

# ST2108

## 8 BIT Microcontroller with 1M bytes ROM

Notice: Sitronix Technology Corp. reserves the right to change the contents in this document without prior notice. This is not a final specification. Some parameters are subject to change.

### 1. FEATURES

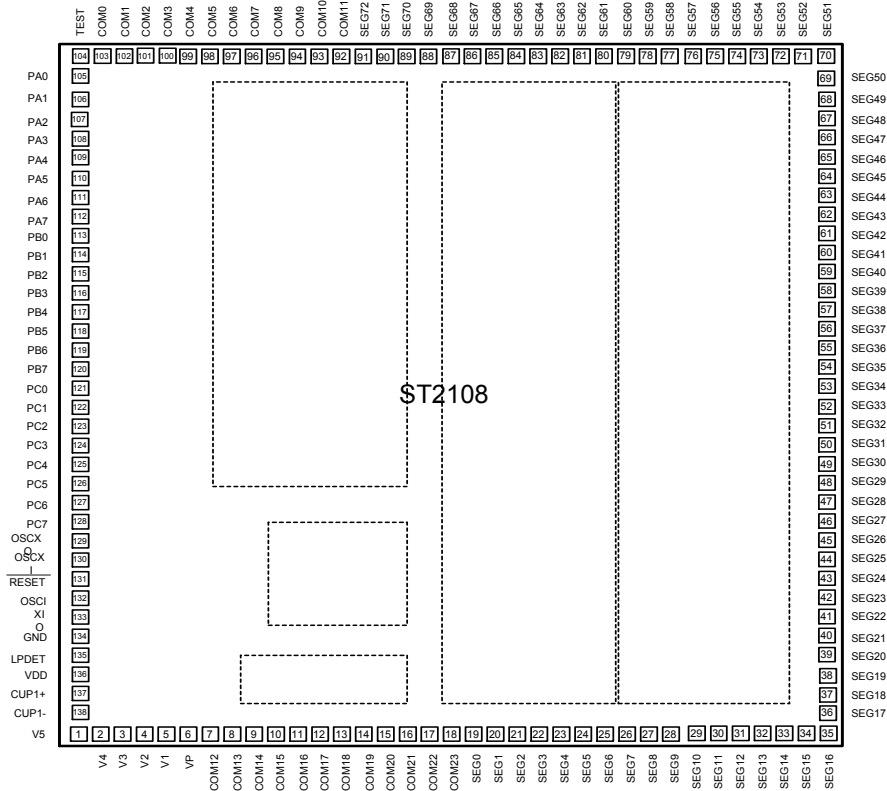
- Totally static RISC CPU
- ROM: 1M x 8-bit
- RAM: 4K x 8-bit
- Stack: Up to 128-level deep
- Operation voltage: 2.4V ~ 3.4V
- I/O ports
  - 24 CMOS bidirectional bit programmable I/O pins
  - 8 output pins (shared with LCD common output)
  - Bit programmable pull-up for input pins
  - Hardware de-bounce option for Port-A
- Low voltage detector
- Timer/Counter:
  - Two 8-bit timer/16-bit event counter
  - One 8-bit Base timer
- 6 Prioritized interrupt sources
  - External interrupt (edge triggered)
  - Timer0 interrupt
  - Timer1 interrupt
  - Base timer interrupt
  - Port-A[7~0] interrupt (transition triggered)
  - DAC reload interrupt
- Dual clock source with warm-up timer
  - Crystal oscillator .....32.768K Hz
  - RC oscillator ..... 500K~4M Hz
  - Resonator oscillator (code option) ..... 455K~4M Hz
- Direct memory access (DMA)
  - Block-to-Block move
  - Block to Single port
- LCD controller
  - 16-level contrast control
  - 1752 (73x24) dots (1/24 duty)
  - 1168 (73x16) dots (1/16 duty)
- Programmable sound generator (PSG)
  - Two channels with three playing modes
  - Tone/noise generator
  - 16-level volume control
- PWM DAC: Three modes up to 8-bit resolution
- Three power down modes:
  - WAI0 mode
  - WAI1 mode
  - STP mode

### 2. GENERAL DESCRIPTION

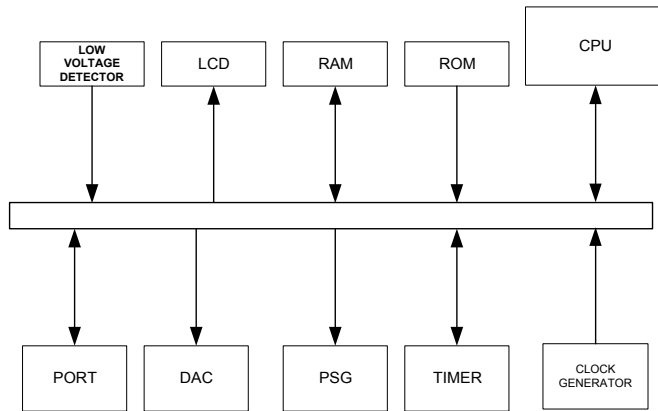
The ST2108 is a RISC CPU based 8-bit microcontroller designed with CMOS silicon gate technology. This single chip microcontroller is useful for translator, databank and other consumer applications. It integrates with SRAM, mask program

ROM, LCD controller/drivers, I/O ports, timers, PSG and PWM DAC. This chip also builds in dual oscillators for the chip performance enhancement.

### 3. PAD DIAGRAM



### 4. BLOCK DIAGRAM



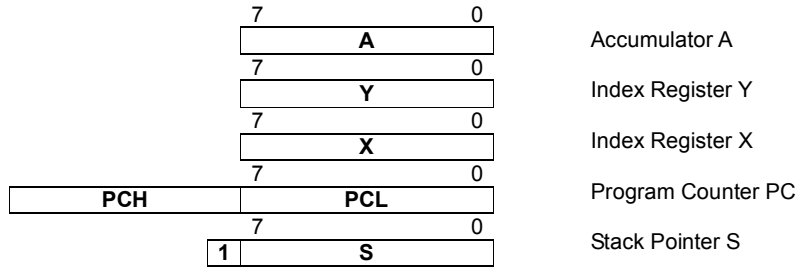
## 5. PAD DESCRIPTION

Pin No.	Designation	I/O	Description
19~91	SEG 0~72	O	LCD segment output
103~92, 7~18	COM 0~23	O	LCD common output
105	PA0 / INTX	I/O I I I	Port-A bit programmable I/O Edge-trigger Interrupt. Transition-trigger Interrupt Programmable Timer1 clock source
106~112	PA 1~7	I/O I	Port-A bit programmable I/O Transition-trigger Interrupt
113,114	PB 0, 1	I/O O	Port-B bit programmable I/O PSG/DAC Output
115~120	PB 2~7	I/O	Port-B bit programmable I/O
121~128	PC 0~7	I/O	Port-C bit programmable I/O
131	$\overline{\text{RESET}}$	I	Pad reset input (low active)
134	GND	P	Ground Input and chip substrate
136	VDD	P	Power supply
129, 130	OSC XO, OSC XI	I/O	OSC I/O pin. For 32768Hz crystal used.
132	OSCI	I	RC oscillator pin, had to be connected to external resistor
133	XIO	I	Resonator input pin
135	NC		
6	VP	O	LCD pumping voltage output
137	CUP1+	I/O	Pump capacitance 1 positive edge
138	CUP1-	I/O	Pump capacitance 1 negative edge
5~1	V1~V5	I	External LCD voltage supply
104	TEST	I	Test pin for chip test, normal to NC.

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

## 6. CPU

### Register Model



#### Accumulator (A)

The Accumulator is a general-purpose 8-bit register that 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.

#### 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 an 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.

#### Stack Pointer (S)

The Stack Pointer is an 8-bit register, which is used to control the addressing of the variable-length stack. Its range from 100H to 1FFH total for 256 bytes (128 level deep). The stack pointer is automatically incremented and decremented under control of the microprocessor to perform stack manipulations under

Accumulator A

Index Register Y

Index Register X

Program Counter PC

Stack Pointer S

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.

#### Program Counter (PC)

The 16-bit Program Counter register provides the address, which steps the microprocessor through sequential program instructions. Each time the microprocessor fetches an 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.

#### Status Register (P)

The 8-bit Processor Status Register contains seven status flags. Some of these flags are controlled by program; others may be controlled both by the program and the CPU. The instruction set contains a member of conditional branch instructions that are designed to allow testing of these flags. Refer to TABLE 6-1

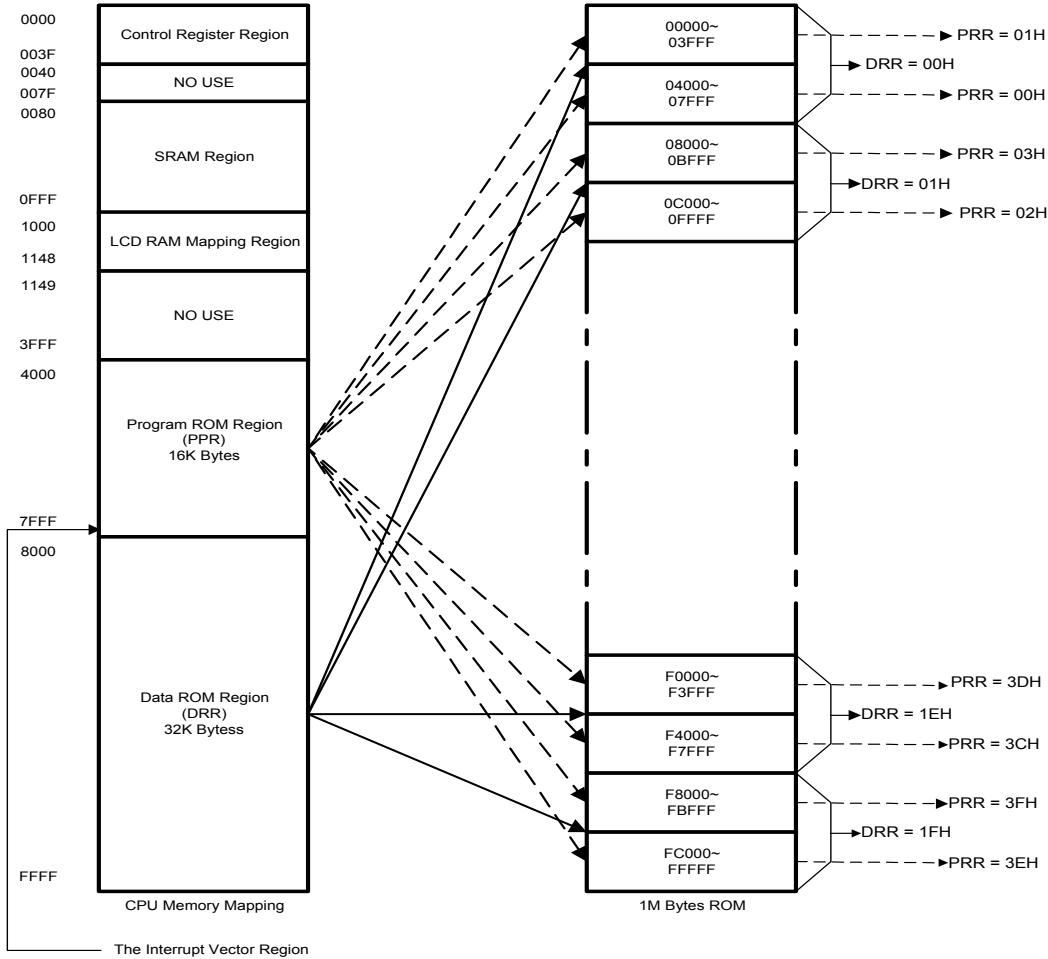
TABLE 6-1 Status register (P)

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: <b>N</b> : Signed flag by arithmetic 1 = Negative 0 = Positive				Bit 3: <b>D</b> : Decimal mode flag 1 = Decimal mode 0 = Binary mode			
Bit 6: <b>V</b> : Overflow of signed Arithmetic flag 1 = Negative 0 = Positive				Bit 2: <b>I</b> : Interrupt disable flag 1 = Interrupt disable 0 = Interrupt enable			
				Bit 1: <b>Z</b> : Zero flag 1 = Zero 0 = Non zero			
Bit 4: <b>B</b> : BRK interrupt flag 1 = BRK interrupt occur 0 = Non BRK interrupt occur				Bit 0: <b>C</b> : Carry flag 1 = Carry 0 = Non carry			

## 7. MEMORY CONFIGURATION

### 7.1 Memory map

ST2108 has total 1M bytes ROM and 4K RAM inside. This ROM can be used as data memory or program memory. PRR is the Program ROM Bank Pointer Register and DRR is the Data ROM Bank Pointer Register. The data ROM address area in ST2108 is from \$8000 to \$FFFF (32K bytes) and program ROM address is from \$4000 to \$7FFF(16K bytes).



## 7.2 ROM

### 7.2.1 Bank Description

Setting corresponding value to register PRR(program memory) or DRR(data memory) when user wants uses different memory bank.

FIGURE 7-1 ROM Control Registers (\$31~\$32)

Address	Register	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PRR	\$31	RW	PRR7	PRR6	PRR5	PRR4	PRR3	PRR2	PRR1	PRR0
DRR	\$32	RW	DRR7	DRR6	DRR5	DRR4	DRR3	DRR2	DRR1	DRR0

## 7.3 RAM

The RAM mapping includes control registers, data RAM and stack RAM.

