

# ST2104

## 8 BIT Microcontroller with 512K bytes ROM

PRELIMINARY

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

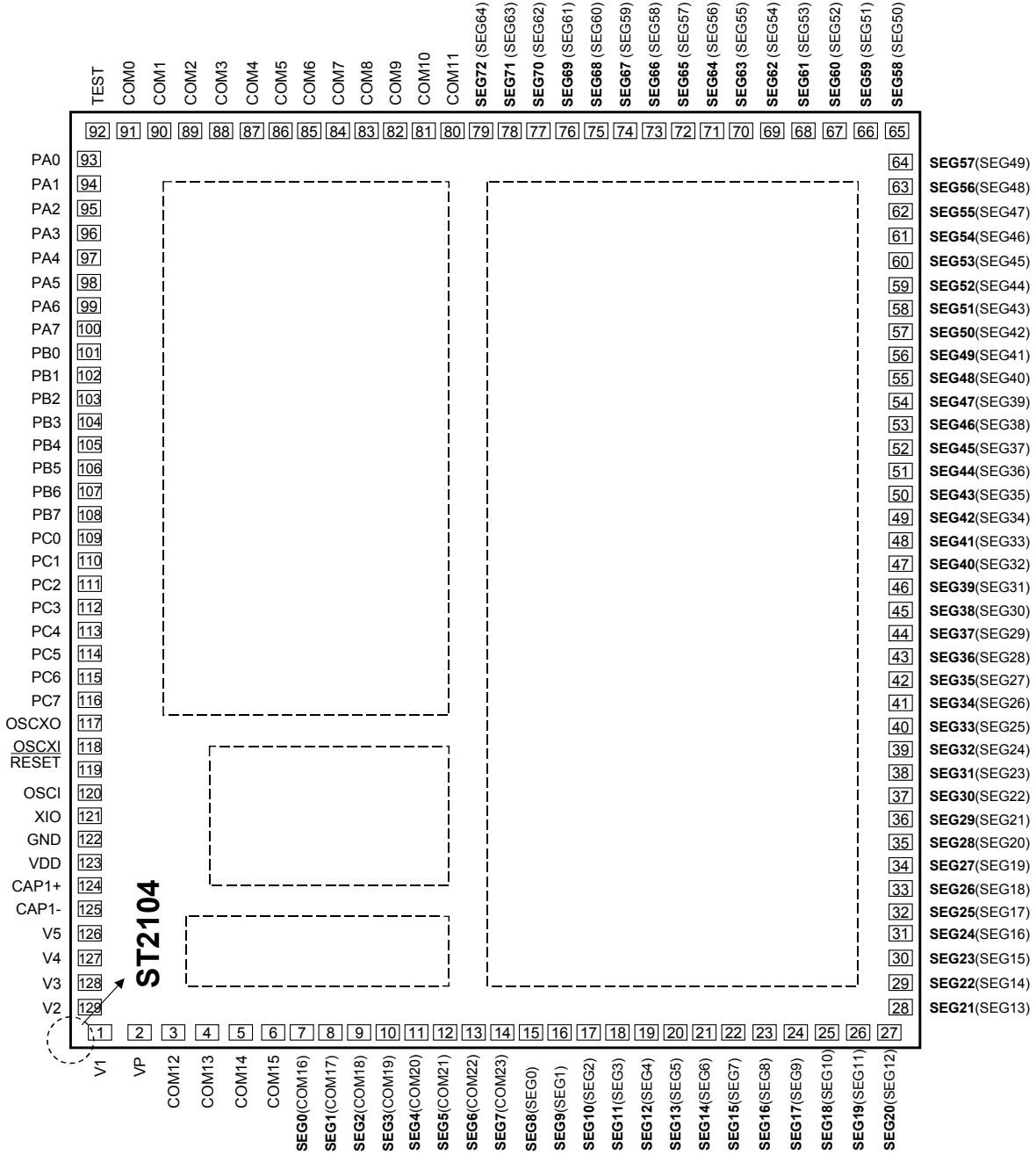
- Totally static 8-bit CPU
- ROM: 512K x 8-bit
- RAM: 4K x 8-bit
- Stack: Up to 128-level deep
- Operation voltage: 2.4V ~ 3.4V
- Operation frequency:
  - 3.0Mhz@2.4V(min.)
  - 4.0Mhz@2.7V(min.)
- Built-in double DC-DC voltage converter for LCD driver
- I/O ports
  - 24 CMOS bidirectional bit programmable I/O pins
  - 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 hardware interrupts with dedicated exception vectors
  - External interrupt (edge triggered)
  - Timer0 interrupt
  - Timer1 interrupt
  - Base timer interrupt
  - Port-A interrupt (transition triggered)
  - DAC reload interrupt
- Dual clock sources with warm-up timer
  - Low frequency crystal oscillator
    - .....32768 Hz
  - RC oscillator ..... 500K ~ 4M Hz
  - High frequency crystal/resonator oscillator (code option)
    - ..... 455K~4M Hz
- Direct memory access (DMA)
  - Block-to-Block move
  - Block to Single port
- LCD controller/driver
  - 16-level contrast control
  - 1168 (73x16) dots (1/16 duty, metal option)
  - 1560 (65x24) dots (1/24 duty, metal option)
- 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

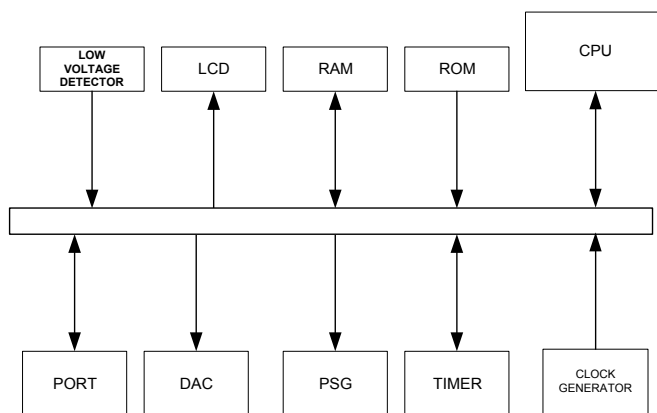
The ST2104 is a W65C02S based 8-bit microcontroller designed with CMOS silicon gate technology. This single chip microcontroller is useful for translator, databank and other consumer applications. It integrates with SRAM, mask ROM,

LCD controller/driver, DC-DC voltage converter, 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



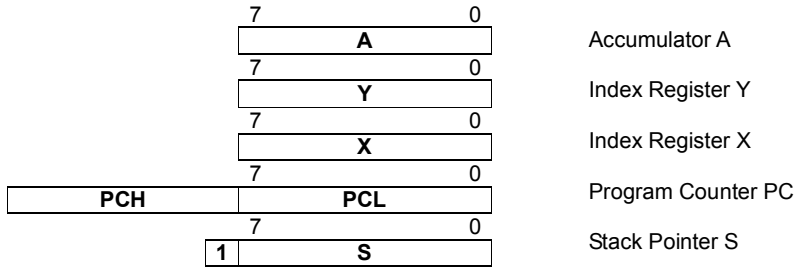
## 5. PAD DESCRIPTION

Pin No.	Designation	I/O	Description
7~14	SEG0(COM16)~ SEG7(COM23)	O	- LCD segment drives 0~7 (1/16 duty mode)
		O	- LCD common drives 16~32 (1/24 duty mode)
15~79	SEG8(SEG0)~ SEG72(SEG64)	O	- LCD segment drives 8~72 (1/16 duty mode)
		O	- LCD segment drives 0~64 (1/24 duty mode)
91~80, 3~6	COM0~11, 12~15	O	LCD common drives 0~15
92	TEST	I	Chip test function. Leave it open.
93	PA0 / INTX	I/O	- Port-A bit programmable I/O
		I	- Edge-trigger Interrupt.
		I	- Transition-trigger Interrupt
		I	- Programmable Timer1 clock source
94~100	PA1~7	I/O	- Port-A bit programmable I/O
		I	- Transition-trigger Interrupt
101,102	PB0, 1	I/O	- Port-B bit programmable I/O
		O	- PSG/DAC Output
103~108	PB2~7	I/O	Port-B bit programmable I/O
109~116	PC0~7	I/O	Port-C bit programmable I/O
117, 118	OSC XO, OSC XI	I/O	Low frequency crystal oscillator I/O pins. Connect to external 32768 Hz crystal.
119	$\overline{\text{RESET}}$	I	Reset signal input (low active)
120	OSCI	I	- RC oscillator input pin. Connected to external resistor
		I	- High frequency crystal/resonator oscillator input pin. Connect to external crystal/resonator.
121	XIO	O	High frequency crystal/resonator oscillator output pin. Connect to external crystal/resonator.
122	GND	P	Ground
123	VCC	P	Power supply
124	CAP1+	I/O	Connect to booster capacitor positive(+) terminal
125	CAP1-	I/O	Connect to booster capacitor negative(-) terminal
1, 129~126	V1, V2~V5	I	Inputs of external power supply for LCD drives
2	VP	O	Voltage output of booster circuit

