being high 50% of the time up to 100% high. As the value goes from 0 to -64, the duty cycle decreases from 50% high to 0%. **PB0** is inverse of **PB1**'s waveform. Figure 13-15 shows the **PB1** wave-forms.

FIGURE 13-15: Single-Pin PWM DAC Wave-form

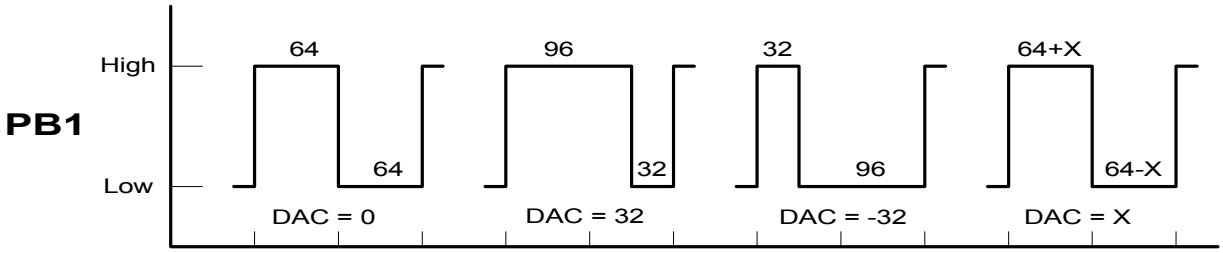
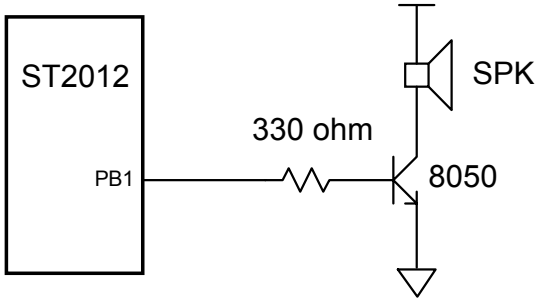


FIGURE 13-16: Single-Pin Application Circuit



13.3.2 Two-Pin Two Ended mode (Accurate to 8 bits)

Two-Pin Two Ended mode is designed for use with a single transistor amplifier. It requires two pins that **PB0** and **PB1**. When the DAC value is positive, **PB1** goes high with a duty cycle proportional to the output value, while **PB0** stays high. When the DAC value is negative, **PB0** goes low with a duty cycle proportional to the output value, while **PB1** stays low. This mode offers a resolution of 8 bits.

Figure 13-17 shows examples of DAC output waveforms with different output values. Each pulse of the DAC is divided into 128 segments per sample period. For a positive output value x=0 to 127, **PB1** goes high for X segments while **PB0** stays high. For a negative output value x=0 to -127, **PB0** goes low for |X| segments while **PB1** stays low.

FIGURE 13-17: Two-Pin Two Ended PWM DAC Wave-form

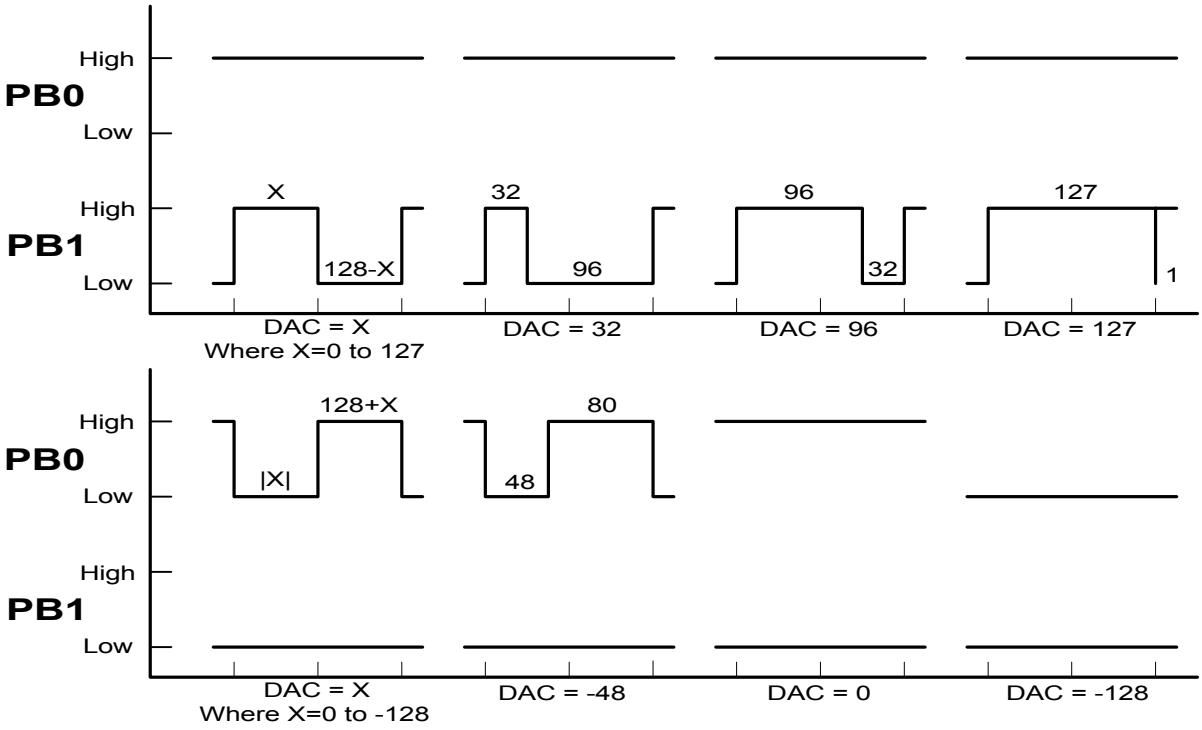
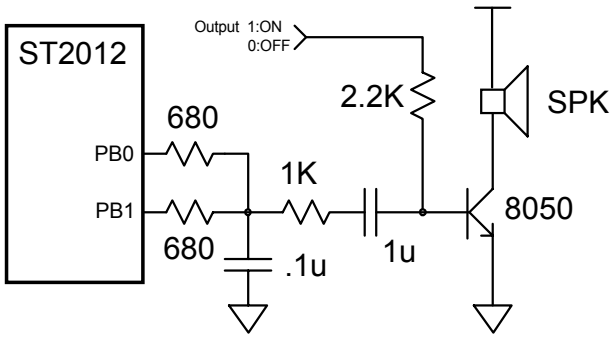