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[7]	PB[6]	PB[5]	PB[4]	PB[3]	PB[2]	PB[1]	PB[0]	1111 1111
\$002	PC	R/W	PC[7]	PC[6]	PC[5]	PC[4]	PC[3]	PC[2]	PC[1]	PC[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
\$009	PCB	R/W	PCB[7]	PCB[6]	PCB[5]	PCB[4]	PCB[3]	PCB[2]	PCB[1]	PCB[0]	0000 0000
\$00A	PCC	R/W	PCC[7]	PCC[6]	PCC[5]	PCC[4]	PCC[3]	PCC[2]	PCC[1]	PCC[0]	0000 0000
\$00F	PMCR	R/W	PULL	PDBN	INTEG	-	-	-	PSGO	PSGB	100 - - -00
\$010	PSG0L	W	PSG0[7]	PSG0[6]	PSG0[5]	PSG0[4]	PSG0[3]	PSG0[2]	PSG0[1]	PSG0[0]	0000 0000
\$011	PSG0H	W	-	-	-	-	PSG0[11]	PSG0[10]	PSG0[9]	PSG0[8]	- - - - 0000
\$012	PSG1L	W	PSG1[7]	PSG1[6]	PSG1[5]	PSG1[4]	PSG1[3]	PSG1[2]	PSG1[1]	PSG1[0]	0000 0000
\$013	PSG1H	W	-	-	-	-	PSG1[11]	PSG1[10]	PSG1[9]	PSG1[8]	- - - - 0000
\$014	DAC	W	DAC[7]	DAC[6]	DAC[5]	DAC[4]	DAC[3]	DAC[2]	DAC[1]	DAC[0]	0000 0000
\$016	PSGC	W	-	PCK[2]	PCK[1]	PCK[0]	PRBS	C1EN	C0EN	DACE=0	-000 0000
		W	-	PCK[2]	PCK[1]	PCK[0]	DMD[1]	DMD[0]	INH	DACE=1	-000 0000
\$017	VOL	W	VOL1[3]	VOL1[2]	VOL1[1]	VOL1[0]	VOL0[3]	VOL0[2]	VOL0[1]	VOL0[0]	0000 0000
\$021	BTM	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	-	-	-	-	-	-
\$024	T0M	R/W	-	-	T0M[5]	T0M[4]	-	T0M[2]	T0M[1]	T0M[0]	- -00 -000
\$025	T0C	R/W	T0C[7]	T0C[6]	T0C[5]	T0C[4]	T0C[3]	T0C[2]	T0C[1]	T0C[0]	0000 0000
\$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
\$028	DMSL	W	DMS[7]	DMS[6]	DMS[5]	DMS[4]	DMS[3]	DMS[2]	DMS[1]	DMS[0]	0000 0000
\$029	DMSH	W	DMS[15]	DMS[14]	DMS[13]	DMS[12]	DMS[11]	DMS[10]	DMS[9]	DMS[8]	0000 0000
\$02A	DMDL	W	DMD[7]	DMD[6]	DMD[5]	DMD[4]	DMD[3]	DMD[2]	DMD[1]	DMD[0]	0000 0000
\$02B	DMDH	W	DMD[15]	DMD[14]	DMD[13]	DMD[12]	DMD[11]	DMD[10]	DMD[9]	DMD[8]	0000 0000
\$02C	DCNTL	W	DCNT[7]	DCNT[6]	DCNT[5]	DCNT[4]	DCNT[3]	DCNT[2]	DCNT[1]	DCNT[0]	0000 0000
\$02D	DCNTH	W	-	-	-	DFIX	DCNT[11]	DCNT[10]	DCNT[9]	DCNT[8]	- - - 0 0000
\$030	SYS	R/W	XSEL	OSTP	XSTP	XBAK	WSKP	WAIT	-	LVDET	0000 00 -0
\$031	PRR	R/W	PRR[7]	PRR[6]	PRR[5]	PRR[4]	PRR[3]	PRR[2]	PRR[1]	PRR[0]	0000 0000
\$032	DRR	R/W	DRR[7]	DRR[6]	DRR[5]	DRR[4]	DRR[3]	DRR[2]	DRR[1]	DRR[0]	0000 0000
\$033	DMR	R/W	DMR[7]	DMR[6]	DMR[5]	DMR[4]	DMR[3]	DMR[2]	DMR[1]	DMR[0]	0000 0000
\$039	SCAN	W	SCAN[7]	SCAN[6]	SCAN[5]	SCAN[4]	SCAN[3]	SCAN[2]	SCAN[1]	SCAN[0]	0000 0000
\$03A	LCTL	W	LPWR	BLANK	REV	DUTY	CTR[3]	CTR[2]	CTR[1]	CTR[0]	0000 0000
\$03B	LCK	W	-	-	-	-	-	LCK[2]	LCK[1]	LCK[0]	- - - - -000
\$03C	IREQ	R/W	-	-	IRBT	IRPT	IRT1	IRT0	IRDAC	IRX	- - 00 0000
\$03E	IENA	R/W	-	-	IEBT	IEPT	IET1	IET0	IEDAC	IEX	- - 00 0000

- Note: 1. Some addresses of I/O area, \$3~\$7, \$B~\$E, \$15, \$18~\$20, \$22, \$2E~\$2F, \$34~\$38, \$3D, \$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. You have to initial the PRR, DRR and DMR registers when system reset.

### 7.3.1 DATA RAM (\$0080~\$0FFF)

DATA RAM are organized in 4K bytes from \$0080~\$0FFF.

### 7.3.2 STACK RAM (\$0100~\$01FF)

STACK RAM is organized in 256 bytes. It provides for a maximum of 128-level subroutine stacks and can be used as data memory.

### 7.3.3 LCD RAM (\$1000~\$1148)

Resident LCD-RAM, accessible through write and read instructions, for 24\*73 and 16\*73 LCD displays. Note that this area can also be used as data memory. Refer to section 14.2 about the detail usage.

## 8. INTERRUPTS

### 8.1 Interrupt description

#### Brk

Instruction 'BRK' will cause software interrupt when interrupt disable flag (I) is cleared. Hardware will push 'PC', 'P' Register to stack and set interrupt disable flag (I). Program counter then will be loaded with the BRK vector from locations \$7FFE and \$7FFF.

#### 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 ST2108 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 be loaded with the restart vector from locations \$7FFC (low byte) and \$7FFD (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 \$7FF8 and \$7FF9.

#### 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 \$7FF6 and \$7FF7.

#### T0 Interrupt

The IRT0 (TIMER0 interrupt request) flag will be set while T0 overflows. With IET0 (TIMER0 interrupt enable) being set, the T0 interrupt will 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 T0 vector from locations \$7FF4 and \$7FF5.

#### 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 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 T1 vector from locations \$7FF2 and \$7FF3.

#### 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 \$7FF0 and \$7FF1.

#### 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 \$7FEE and \$7FEF.

All interrupt vectors address are listing as TABLE 8-1

TABLE 8-1 Interrupt Vectors

Name	Signal	Vector address	Priority	Comment
BRK	Internal	\$7FFF,\$7FFE	8	Software BRK operation vector
RESET	External	\$7FFD,\$7FFC	1	RESET vector
-	-	\$7FFB,\$7FFA	-	Reserved
INTX	External	\$7FF9,\$7FF8	2	PA0 edge interrupt
DAC	Internal	\$7FF7,\$7FF6	3	Reload DAC data interrupt
T0	INT/EXT	\$7FF5,\$7FF4	4	Timer0 interrupt
T1	INT/EXT	\$7FF3,\$7FF2	5	Timer1 interrupt
PT	External	\$7FF1,\$7FF0	6	Port-A transition interrupt
BT	Internal	\$7FEF,\$7FEE	7	Base Timer interrupt

## 8.2 Interrupt Request Flag

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

**TABLE 8-2 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	IRT0	IRDAC	IRX	-- 00 0000
Bit 5: <b>IRBT</b> : Base Timer Interrupt Request bit 1 = Time base interrupt occurs 0 = Time base interrupt doesn't occur						Bit 2: <b>IRT0</b> : Timer0 Interrupt Request bit 1 = Timer0 overflow interrupt occurs 0 = Timer0 overflow interrupt doesn't occur					
Bit 4: <b>IRPT</b> : Port-A Interrupt Request bit 1 = Port-A transition interrupt occurs 0 = Port-A transition interrupt doesn't occur						Bit 1: <b>IRDAC</b> : DAC reload Interrupt Request bit 1 = DAC time out interrupt occurs 0 = DAC time out interrupt doesn't occur					
Bit 3: <b>IRT1</b> : Timer1 Interrupt Request bit 1 = Timer1 overflow interrupt occurs 0 = Timer1 overflow interrupt doesn't occur						Bit 0: <b>IRX</b> : INTX Interrupt Request bit 1 = INTX edge interrupt occurs 0 = INTX edge interrupt doesn't occur					

**TABLE 8-3 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	IET0	IEDAC	IEX	-- 00 0000
Bit 5: <b>IEBT</b> : Base Timer Interrupt Enable bit 1 = Time base interrupt enable 0 = Time base interrupt disable						Bit 2: <b>IET0</b> : Timer0 Interrupt Enable bit 1 = Timer0 overflow interrupt enable 0 = Timer0 overflow interrupt disable					
Bit 4: <b>IEPT</b> : Port-A Interrupt Enable bit 1 = Port-A transition interrupt enable 0 = Port-A transition interrupt disable						Bit 1: <b>IEDAC</b> : DAC reload Interrupt Enable bit 1 = DAC time out interrupt enable 0 = DAC time out interrupt disable					
Bit 3: <b>IET1</b> : Timer1 Interrupt Enable bit 1 = Timer1 overflow interrupt enable 0 = Timer1 overflow interrupt disable						Bit 0: <b>IEX</b> : INTX Interrupt Enable bit 1 = INTX edge interrupt enable 0 = INTX edge interrupt disable					

\* These registers can be read and written on real chip, but can only be written on Emulation Board.



## 9. I/O PORTS

### 9.1 Description

ST2108 has four I/O ports, PORT-A, PORT-B, PORT-C and COMMON-PORT. In total, ST2108 provides for a maximum of 32 I/O pins with COMMON-PORT being programmed as output ports. For detail pin assignment, please refer to TABLE 9-1

**TABLE 9-1 I/O Description**

PORT NAME	PAD NAME	PAD NUMBER	PIN TYPE	FEATURE
Port-A	PA0/INTX	105	I/O	Programmable input/output pin
	PA1	106	I/O	
	PA2	107	I/O	
	PA3	108	I/O	
	PA4	109	I/O	
	PA5	110	I/O	
	PA6	111	I/O	
	PA7	112	I/O	
Port-B	PB0	113	I/O	Programmable input/output pin
	PB1	114	I/O	
	PB2	115	I/O	
	PB3	116	I/O	
	PB4	117	I/O	
	PB5	118	I/O	
	PB6	119	I/O	
	PB7	120	I/O	
Port-C	PC0	121	I/O	Programmable input/output pin
	PC1	122	I/O	
	PC2	123	I/O	
	PC3	124	I/O	
	PC4	125	I/O	
	PC5	126	I/O	
	PC6	127	I/O	
	PC7	128	I/O	
Common-Port	COM0	103	O	These 8 common pins can be programmed as output ports (open drain type).
	COM1	102	O	
	COM2	101	O	
	COM3	100	O	
	COM4	99	O	
	COM5	98	O	
	COM6	97	O	
	COM7	96	O	

## 9.2 Port-A

### 9.2.1 Port-A Description

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 de-bounce and interrupt edge selection (PA0 only).

TABLE 9-2 Summary Of 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	IRT0	IRDAC	IRX	-- 00 0000
\$03E	IENA	R/W	-	-	IEBT	IEPT	IET1	IET0	IEDAC	IEX	-- 00 0000

### 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 9-3 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: <b>PCA[7~0]</b> : 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 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 disabled 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 FIGURE 9-1.

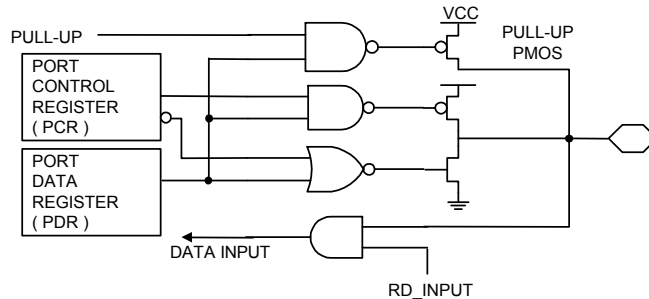


FIGURE 9-1 Port-A Block Diagram

TABLE 9-4 Port Function 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: <b>PULL</b> : Enable all pull-up function bit 1 = Enable pull-up function 0 = Disable pull-up function  Bit 6: <b>PDBN</b> : Enable Port-A interrupt de-bounce bit 1 = De-bounce for Port-A interrupt 0 = No de-bounce for Port-A interrupt  Bit 5: <b>INTEG</b> : 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 would 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 FIGURE 9-1.

Operate Port-A interrupt steps:

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
    .
  
```

Interrupt subroutine

```

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

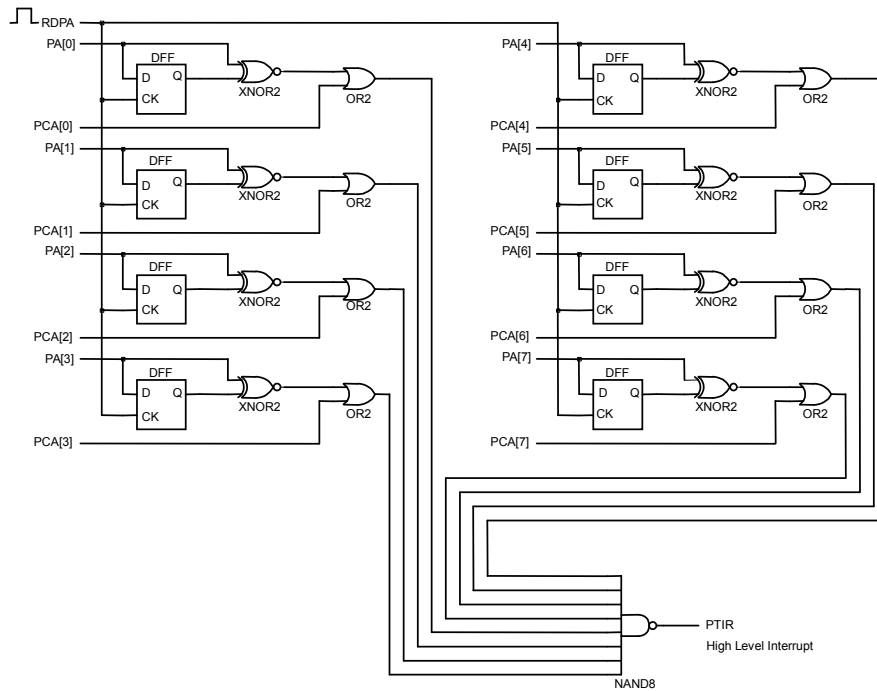


FIGURE 9-2 Port Interrupt Logic Diagram

9.2.5 Port-A Interrupt De-bounce

ST2108 has hardware de-bounce option for Port-A interrupt. The de-bounce will be enabled with "1" and disable with "0" for PDBN. The de-bounce will active when Port-A transition occurs,

PDBN enable and **OSCX enable**. The de-bounce time is **OSCX x 512 cycles (about 16 ms)**. Refer to TABLE 9-6.

TABLE 9-5 Port Function 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: <b>PDBN</b> : Enable Port-A interrupt de-bounce bit 1 = De-bounce for Port-A interrupt 0 = No de-bounce for Port-A interrupt											

9.2.6 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 Figure 9-3.

When programmer enables INTX and PT interrupts, PA0 trigger will occur. Both INTX and PT interrupts will happen sequentially. Please refer to 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
.
.
    
