Sitronix

ST2100

8 BIT MICROCONTROLLER WITH 2M 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.

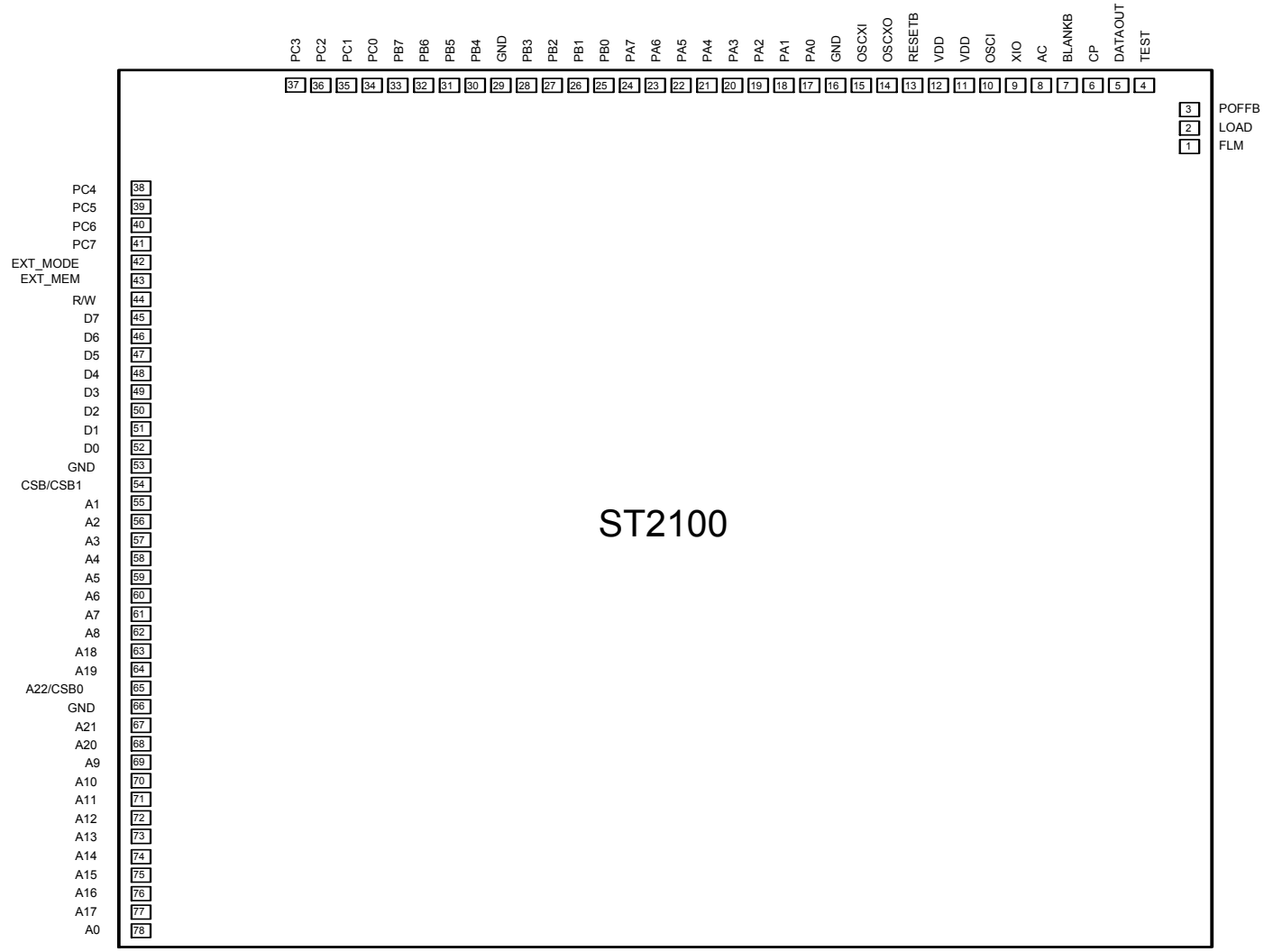
1. FEATURES

- Totally static 8-bit CPU
- ROM: 2M x 8 bits
- RAM: 4K x 8 bits
- Stack: Up to 128-level deep
- Operation voltage: 2.4V ~ 5.5V
- Operation frequency:
  - 3.0Mhz@2.4V(min.)
  - 4.0Mhz@2.7V(min.)
- External memory control up to 8M x 8 bits
- 24 CMOS bidirectional bit programmable I/O pins
- Hardware de-bounce option for input port
- Bit programmable pull-up for input port
- Timer/Counter :
  - Two 8-bit timer/16-bit event counter
  - One 8-bit Base timer
- Six powerful interrupt sources :
  - External interrupt (edge trigger)
  - TIMER0 interrupt
  - TIMER1 interrupt
  - BASE timer interrupt
  - PORTA[7~0] interrupt (transition trigger)
  - DAC reload interrupt
- Dual clock sources with warm-up timer :
  - OSCX: Crystal oscillator 32.768K Hz
  - OSCI: RC oscillator 500K ~ 4M Hz or
    OSCI,XIO: Resonator 500K ~ 4M Hz (code option)
- Direct Memory Access (DMA)
- LCD controller with two panel size :
  - 5376 (48x112) dots
  - 4096 (32X128) dots
- Programmable Sound Generator (PSG) includes :
  - Dual Tone generator
  - Noise generator
  - 16 level volume sound effect generator
  - Digital DAC for speech / tone
- Three power down modes :
  - WAI0 mode
  - WAI1 mode
  - STP mode

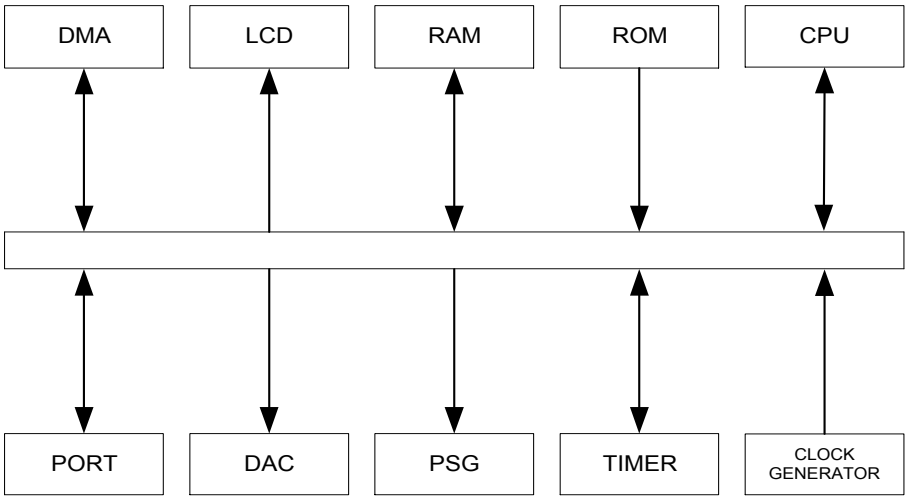
2. GENERAL DESCRIPTION

The ST2100 is a single chip micro-controller designed with CMOS silicon gate technology. This single chip micro-controller is useful for business equipment and other consumer applications. It integrates with 8-bit CPU core, SRAM, timer, LCD driver, I/O port and mask program ROM. This chip built-in a dual-oscillator to enhance the chip performance.