FIGURE 13-18: Two-Pin Two Ended mode Application Circuit



13.3.3 Two-Pin Push Pull mode (Accurate to 8 bits)

Two-Pin Push Pull mode is designed for buzzer. It requires two pins that **PB0** and **PB1**. When the DAC value is 0, both pins are low. When the DAC value is positive, **PB1** goes high with a duty cycle proportional to the output value, while **PB0** stays low. When the DAC value is negative, **PB0** goes high with a duty cycle proportional to the output value, while **PB1** stays low. This mode offers a resolution of 8 bits.

Figure 13-19 shows examples of DAC output waveforms with different output values. Each pulse of the DAC is divided into 128 segments per sample period. For a positive output value x=0 to 127, **PB1** goes high for X segments while **PB0** stays low. For a negative output value x=0 to -127, **PB0** goes high for |X| segments while **PB1** stays low.

FIGURE 13-19: Two-Pin Push Pull PWM DAC Wave-form

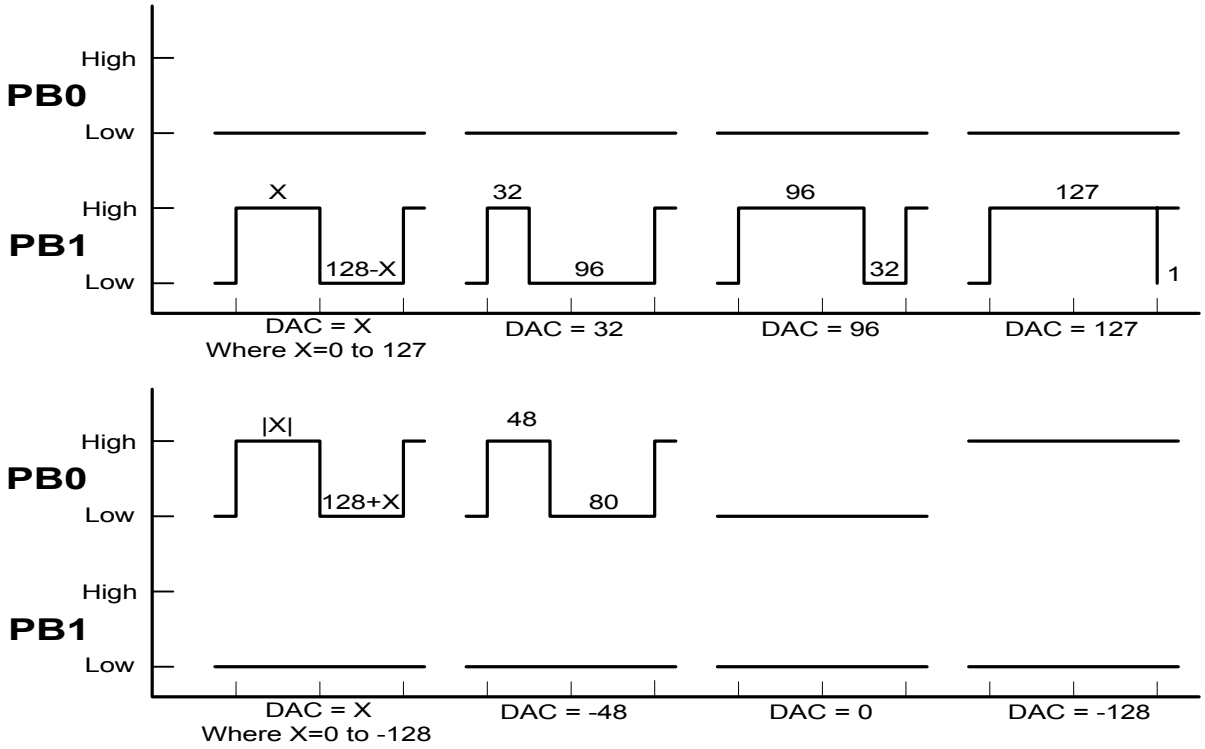
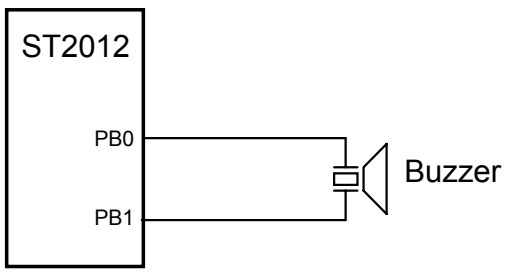