```

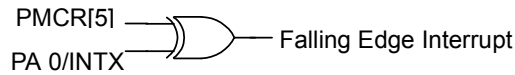


FIGURE 9-3 INTX Logic Diagram

### 9.3 Port-B and Port-C

#### 9.3.1 General Description

Port -B and Port-C are bit programmable bi-direction I/O port, which is controlled by PCB and PCC registers. It also provides

User with bit-programmable pull-up MOS and sound output port separately.

**TABLE 9-6 Summary of Port-B AND Port-C 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[7]	PB[6]	PB[5]	PB[4]	PB[3]	PB[2]	PB[1]	PB[0]	1111 1111
\$002	PC	R/W	PC[7]	PC[6]	PC[5]	PC[4]	PC[3]	PC[2]	PC[1]	PC[0]	1111 1111
\$009	PCB	R/W	PCB[7]	PCB[6]	PCB[5]	PCB[4]	PCB[3]	PCB[2]	PCB[1]	PCB[0]	0000 0000
\$00A	PCC	R/W	PCC[7]	PCC[6]	PCC[5]	PCC[4]	PCC[3]	PCC[2]	PCC[1]	PCC[0]	0000 0000
\$00F	PMCR	R/W	PULL	PDBN	INTEG	-	-	-	PSG0	PSGB	100 - -00

#### 9.3.2 Input/Output Control

PCB (or PCC) controls direction of Port-B (or Port-C). Every bit of PCB[7~0] (or PCC[7~0]) is mapped into the I/O

Direction of PB[7~0] (or PC[7~0]) correspondingly, with "1" for output mode, and "0" for input mode.

**TABLE 9-7 PORT-B Control Register (PCB)**

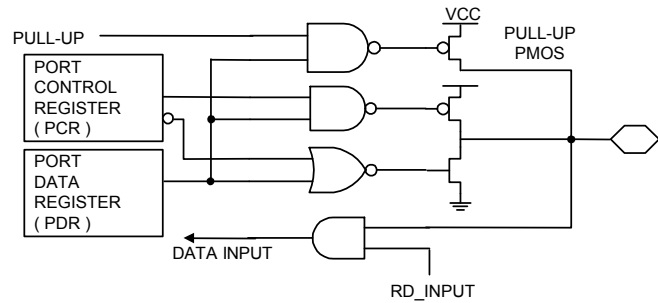
Address	Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default
\$009	PCB	R/W	PCB[7]	PCB[6]	PCB[5]	PCB[4]	PCB[3]	PCB[2]	PCB[1]	PCB[0]	0000 0000
Bit 7~0: <b>PCB[7~0]</b> : Port-B directional bits 1 = Output mode 0 = Input mode											

**TABLE 9-8 PORT-C Control Register (PCC)**

Address	Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default
\$00A	PCC	R/W	PCC[7]	PCC[6]	PCC[5]	PCC[4]	PCC[3]	PCC[2]	PCC[1]	PCC[0]	0000 0000
Bit 7~0: <b>PCC[7~0]</b> : Port-C directional bits 1 = Output mode 0 = Input mode											

**9.3.3 PORT-B and PORT-C 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, PC) when an I/O is used as input. The PULL control bit of PMCR also controls the ON/OFF of all pull-up MOS simultaneously. Please refer to FIGURE 9-4.



**FIGURE 9-4 Port-B and Port-C Block Diagram**

**TABLE 9-9 Port 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
<p>Bit 7: <b>PULL</b> : Enable all pull-up functions bit                      1 = Enable pull-up function                      0 = Disable pull-up function</p> <p>Bit 1: <b>PSG0</b> : 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</p> <p>Bit 0: <b>PSGB</b> : 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</p>											

### 9.4 Common-Port

The COM0~COM7 can be used as LCD drivers or output ports (open drain type). In output port mode, SCAN[7~0] will be map to COM7~COM0 output ports, which pin assignment will be decided by Bit 4 of LCTL[4], Please refer to the following table.

TABLE 9-10 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	REV	DUTY	CTR[3]	CTR[2]	CTR[1]	CTR[0]	0000 0000

Bit 4: **DUTY** : LCD duty control bit  
 1 = 1/16 duty and COM7~COM1 will be general-purpose output pins.  
 0 = 1/24 duty.

TABLE 9-11 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
\$039	SCAN	W	SCAN[7]	SCAN[6]	SCAN[5]	SCAN[4]	SCAN[3]	SCAN[2]	SCAN[1]	SCAN[0]	0000 0000

Bit x: **SCAN[x]** : COMx scan output bit  
 1 = COMx output =high impedance  
 0 = COMx output =LOW  
 where x = 0~7

## 10. OSCILLATOR

ST2108 is a 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 confirming SYSCK really switched when read XSEL bit. LCD driver, Timer1, Base Timer and PSG can utilize these two clock sources as well.

TABLE 10-1 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	-	LVDET	0000 00-0
Bit 7: <b>XSEL</b> : System clock (SYSCK) select (write) / confirm (read) bit 1 = OSCX 0 = OSC  Bit 6: <b>OSTP</b> : OSC stop control bit 1 = Disable OSC 0 = Enable OSC  Bit 5: <b>XSTP</b> : OSCX stop control bit 1 = Disable OSCX 0 = Enable OSCX  Bit 4: <b>XBAK</b> : OSCX driver heavy load bit 1 = OSCX normal load 0 = OSCX heavy load											

**Note:**

1. The XSEL (SYS[7]) bit will show which real working mode is when it is read.
2. System warm-up to 16 or 256 oscillation cycles is when system clock (SYSCK) is change or power on reset.
3. OSCX must be in heavy load mode for general crystals. So XBAK must keep "0" when normal operation.

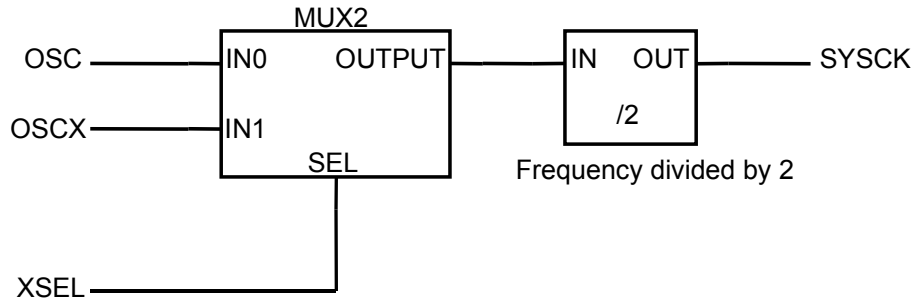


FIGURE 10-1 System Clock Diagram



# 11. TIMER/EVENT COUNTER

## 11.1 Prescaler

### 11.1.1 Function Description

The ST2108 has three timers: Base timer, Timer 0 and Timer 1 with two prescalers PRES and PREW. There are the two clock

sources for PRES and one clock source (OSCX) for PREW. Refer to FIGURE 11-1

TABLE 11-1 Summary of 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	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	-	-	-	-	-	-
\$024	TOM	*R/W	-	-	TOM[5]	TOM[4]	-	TOM[2]	TOM[1]	TOM[0]	- -00 -000
\$025	T0C	R/W	T0C[7]	T0C[6]	T0C[5]	T0C[4]	T0C[3]	T0C[2]	T0C[1]	T0C[0]	0000 0000
\$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	IRT0	IRDAC	IRX	- -00 0000
\$03E	IENA	*R/W	-	-	IEBT	IEPT	IET1	IET0	IEDAC	IEX	- -00 0000

Note: These registers can be read and written on real chip, but can only be written on Emulation Board.

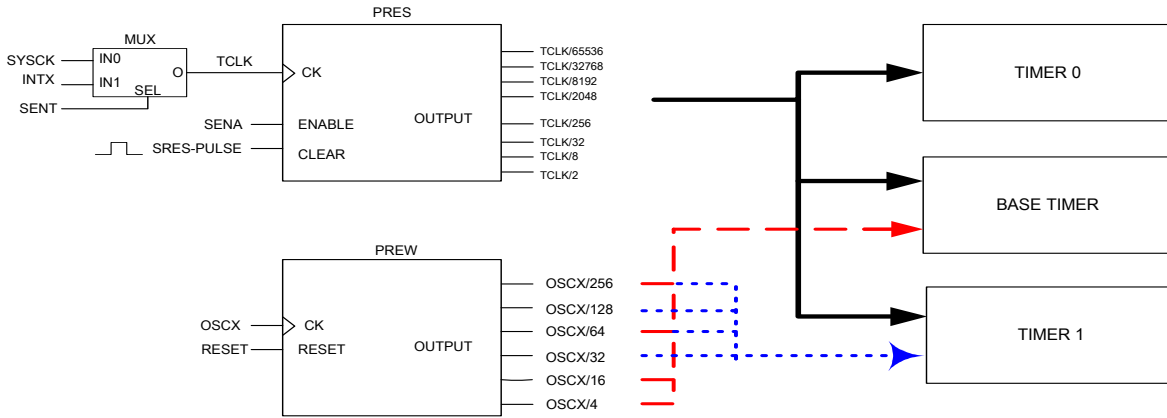


FIGURE 11-1 Structure Of Two Prescalers

11.1.2 PRES

The prescaler PRES is an 16-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 11-2 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]** :The low byte value 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"

11.1.3 PREW

The prescaler PREW is an 8-bits counter as shown in Figure 11-6. PREW provides four clocks source for base timer and

timer1. It stops counting only if OSCX stops or hardware reset occurs.

## 11.2 Base timer

### 11.2.1 Function Description

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 11-2

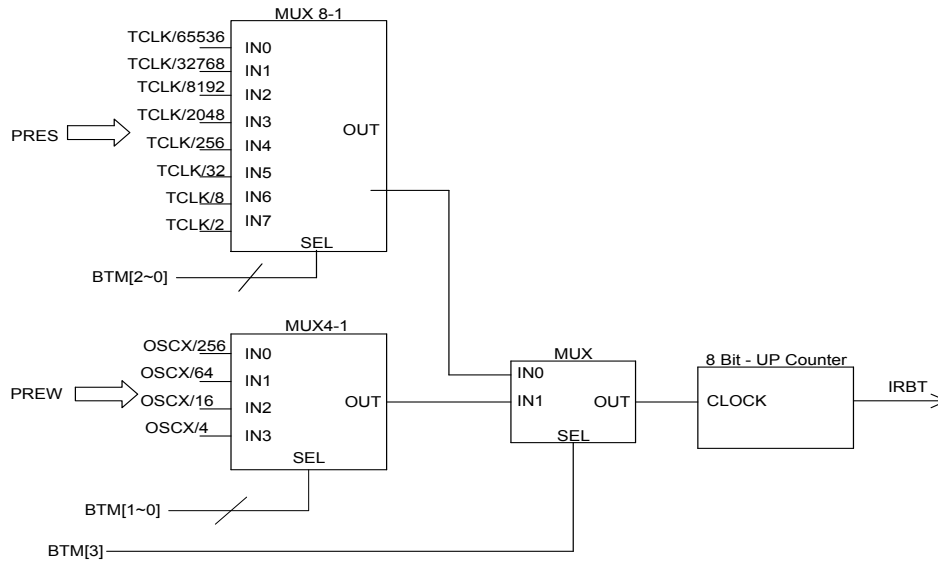


FIGURE 11-2 Structure Of Base Timer

### 11.2.2 Base Timer Clock Source Control

Several clock sources can be selected for Base Timer. Please refer to TABLE 11-3

TABLE 11-3 Clock Sources Of Base Timer

* SENA	BTM[3]	BTM[2]	BTM[1]	BTM[0]	Base Timer source clock
0	X	X	X	X	STOP
1	0	0	0	0	TCLK / 65536
1	0	0	0	1	TCLK / 32768
1	0	0	1	0	TCLK / 8192
1	0	0	1	1	TCLK / 2048
1	0	1	0	0	TCLK / 256
1	0	1	0	1	TCLK / 32
1	0	1	1	0	TCLK / 8
1	0	1	1	1	TCLK / 2
X	1	0	0	0	OSCX / 256
X	1	0	0	1	OSCX / 64
X	1	0	1	0	OSCX / 16
X	1	0	1	1	OSCX / 4

Note: TCLK will stop when an '0' is written to SENA (PRS[6]).