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

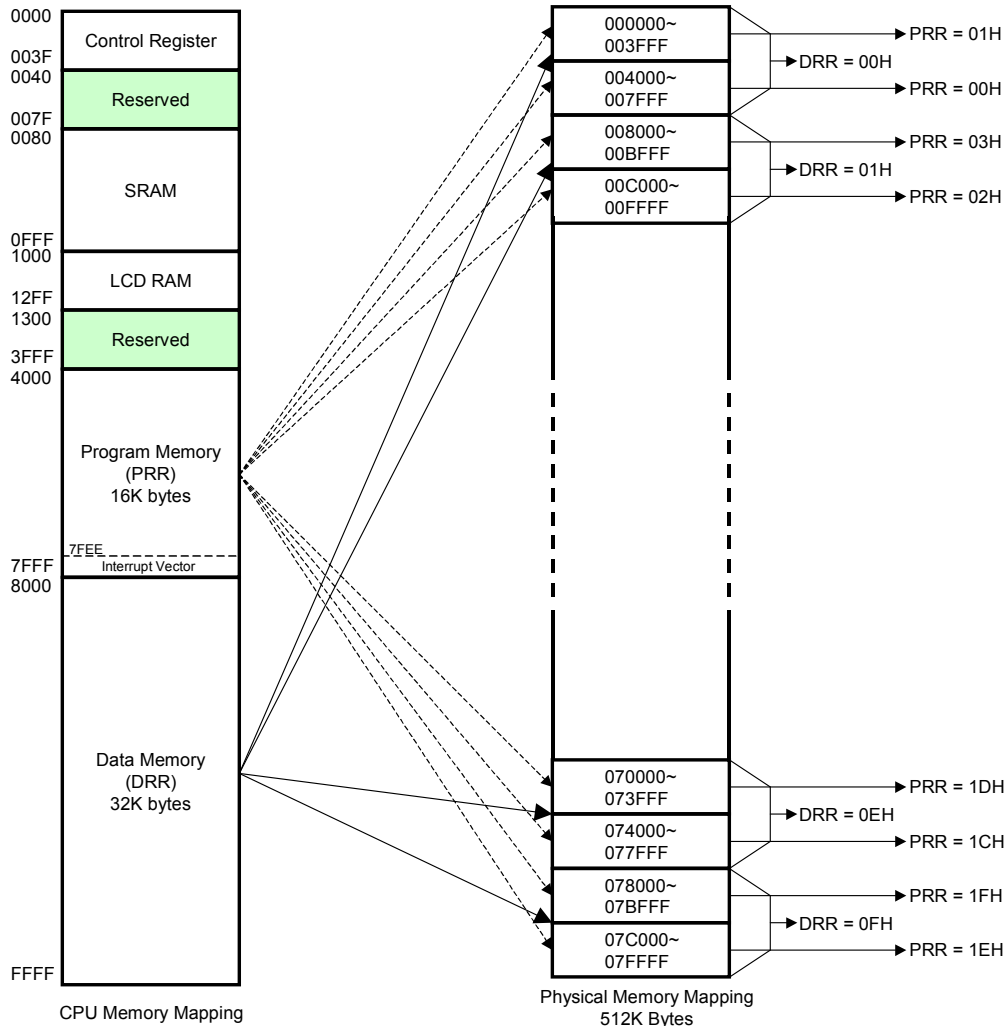
TABLE 6-1 Status Register (P)

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

ST2104 builds in 512K bytes ROM and 4K bytes RAM. The internal ROM can be used as data memory or program memory. PRR is the Program ROM Bank Register and DRR is the Data ROM Bank Register. The logical program ROM address is from \$4000 to \$7FFF(16K bytes), and \$8000 to \$FFFF (32K bytes) is for logical data ROM address.



## 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	-	-	-	PRR4	PRR3	PRR2	PRR1	PRR0
DRR	\$32	RW	-	-	-	-	DRR3	DRR2	DRR1	DRR0

### 7.3 RAM

Internal static RAM is for control registers, data RAM, stack RAM and the LCD frame buffer.

#### 7.3.1 Control 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
\$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	R/W	PSG0[7]	PSG0[6]	PSG0[5]	PSG0[4]	PSG0[3]	PSG0[2]	PSG0[1]	PSG0[0]	0000 0000
\$011	PSG0H	R/W	-	-	-	-	PSG0[11]	PSG0[10]	PSG0[9]	PSG0[8]	---- 0000
\$012	PSG1L	R/W	PSG1[7]	PSG1[6]	PSG1[5]	PSG1[4]	PSG1[3]	PSG1[2]	PSG1[1]	PSG1[0]	0000 0000
\$013	PSG1H	R/W	-	-	-	-	PSG1[11]	PSG1[10]	PSG1[9]	PSG1[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	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	R/W	VOL1[3]	VOL1[2]	VOL1[1]	VOL1[0]	VOL0[3]	VOL0[2]	VOL0[1]	VOL0[0]	0000 0000
\$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	-	-	-	-	-	-
\$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	R/W	DMS[7]	DMS[6]	DMS[5]	DMS[4]	DMS[3]	DMS[2]	DMS[1]	DMS[0]	0000 0000
\$029	DMSH	R/W	DMS[15]	DMS[14]	DMS[13]	DMS[12]	DMS[11]	DMS[10]	DMS[9]	DMS[8]	0000 0000
\$02A	DMDL	R/W	DMD[7]	DMD[6]	DMD[5]	DMD[4]	DMD[3]	DMD[2]	DMD[1]	DMD[0]	0000 0000
\$02B	DMDH	R/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	-	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
\$03A	LCTL	R/W	LPWR	BLANK	REV	-	CTR[3]	CTR[2]	CTR[1]	CTR[0]	000- 0000
\$03B	LCK	R/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. Undefined bytes and bits should not be used and keep it "0".

2. Do not use bit modification instructions for write-only registers, such as RMBx, SMBx.

#### 7.3.2 Data RAM (\$0080~\$0FFF)

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

#### 7.3.3 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.

#### 7.3.4 LCD Frame Buffer (\$1000~\$1148)

LCD frame buffer is accessible by both read/write instructions and the LCD controller. Note that this area can also be used as data memory. Pixels of LCD panels, 73\*24 or 65\*16, are directly mapped into this area. Refer to section 14.3 for the detail mapping.

## 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 make an initialization sequence to begin. After the system has been operating, a low on this line of a least two clock cycles will cease ST2104 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 \$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 are listed in 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					

## 9. I/O PORTS

### 9.1 Description

ST2104 can supply total 32 GPIOs divided into three I/O ports, refer to TABLE 9-1 Port-A, Port-B, and Port-C. For detail pin assignment, please

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

PORT NAME	PAD NAME	PAD NUMBER	PIN TYPE	FEATURE
Port-A	PA0/INTX	93	I/O	Programmable input/output pin
	PA1	94	I/O	
	PA2	95	I/O	
	PA3	96	I/O	
	PA4	97	I/O	
	PA5	98	I/O	
	PA6	99	I/O	
	PA7	100	I/O	
Port-B	PB0	101	I/O	Programmable input/output pin
	PB1	102	I/O	
	PB2	103	I/O	
	PB3	104	I/O	
	PB4	105	I/O	
	PB5	106	I/O	
	PB6	107	I/O	
	PB7	108	I/O	
Port-C	PC0	109	I/O	Programmable input/output pin
	PC1	110	I/O	
	PC2	111	I/O	
	PC3	112	I/O	
	PC4	113	I/O	
	PC5	114	I/O	
	PC6	115	I/O	
	PC7	116	I/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 also provides bit programmable pull-up resistors for each input pin. Two interrupts can be

triggered by Port-A, de-bounced interrupt for keyboard scan and edge sensitive interrupt (PA0 only) for external event.

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. Each bit of PCA controls the direction of one single I/O of Port-A respectively,

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

### 9.2.3 Port-A Pull-Up Option

Port-A contains PMOS transistors of pull-up resistor controlled by software in bit-manner. In case of input direction, on/off of the pull-up PMOS transistor is controlled by the data wrote to data register, PA. "1" is for enable and "0" is for disable. Above all, whole pull-up control is by PULL bit of PMCR. Refer to FIGURE 9-1 for the block description.

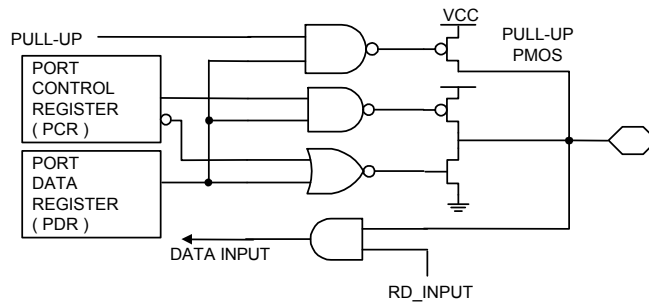


FIGURE 9-1 Port-A Block Diagram

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

Bit 6: **PDBN** : Enable Port-A interrupt de-bounce bit  
 1 = De-bounce for Port-A interrupt  
 0 = No de-bounce 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 is suitable for the return line inputs of keyboard scan because of the port transition interrupt function. Difference between current value and the data kept previously of Port-A will generate an interrupt request. The last state of Port-A must be latched before transition, and this can be done by one read

instruction to Port-A. If both INTX and PT interrupts are enabled, signal edge of PA0 may trigger PT interrupt as well as INTX. Steps and program example are shown below. Also refer to FIGURE 9-2 for the block diagram.

#### 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 ISR

#### Example:

```

.
.
STZ <PCA      ; Set input mode.
LDA  #$FFF
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

```

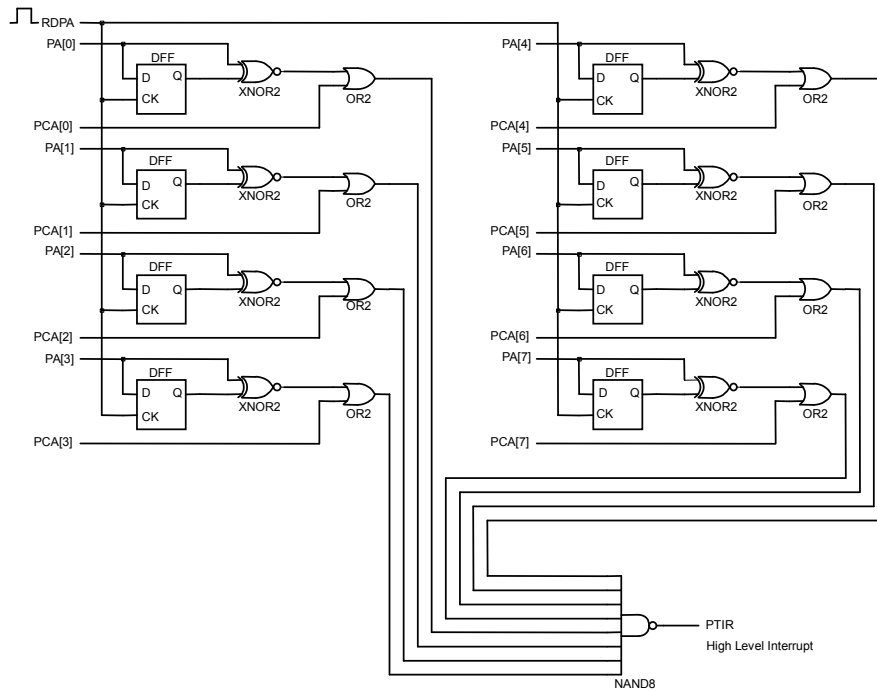


FIGURE 9-2 Port Interrupt Logic Diagram

**9.2.5 Port-A Interrupt De-bounce**

ST2104 has hardware de-bounce block for Port-A interrupt. It is enabled with "1" and disable with "0" of PDBN(PMCR[6]). The de-bounce function is activated by Port-A transition. It uses

OSC<sub>X</sub> as the sampling clock. The de-bounce time is **OSC<sub>X</sub> x 512 cycles (about 16 ms)**. Data filtered by de-bounce presents a stable state, then the interrupt can be issued.