FIGURE 13-20: Two-Pin Push Pull Application Circuit

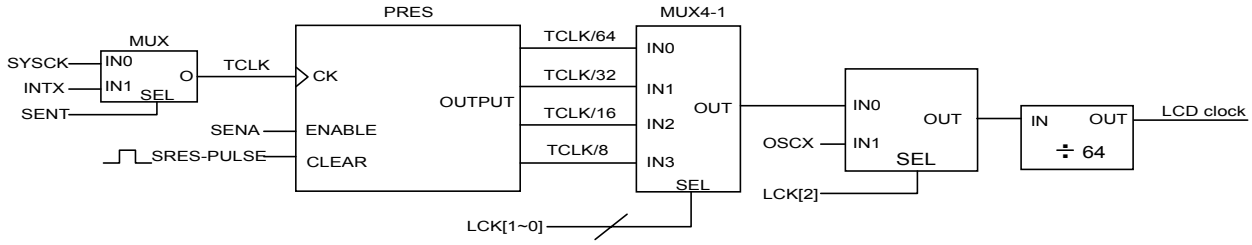


14. LCD

The ST2012 can drive up to 320 dots of LCD panel directly. The LCD driver can control by 1/4 duty(160 dots) and 1/8 duty (320 dots). LCD block include display RAM ($200~ $227) for storing the display data, 40-segment output pins (SEG0~SEG39), 8-common output pins (COM0~COM7).

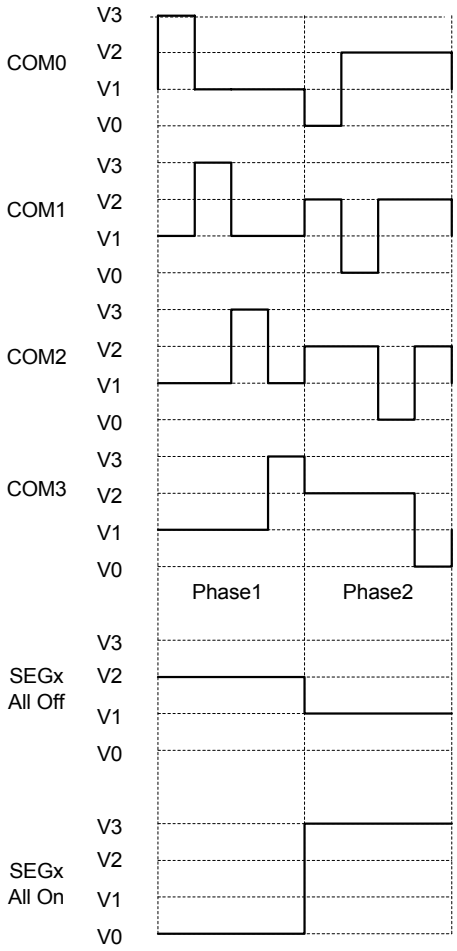
All LCD RAM are random after power on reset. The bias voltage circuits of the LCD display is built-in and no external resistor is needs.