## 11.3 Timer 0

### 11.3.1 Function Description

The Timer0 is an 8-bit up counter. It can be used as a timer or an event counter. TOC(\$25) is a real time read/write counter. When an overflow from \$FF to \$00, a timer interrupt request IRT0 will

be generated. Timer0 will stop counting when system clock stops. Please refer to FIGURE 11-3.

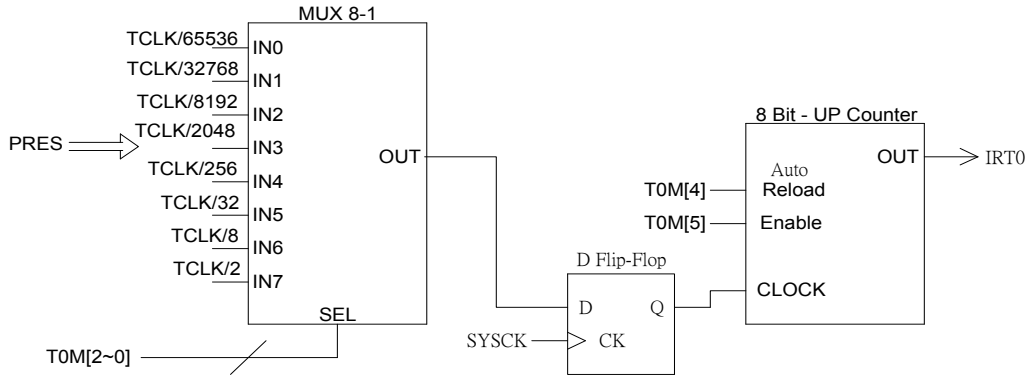


FIGURE 11-3 Timer0 Structure

### 11.3.2 Timer0 Clock Source Control

Several clock sources can be chosen from for Timer0. It's very important that Timer0 can keep counting as long as SYSCK stays active. Refer to TABLE 11-4.

TABLE 11-4 Clock Sources Of Timer0

TOM[2]	TOM[1]	TOM[0]	T0 Timer Clock Source
0	0	0	TCLK/65536
0	0	1	TCLK/32768
0	1	0	TCLK/8192
0	1	1	TCLK/2048
1	0	0	TCLK/256
1	0	1	TCLK/32
1	1	0	TCLK/8
1	1	1	TCLK/2

TOM[4] : Control automatic reload operation  
 0 : No auto reload  
 1 : Auto reload

TOM[5] : Control Timer 0 enable/disable  
 0 : Disable counting  
 1 : Enable counting

SENA : Prescaler enable bit  
 0 : TCLK stop  
 1 : TCLK counting

TABLE 11-5 Timer0 Register (T0C)

Address	Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default
\$025	T0C	R/W	T0C[7]	T0C[6]	T0C[5]	T0C[4]	T0C[3]	T0C[2]	T0C[1]	T0C[0]	0000 0000
Bit 7-0: T0C[7-0] : Timer0 up counter register											

### 11.4 Timer 1

The Timer1 is an 8-bit up counter. It used as timer/counter as program specified. The difference between base timer is that Timer1 will halt during CPU SBY, but base timer will not. It is shown in FIGURE 11-4.

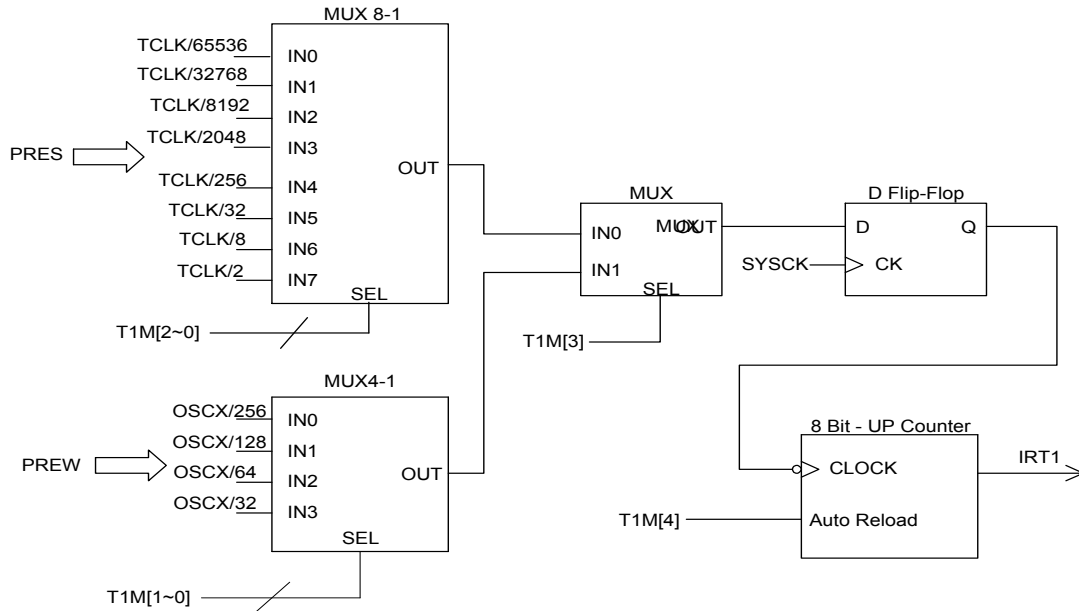


FIGURE 11-4 Timer1 Structure

TABLE 11-6 Timer1 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
Bit 7-0: T1C[7-0] : Timer1 up counter register											

TABLE 11-7 Clock Sources Of Timer1

T1M[3]	T1M[2]	T1M[1]	T1M[0]	T1 Timer Clock Source
0	0	0	0	TCLK/65536
0	0	0	1	TCLK/32768
0	0	1	0	TCLK/8192
0	0	1	1	TCLK/2048
0	1	0	0	TCLK/256
0	1	0	1	TCLK/32
0	1	1	0	TCLK/8
0	1	1	1	TCLK/2
1	0	0	0	OSCX/256
1	0	0	1	OSCX/128
1	0	1	0	OSCX/64
1	0	1	1	OSCX/32

T1M[4]: Control automatic reload operation  
 0: No auto reload  
 1: auto reload  
 SENA : Prescaler enable bit  
 0 : TCLK stop  
 1 : TCLK counting

# 12. PSG

## 12.1 Function description

The built-in dual channel Programmable Sound Generator (PSG) is controlled by register file directly. Its flexibility makes it useful in applications such as music synthesis, sound effects generation, audible alarms and tone signaling. In order to generate sound effects while allowing the processor to perform other tasks, the PSG can continue to produce sound after the initial commands have been given by the CPU. The structure of PSG was shown in FIGURE 12-2 and the PSG clock source is shown in FIGURE 12-1. The ST2108 has three PSG playing type. One for channel0(C0) & channel1(C1) square type tone sound playing. One for ch0 square tone sound and ch1 noise sound. The third sound playing type is DAC PCM playing.

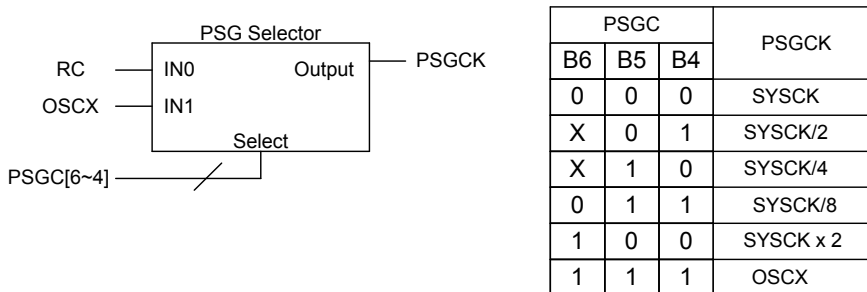


FIGURE 12-1 PSG Clock Source Control

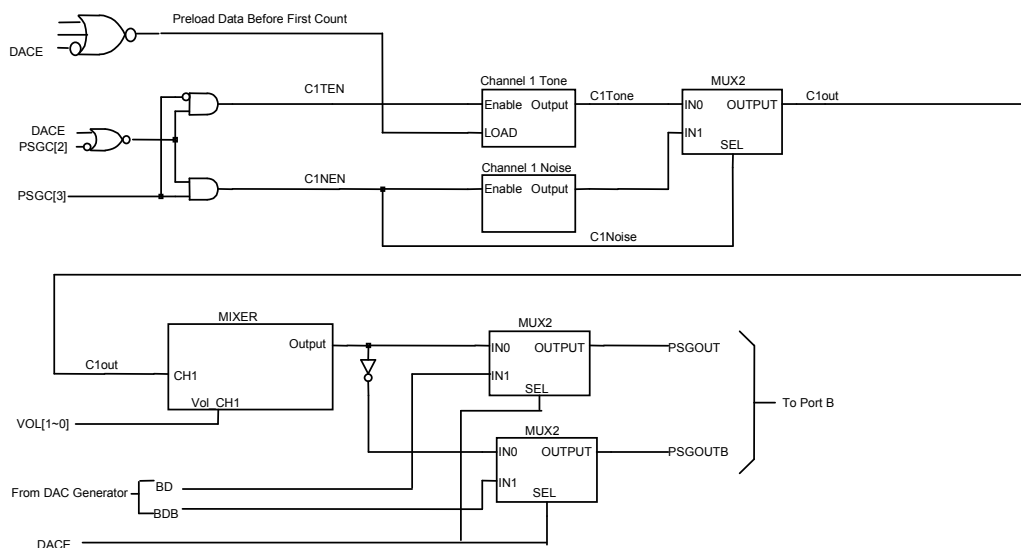


FIGURE 12-2 PSG Block Diagram

TABLE 12-1 Summary Of 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
\$010	PSG0L	W	PSG0[7]	PSG0[6]	PSG0[5]	PSG0[4]	PSG0[3]	PSG0[2]	PSG0[1]	PSG0[0]	0000 0000
\$011	PSG0H	W	-	-	-	-	PSG0[11]	PSG0[10]	PSG0[9]	PSG0[8]	---- 0000
\$012	PSG1L	W	PSG1[7]	PSG1[6]	PSG1[5]	PSG1[4]	PSG1[3]	PSG1[2]	PSG1[1]	PSG1[0]	0000 0000
\$013	PSG1H	W	-	-	-	-	PSG1[11]	PSG1[10]	PSG1[9]	PSG1[8]	---- 0000
\$016	PSGC	W	-	PCK[2]	PCK[1]	PCK[0]	PRBS	C1EN	C0EN	DACE=0	- 000 0000
		W	-	PCK[2]	PCK[1]	PCK[0]	DMD[1]	DMD[0]	INH	DACE=1	- 000 0000
\$017	VOL	W	VOL1[3]	VOL1[2]	VOL1[1]	VOL1[0]	VOL0[3]	VOL0[2]	VOL0[1]	VOL0[0]	0000 0000

TABLE 12-2 PSG 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	-	-	-	PSGO	PSGB	100 - - -00
<p>Bit 1: <b>PSGO</b> : PSG output enable bit                      1 = PSG data output pin if PB1 is set in output mode                      0 = PB1 is normal I/O pin</p> <p>Bit 0: <b>PSGB</b> : 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</p>											

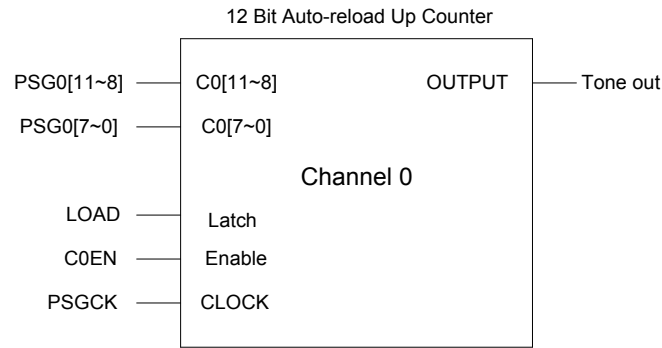
TABLE 12-3 PSG Volume Control Register (VOL)

Address	Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default
\$017	VOL	W	VOL1[3]	VOL1[2]	VOL1[1]	VOL1[0]	VOL0[3]	VOL0[2]	VOL0[1]	VOL0[0]	0000 0000
<p>Bit 3~0: <b>VOL0[3~0]</b> : PSG channel 0 volume control bit                      0000 = No sound output                      0001 = 1/16 volume (PSGCK must &gt;= 320K Hz)                      :                      0100 = 4/16 volume                      :                      1000 = 8/16 volume                      :                      1111 = Maximum volume (PSGCK must &gt;= 20K Hz)</p> <p>Bit 7~4: <b>VOL1[3~0]</b> : PSG channel 1 volume control bit                      0000 = No sound output                      0001 = 1/16 volume (PSGCK must &gt;= 320K Hz)                      :                      0100 = 4/16 volume                      :                      1000 = 8/16 volume                      :                      1111 = Maximum volume (PSGCK must &gt;= 20K Hz)</p> <p><b>Note: If single channel is enable, then PSG volume control can be double. (16 + 16 = 32 level volume control)</b></p>											

## 12.2 Tone Generator

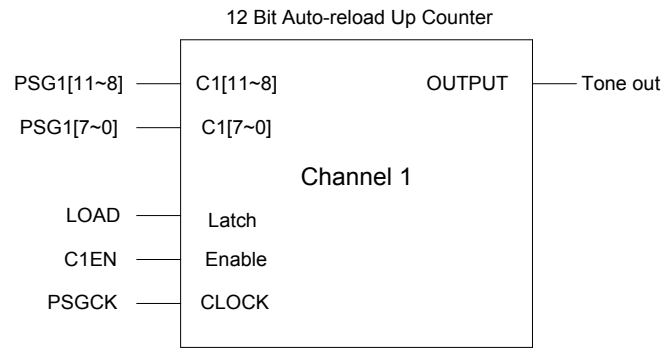
### 12.2.1 General Description

The tone frequency is decided by PSGCK and 12-bit programmable divider (PSG[11~0]). Please refer to FIGURE 12-3 and FIGURE 12-4.



$$\text{Frequency of Channel 0 Tone} = \text{PSGCK} / (1000\text{H} - \text{PSG0}[11\sim 0]) / 2$$

**FIGURE 12-3 Tone Generator Channel 0**



$$\text{Frequency of Channel 1 Tone} = \text{PSGCK} / (1000\text{H} - \text{PSG1}[11\sim 0]) / 2$$

**FIGURE 12-4 Tone Generator Channel 1**

**12.2.2 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 12-4 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	W	-	PCK[2]	PCK[1]	PCK[0]	PRBS	C1EN	C0EN	DACE=0	- 000 0000
		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 1: **C0EN** : PSG channel 0 (Tone) enable bit  
 1 = PSG0 (Tone) enable  
 0 = PSG0 (Tone) disable

Bit 2: **C1EN** : PSG channel 1 (Tone or Noise) enable bit  
 1 = PSG1 (Tone or Noise) enable  
 0 = PSG1 (Tone or Noise) disable

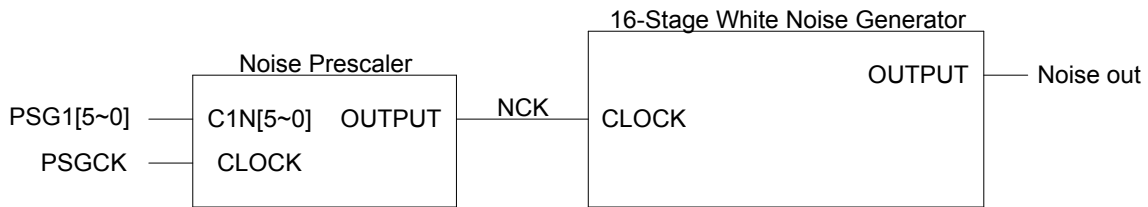
Bit 3: **PRBS** : Tone or Noise generator selection bit  
 1 = Noise generator  