**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: **PDBN** : 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 plays another function of external edge-sensitive interrupt source. Falling or rising edge is controlled by INTEG(PMCR[5]). Please refer to FIGURE 9-3. If both INTX and PT interrupts are

enabled, signal edge of PA0 may trigger PT interrupt as well as INTX. Steps and program example are shown below.

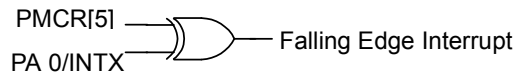
**Steps for INTX interrupt operation:**

1. Set PA0 to 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 ports, controlled by PCB and PCC registers. There is also bit

programmable pull-up resistor for each input pin. PB[1:0] also have the function of PWM output for sound generation.

**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/PCC controls the I/O direction of Port-B/C. Each bit of PCB[7~0]/PCC[7~0] controls the direction of one single bit of

Port-B/C respectively, with “1” for output mode, and “0” for input mode.

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

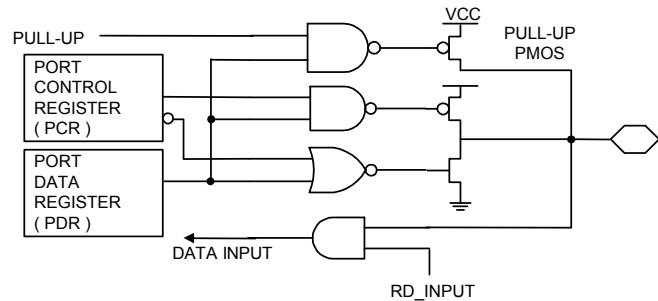
Address	Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default
\$009	PCB	R/W	PCB[7]	PCB[6]	PCB[5]	PCB[4]	PCB[3]	PCB[2]	PCB[1]	PCB[0]	0000 0000
Bit 7~0: <b>PCB[7~0]</b> : Port-B directional bits 1 = Output mode 0 = Input mode											

**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**

Port-B/C contains PMOS transistors of pull-up resistor controlled by software in bit-manner. In case of input direction, on/off of the pull-up PMOS transistor is controlled by the data wrote to data register, PB/PC. "1" is for enable and "0" is for disable. Above all, whole pull-up control is by PULL bit of PMCR. Refer to FIGURE 9-4 for the block description.



**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>											

## 10. OSCILLATOR

ST2104 has dual clock sources, OSC (RC) and OSCX (32768Hz crystal). The system clock (SYSCK) can be switched between OSC and OSCX, and is controlled by XSEL (SYS[7]). When system clock is switched, the warm-up cycles occur at the same time. Clock source being used is shown at

XSEL (read). Read and test XSEL to confirm SYSCK is already switched over. Other blocks, such as LCD controller, 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	-	WSKP	WAIT	-	LVDET	0000 00-0

Bit 7: **XSEL** : System clock (SYSCK) 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

**Note:**

1. XSEL (SYS[7]) shows which clock source is used for SYSCK when it is read.
2. System warm-up of 16 or 256 oscillation cycles occurs when system clock (SYSCK) is changed or power on reset.

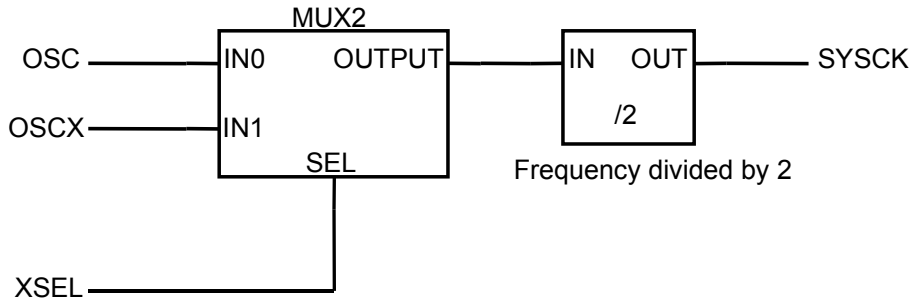


FIGURE 10-1 System Clock Diagram



# 11. TIMER/EVENT COUNTER

## 11.1 Prescaler

### 11.1.1 Function Description

The ST2104 has three timers, Base timer, Timer 0 and Timer 1, and two prescalers PRES and PREW. There are two clock sources, SYSCK and INTX, 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	-	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

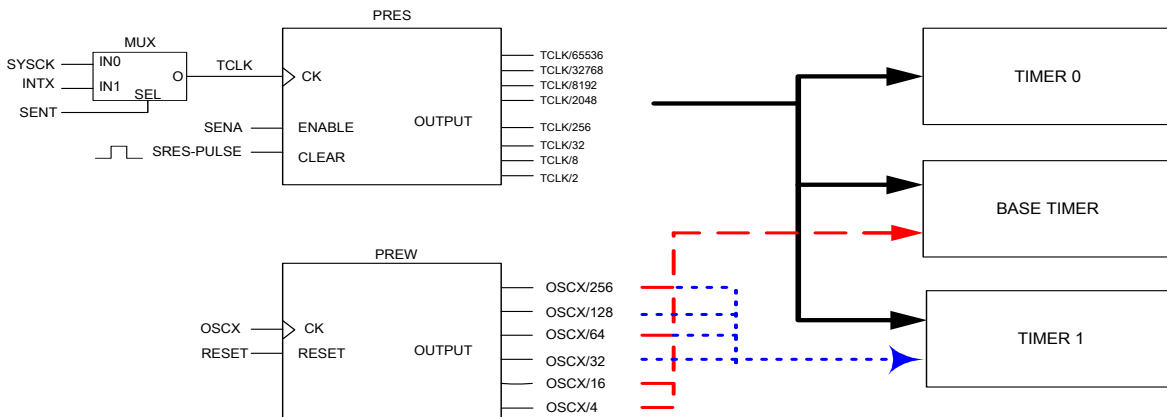


FIGURE 11-1 Structure Of Two Prescalers

11.1.2 PRES

The prescaler PRES is an 16-bits counter as shown in FIGURE 11-1. 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 prescaler 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 11-1. 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

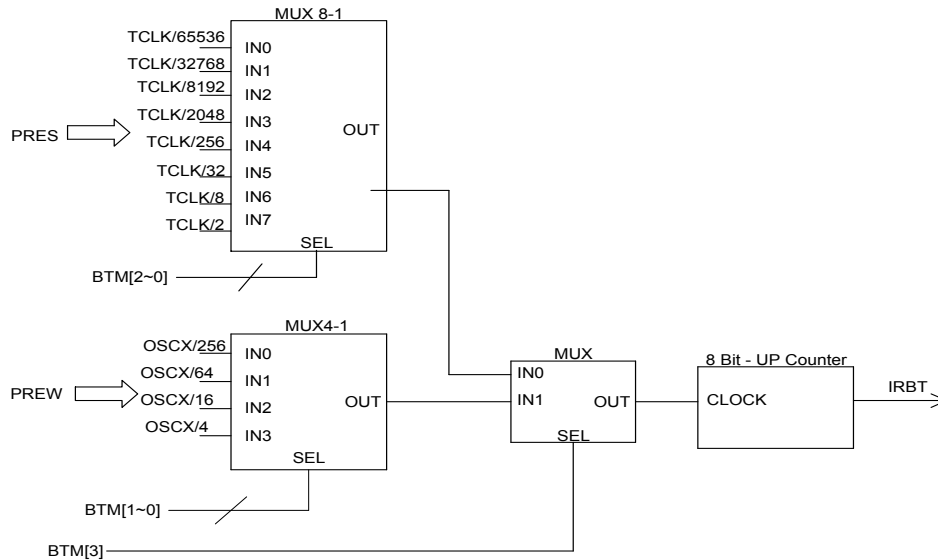


FIGURE 11-2 Structure Of Base Timer

### 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.

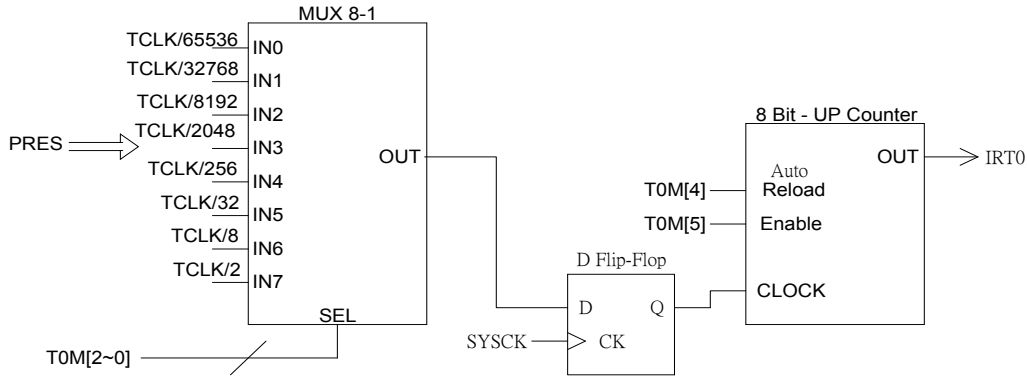


FIGURE 11-3 Timer0 Structure

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.