FIGURE 14-21: Clock source of LCD

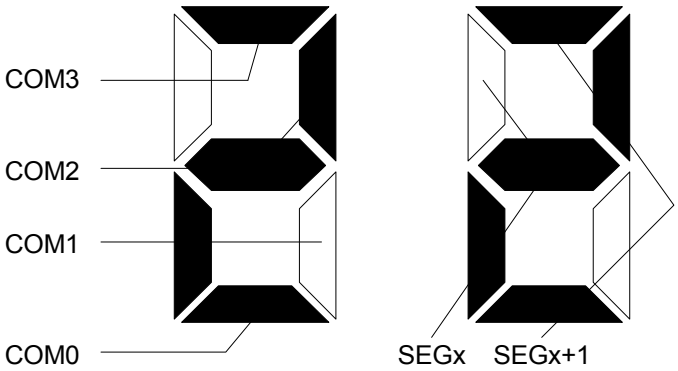


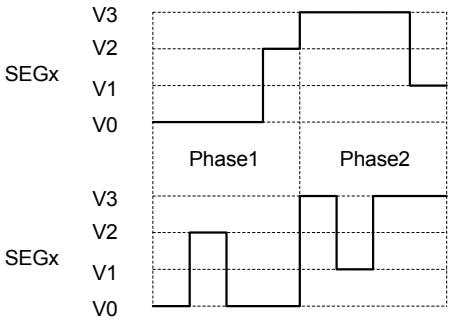
14.2 LCD driver 1/4 duty output

1/4 duty , 1/3 bias LCD signal



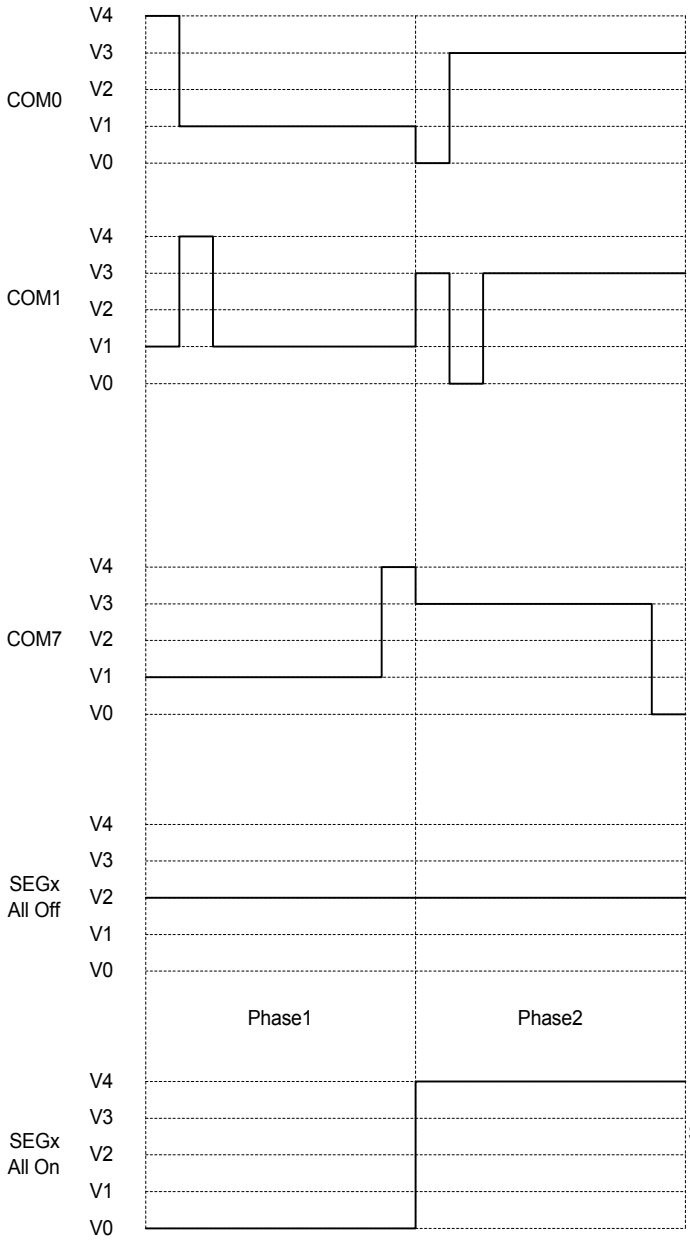
Example



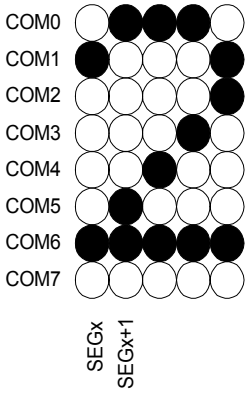


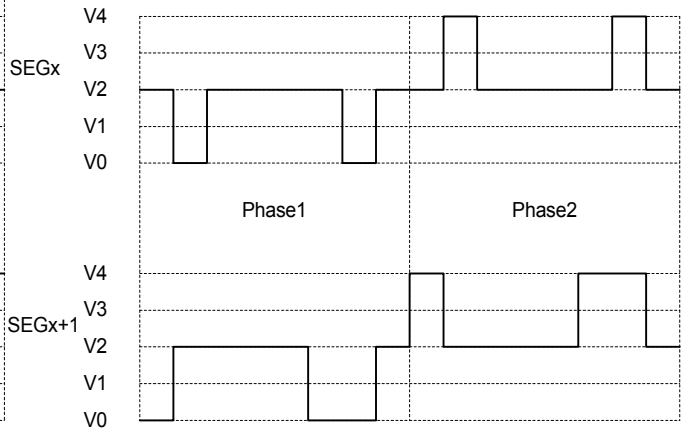
14.3 LCD driver 1/8 duty output

1/8 duty , 1/4 bias LCD signal



Example





14.4 LCD control register

TABLE 14-33: LCD CONTROL REGISTERS

Address	Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default
$020	**LCK**	R/W	-	-	-	-	-	LCK[2]	LCK[1]	LCK[0]	- - - - - 100
$023	**PRS**	R/W	SRES	SENA	SENT	-	-	-	-	-	000 - - - - -
$03A	**LCTL**	W	LPWR	BLANK	COMO	LENH	SEGO	-	-	DUTY	0000 0- -0

TABLE 14-34: LCD FREQUENCY REGISTER (LCK)

Address	Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default
$020	**LCK**	R/W	-	-	-	-	-	LCK[2]	LCK[1]	LCK[0]	- - - - - 100

Bit 2~0: **LCK[2~0] :** LCD clock source
000 = TCLK / 4096 (LCD frame clock = TCLK / 32768) *
001 = TCLK / 2048 (LCD frame clock = TCLK / 16384) *
010 = TCLK / 1024 (LCD frame clock = TCLK / 8192) *
011 = TCLK / 512 (LCD frame clock = TCLK / 4096) *
1XX = OSCX / 64 (LCD frame clock = 64)

*** SENA must switch "1". (refer to FIGURE 14-21)**

TABLE 14-35: LCD CONTROL REGISTER (LCTL)

Address	Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default
$03A	**LCTL**	W	LPWR	BLANK	COMO	LENH	SEGO	-	-	DUTY	0000 0- -0

Bit 7: **LPWR :** LCD power ON/OFF bit
1 = LCD power OFF
0 = LCD power ON

Bit 6: **BLANK :** LCD display ON/OFF bit
1 = Disable LCD display (Common line is still scanning)
0 = Enable LCD display

Bit 5: **COMO :** Output mode for LCD common
1 = COM7~COM4 will be general purpose output only pin
0 = All common output is used as LCD common driver.