 0 = Tone generator

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

## 12.3 Noise Generator Control

### 12.3.1 General description

Noise generator is shown in FIGURE 12-5, which base frequency is controlled by PSG1[5~0].



$$NCK \text{ Frequency} = \text{PSGCK} / (40H - \text{PSG1}[5\sim 0])$$

FIGURE 12-5 Noise Generator

### 12.3.2 Noise Generator Programming

To program noise generator, PSG0 (PMCR[1]) or PSGB (PMCR[0]) should be set to "1" for PB1 or PB0 in order to be in PSG output. DACE defines noise or DAC function.

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

### 13. PWM DAC

A built-in PWM DAC is for analog sampling data or voice signals. The structure of DAC is shown in TABLE 13-1. 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 frequency of RC oscillator can't be less than 2M Hz.

TABLE 13-1 Summary Of 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	-	-	-	PSGO	PSGB	100 - - 00
\$010	PSG0L	W	PSG0[7]	PSG0[6]	PSG0[5]	PSG0[4]	PSG0[3]	PSG0[2]	PSG0[1]	PSG0[0]	0000 0000
\$011	PSG0H	W	-	-	-	-	PSG0[11]	PSG0[10]	PSG0[9]	PSG0[8]	- - - - 0000
\$012	PSG1L	W	PSG1[7]	PSG1[6]	PSG1[5]	PSG1[4]	PSG1[3]	PSG1[2]	PSG1[1]	PSG1[0]	0000 0000
\$013	PSG1H	W	-	-	-	-	PSG1[11]	PSG1[10]	PSG1[9]	PSG1[8]	- - - - 0000
\$016	PSGC	W	-	PCK[2]	PCK[1]	PCK[0]	PRBS	C1EN	C0EN	DACE=0	- 000 0000
		W	-	PCK[2]	PCK[1]	PCK[0]	DMD[1]	DMD[0]	INH	DACE=1	- 000 0000
\$017	VOL	W	VOL1[3]	VOL1[2]	VOL1[1]	VOL1[0]	VOL0[3]	VOL0[2]	VOL0[1]	VOL0[0]	0000 0000

TABLE 13-2 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	W	DAC[7]	DAC[6]	DAC[5]	DAC[4]	DAC[3]	DAC[2]	DAC[1]	DAC[0]	0000 0000
<p>Bit 7~0: <b>DAC[7~0]</b> : DAC output data</p> <p>Note: For Single-Pin Single Ended mode, the effective output resolution is 7 bit.</p>											

TABLE 13-3 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	W	-	PCK[2]	PCK[1]	PCK[0]	PRBS	C1EN	C0EN	DACE=0	- 000 00-0
		W	-	PCK[2]	PCK[1]	PCK[0]	DMD[1]	DMD[0]	INH	DACE=1	- 000 0000
<p>Bit 0: <b>DACE</b> : 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</p> <p>Bit 1: <b>INH</b> : DAC output inhibit control bit                      1 = DAC output inhibit                      0 = DAC output enable</p> <p>Bit 3~2: <b>DMD[1~0]</b> : 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</p> <p>Bit 6~4: <b>PCK[2~0]</b> : PSGCK selection for PSG and DAC                      000 = SYSCK                      X01 = SYSCK / 2                      X10 = SYSCK / 4                      011 = SYSCK / 8                      100 = SYSCK x 2 (= frequency of RC oscillator)                      111 = OSCX</p> <p><b>Note: In DAC mode, PSGCK must select SYSCK x 2 (PCK[2~0]=100) under RC=2MHz.</b></p>											

### 13.2 Sample Rate Control

PSG1L and PSG1H control the sample rate. PSG1[11~6] controls PWM repeat times (usually set=111100 for four times of DAC reload) and PSG1[5~0] usually set '1'. The

input clock source is controlled by PCK[2~0]. The block diagram is shown as the following:

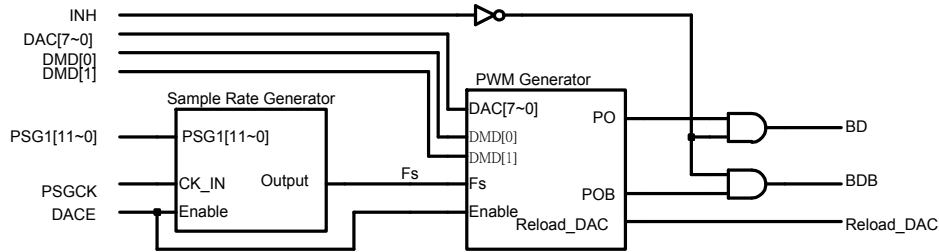


FIGURE 13-1 DAC Diagram

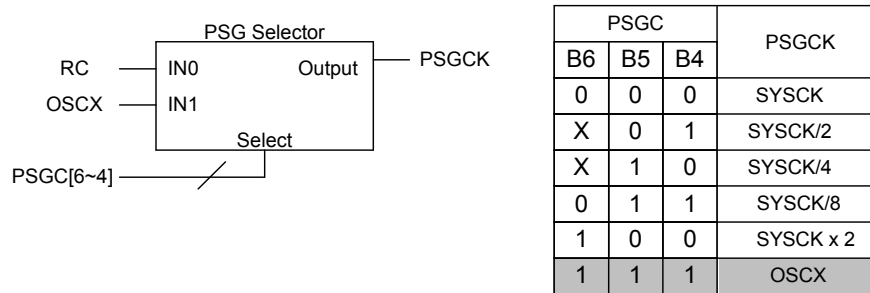


FIGURE 13-2 DAC Clock Source Control

TABLE 13-4 DAC Sample Rate Description (RCosc = 2MHz)

DAC interrupt frequency	PWM frequency	PSGC b6, b5, b4	PSG1H, PSG1L
8K	32K	100	00001111, 00111111
6K	12K	100	00001111, 10111111

### 13.3 PWM DAC Mode Options

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 13-3)

mode. They are depended on the application used. The DAC mode is controlled by DMD[1~0]. (TABLE 13-3)

#### 13.3.1 Single-Pin Mode (7-bit Accuracy)

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 13-3 shows the **PB1** waveforms.

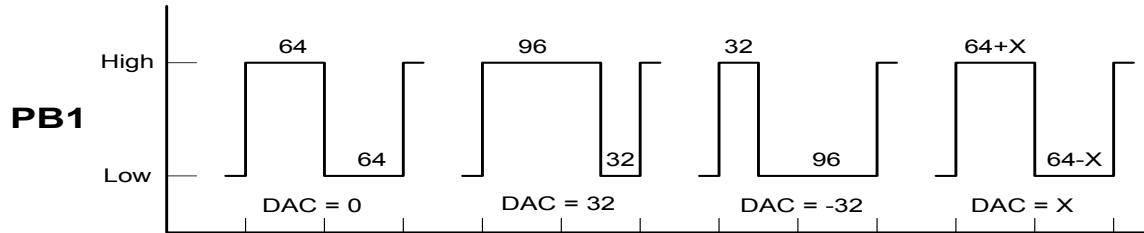


FIGURE 13-3 Single-Pin Mode Wave Form

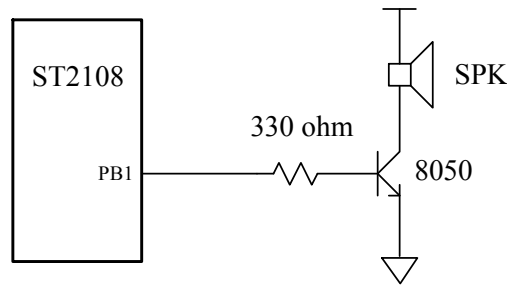


FIGURE 13-4 Single-Pin Mode Application Circuit

### 13.3.2 Two-Pin Two Ended Mode (8-bit Accuracy)

Two-Pin Two-Ended mode is designed for use with a single transistor amplifier. It requires two pin 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 13-5 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.

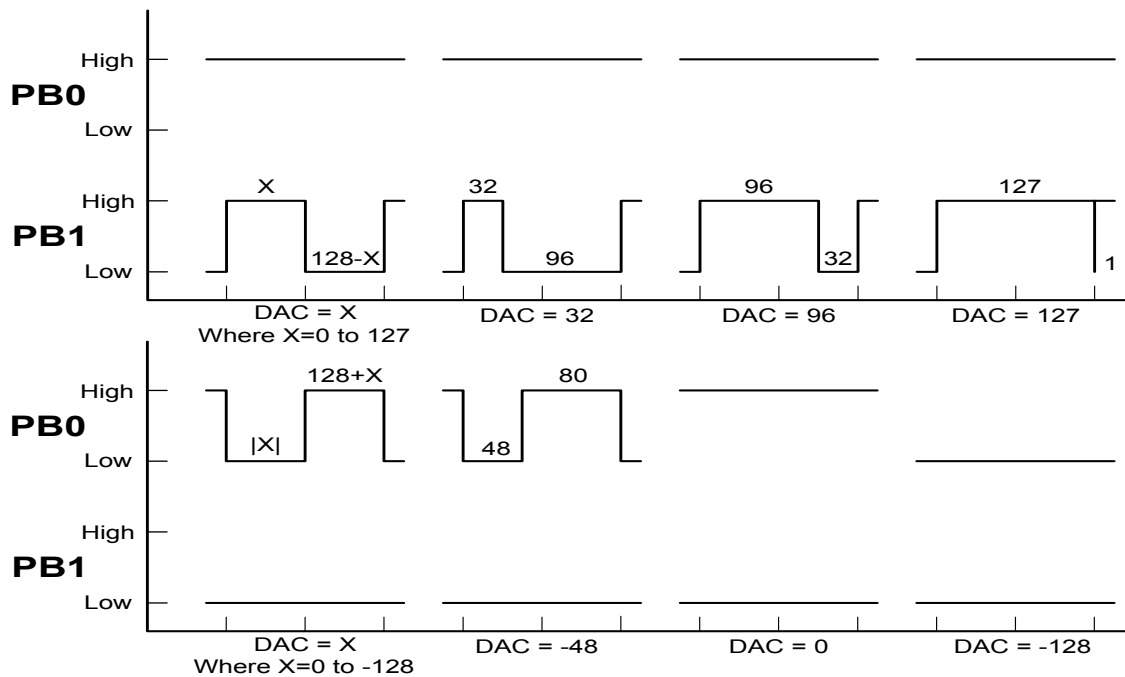


FIGURE 13-5 Two-Pin Two Ended Mode Wave-Form

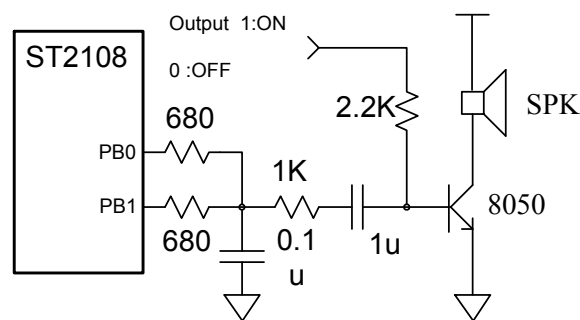


FIGURE 13-6 Two-Pin Two Ended Mode Application Circuit

### 13.3.3 Two-Pin Push Pull Mode (8-bit Accuracy)

Two-Pin Push Pull mode is designed for buzzer. It requires two pin 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 13-7 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.

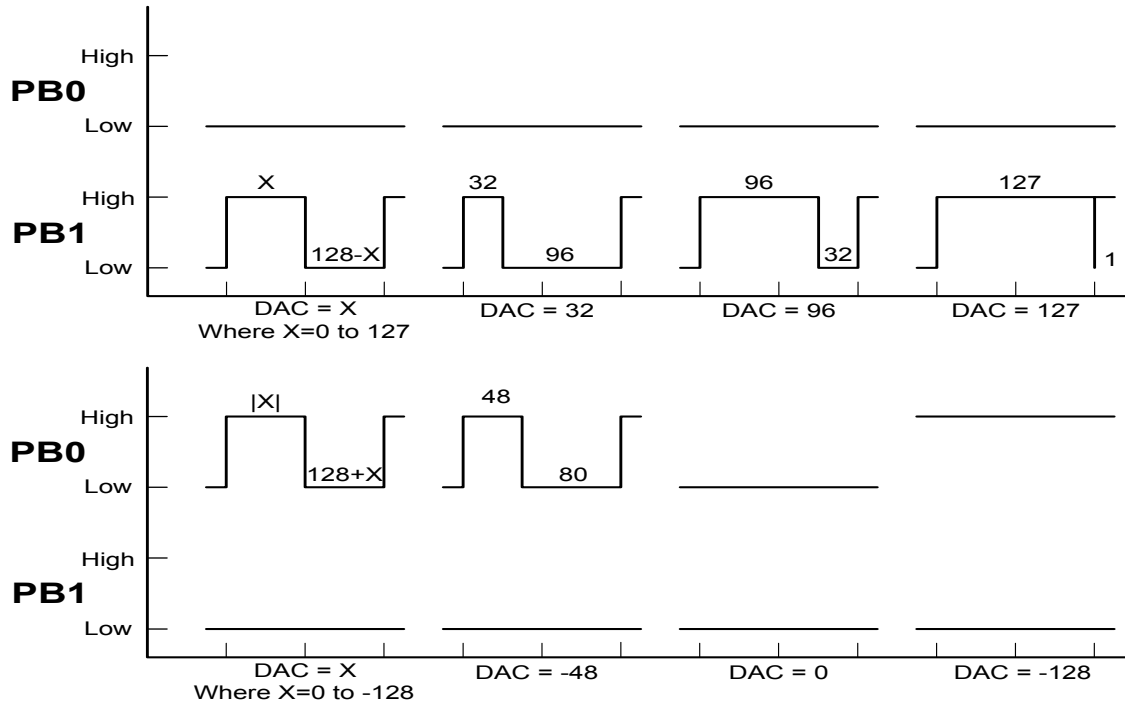


FIGURE 13-7 Two-Pin Push Pull Mode Wave Form

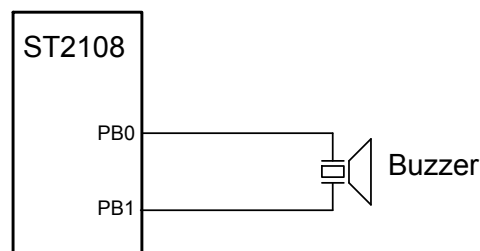


FIGURE 13-8 Two-Pin Push Pull Mode Application Circuit

## 14. LCD

The ST2108 is capable of driving up to 1752 dots of LCD panel directly. It supports two kinds of duty: 1/24 and 1/16. LCD block include display RAM (\$1000~ \$1148) for storing the display data, 73-segment output pins (SEG0~SEG72), 24-common output pins (COM0~COM23) or 16-common output pins (COM8~COM23).

All LCD RAM are random after power on reset. The LCD contrast is setting by software option.

**TABLE 14-1 Driver Output Levels**

Driver	Mode	Data	Display data output level
Common	Selected	H	VP
		L	V5
	Non-selected	H	V1
		L	V4
Segment	Selected	H	VP
		L	V5
	Non-selected	H	V2
		L	V3

### 14.1 LCD Timing

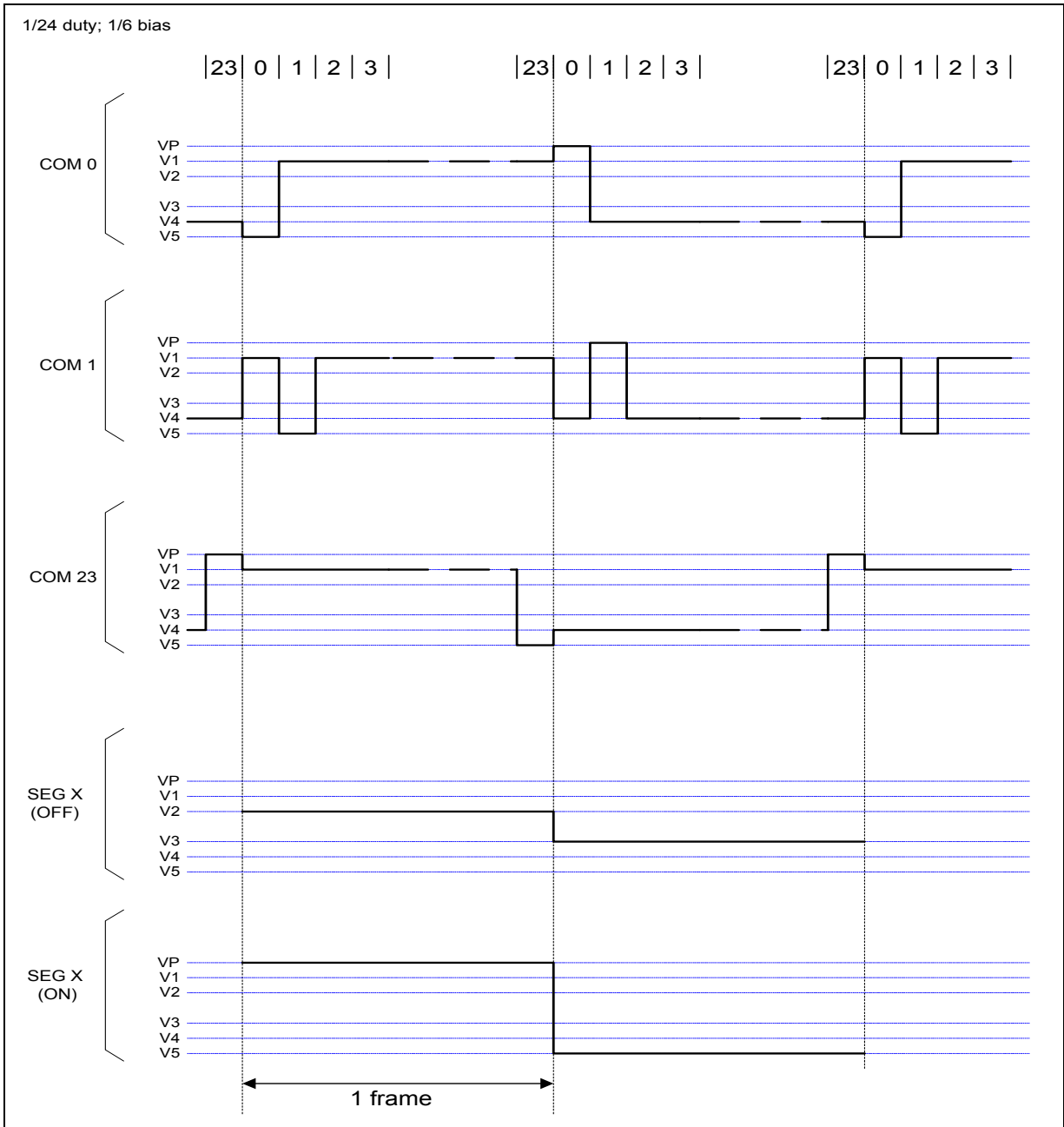


FIGURE 14-1 LCD Timing



## 14.2 LCD Control Register

TABLE 14-2 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	REV	DUTY	CTR[3]	CTR[2]	CTR[1]	CTR[0]	0000 0000
<p>Bit 7: <b>LPWR</b> : LCD power ON/OFF bit            1 = LCD power OFF            0 = LCD power ON</p> <p>Bit 6: <b>BLANK</b> : LCD display ON/OFF bit            1 = Disable LCD display (Common line is still scanning)            0 = Enable LCD display</p> <p>Bit 5: <b>REV</b> : LCD display reverse            1 = Reverse display            0 = Normal display</p> <p>Bit 4: <b>DUTY</b> : LCD duty control bit            1 = 1/16 duty and COM7~COM0 will be general-purpose output pins.            0 = 1/24 duty.</p> <p>Bit 3~0: <b>CTR[3~0]</b> : LCD contrast control            0000 = contrast level 16 (maximum)            0001 = contrast level 15            0010 = contrast level 14            0011 = contrast level 13            0100 = contrast level 12            0101 = contrast level 11            0110 = contrast level 10            0111 = contrast level 9            1000 = contrast level 8            1001 = contrast level 7            1010 = contrast level 6            1011 = contrast level 5            1100 = contrast level 4            1101 = contrast level 3            1110 = contrast level 2            1111 = contrast level 1 (minimum)</p>											