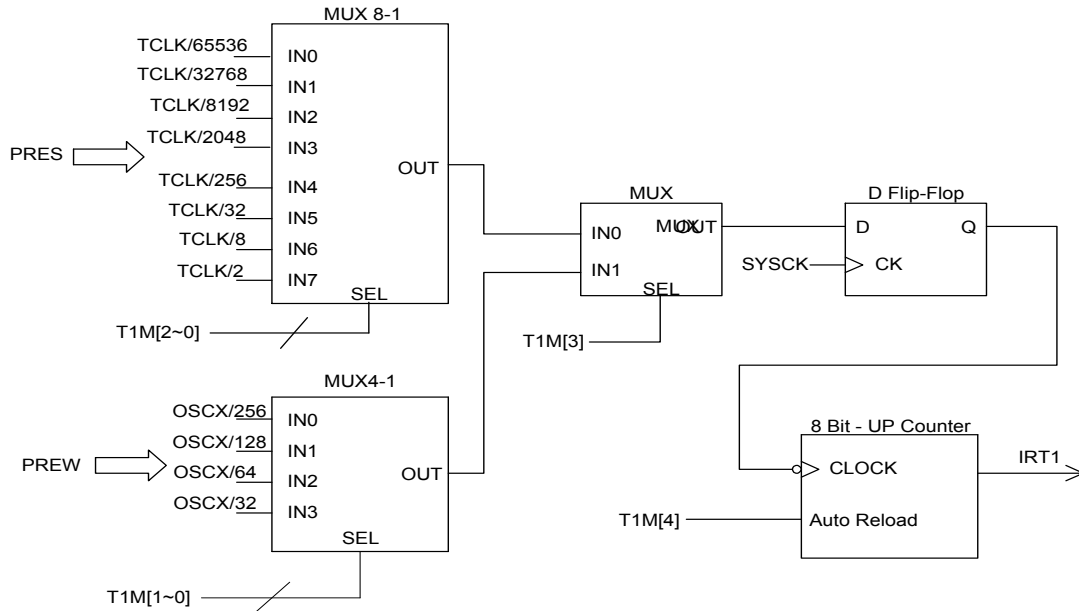


FIGURE 11-4 Timer1 Structure

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 ST2104 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.

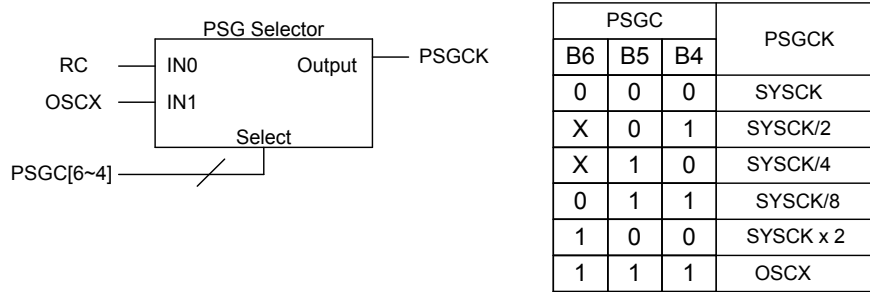


FIGURE 12-1 PSG Clock Source Control

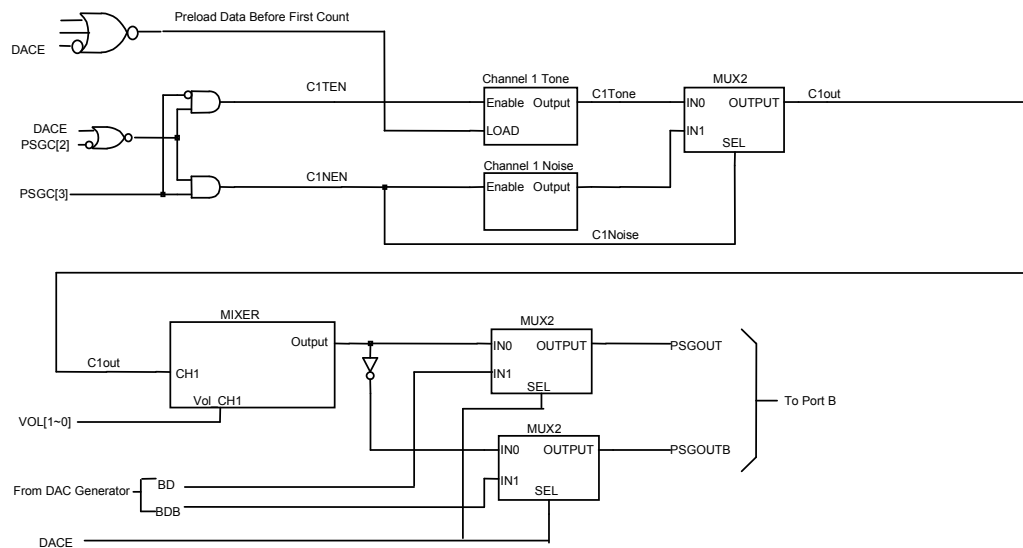


FIGURE 12-2 PSG Block Diagram

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>											

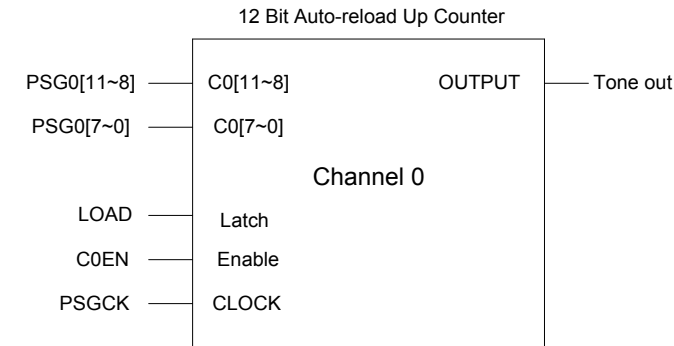
TABLE 12-3 PSG Volume Control Register (VOL)

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

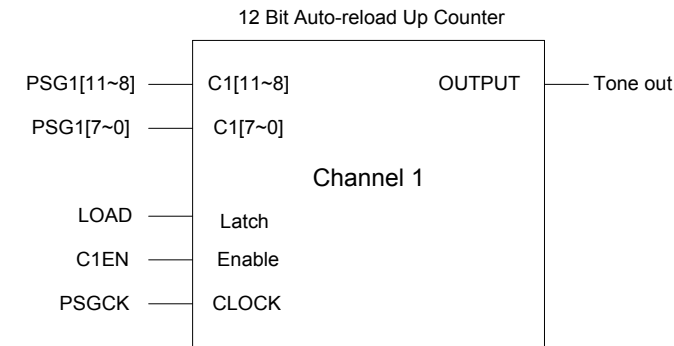
### 12.2.1 General Description

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.



$$\text{Frequency of Channel 0 Tone} = \text{PSGCK} / (1000\text{H} - \text{PSG0}[11\sim0]) / 2$$

FIGURE 12-3 Tone Generator Channel 0



$$\text{Frequency of Channel 1 Tone} = \text{PSGCK} / (1000\text{H} - \text{PSG1}[11\sim0]) / 2$$

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

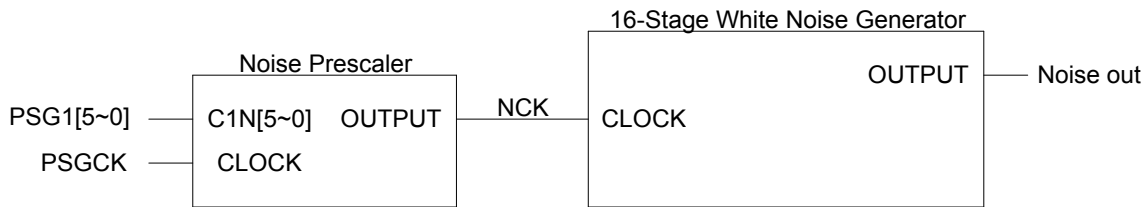
Bit 3: **PRBS** : Tone or Noise generator selection bit  
 1 = Noise generator  
 0 = Tone generator

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 less 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	-	-	-	PSG0	PSGB	100 - - - 00
\$012	PSG1L	W	PSG1[7]	PSG1[6]	PSG1[5]	PSG1[4]	PSG1[3]	PSG1[2]	PSG1[1]	PSG1[0]	00000000
\$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]	00000000
\$016	PSGC	W	-	PCK[2]	PCK[1]	PCK[0]	PRBS	C1EN	C0EN	DACE=0	- 00000 - 0
		W	-	PCK[2]	PCK[1]	PCK[0]	DMD[1]	DMD[0]	INH	DACE=1	- 0000000

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

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

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

TABLE 13-3 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

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  
 X01 = SYSCK / 2  
 X10 = SYSCK / 4  
 011 = SYSCK / 8  
 100 = SYSCK x 2 (= frequency of RC oscillator)  
 111 = OSCX

**Note: In DAC mode, PSGCK must select SYSCK x 2 (PCK[2~0]=100) under RC=2MHz.**

### 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:

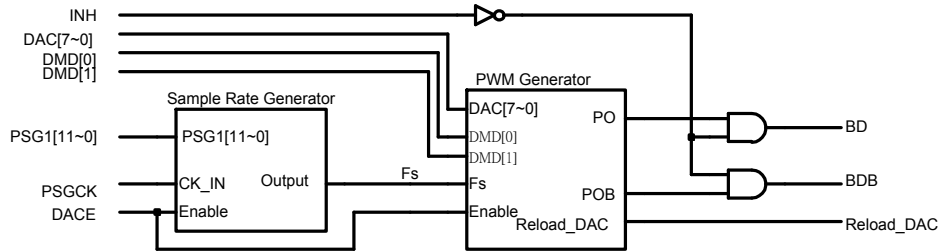


FIGURE 13-1 DAC Diagram

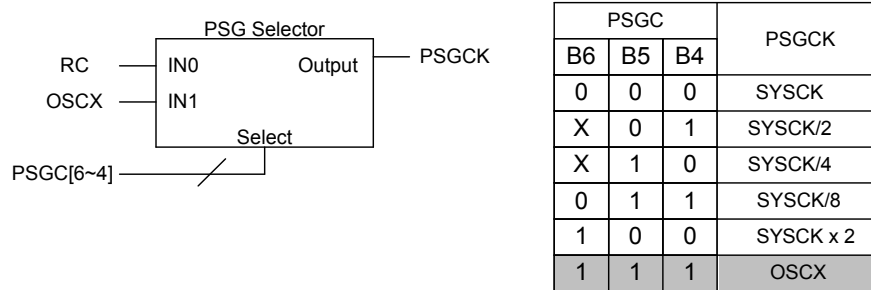


FIGURE 13-2 DAC Clock Source Control

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
16K	32K	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.