3. PAD DIAGRAM



4. BLOCK DIAGRAM

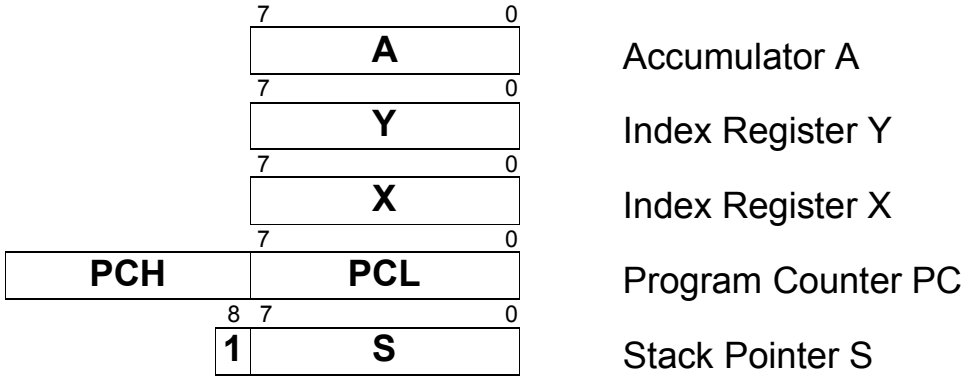


Pad Description

Pin No.	Designation	I/O	Description
13	$\overline{\text{RESET}}$	I	Pad reset input (active low)
16,29,53,66	GND	P	Ground Input and chip sub-strate
17-24	PORTA[0-7]	I/O	Programmable I/O, Transition Interrupt(edge active), INTX Interrupt , Timer Prescaler PRE16 clock source
25-28,30-33	PORTB [0-7]	I/O	Bit programmable I/O,PSG output, DAC output
34-41	PORTC [0-7]	I/O	Bit programmable I/O
55-64,67-78	A[0-21]	O	Address bus for expand memory
65	A[22]/CSB0	O	Address bus for expand memory / Expand memory chip select signal
45-52	D[0-7]	I/O	Data bus for expand memory
44	R/W	O	Read or write signal for expand memory
54	CSB/CSB1	O	Expand memory chip select signal
43	EXT_MEM	O	External memory enable/disable control signal
11	VDD	P	Power supply pin
14,15	OSCXO, OSCXI	I/O	OSCX I/O pin. For 32768Hz crystal used.
10	OSCI	I	RC oscillator pin, had to be connected to external resistor
4	TEST	I	Test pin for chip test, normal to NC.
1	FLM	O	First line mark for common signal(to LCD driver ST2101)
2	LOAD	O	Load data into Segment or common driver's data latch (to LCD driver ST2101)
8	AC	O	LCD alternating signal (connect to LCD driver ST2101)
3	POFFB	O	Control the power generator of voltage pumping circuit (to LCD driver ST2101)
9	XIO	O	OSCI,XIO for resonator 500K ~ 4M Hz (code option)
6	CP	O	Shift clock pulse for segment driver (to LCD driver ST2101)
5	DATAOUT	O	Output serial data for segment driver (to LCD driver ST2101)
7	BLANKB	O	LCD display can be turn off directly by external control. When BLANKB is low, the LCD display automatically set to blank state (to LCD driver ST2101).