Bit 4: **LENH :** heavy load control for LCD display
1 = Enhanced driving (For large size LCD panel with more power consumption)
0 = Normal driving

Bit 3: **SEGO :** mode control for LCD segment output
1 = SEG3~SEG0 will be general purpose output pin only
0 = All segment output is used as LCD segment driver

Bit 0: **DUTY :** LCD duty control bit
1 = 1/8 duty (1/4 bias) *
0 = 1/4 duty (1/3 bias)

* Under 1/8 duty condition with writing a "1" to COMO, LCD output pin COM7~COM4 will be controlled by the SCAN register. (Please refer to 9.5 Common port)

14.5 LCD RAM map

The LCD RAM map is shown as following:

TABLE 14-36: LCD RAM MAPPING

SEG	ADDRESS	COM0	COM1	COM2	COM3	COM4	COM5	COM6	COM7
0	200H	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
1	201H	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
2	202H	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
3	203H	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
4	204H	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
5	205H	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
6	206H	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
7	207H	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
8	208H	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
9	209H	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
10	20AH	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
11	20BH	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
12	20CH	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
13	20DH	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
14	20EH	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
15	20FH	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
16	210H	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
17	211H	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
18	212H	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
19	213H	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
20	214H	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
21	215H	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
22	216H	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
23	217H	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
24	218H	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
25	219H	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
26	21AH	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
27	21BH	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
28	21CH	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
29	21DH	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
30	21EH	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
31	21FH	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
32	220H	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
33	221H	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
34	222H	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
35	223H	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
36	224H	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
37	225H	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
38	226H	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
39	227H	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7

Note:

1. The LCD RAM address is allocated at page 2 of memory map. Only bit0 ~ bit3 is useful when it is 1/4 duty mode.
2. The LCD RAM can be **write & read** as like general purpose RAM.

15. Power Down Mode

The ST2012 has three power down modes: WAI-0, WAI-1 and STP. The instruction WAI will enable mode WAI-0 or WAI-1, which are controlled by WAIT(SYS[2]). The instruction WAI (WAI-0 and WAI-1 modes) can be wake-up by interrupt. However, the instruction of STP can only be wake-up by hardware reset.

15.1 WAI-0 Mode:

When **WAIT** is cleared, WAI instruction lets MCU enter WAI-0 mode. In the mean time, oscillator circuit is be active and interrupts, timer/counter, and PSG will all be working. Under such circumstance, CPU stops and the related instruction execution will stop. All registers, RAM, and I/O pins will retain their states before the MCU enter standby mode. WAI-0 mode can be wake-up by reset or interrupt request. If user disable interrupt(CPU register **I**='1'), MCU will still be wake-up but not go into the interrupt service routine. If interrupt is enabled(CPU register **I**='0'), the corresponding interrupt vector will be fetched and interrupt service routines will executed.
The sample program is showed as followed:

```
LDA #$00
STA SYS
WAI          ; WAI 0 mode
```

15.2 WAI-1 Mode:

When **WAIT** is set, WAI instruction let MCU to enter WAI-1 mode. In this mode, the CPU will stop, but PSG, timer/counter won't stop if the clock source is from OSCX. The wake-up procedure is the same as the one for WAI-0. But the warm-up cycles are occur when WAI-1 wake-up.
The sample program is shown as the following:

```
LDA #$04
STA SYS
WAI          ; WAI 1 mode
```

15.3 STP Mode:

STP instruction will force MCU to enter stop mode. In this mode, MCU stops, but PSG, timer/counter won't stop if the clock source is from OSCX. In power-down mode, MCU only be wake-up by hardware reset, and the warm-up cycles are occur at the same time.
The sample program is showed as the following:

```
.
.
STP
.
.
```

TABLE 15-37: STATUS UNDER POWER DOWN MODE

(SYSCK source from OSC)

Mode	Timer1	SYSCK	OSC	OSCX	Base Timer	RAM	REG.	LCD	I/O	Wake-up condition
WAI-0	Retain									Reset, Any interrupt
WAI-1	Stop	Stop	Stop	Retain						Reset, Any interrupt
STP	Stop	Stop	Stop	Retain						Reset

(SYSCK source from OSCX)

Mode	Timer1	SYSCK	OSC	OSCX	Base Timer	RAM	REG.	LCD	I/O	Wake-up condition
WAI-0	Retain									Reset, Any interrupt
WAI-1	Stop	Stop	Retain							Reset, Any interrupt
STP	Stop	Stop	Retain							Reset

16. ST2012 Electrical Characteristics

DC Supply Voltage ---------- -0.3V to +5.0V

Operating Ambient Temperature ---------- -10°C to +60°C

Storage Temperature ---------- -10°C to +125°C

***Notice:** Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. All the ranges 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. Exposed to the absolute maximum rating conditions for extended periods may affect device reliability.

16.1 DC Electrical Characteristics(TBD.)

Standard operation conditions: V_{DD} = 3.0V, GND = 0V, T_A = 25°C, OSC = 2M Hz, unless otherwise specified