TABLE 14-3 LCD Clock Control Register

Address	Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default
\$03B	LCK	W	-	-	-	-	-	LCK[2]	LCK[1]	LCK[0]	---- -000
<p>Bit 2~0 : <b>LCK[2~0]</b> : LCD frame clock control            000 = RC / 64 ( 2,000,000/24/80/64 = 16.3 frames/sec * )            001 = RC / 32 ( 2,000,000/24/80/32 = 32.6 frames/sec * )            010 = RC / 16 ( 2,000,000/24/80/16 = 65.1 frames/sec * )            011 = RC / 8 ( 2,000,000/24/80/8 = 130.2 frames/sec * )            100 = RC / 4 ( 2,000,000/24/80/4 = 260.4 frames/sec * )            101 = X            110 = X            111 = X</p>											

\* Under RC = 2M Hz condition.

### 14.3 Display RAM

Since two kinds of duty are supported, different memory mappings should be referred. Please see TABLE 14-4 and

TABLE 14-5.

**TABLE 14-4 1/24 Duty LCD Memory Mapping**

	SEG0	SEG1	SEG2	SEG3	SEG4	SEG5		SEG72
<b>Address</b>	<b>1000H</b>	<b>1001H</b>	<b>1002H</b>	<b>1003H</b>	<b>1004H</b>	<b>1005H</b>		<b>1048H</b>
<b>COM0</b>	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7	.....	Bit7
<b>COM1</b>	Bit6	Bit6	Bit6	Bit6	Bit6	Bit6		Bit6
<b>COM2</b>	Bit5	Bit5	Bit5	Bit5	Bit5	Bit5		Bit5
<b>COM3</b>	Bit4	Bit4	Bit4	Bit4	Bit4	Bit4	.....	Bit4
<b>COM4</b>	Bit3	Bit3	Bit3	Bit3	Bit3	Bit3		Bit3
<b>COM5</b>	Bit2	Bit2	Bit2	Bit2	Bit2	Bit2		Bit2
<b>COM6</b>	Bit1	Bit1	Bit1	Bit1	Bit1	Bit1	.....	Bit1
<b>COM7</b>	Bit0	Bit0	Bit0	Bit0	Bit0	Bit0		Bit0
<b>Address</b>	<b>1080H</b>	<b>1081H</b>	<b>1082H</b>	<b>1083H</b>	<b>1084H</b>	<b>1085H</b>		<b>10C8H</b>
<b>COM8</b>	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7	.....	Bit7
<b>COM9</b>	Bit6	Bit6	Bit6	Bit6	Bit6	Bit6		Bit6
<b>COM10</b>	Bit5	Bit5	Bit5	Bit5	Bit5	Bit5		Bit5
<b>COM11</b>	Bit4	Bit4	Bit4	Bit4	Bit4	Bit4	.....	Bit4
<b>COM12</b>	Bit3	Bit3	Bit3	Bit3	Bit3	Bit3		Bit3
<b>COM13</b>	Bit2	Bit2	Bit2	Bit2	Bit2	Bit2		Bit2
<b>COM14</b>	Bit1	Bit1	Bit1	Bit1	Bit1	Bit1	.....	Bit1
<b>COM15</b>	Bit0	Bit0	Bit0	Bit0	Bit0	Bit0		Bit0
<b>Address</b>	<b>1100H</b>	<b>1101H</b>	<b>1102H</b>	<b>1103H</b>	<b>1104H</b>	<b>1105H</b>		<b>1148H</b>
<b>COM16</b>	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7	.....	Bit7
<b>COM17</b>	Bit6	Bit6	Bit6	Bit6	Bit6	Bit6		Bit6
<b>COM18</b>	Bit5	Bit5	Bit5	Bit5	Bit5	Bit5		Bit5
<b>COM19</b>	Bit4	Bit4	Bit4	Bit4	Bit4	Bit4	.....	Bit4
<b>COM20</b>	Bit3	Bit3	Bit3	Bit3	Bit3	Bit3		Bit3
<b>COM21</b>	Bit2	Bit2	Bit2	Bit2	Bit2	Bit2		Bit2
<b>COM22</b>	Bit1	Bit1	Bit1	Bit1	Bit1	Bit1	.....	Bit1
<b>COM23</b>	Bit0	Bit0	Bit0	Bit0	Bit0	Bit0		Bit0

**TABLE 14-5 1/16 Duty LCD Memory Mapping**

	SEG0	SEG1	SEG2	SEG3	SEG4	SEG5		SEG72
<b>Address</b>	<b>1000H</b>	<b>1001H</b>	<b>1002H</b>	<b>1003H</b>	<b>1004H</b>	<b>1005H</b>		<b>1048H</b>
<b>COM8</b>	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7	.....	Bit7
<b>COM9</b>	Bit6	Bit6	Bit6	Bit6	Bit6	Bit6		Bit6
<b>COM10</b>	Bit5	Bit5	Bit5	Bit5	Bit5	Bit5		Bit5
<b>COM11</b>	Bit4	Bit4	Bit4	Bit4	Bit4	Bit4	.....	Bit4
<b>COM12</b>	Bit3	Bit3	Bit3	Bit3	Bit3	Bit3		Bit3
<b>COM13</b>	Bit2	Bit2	Bit2	Bit2	Bit2	Bit2		Bit2
<b>COM14</b>	Bit1	Bit1	Bit1	Bit1	Bit1	Bit1	.....	Bit1
<b>COM15</b>	Bit0	Bit0	Bit0	Bit0	Bit0	Bit0		Bit0
<b>Address</b>	<b>1080H</b>	<b>1081H</b>	<b>1082H</b>	<b>1083H</b>	<b>1084H</b>	<b>1085H</b>		<b>10C8H</b>
<b>COM16</b>	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7	.....	Bit7
<b>COM17</b>	Bit6	Bit6	Bit6	Bit6	Bit6	Bit6		Bit6
<b>COM18</b>	Bit5	Bit5	Bit5	Bit5	Bit5	Bit5		Bit5
<b>COM19</b>	Bit4	Bit4	Bit4	Bit4	Bit4	Bit4	.....	Bit4
<b>COM20</b>	Bit3	Bit3	Bit3	Bit3	Bit3	Bit3		Bit3
<b>COM21</b>	Bit2	Bit2	Bit2	Bit2	Bit2	Bit2		Bit2
<b>COM22</b>	Bit1	Bit1	Bit1	Bit1	Bit1	Bit1	.....	Bit1
<b>COM23</b>	Bit0	Bit0	Bit0	Bit0	Bit0	Bit0		Bit0

Notice: Can not use undefined RAM area (\$1049~\$107F, \$10C9~\$10FF).

## 15. DIRECT MEMORY ACCESS (DMA)

To speed up the memory access of this system, a sequential direct memory access(DMA) controller is designed-in. DMA can perform memory transfer function more efficient than CPU does. While DMA working, data ROM register (DRR) will disable and DMA use DMA memory bank register (DMR) to access ROM. After DMA

complete, ROM bank control still return to DRR. With the help of DMR can make DMS across bank boundary smoothly, but DMR is only valid for DMS. **The DMR can automatic increase when DMS across bank boundary.**

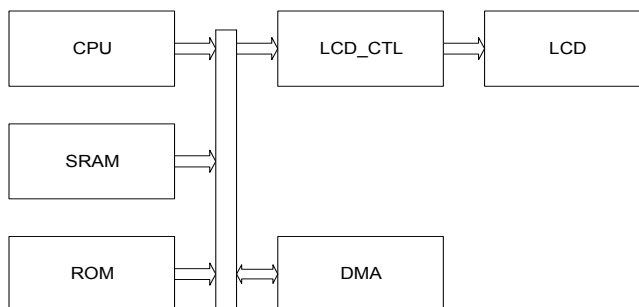


FIGURE 15-1 System Block Diagram

### 15.1 DMA Control Register

The control register is shown as following:

TABLE 15-2 DMA Control Register (LCTL)

Address	Register	R/W	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	COMMENT
\$028	DMSL	W	DMS7	DMS6	DMS5	DMS4	DMS3	DMS2	DMS1	DMS0	DMA Source register low byte
\$029	DMSH	W	DMS15	DMS14	DMS13	DMS12	DMS11	DMS10	DMS9	DMS8	DMA Source register high byte
\$02A	DMDL	W	DMD7	DMD6	DMD5	DMD4	DMD3	DMD2	DMD1	DMD0	DMA Desitination register low byte
\$02B	DMDH	W	DMD15	DMD14	DMD13	DMD12	DMD11	DMD10	DMD9	DMD8	DMA Desitination register high byte
\$02C	DCNTL	W	DCNT7	DCNT6	DCNT5	DCNT4	DCNT3	DCNT2	DCNT1	DCNT0	DMA Counter low byte
\$02D	DCNTH	W	-	-	-	DFIX	DCNT11	DCNT10	DCNT9	DCNT8	DMA Counter high byte
\$033	DMR	R/W	DMR7	DMR6	DMR5	DMR4	DMR3	DMR2	DMR1	DMR0	DMA memory bank register

DCNTH[4]: DFIX DMA destination counter mode

- 0: increase mode (DMS++ and DMD++ after every move)
- 1: fixed mode (DMS++ but DMD is fixed)

The DMA always move (DCNT+1) bytes of data.

DMA will start right after CPU write data into register DCNTL. During the DMA operation, the CPU hold, until the DMA transfer completed.

The DMR register reset to "\$00" on real chip, but Emulation Board is "unknown", so recommend initial DMR register before use.

**Before Read/Write you have to initial the PRR, DRR, DMR register when system reset.**

## 15.2 DMA Programming Flow

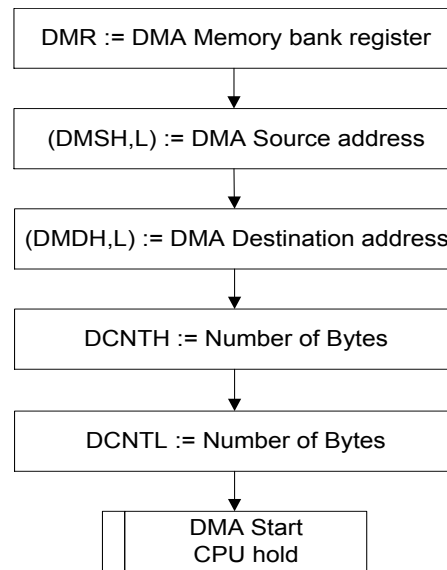


FIGURE 15-2 DMA Programming Flow

## 15.3 Example Program 1:

This program fills "00" to address \$1000~\$12FF.