42	EXT_MODE	I	Select {CSB,A[22]} or {CSB0,CSB1}
12	VDD	P	Power supply pin

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

5. CPU



CPU REGISTER MODEL

5.1 Accumulator (A)

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

5.2 Index Registers (X,Y)

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

5.3 Stack Pointer (S)

The stack Pointer is an 8-bit register which is used to control the addressing of the variable-length stack. It's range from 100H to 1FFH total for 256 bytes (128 level deep). The stack pointer is automatically increment and decrement under control of the microprocessor to perform stack manipulations under direction of either the program or interrupts (IRQ). The stack allows simple implementation of nested subroutines and multiple level interrupts. The stack pointer is initialized by the user's software.

5.4 Program Counter (PC)

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

5.5 Status Register (P)

The 8-bit Processor Status Register contains seven status flags. Some of the flags are controlled by the program, others may be controlled both by the program and the CPU. The instruction set contains a member of conditional branch instructions which are designed to allow testing of these flags. Refer to TABLE 5-1:

TABLE 5-1:STATUS REGISTER (P)

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

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

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

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

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

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

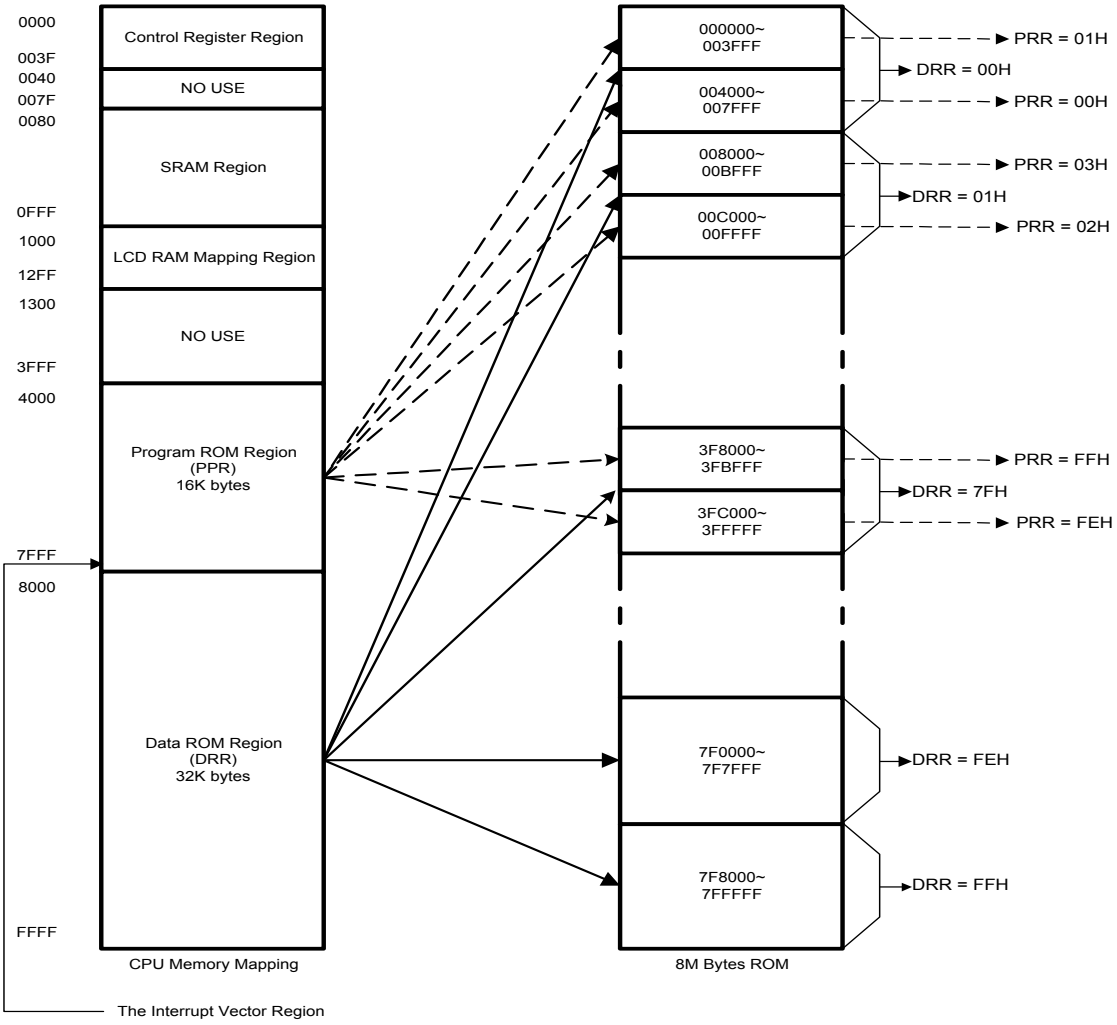
Bit 1: **Z :** Zero flag
1 = Zero
0 = Non zero