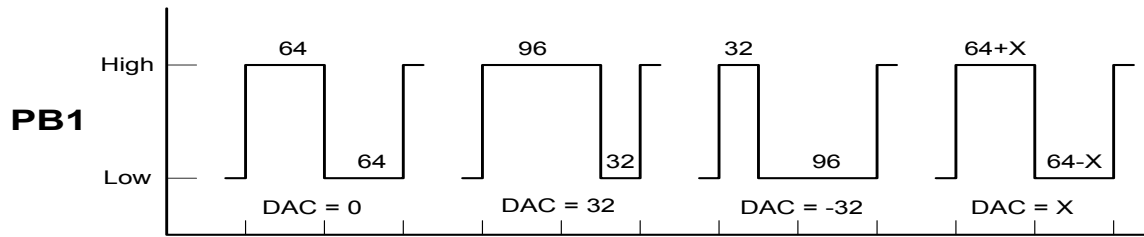


FIGURE 13-3 Single-Pin Mode Wave Form

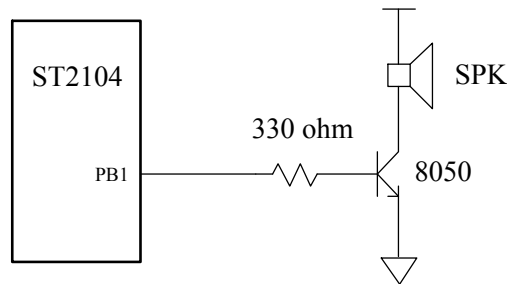
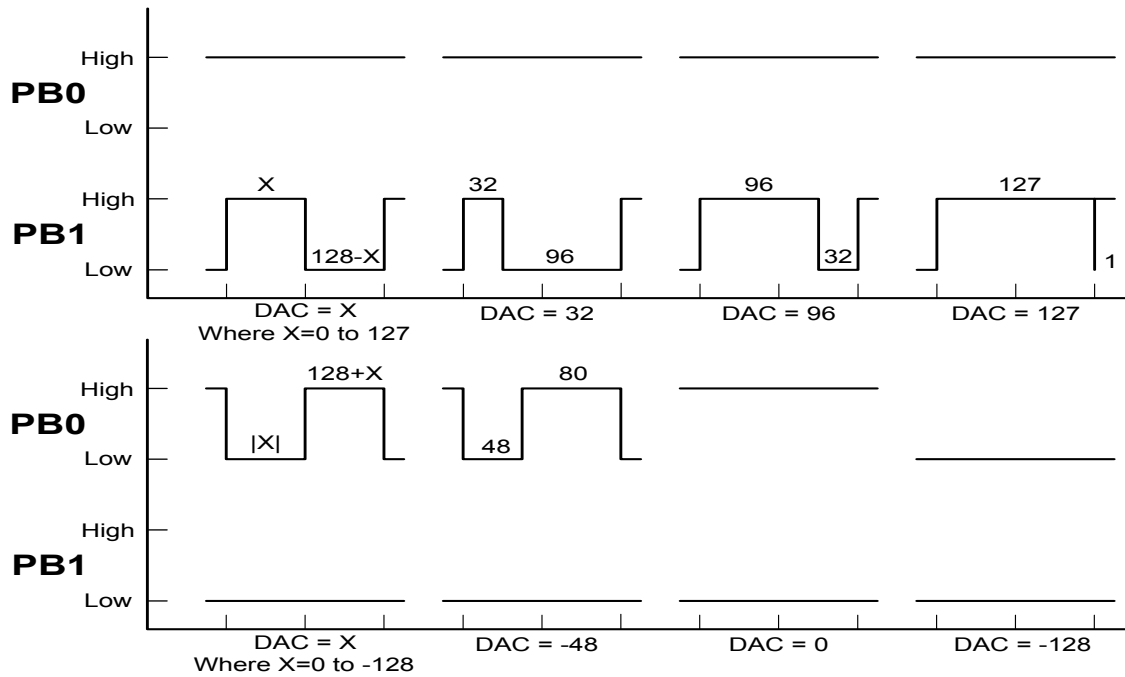


FIGURE 13-4 Single-Pin Mode Application Circuit

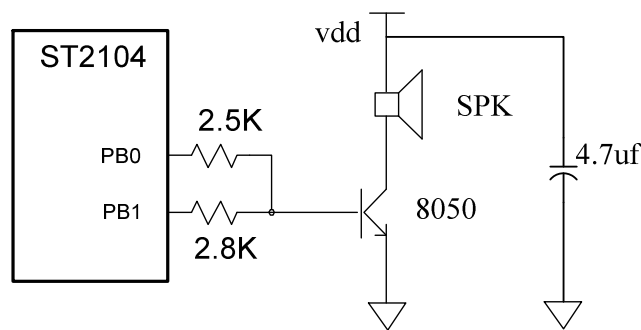
**13.3.2 Two-Pin Two Ended Mode (8-bit Accuracy)**

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.

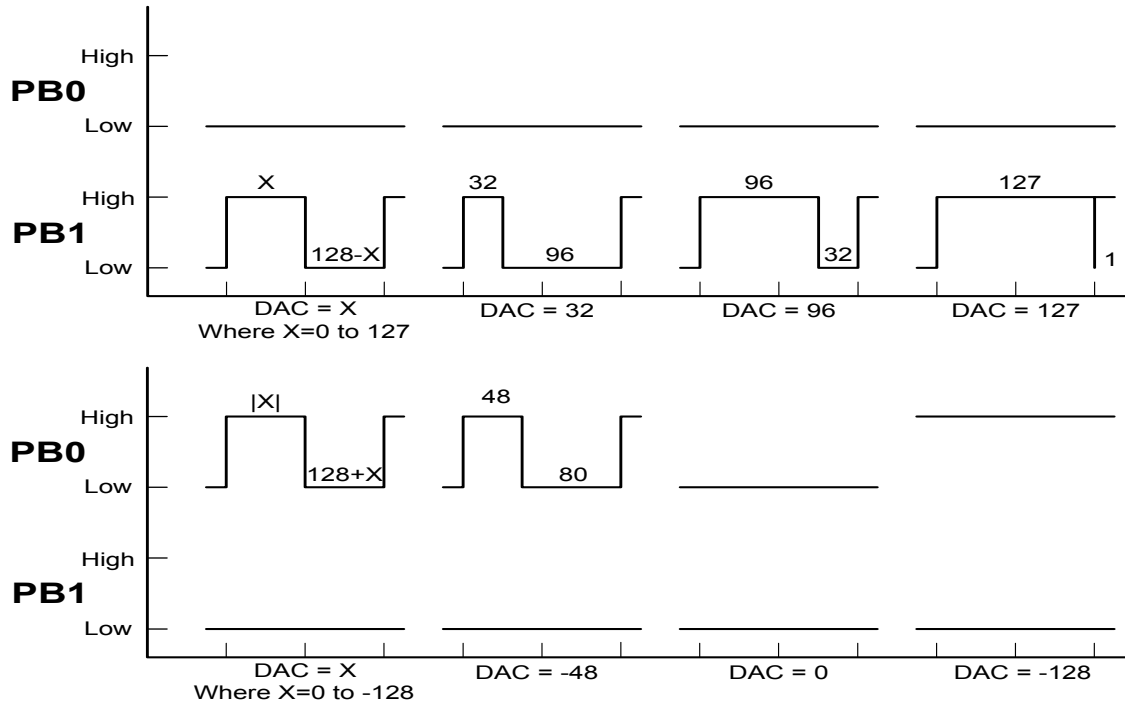


FIGURE 13-7 Two-Pin Push Pull Mode Wave Form

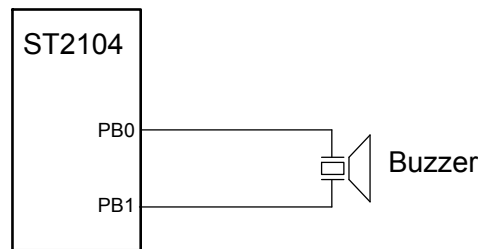


FIGURE 13-8 Two-Pin Push Pull Mode Application Circuit

## 14. LCD

The ST2104 is capable of driving up to 1560 dots of LCD panel directly. It supports two kinds of duty: 1/24 and 1/16. LCD block includes display frame buffer (\$1000~ \$1148) for storing the display data, and 89 LCD drives for different panels: 73x16 and 65x24.

Data in frame buffer is undefined after power on, therefore correct frame data should be filled in before turn on display. Contrast control is supported by software. The initial value is [0000] (maximum).