Parameter	Symbol	Min.	Typ.	Max.	Unit	Condition
Operating Voltage	V_{DD}	2.4	3	3.4	V	
Operating Current	I_{OP}		904	994	μA	All output pins unload, execute NOP instruction, LCD on
Standby Current	I_{SB0}	-	0.6	-	μA	All output pins unload, OSCX off, LCD off (WAIT1/STOP mode)
Standby Current	I_{SB2}	-	2.8	-	μA	All output pins unload, OSCX on, LCD on (Normal) (WAIT1/STOP mode)
Standby Current	I_{SB3}	-	4.5	-	μA	All output pins unload, OSCX on, LCD on (Enhance) (WAIT1/STOP mode)
Standby Current	I_{SB4}	-	145	-	μA	All output pins unload, OSCX off, LCD on (WAIT0 mode)
LCD glass consumption	I_{LCD}	-	2	-	μA	Normal size LCD
Input High Voltage	V_{IH}	$0.7V_{DD}$	-	V_{DD} + 0.3	V	PORT A, PORT B
		$0.85V_{DD}$	-		V	$\overline{RESET}$, $\overline{INT}$
Input Low Voltage	V_{IL}	GND -0.3	-	$0.3V_{DD}$	V	PORT A, PORT B
			-	$0.15V_{DD}$	V	$\overline{RESET}$, $\overline{INT}$
Pull-up resistance	R_{OH}	60	80	100	KΩ	PORTA, PORTB (IOH = -37uA, VOH=0).
Output high voltage	V_{OH1}	0.7VDD	-		V	PORTA, PORTB (IOH = -3mA).
Output low voltage	V_{OL1}			0.8	V	PORTA, PORTB (IOL= 3mA).
Output high voltage	V_{OH2}	0.7 VDD			V	PSG, IOH = -5mA.
Output low voltage	V_{OL2}			0.8	V	PSG, IOL= 5mA.
Output high voltage	V_{OH3}	2.8			V	SEGx, Ioh = -800uA, C=50P,rise time < 200ns
Output low voltage	V_{OL3}			0.2	V	SEGx, Iol = 800uA
Output low voltage	V_{OL4}			0.8	V	SEG 0~3 to be output port, Iol = 2.5mA
Output low voltage	V_{OL5}			0.8	V	COM 0~3 to be output port, Iol = 2.5mA
Output high voltage	V_{OH6}	VDD-0.6			V	COMx, Ioh = -1 mA.
Output low voltage	V_{OL6}			0.8	V	COMx, Iol = 1 mA
Oscillation start time	T_{STT}	-	1	3	s	
Frequency stability	Δ F / F			1	PPM	[F(3.0)-F(2.5)]/F(3.0)(crystal oscillator)

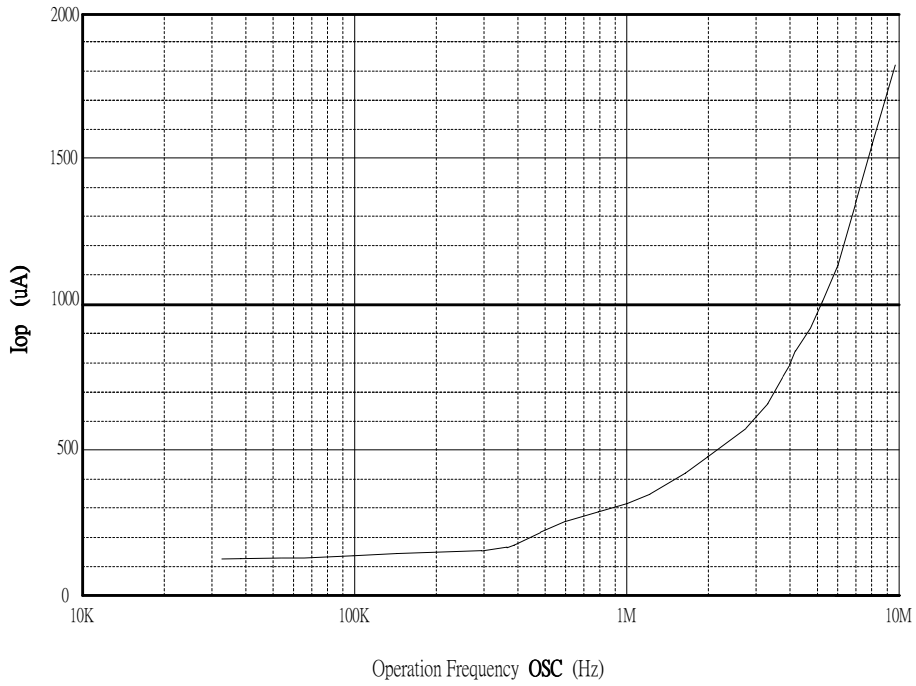
TABLE 16-38: R vs. OSC

Frequency variation	Δ F / F	-10	3	10	PPM	C1= 15 – 30P.

OSCI Resistance	OSC Frequency
51 K	4 MHz
120 K	2 MHz
260 K	1 MHz
470 K	600 KHz
820 K	350 KHz

Note: Average value for reference.

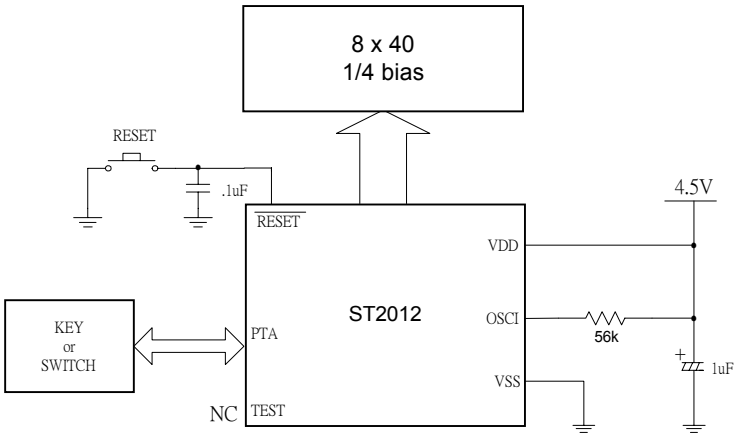
FIGURE 16-22: Typical Operation Current vs. OSC