```

STZ   $1000           ;; "00" to $1000
STZ   <DMSL
LDA   #$10
STA   <DMSH           ;; source = $1000
STA   <DMDH
LDA   #$01
STA   <DMDL           ;; destination = $1001
LDA   #$02
STA   <DCNTH
LDA   #$FE
STA   <DCNTL         ;; move $2FF bytes
:
:
  
```

## 15.4 Example Program 2:

This program moves data in address \$1080~\$12FF to \$1000~\$127F.

```
LDA    #$80
STA    <DMSL
LDA    #$10
STA    <DMSH    ;; source = $1080
STA    <DMDH
STZ    <DMDL    ;; destination = $1000
LDA    #$02
STA    <DCNTH
LDA    #$7F
STA    <DCNTL    ;; move $280 bytes
:
:
```

## 15.5 Application Program 3:

This program moves data in address \$8000~\$803F one single port at \$0200.

```
STZ    <DMSL
LDA    #$80
STA    <DMSH    ;; source = $8000
STZ    <DMDL
LDA    #$02
STA    <DMDH    ;; destination = $0200
LDA    #$10
STA    <DCNTH
LDA    #$3F
STA    <DCNTL    ;; move $40 bytes
:
:
```

## 16. POWER DOWN MODES

ST2108 has three power down modes: WAI-0, WAI-1 and STP. The instruction WAI will enable either WAI-0 or WAI-1, which is controlled by **WAIT**(SYS[2]). And the instruction

STP will enable **STP** mode in the same manner. WAI-0 and WAI-1 modes can be waked up by interrupt. However, **STP** mode can only be waked up by hardware reset.

TABLE 16-1 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	-	LVDET	0000 00-0
<p>Bit 3: <b>WSKP</b> : System warm-up control bit                      1 = Warm-up to 16 oscillation cycles                      0 = Warm-up to 256 oscillation cycles</p> <p>Bit 2: <b>WAIT</b> : WAI-0 / WAI-1mode select bit                      1 = WAI instruction causes the chip to enter WAI-1 mode                      0 = WAI instruction causes the chip to enter WAI-0 mode</p>											

### 16.1 WAI-0 Mode:

If **WAIT** is cleared, WAI instruction makes MCU enter WAI-0 mode. In the mean time, the oscillator, interrupts, timer/counter, and PSG are still working. On the other hand CPU and the related instruction execution stop. All registers, RAM, and I/O pins will retain the same states as those before the MCU entered power down mode. WAI-0 mode

can be waked up by reset or interrupt request even if user sets interrupt disable flag I. In that case MCU will be waked up but not entering interrupt service routine. If interrupt disable flag is cleared (I='0'), the corresponding interrupt vector will be fetched and the service routine will be executed. The sample program is shown below:

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

### 16.2 WAI-1 Mode:

If **WAIT** is set, WAI instruction makes MCU enter WAI-1 mode. In this mode, CPU stops, but the PSG, timer/counter keep running if their clock sources are from OSCX. The wake-up procedure is the same as for WAI-0. The difference is that the warm-up cycles occurs when waking

from WAI-1. Sample program is shown as following:  
**Notice: If using resonator for OSC, only OSCX can be used for system clock (SYSCK) before entering WAI-1.**

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

### 16.3 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

can only be waked up by hardware reset, and the warm-up cycles occurs at the same time.

FIGURE 16-1 Status Under Power Down Modes

**SYSCK source is OSC:**

Mode	Timer0,1	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 is OSCX:**

Mode	Timer0,1	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

## 17. LOW VOLTAGE DETECTOR

ST2108 has a built-in low voltage detector for power management. The active range of voltage detection is from 2.4V to 2.7V. When **LVDET** is set, detector circuit is enabled and the detection result will be outputted at the same bit after 3  $\mu$ s. Using read instruction twice can get this

result: both equal '1' represents 'low voltage'. Once low voltage detector is enabled, it keeps on consuming power. So it is important that remember to write "0" to LVDET to disable the detector after detection is completed. One sample program is shown below:

**Start:**

SMB0 <SYS ; enable detector

:

Wait 3  $\mu$ s

:

CLC

BBR0 <SYS,Normal\_Voltage

BBR0 <SYS,Normal\_Voltage

**Low\_Voltage:**

SEC

**Normal\_Voltage:**

RMB0 <SYS ; disable detector

TABLE 17-1 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	-	LVDET	0000 00-0

Bit 0: **LVDET** : Low voltage detect  
 1 = Enable detector (write) / Low voltage (read)  
 0 = Disable detector (write) / Normal voltage (read)

## 18. ELECTRICAL CHARACTERISTICS

DC Supply Voltage ----- -0.3V to +4.5V  
 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.

### 18.1 DC Electrical Characteristics

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

Parameter	Symbol	Min.	Typ.	Max.	Unit	Condition
Operating Voltage	$V_{DD}$	2.4	3	3.4	V	
Operating Current	$I_{OP}$		800	1000	$\mu A$	All output pins unload, execute NOP instruction, LCD on
Standby Current	$I_{SB0}$	-	0.05	-	$\mu A$	All output pins unload, OSCX off, LCD off (WAIT1/STOP mode)
Standby Current	$I_{SB1}$	-	0.5	-	$\mu A$	All output pins unload, OSCX on, LCD off (WAIT1/STOP mode)
Standby Current	$I_{SB2}$	-	60	-	$\mu A$	All output pins unload, OSCX off, LCD off (WAIT0 mode)
LCD consumption	$I_{LCD}$	-	200	300	$\mu A$	Bias resistor = 60k, LCD size=1.5cm x 4.5cm
Input High Voltage	$V_{IH}$	$0.7V_{DD}$ $0.85V_{DD}$	-	$V_{DD} + 0.3$	V	PORT A, PORT B, PORT C $\overline{RESET}$ , $\overline{INT}$
Input Low Voltage	$V_{IL}$	$GND - 0.3$	-	$0.3V_{DD}$ $0.15V_{DD}$	V	PORT A, PORT B, PORT C $\overline{RESET}$ , $\overline{INT}$
Pull-up resistance	$R_{IH}$	250	300	450	$K\Omega$	PORTA, PORTB, PORT C ( $I = -6\mu A$ , $V_{IH} = 0.3V_{DD}$ ).
Output high voltage	$V_{OH1}$	$0.7V_{DD}$	-		V	PORTA, PORTB, PORT C ( $I_{OH} = -3mA$ ).
Output low voltage	$V_{OL1}$			$0.3V_{DD}$	V	PORTA, PORTB, PORT C ( $I_{OL} = 6mA$ ).
Output high voltage	$V_{OH2}$	$0.7 V_{DD}$			V	PB0/1 as PSG/DAC, $I_{OH} = -4mA$ .
Output low voltage	$V_{OL2}$			$0.3V_{DD}$	V	PB0/1 as PSG/DAC, $I_{OL} = 8mA$ .
Output high voltage	$V_{OH3}$	2.8			V	SEGx, $I_{oh} = -800\mu A$ , $C = 50P$ , rise time < 200ns
Output low voltage	$V_{OL3}$			0.2	V	SEGx, $I_{ol} = 800\mu A$
Output high voltage	$V_{OH5}$	$0.7V_{DD}$			V	COM0~7, $I_{oh} = -1 mA$ .
Output low voltage	$V_{OL5}$			$0.3V_{DD}$	V	COM0~7, $I_{ol} = 1 mA$ .
Oscillation start time	$T_{STT}$	-	1	2	s	
Frequency stability	$\Delta F / F$			1	PPM	$[F(3.0) - F(2.5)] / F(3.0)$ (crystal oscillator)
Frequency variation	$\Delta F / F$	-10	3	10	PPM	$C1 = 15 - 30P$ .
Low voltage detector current	$I_{Vdet}$		11		$\mu A$	No detector voltage adjustment



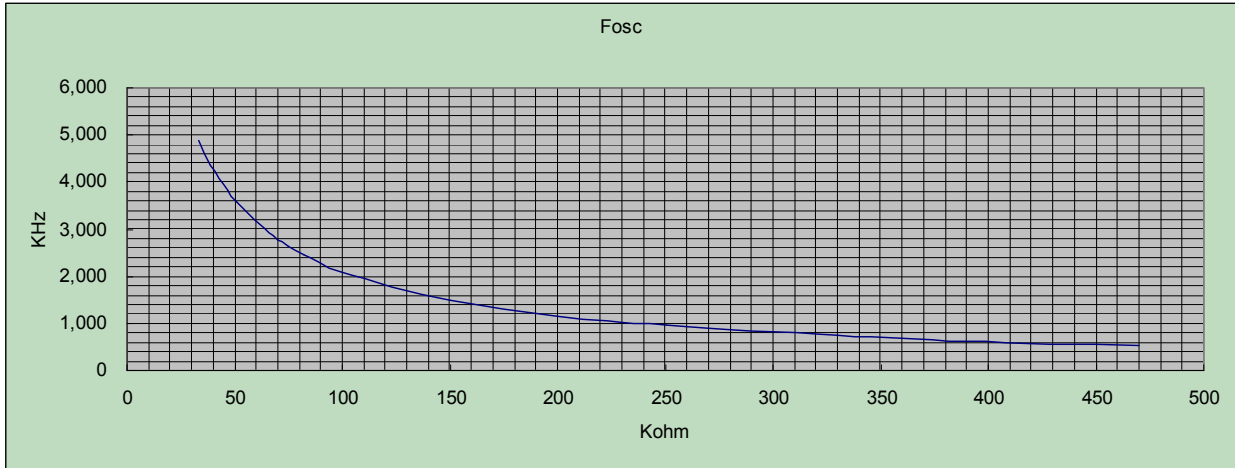


FIGURE 18-1 Oscillation Resistor VS. Frequency

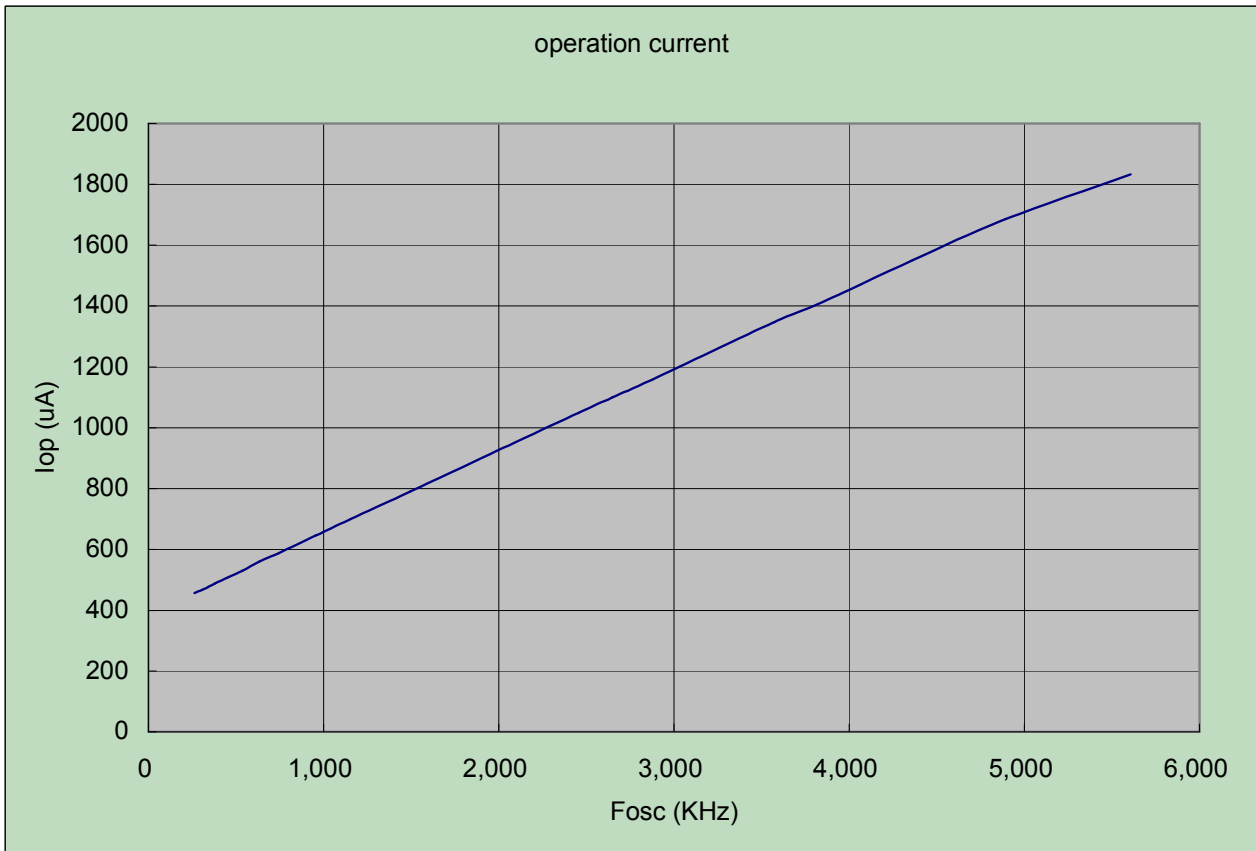
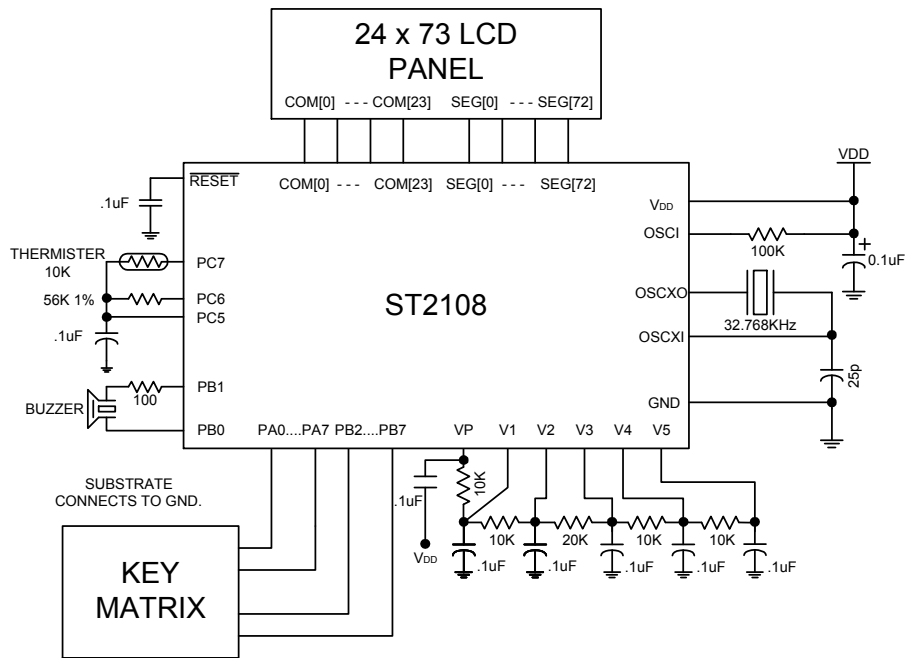


FIGURE 18-2 Typical Operation Current VS. Operation Frequency

# 19. APPLICATION CIRCUITS

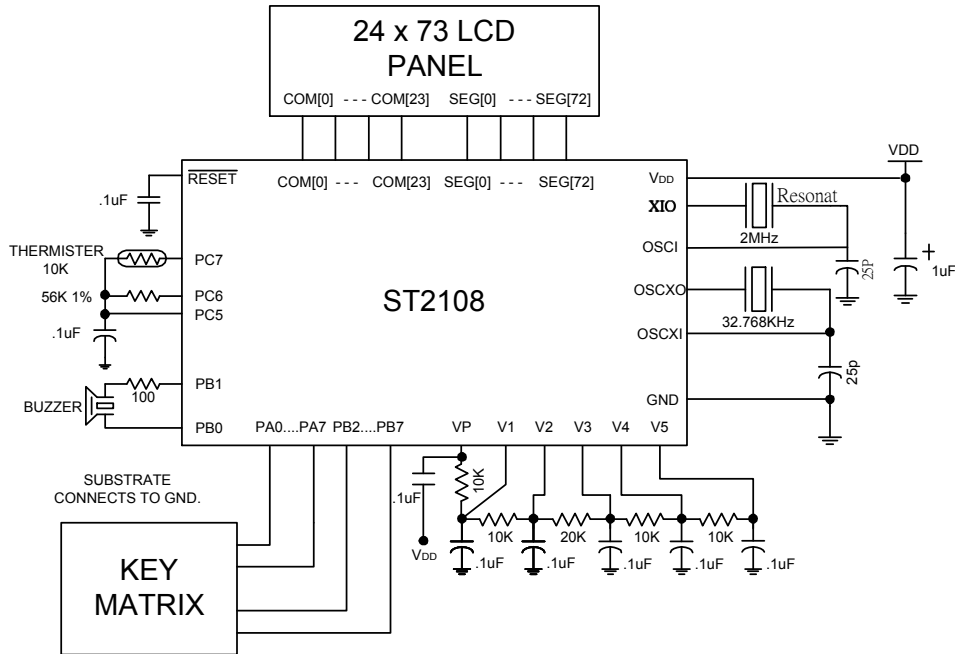
## 19.1 Application 1:

- VDD : 3.0V
- CLOCK : RC 2M Hz and crystal 32.768K Hz
- LCD : 24 x 73
- KEY : 48 key



19.2 Application 2:

- VDD : 3.0V
- CLOCK : Use resonator 2MHz and crystal 32.768K Hz
- LCD : 24 x 73
- KEY : 48 key



## 20. PAD DIAGRAM

Unit: um

Pad No.	Name	X	Y	Pad No.	Name	X	Y	Pad No.	Name	X	Y
1	V5	-2063	-2085	47	SEG28	2066	-674	93	COM10	-713	2115
2	V4	-1943	-2114	48	SEG29	2066	-554	94	COM9	-833	2115
3	V3	-1823	-2114	49	SEG30	2066	-434	95	COM8	-953	2115
4	V2	-1703	-2114	50	SEG31	2066	-314	96	COM7	-1073	2115
5	V1	-1583	-2114	51	SEG32	2066	-194	97	COM6	-1193	2115
6	VP	-1463	-2114	52	SEG33	2066	-74	98	COM5	-1313	2115
7	COM12	-1343	-2114	53	SEG34	2066	46	99	COM4	-1433	2115
8	COM13	-1223	-2114	54	SEG35	2066	166	100	COM3	-1553	2115
9	COM14	-1103	-2114	55	SEG36	2066	286	101	COM2	-1673	2115
10	COM15	-983	-2114	56	SEG37	2066	406	102	COM1	-1793	2115
11	COM16	-863	-2114	57	SEG38	2066	526	103	COM0	-1913	2115
12	COM17	-743	-2114	58	SEG39	2066	646	104	TEST	-2033	2115
13	COM18	-623	-2114	59	SEG40	2066	766	105	PA0	-2063	1995
14	COM19	-503	-2114	60	SEG41	2066	886	106	PA1	-2063	1875
15	COM20	-383	-2114	61	SEG42	2066	1006	107	PA2	-2063	1755
16	COM21	-263	-2114	62	SEG43	2066	1126	108	PA3	-2063	1635
17	COM22	-143	-2114	63	SEG44	2066	1246	109	PA4	-2063	1515
18	COM23	-23	-2114	64	SEG45	2066	1366	110	PA5	-2063	1395
19	SEG0	97	-2114	65	SEG46	2066	1486	111	PA6	-2063	1275
20	SEG1	217	-2114	66	SEG47	2066	1606	112	PA7	-2063	1155
21	SEG2	337	-2114	67	SEG48	2066	1726	113	PB0	-2063	1035
22	SEG3	457	-2114	68	SEG49	2066	1846	114	PB1	-2063	915
23	SEG4	577	-2114	69	SEG50	2066	1966	115	PB2	-2063	795
24	SEG5	697	-2114	70	SEG51	2047	2115	116	PB3	-2063	675
25	SEG6	817	-2114	71	SEG52	1927	2115	117	PB4	-2063	555
26	SEG7	937	-2114	72	SEG53	1807	2115	118	PB5	-2063	435
27	SEG8	1057	-2114	73	SEG54	1687	2115	119	PB6	-2063	315
28	SEG9	1177	-2114	74	SEG55	1567	2115	120	PB7	-2063	195
29	SEG10	1297	-2114	75	SEG56	1447	2115	121	PC0	-2063	75