**TABLE 14-1 Driver Output Levels**

Driver	Mode	Alternation	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

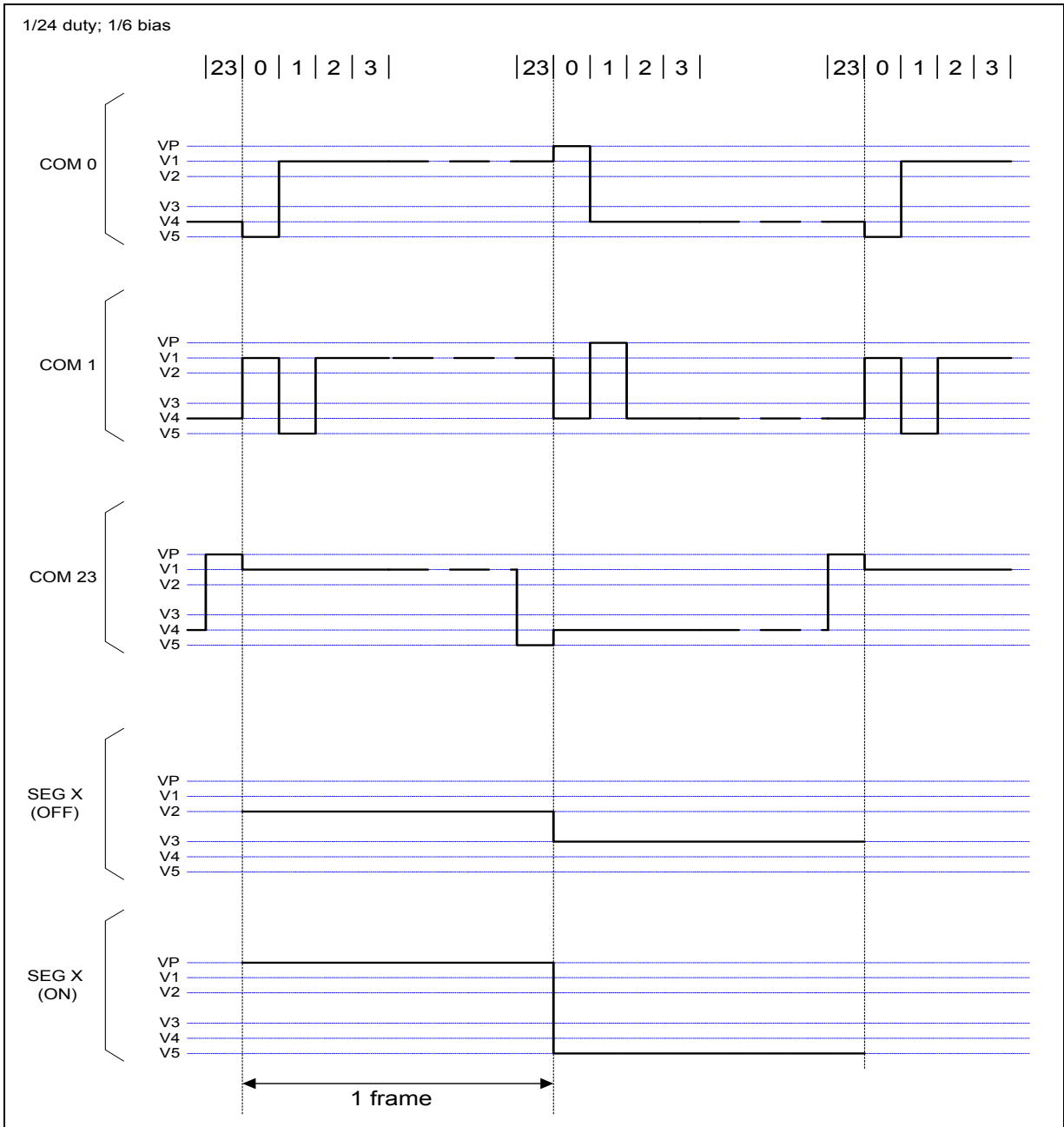


FIGURE 14-1 LCD Timing



## 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	-	CTR[3]	CTR[2]	CTR[1]	CTR[0]	0000 0000

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: **REV** : LCD display reverse  
 1 = Reverse display  
 0 = Normal display

Bit 3~0: **CTR[3~0]** : LCD contrast control

DUTY = 1/16		DUTY = 1/24	
0000 = contrast level	16 (maximum)	0000 = contrast level	14
0001 = contrast level	15	0001 = contrast level	14
0010 = contrast level	14	0010 = contrast level	14 (maximum)
0011 = contrast level	13	0011 = contrast level	13
0100 = contrast level	12	0100 = contrast level	12
0101 = contrast level	11	0101 = contrast level	11
0110 = contrast level	10	0110 = contrast level	10
0111 = contrast level	9	0111 = contrast level	9
1000 = contrast level	8	1000 = contrast level	8
1001 = contrast level	7	1001 = contrast level	7
1010 = contrast level	6	1010 = contrast level	6
1011 = contrast level	5	1011 = contrast level	5
1100 = contrast level	4	1100 = contrast level	4
1101 = contrast level	3	1101 = contrast level	3
1110 = contrast level	2	1110 = contrast level	2
1111 = contrast level	1 (minimum)	1111 = contrast level	1 (minimum)

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

Bit 2~0 : **LCK[2~0]** : LCD frame clock control

DUTY = 1/16		DUTY = 1/24	
000 = RC / 64 ( 2M/16/80/64 = 24.4 Hz )		000 = RC / 64 ( 2M/24/72/64 = 18.1 Hz )	
001 = RC / 32 ( 2M/16/80/32 = 48.8 Hz )		001 = RC / 32 ( 2M/24/72/32 = 36.2 Hz )	
010 = RC / 16 ( 2M/16/80/16 = 97.7 Hz )		010 = RC / 16 ( 2M/24/72/16 = 72.3 Hz )	
011 = RC / 8 ( 2M/16/80/8 = 195.3Hz )		011 = RC / 8 ( 2M/24/72/8 = 144.7 Hz )	
100 = RC / 4 ( 2M/16/80/4 = 390.6 Hz )		100 = RC / 4 ( 2M/24/72/4 = 289.4 Hz )	
101 = X		101 = X	
110 = X		110 = X	
111 = X		111 = X	

\* Under RC = 2M Hz condition.

### 14.3 Frame Buffer

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 Frame Buffer Memory Mapping**

	SEG0	SEG1	SEG2	SEG3	SEG4	SEG5		SEG64	User RAM
<b>Address</b>	<b>1000H</b>	<b>1001H</b>	<b>1002H</b>	<b>1003H</b>	<b>1004H</b>	<b>1005H</b>		<b>1040H</b>	<b>1041H ~ 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>10C0H</b>	<b>10C1H ~ 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>1140H</b>	<b>1141H ~ 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 Frame Buffer 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 ~ 1148H</b>							
<b>User RAM</b>								

Note: Can not use undefined RAM area (\$1x49~\$1x7F, \$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.**

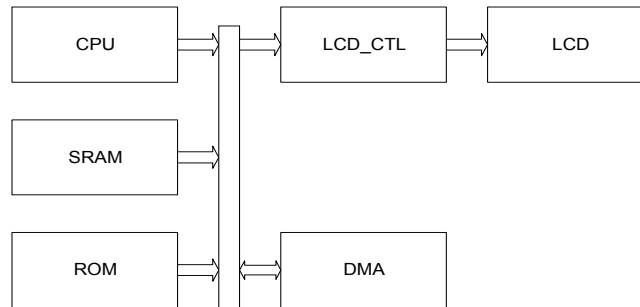


FIGURE 15-1 System Block Diagram

### 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

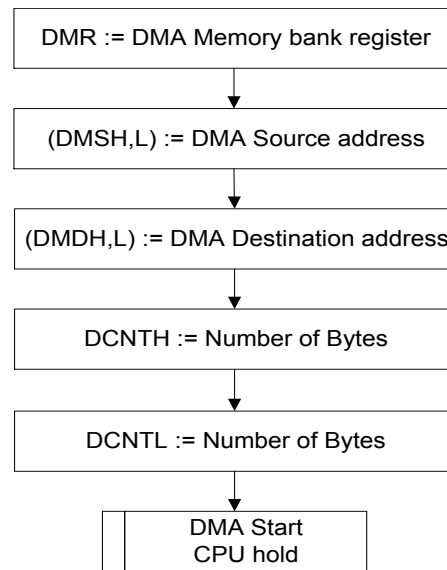


FIGURE 15-2 DMA Programming Flow

## 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

ST2104 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	<b>SYS</b>	R/W	XSEL	OSTP	XSTP	-	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:

```
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.

\*\*\*notice:

If high frequency was optioned to crystal mode (not RC oscillator), system clock need to be switched to low

frequency (32768Hz) before into WAI1 mode.

```
smb7    <SYS    ;; Change to OSCX
bbr7    <SYS,$  ;; Wait OSCX stable
smb6    <SYS    ;; OSC OFF
wai                    ;; sleep
```