17. Application Circuits

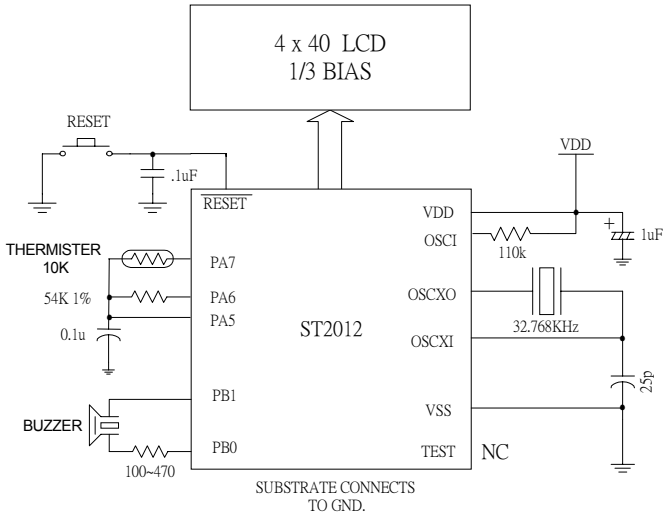
17.1 Application 1:

VDD : 4.5V
CLOCK : RC 4.0M
LCD : 1/8 duty, 1/4 bias;
I/O : PORT A



17.2 Application 2:

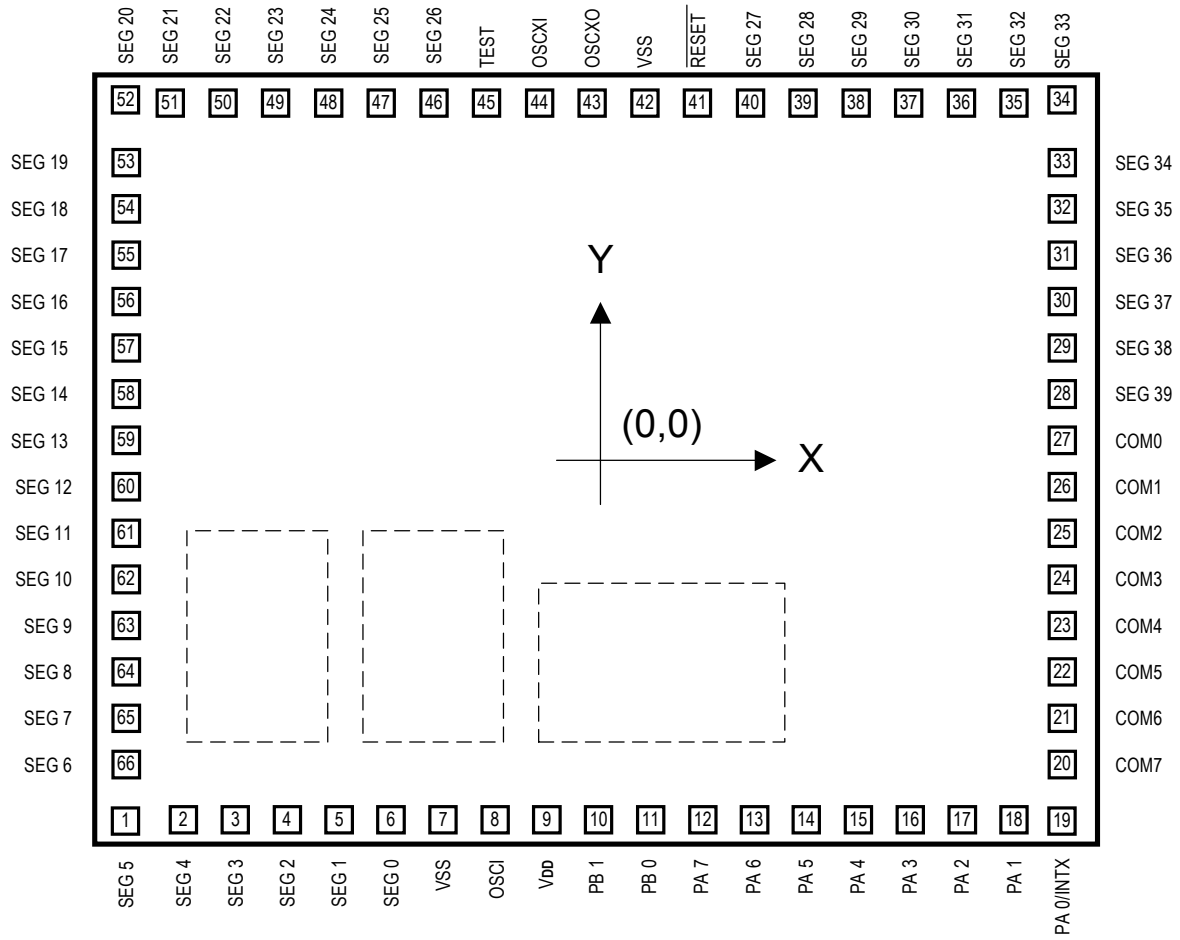
Clock : 32.768KHz crystal and 2.0M RC oscillator
LCD : 1/4 duty, 1/3 bias
I/O : PORT A
ALARM : PB.0, PB.1



18. Bonding Information

These are bonding diagram and bonding description.

18.1 Bonding Diagram:



Chip Size: 2400 x 2070 um

* The chip substrate should be connected with the VSS pin.