30	SEG11	1417	-2114	76	SEG57	1327	2115	122	PC1	-2063	-45
31	SEG12	1537	-2114	77	SEG58	1207	2115	123	PC2	-2063	-165
32	SEG13	1657	-2114	78	SEG59	1087	2115	124	PC3	-2063	-285
33	SEG14	1777	-2114	79	SEG60	967	2115	125	PC4	-2063	-405
34	SEG15	1897	-2114	80	SEG61	847	2115	126	PC5	-2063	-525
35	SEG16	2017	-2114	81	SEG62	727	2115	127	PC6	-2063	-645
36	SEG17	2066	-1994	82	SEG63	607	2115	128	PC7	-2063	-765
37	SEG18	2066	-1874	83	SEG64	487	2115	129	OSC XO	-2063	-885
38	SEG19	2066	-1754	84	SEG65	367	2115	130	OSC XI	-2063	-1005
39	SEG20	2066	-1634	85	SEG66	247	2115	131	/RESET	-2063	-1125
40	SEG21	2066	-1514	86	SEG67	127	2115	132	OSCI	-2063	-1245
41	SEG22	2066	-1394	87	SEG68	7	2115	133	XIO	-2063	-1365
42	SEG23	2066	-1274	88	SEG69	-113	2115	134	GND	-2063	-1485
43	SEG24	2066	-1154	89	SEG70	-233	2115	135	NC	-2063	-1605
44	SEG25	2066	-1034	90	SEG71	-353	2115	136	VDD	-2063	-1725
45	SEG26	2066	-914	91	SEG72	-473	2115	137	CUP1+	-2063	-1845
46	SEG27	2066	-794	92	COM11	-593	2115	138	CUP1-	-2063	-1965

Chip size = 4340 um x 4440 um  
Substrate connect to ground

(there is a number beside pad 1 for easy locate)



FIGURE 20-1 The PCB 113-1 of ST2108 EVB



Project name \_\_\_\_\_ / \_\_\_\_\_ / \_\_\_\_\_

	Confirmed Item	Check	Note
1	Control register \$38“MUST”be reset to“0” .		
2	After power on , initial user RAM and confirm control register .		
3	Confirm LCD panel’s V <sub>OP</sub> ( contrast level ) 、 Duty and Bias .		
4	Confirm the difference between E.V. Board and real chip ( ex. V <sub>OP</sub> 、 driving ability 、 F <sub>OSC</sub> 、 power consumption 、 noise...etc. )		
5	Before entry power down mode , turn off un-used peripheral. ( LCD driver 、 PSG 、 OSC or OSCX )		
6	Make sure power down mode work .		
7	Calculate average operating current . ( Wake up time ratio )		
8	Confirm I/O directions and set pull-up for un-used input pins .		
9	For input mode with pull-up function , Please set bit 7 of port condition control register ( PMCR[7] ) and each bit of port data register .		
10	If use I/O for pin option , please re-configure I/O status after reading . ( directions and pull-up resistor )		
11	Pay attention to“bit instructions” , because some registers have different function for read and write acting. ex. PA 、 PB 、 PRS 、 SYS and control register for write only .		
12	Disable un-used function’s control register and put“RTI” Instruction at un-used interrupt vector .		
13	Make sure timer counting correct .		
14	Low voltage detect program have to read twice , and disable this function after reading . ( Refer to specification please )		
15	Make sure software key de-bounce work . ( 10 ~ 50 mS )		
16	Make sure calendar counting correct . ( include user setting )		
17	Make sure stack memory will not overflow .		
18	Under test mode , every functions / parts must be tested . ex. LCD 、 LED 、 speaker / buzzer 、 key 、 motor and senser...etc.		
19	Please use same parts when developing and producing .		
20	Please select general parts for production.		
21	When testing , write every unusual situation down and find out the reasons indeed.		
22	Make sure the program accept un-normal operatings and system will not hold or crash down .		
23	Please write specification’s version number down .		
24	When you set I/O port as input mode , please make sure signal level stable before reading . ex. When key scan , please delay 12 uS then get key code .		
25	32768Hz of crystal must working with heavy load. The difference of power consumption between Normal and Heavy is only 1uA		

Engineer \_\_\_\_\_

Manager \_\_\_\_\_

Reference table for LCD panel's parameters :

According to your setting of contrast level , mapping to LCD panel's parameter .

[ST2108] – 73 x 24

Contrast		Equivalent Duty	Optimal Bias	R2	
Level	CTR[3:0]			n x R1	$\Omega$ (R1=10k)
<input type="checkbox"/> 1(light)	1111	93.7	10.7	6.7 x	66.8 k
<input type="checkbox"/> 2	1110	78.4	9.9	5.9 x	58.5 k
<input type="checkbox"/> 3	1101	67.4	9.2	5.2 x	52.1 k
<input type="checkbox"/> 4	1100	59.1	8.7	4.7 x	46.9 k
<input type="checkbox"/> 5	1011	52.6	8.3	4.3 x	42.5 k
<input type="checkbox"/> 6	1010	47.4	7.9	3.9 x	38.9 k
<input type="checkbox"/> 7	1001	43.1	7.6	3.6 x	35.7 k
<input type="checkbox"/> 8	1000	39.6	7.3	3.3 x	32.9 k
<input type="checkbox"/> 9	0111	36.6	7.0	3.0 x	30.5 k
<input type="checkbox"/> 10	0110	34.0	6.8	2.8 x	28.3 k
<input type="checkbox"/> 11	0101	31.7	6.6	2.6 x	26.3 k
<input type="checkbox"/> 12	0100	29.8	6.5	2.5 x	24.6 k
<input type="checkbox"/> 13	0011	28.0	6.3	2.3 x	22.9 k
<input type="checkbox"/> 14	0010	26.5	6.1	2.1 x	21.5 k
<input type="checkbox"/> 15	0001	25.1	6.0	2.0 x	20.1 k
<input type="checkbox"/> 16(dark)	0000	24.0	5.9	1.9 x	19.0 k

[ST2108] – 73 x 16

Contrast		Equivalent Duty	Optimal Bias	R2	
Level	CTR[3:0]			n x R1	$\Omega$ (R1=10k)
<input type="checkbox"/> 1(light)	1111	62.4	8.9	4.9 x	49.0 k
<input type="checkbox"/> 2	1110	52.2	8.2	4.2 x	42.3 k
<input type="checkbox"/> 3	1101	44.9	7.7	3.7 x	37.0 k
<input type="checkbox"/> 4	1100	39.4	7.3	3.3 x	32.8 k
<input type="checkbox"/> 5	1011	35.1	6.9	2.9 x	29.2 k
<input type="checkbox"/> 6	1010	31.6	6.6	2.6 x	26.2 k
<input type="checkbox"/> 7	1001	28.8	6.4	2.4 x	23.6 k
<input type="checkbox"/> 8	1000	26.4	6.1	2.1 x	21.4 k
<input type="checkbox"/> 9	0111	24.4	5.9	1.9 x	19.4 k
<input type="checkbox"/> 10	0110	22.7	5.8	1.8 x	17.6 k
<input type="checkbox"/> 11	0101	21.2	5.6	1.6 x	16.0 k
<input type="checkbox"/> 12	0100	19.8	5.5	1.5 x	14.5 k
<input type="checkbox"/> 13	0011	18.7	5.3	1.3 x	13.2 k
<input type="checkbox"/> 14	0010	17.7	5.2	1.2 x	12.0 k
<input type="checkbox"/> 15	0001	16.7	5.1	1.1 x	10.9 k
<input type="checkbox"/> 16(dark)	0000	16.0	5.0	1.0 x	10.0 k



## 21. APPENDIX

### Version 1.2

Page 1 .....Feature Change.....2008/9/17

### Version 1.1

Page 18 Modify PRES 8-bits counter to 16 bit and add description low byte

**PRS[7~0] : The low byte value of PRES counter**

Page 45 Add ST2108 EVB photo

Page 46 Add checklist for customer to confirm ST2108 EVB PCB number.....2007/5/21

### Version 0.16

Page 2 ..... Modify PAD diagram.

### Version 0.15a

Page 34 ..... Change number: '1752 dots', 'seg0~seg72', '\$1000~\$1148' .

Page 36 ..... Line 23 'com1' change to 'com0'.

### Version 0.15

Page 2 ..... Modify PAD diagram.

..... Modify LCD segment size from 72 to 73.

### Version 0.14

Page 1 ..... Add 16x72 LCD & low voltage detector feature

Page 8, 35 ..... Modify 'LCTL', 'SCAN'.

Page 1, 12, 18 ..... Add COMMON-PORT output function.

Page 19 ..... Add low voltage flag description.

Page 36 ..... Add 1/16 duty LCD map.

Page 40 ..... Modify circuit of application 1

### Version 0.161-

Page 35 ..... LCD driver timing chart

### Version 0.162

Page 8 (section 7.3)..... Add DMA registers

Page 30(section 13.2).... Modify DAC sample rate description table

Page 36(table 14-2) ..... Change LCD frame clock control

Page 39(section 16)..... Add low power detector description

### Version 0.163

Page 8 (section 7.3)..... Add DMR, DFIX and some initial value

### Version 0.164

Page 2 (section 3) ..... Modify pad allocation

### Version 0.165

Page 43,44(section 19).. Pad diagram, bonding diagram

### Version 0.17

Page 38(section 15)..... DMA

### Version 0.171

Page 2 ..... Add information to easy locate pad 1 for bonding

### Version 0.2

Application ..... Add 0.1uF cap between Vp and ground

### Version 0.21

Section 5,9..... Correct pad number

### Version 0.3 ..... Operation frequency change from 4MHz to 2MHz

**Version 0.31** .....DAC operation at RC=2MHz (section 13)

**Version 0.32** ..... Update R vs. Fosc vs. Iop figure 18-2/3  
..... Section 5, 9.4, 12.1, 12.2.2, 13.2 and Table13-3

**Version 0.33**

Page1(section 1) .....Add resonator oscillator feature  
Page3(section 5) ..... Remove low power detector adjust pin LVDET  
Page38(section 16.2).....Add notice item  
Page39(section 17).....Low voltage detecto

**Version 0.35**

Page43 .....Add resonator oscillator application.

**Version 0.36**

Page18 .....Modify PRES counter description.

**Version 0.4**

Page16 .....Add one note after TABLE 10-1. **XBAK** should be "0" when normal operation

---

The above information is the exclusive intellectual property of Sitronix Technology Corp. and shall not be disclosed, distributed or reproduced without permission from Sitronix. Sitronix Technology Corp. reserves the right to change this document without prior notice and makes no warranty for any errors which may appear in this document. Sitronix products are not intended for use in life support, critical care, medical, safety equipment, or similar applications where products failure could result in injury, or loss of life, or personal or physical harm, or any military or defense application, or any governmental procurement to which special terms or provisions may apply.