```
;;after wake up.
rmb6    <SYS    ;; OSC ON.
jsr     WAIT_8ms ;; For OSC stable.
rmb7    <SYS    ;; Change to OSC
bbs7    <SYS,$  ;; Wait OSC stable.
```

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

ST2104 has a built-in low voltage detector for power management. 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: first read will enable initial stableness control.

Second read 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,$+3
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	-	WSKP	WAIT	-	LVDET	0000 00-0
Bit 0: <b>LVDET</b> : 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: VCC = 3.0V, GND = 0V, T<sub>A</sub> = 25°C, OSC = 2M Hz, unless otherwise specified

Parameter	Symbol	Min.	Typ.	Max.	Unit	Condition
Operating Voltage	VCC	2.4		3.4	V	Built-in double DC-DC voltage converter for LCD driver:
Operating Current	I <sub>OP</sub>		1046		μA	All I/O port are input and pull-up, execute NOP instruction, LCD on
Standby Current	I <sub>SB0</sub>		1		μA	All I/O port are input and pull-up, OSCX off, LCD off (WAIT1/STOP mode)
Standby Current	I <sub>SB1</sub>		2		μA	All I/O port are input and pull-up, OSCX on, LCD off (WAIT1/STOP mode)
Standby Current	I <sub>SB2</sub>		78		μA	All I/O port are input and pull-up, OSCX off, LCD off (WAIT0 mode)
LCD consumption	I <sub>LCD</sub>		399		μA	Bias resistor = 10k, LCD size=1.5cm x 4.5cm
Input High Voltage	V <sub>IH</sub>	0.7V <sub>CC</sub>		V <sub>CC</sub> +0.3	V	PORT A, PORT B, PORT C
		0.85V <sub>CC</sub>			V	$\overline{\text{RESET}}$ , $\overline{\text{INT}}$
Input Low Voltage	V <sub>IL</sub>	GND-0.3		0.3V <sub>CC</sub>	V	PORT A, PORT B, PORT C
					0.15V <sub>CC</sub>	V
Pull-up resistance	R <sub>IH</sub>		250		KΩ	PORTA, PORTB, PORT C (I = -6uA, V <sub>IH</sub> =0.3V <sub>CC</sub> ).
Output high voltage	V <sub>OH1</sub>	0.7V <sub>CC</sub>			V	PORTA, PORTB, PORT C (I <sub>OH</sub> = -3mA).
Output low voltage	V <sub>OL1</sub>			0.3V <sub>CC</sub>	V	PORTA, PORTB, PORT C (I <sub>OL</sub> = 6mA).
Output high voltage	V <sub>OH2</sub>	0.7V <sub>CC</sub>			V	PB0/1 as PSG/DAC, I <sub>OH</sub> = -4mA.
Output low voltage	V <sub>OL2</sub>			0.3V <sub>CC</sub>	V	PB0/1 as PSG/DAC, I <sub>OL</sub> = 8mA.
Output high voltage	V <sub>OH3</sub>	2.8			V	SEGx, I <sub>oh</sub> = -800uA, C=50P, rise time < 200ns
Output low voltage	V <sub>OL3</sub>			0.2	V	SEGx, I <sub>ol</sub> = 800uA
Output high voltage	V <sub>OH5</sub>	0.7V <sub>CC</sub>			V	COM0~7, I <sub>oh</sub> = -1 mA.
Output low voltage	V <sub>OL5</sub>			0.3V <sub>CC</sub>	V	COM0~7, I <sub>ol</sub> = 1 mA.
Oscillation start time	T <sub>STT</sub>		1	2	S	
Frequency stability	Δ F / F			1	PPM	[F(3.0)-F(2.5)]/F(3.0)(crystal oscillator)
Frequency variation	Δ F / F	-10	3	10	PPM	C1= 15 - 30P.
Low voltage detector current	I <sub>LVD</sub>		147		uA	No detector voltage adjustment
Low voltage detector voltage range	V <sub>LVD</sub>	2.4	2.55	2.7	V	Detect V <sub>CC</sub> level.

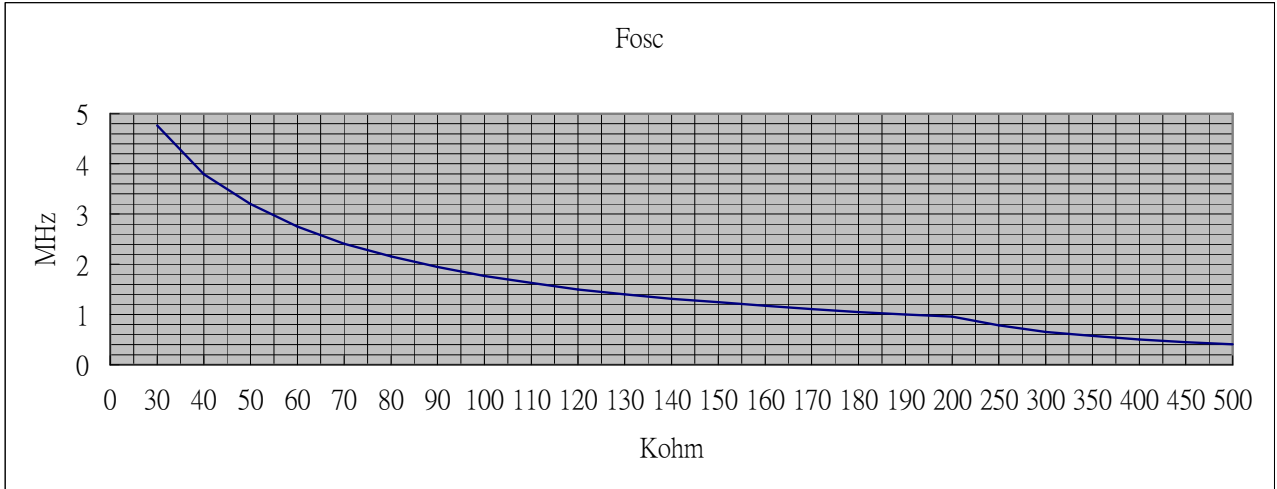


FIGURE 18-1 Oscillation Resistor V.S. Frequency

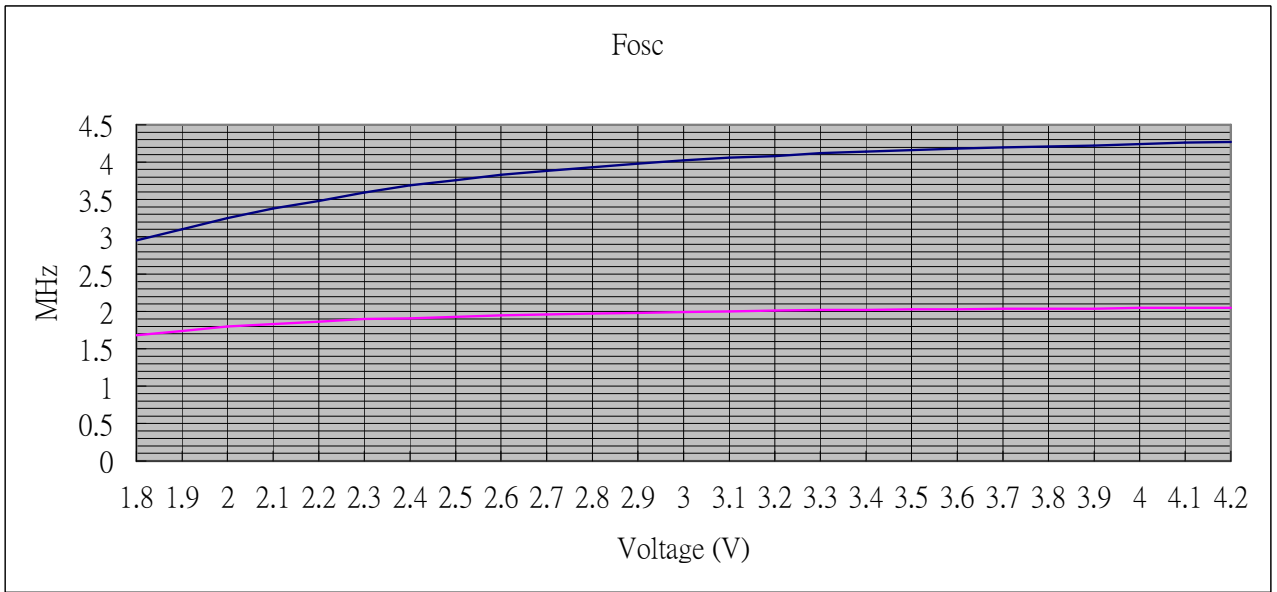
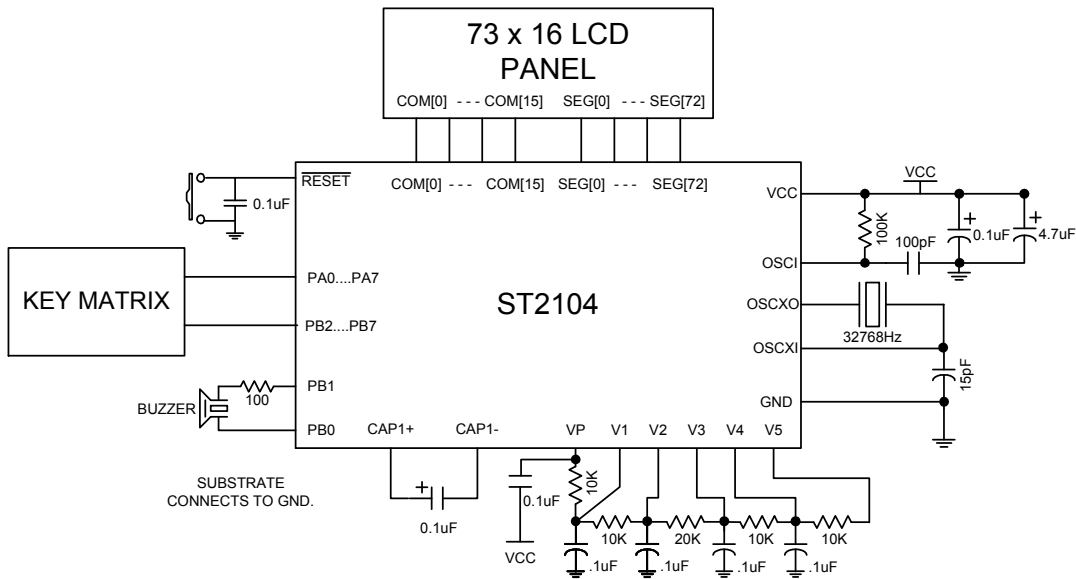


FIGURE 18-2 Operation Voltage VS. Operation Frequency

## 19. APPLICATION CIRCUITS

### 19.1 APPLICATION 1:

VCC : 3.0V  
 CLOCK : RC 2M Hz and crystal 32768 Hz  
 LCD : 73 x 16  
 KEY : 48 key



ST2104 EVB PCB113-1



FIGURE 19-1 The PCB 113-1 of ST2104 EVB

## 20. PAD CENTER COORDINATES

- Chip size: 3270μm x 4440μm
- Coordinate: Pad center (μm)
- Origin: Chip center
- Pad pitch: 110μm, 120μm
- Substrate connection: GND

Unit: μm

Pad No.	Name	X	Y	Pad No.	Name	X	Y	Pad No.	Name	X	Y
1	V1	-1485.7	-2139.7	44	SEG37(SEG29)	1554.8	-219.7	87	COM4	-935.2	2140.3
2	VP	-1340.4	-2139.7	45	SEG38(SEG30)	1554.8	-109.7	88	COM3	-1045.2	2140.3
3	COM12	-1220.4	-2139.7	46	SEG39(SEG31)	1554.8	0.3	89	COM2	-1165.2	2140.3
4	COM13	-1100.4	-2139.7	47	SEG40(SEG32)	1554.8	110.3	90	COM1	-1285.2	2140.3
5	COM14	-980.4	-2139.7	48	SEG41(SEG33)	1554.8	220.3	91	COM0	-1405.2	2140.3
6	COM15	-870.4	-2139.7	49	SEG42(SEG34)	1554.8	330.3	92	TEST	-1525.2	2140.3
7	SEG0(COM16)	-760.4	-2139.7	50	SEG43(SEG35)	1554.8	440.3	93	PA0	-1555.2	2020.3
8	SEG1(COM17)	-650.4	-2139.7	51	SEG44(SEG36)	1554.8	550.3	94	PA1	-1555.2	1900.3
9	SEG2(COM18)	-540.4	-2139.7	52	SEG45(SEG37)	1554.8	660.3	95	PA2	-1555.2	1780.3
10	SEG3(COM19)	-430.4	-2139.7	53	SEG46(SEG38)	1554.8	770.3	96	PA3	-1555.2	1660.3
11	SEG4(COM20)	-320.4	-2139.7	54	SEG47(SEG39)	1554.8	880.3	97	PA4	-1555.2	1540.3
12	SEG5(COM21)	-210.4	-2139.7	55	SEG48(SEG40)	1554.8	990.3	98	PA5	-1555.2	1430.3
13	SEG6(COM22)	-100.4	-2139.7	56	SEG49(SEG41)	1554.8	1100.3	99	PA6	-1555.2	1320.3
14	SEG7(COM23)	9.6	-2139.7	57	SEG50(SEG42)	1554.8	1210.3	100	PA7	-1555.2	1210.3
15	SEG8(SEG0)	119.6	-2139.7	58	SEG51(SEG43)	1554.8	1320.3	101	PB0	-1555.2	1100.3
16	SEG9(SEG1)	229.6	-2139.7	59	SEG52(SEG44)	1554.8	1430.3	102	PB1	-1555.2	990.3
17	SEG10(SEG2)	339.6	-2139.7	60	SEG53(SEG45)	1554.8	1540.3	103	PB2	-1555.2	880.3
18	SEG11(SEG3)	449.6	-2139.7	61	SEG54(SEG46)	1554.8	1660.3	104	PB3	-1555.2	770.3
19	SEG12(SEG4)	559.6	-2139.7	62	SEG55(SEG47)	1554.8	1780.3	105	PB4	-1555.2	660.3
20	SEG13(SEG5)	669.6	-2139.7	63	SEG56(SEG48)	1554.8	1900.3	106	PB5	-1555.2	550.3
21	SEG14(SEG6)	779.6	-2139.7	64	SEG57(SEG49)	1554.8	2020.3	107	PB6	-1555.2	440.3
22	SEG15(SEG7)	889.6	-2139.7	65	SEG58(SEG50)	1524.8	2140.3	108	PB7	-1555.2	330.3
23	SEG16(SEG8)	999.6	-2139.7	66	SEG59(SEG51)	1404.8	2140.3	109	PC0	-1555.2	220.3
24	SEG17(SEG9)	1119.6	-2139.7	67	SEG60(SEG52)	1284.8	2140.3	110	PC1	-1555.2	110.3
25	SEG18(SEG10)	1239.6	-2139.7	68	SEG61(SEG53)	1164.8	2140.3	111	PC2	-1555.2	0.3
26	SEG19(SEG11)	1359.6	-2139.7	69	SEG62(SEG54)	1044.8	2140.3	112	PC3	-1555.2	-109.7
27	SEG20(SEG12)	1479.6	-2139.7	70	SEG63(SEG55)	934.8	2140.3	113	PC4	-1555.2	-219.7
28	SEG21(SEG13)	1554.8	-2019.7	71	SEG64(SEG56)	824.8	2140.3	114	PC5	-1555.2	-329.7
29	SEG22(SEG14)	1554.8	-1899.7	72	SEG65(SEG57)	714.8	2140.3	115	PC6	-1555.2	-439.7