18.2 Bonding Description

Unit: um

PAD #	NAME	PAD CENTER	
		X	Y
1	SEG5	-1109.10	-948.30
2	SEG4	-961.90	-943.30
3	SEG3	-841.90	-943.30
4	SEG2	-721.90	-943.30
5	SEG1	-601.90	-943.30
6	SEG0	-481.90	-943.30
7	VSS	-361.90	-943.30
8	OSCI	-240.00	-943.30
9	VDD	-118.05	-943.30
10	PB1	1.95	-943.30
11	PB0	121.95	-943.30
12	PA7	241.95	-943.30
13	PA6	361.95	-943.30
14	PA5	481.95	-943.30
15	PA4	601.95	-943.30
16	PA3	721.95	-943.30
17	PA2	841.95	-943.30
18	PA1	961.95	-943.30
19	PA0/INTX	1109.10	-943.30
20	COM7	1109.10	-780.00
21	COM6	1109.10	-660.00
22	COM5	1109.10	-540.00
23	COM4	1109.10	-420.00
24	COM3	1109.10	-300.00
25	COM2	1109.10	-180.00
26	COM1	1109.10	-60.00
27	COM0	1109.10	60.00
28	SEG39	1109.10	180.00
29	SEG38	1109.10	300.00
30	SEG37	1109.10	420.00
31	SEG36	1109.10	540.00
32	SEG35	1109.10	660.00
33	SEG34	1109.10	780.00

PAD #	NAME	PAD CENTER	
		X	Y
34	SEG33	1109.10	948.30
35	SEG32	960.00	943.30
36	SEG31	840.00	943.30
37	SEG30	720.00	943.30
38	SEG29	600.00	943.30
39	SEG28	480.00	943.30
40	SEG27	360.00	943.30
41	/RESET	240.00	943.30
42	VSS	120.00	943.30
43	OSCXO	0.00	943.30
44	OSCXI	-120.00	943.30
45	TEST	-240.00	943.30
46	SEG26	-360.00	943.30
47	SEG25	-480.00	943.30
48	SEG24	-600.00	943.30
49	SEG23	-720.00	943.30
50	SEG22	-840.00	943.30
51	SEG21	-960.00	943.30
52	SEG20	-1109.10	948.30
53	SEG19	-1109.10	780.00
54	SEG18	-1109.10	660.00
55	SEG17	-1109.10	540.00
56	SEG16	-1109.10	420.00
57	SEG15	-1109.10	300.00
58	SEG14	-1109.10	180.00
59	SEG13	-1109.10	60.00
60	SEG12	-1109.10	-60.00
61	SEG11	-1109.10	-180.00
62	SEG10	-1109.10	-300.00
63	SEG9	-1109.10	-420.00
64	SEG8	-1109.10	-540.00
65	SEG7	-1109.10	-660.00
66	SEG6	-1109.10	-780.00

19. Appendix

19.1 ST2012B LCD heavy driving version

19.1.1 Function description

ST2012B is ST2012 family of LCD heavy driving version. It has more driving power than ST2012. It can support large size LCD for game application. The ST2012B has extra control register XLCD($38) that control LCD heavy driving mode.

TABLE 19-39: HEAVY DRIVING LCD MODE CONTROL

Address	Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default
$038	XLCD	W	-	-	-	-	-	-	HEAV	1	- - - - - -??

Bit 1: **HEAV :** LCD driving mode control
1 = Heavy driving mode
0 = Normal driving mode

Note: The bit-0 must set '1' for ST2012B.

19.1.2 LCD driving level control

The HEAV(XLCD[1]) also can be select 4 level LCD driving with LENH(LCTL[4]). The ST2012B power consumption reference table is shown as the following:

TABLE 19-40: ST2012B POWER CONSUMPTION REFERENCE

LCD driving level	HEAV	LENH	1/4 Duty consumption	1/8 Duty consumption
Normal	0	0	**14**	**15**
Middle	0	1	**17.3**	**20**
Big	1	0	**35.5**	**47.3**
Maximum	1	1	**114.4**	**86.9**

Condition: OSCX on, LCD on, WAI-1, All I/O no load.

20. <Revisions>

Version 1.30 -Page44 modify DC electrical characteristic talbe
-Page47 modify TABLE 19-40: ……2003/3/4
Version 1.20 -Page1 modify operation voltage.
-Page 8 increase note 4,note 5.
-Page 42 fix ST2012 **Electrical Characteristics.**
-Page 47 fix TABLE 19-40.
Version 0.44 - Modify all control register **'R/W'** define.
- Page 47 increase ST2012B appendix.
Version 0.43 - Page 2 modify **Block Diagram**.
- Change **$3C(IREQ)** power on default.
Version 0.42 - Page 43 increase Bonding information .
- Page 42 modify **Application1, Application 2**.
Version 0.41 - Page 38 fix TABLE 14-34.
- Page 41,42 update **Electrical Characteristics**.
Version 0.40 - Change **$3A(LCTL)** be write only register.
- Page 42 modify **Application1, Application 2**.
- Page 40 fix TABLE 15-37.
Version 0.39 - Page 1 RC oscillator 500k~4M Hz.
- Page 1 PSG increase 4 level volume description.
- Page 28 12.2 line4 spell 'writing' bug.
- Page 30 description of Digital DAC.
- Page 30 increase last line.
- Page 36 **COM1~COM4** fix to **COM0~COM3**.
- Page 38 last line **3.1.5** fix to **9.5**.
- Page 40 modify TABLE 15-37.
Version 0.38 - Page 1,21 description of 16-bit event counter.
- Modify LCD description.
- Page 14 Modify debounce time (16 ms).
Version 0.37 - Page 20 increase warm-up cycles description of SYSCK.
- Page 37 increase warm-up cycles description of WAI-1 & STP mode.
Version 0.36 - Change register $20 (LCK[2]) default '0' to '1'.
- Page 5,6,8 cancel '**BRK**' instruction.
- Page 14 description of debounce.
- Page 20 description of **XSEL** bit.
- Page 20 redraw Figure 10-5.
- Page 21 redraw Figure 11-6.
- Page 28 description of **PCK[2~0]**.
- Page 31 redraw Figure 13-13.
- Page 33 Increase note 2.
- Page 36 description of LCD frame clock.
- Page 39 redraw Application2.