30	SEG23(SEG15)	1554.8	-1779.7	73	SEG66(SEG58)	604.8	2140.3	116	PC7	-1555.2	-549.7
31	SEG24(SEG16)	1554.8	-1659.7	74	SEG67(SEG59)	494.8	2140.3	117	OSC XO	-1555.2	-659.7
32	SEG25(SEG17)	1554.8	-1539.7	75	SEG68(SEG60)	384.8	2140.3	118	OSC XI	-1555.2	-769.7
33	SEG26(SEG18)	1554.8	-1429.7	76	SEG69(SEG61)	274.8	2140.3	119	RESET	-1555.2	-879.7
34	SEG27(SEG19)	1554.8	-1319.7	77	SEG70(SEG62)	164.8	2140.3	120	OSCI	-1555.2	-989.7
35	SEG28(SEG20)	1554.8	-1209.7	78	SEG71(SEG63)	54.8	2140.3	121	XIO	-1555.2	-1099.7
36	SEG29(SEG21)	1554.8	-1099.7	79	SEG72(SEG64)	-55.2	2140.3	122	GND	-1555.2	-1209.7
37	SEG30(SEG22)	1554.8	-989.7	80	COM11	-165.2	2140.3	123	VCC	-1555.2	-1319.7
38	SEG31(SEG23)	1554.8	-879.7	81	COM10	-275.2	2140.3	124	CAP1+	-1555.2	-1429.7
39	SEG32(SEG24)	1554.8	-769.7	82	COM9	-385.2	2140.3	125	CAP1-	-1555.2	-1539.7
40	SEG33(SEG25)	1554.8	-659.7	83	COM8	-495.2	2140.3	126	V5	-1555.2	-1659.7
41	SEG34(SEG26)	1554.8	-549.7	84	COM7	-605.2	2140.3	127	V4	-1555.2	-1779.7
42	SEG35(SEG27)	1554.8	-439.7	85	COM6	-715.2	2140.3	128	V3	-1555.2	-1899.7
43	SEG36(SEG28)	1554.8	-329.7	86	COM5	-825.2	2140.3	129	V2	-1555.2	-2019.7

## 21. ROM CODE CHECKLIST

<b>CHECKLIST</b>	
<b>ST2104</b> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	
8-Bit Microcontroller With 512K Bytes ROM	
<b>OSCILLATOR</b>	<input type="checkbox"/> 32768 Hz Crystal <input type="checkbox"/> R-OSC _____ MHz (Resistor= _____ KΩ) <input type="checkbox"/> Resonator <input type="checkbox"/> Crystal _____ MHz
<b>OPERATING VOLTAGE</b>	<input type="checkbox"/> 2.4V ~ 3.4V <input type="checkbox"/> 3.0V ± 10% <input type="checkbox"/> Other Range _____ ~ _____ V
<b>Note:</b> Maximum operating frequency = 4.0 Mhz@2.7V, 3.0 Mhz@2.4V	
<b>BATTERY</b>	<input type="checkbox"/> CR20 _____ x <input type="checkbox"/> AAx _____ <input type="checkbox"/> AAAx <input type="checkbox"/> _____
<b>POWER DOWN MODES</b>	<input type="checkbox"/> WAI-0 <input type="checkbox"/> WAI-1
<b>LOW VOLTAGE DETECTOR</b>	<input type="checkbox"/> Disabled <input type="checkbox"/> Enabled
<b>MASK OPTION</b>	<input type="checkbox"/> <b>ST2104A</b> : 73 SEGs x 16 COMs . <input type="checkbox"/> <b>ST2104B</b> : 65 SEGs x 24 COMs .
<b>LCD panel V<sub>OP</sub></b>	_____ <b>Voltage.</b> = Operation voltage x 2 x 0.96
<b>DATA SHEET:</b> ST2104_datasheet_v _____	
<b>CODE FILE:</b> _____ .BIN	<b>DATE(Y/M/D):</b> 20____/____/____
<b>ST2104 EVB</b>	<b>PCB</b> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> - <input type="checkbox"/>
<b>CHECK SUM</b> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> H (Byte Mode)	
<b>Note:</b> a. File format must be binary and the extension should be ".BIN". b. File should be wrapped in ZIP format for transferring or e-mailing. c. Only single file is allowed. d. File length is 512K bytes. e. Functions should be checked on the emulation board or by real chip. f. Electric characteristics of the emulation board are not identical with those of the real chip.	
<b>CUSTOMER</b>	
COMPANY	
SIGNATURE	
<b>SITRONIX</b>	
FAE/SA	
SALES	

Project Name:

Date:

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	Undefined byte and bit of register should not be used and keep it "0". Especially SYS[4] bit as "0" to ensure OSCX working normally.		
26	If high frequency was optioned to crystal mode (not RC oscillator), system clock need to be switched to low frequency (32768Hz) before into WAI1 mode. Example: smb7 <SYS ;; Change to OSCX bbr7 <SYS,\$ ;; Wait OSCX stable. smb6 <SYS ;; OSC OFF		
Engineer		Manager	

Reference table for LCD panel's parameters :

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

[ST2104A] - 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 checked="" 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

[ST2104B] - 65x24 LCD 規格參考表

Contrast		Equivalent Duty	Optimal Bias	R2	
Level	CTR[3:0]			n x R1	$\Omega$ (R1=10k)
<input type="checkbox"/> 1(light)	1111	84.3	10.2	6.2 x	61.8 k
<input type="checkbox"/> 2	1110	70.5	9.4	5.4 x	54.0 k
<input type="checkbox"/> 3	1101	60.6	8.8	4.8 x	47.9 k
<input type="checkbox"/> 4	1100	53.2	8.3	4.3 x	42.9 k
<input type="checkbox"/> 5	1011	47.3	7.9	3.9 x	38.8 k
<input type="checkbox"/> 6	1010	42.7	7.5	3.5 x	35.3 k
<input type="checkbox"/> 7	1001	38.8	7.2	3.2 x	32.3 k
<input checked="" type="checkbox"/> 8	1000	35.6	7.0	3.0 x	29.7 k
<input type="checkbox"/> 9	0111	32.9	6.7	2.7 x	27.4 k
<input type="checkbox"/> 10	0110	30.6	6.5	2.5 x	25.3 k
<input type="checkbox"/> 11	0101	28.6	6.3	2.3 x	23.4 k
<input type="checkbox"/> 12	0100	26.8	6.2	2.2 x	21.8 k
<input type="checkbox"/> 13	0011	25.2	6.0	2.0 x	20.2 k
<input type="checkbox"/> 14(dark)	0010	24	5.9	1.9 x	19.0 k



## 22. REVISIONS

Version 1.4	Page 1/40/45	Modify operation voltage MAX to 3.4V	.....2009/2/11
Version 1.3	Page 38/46	Add warm-up process in check list	.....2008/9/15
Version 1.2	Page 18	Modify PRES 8-bits counter to 16 bit and add description low byte	
		<b>PRS[7~0] : The low byte value of PRES counter</b>	
	Page 43	Add ST2104 EVB photo	
	Page 45	Add checklist for customer to confirm ST2104 EVB PCB number	.....2007/5/21
Version 1.1:	Page 7/16/17/38/39	Cancel the normal mode driving of OSCX(32768).	
Version 1.0:	Page 29	Modify Two-Pin Two Ended Mode Application Circuit.	
Version 0.8	Page 40	Add LVD voltage range in DC electrical characteristic.	
Version 0.7	Page 34	Modify TABLE 14-5 COM8~COM23 → COM0~COM16	
Version 0.6	Page 18	Modify the PRES reading value.	
Version 0.5	Page 1	Modify the voltage features about operation voltage.	
	Page 4	Delete the duty bit of the LCTL register	
	Page 18	Modify the figure number 11-6 to 11-1.	
	Page 40	Update the electrical characteristic	
	Page 41	Figure 18-1 and Figure 18-2 is changed.	
	Page 33	Modify the formula for LCD frames rate.	
Version 0.4:	Page 34	LCD memory mapping	
	Page 2/4/43	Segment outputs names	
Version 0.3:	Page 6	Memory map	
	Page 7	Register table	
	Page 33	LCD contrast control (1/24 Duty)	

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.