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

6. MEMORY CONFIGURATION

6.1 Memory map

ST2100 has total 2M bytes ROM and 4K bytes 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 ST2100 is from $8000 to $FFFF (32K bytes) and program ROM address is from $4000 to $7FFF(16K bytes).



.

6.2 ROM

6.2.1 Bank Description

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

TABLE 6-1: ROM CONTROL REGISTERS ($31~$32)

Address	Register	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PRR	$31	RW	PRR[7]	PRR[6]	PRR[5]	PRR[4]	PRR[3]	PRR[2]	PRR[1]	PRR[0]
DRR	$32	RW	DRR[7]	DRR[6]	DRR[5]	DRR[4]	DRR[3]	DRR[2]	DRR[1]	DRR[0]

6.2.2 Memory Modes

The internal 2M bytes memory can be enabled or disabled by selecting input level of EXT_MEM. When EXT_MEM equals “1”, the internal memory is enabled. This is the normal option of one ST2100 in common products. When EXT_MEM equals “0”, the internal memory is disabled and the first 2M bytes area will be mapped to external. This mode is for code generation when developing system using a real-chip board.

The external memory is accessed by bus signals: CSB/CSB1, A22/CSB0, A[21:0], R/W and D[7:0]. Two kinds of external memory mapping are further defined by EXT_MODE. Refer to TABLE 6-2: for memory modes and the mapping of both internal and external memory.

TABLE 6-2: Memory Modes and External Address Mapping

<table>
<tr><td></td><td></td><td>Internal Memory</td><td colspan="3">External Memory</td></tr>
<tr><td></td><td></td><td></td><td colspan="3">EXT_MODE</td></tr>
<tr><td></td><td></td><td></td><td colspan="2">1</td><td>0</td></tr>
<tr><td></td><td></td><td></td><td>CSB/CSB1</td><td>A22/CSB0</td><td>CSB/CSB1</td></tr>
<tr><td rowspan="2">EXT_MEM</td><td>1</td><td>1~2MB
000000
~1FFFFF
(CSB,A[22:0]=1)</td><td>3~8MB
200000~7FFFFF
CSB | A22 | A21 | Address
0 | 0 | 0 | 3~4MB
0 | 0 | 1 | 5~6MB
0 | 1 | 0 | 7~8MB</td><td>3~6MB
200000~5FFFFF
CSB0 | A21 | Address
0 | 0 | 3~4MB
0 | 1 | 5~6MB</td><td>7~8MB
600000~7FFFFF
CSB1 | A21 | Address
0 | 0 | 7~8MB</td></tr>
<tr><td>0</td><td>Disable</td><td>1~8MB
000000~7FFFFF
CSB | A22 | A21 | Address
0 | 0 | 0 | 1~2MB
0 | 0 | 1 | 3~4MB
0 | 1 | 0 | 5~6MB
0 | 1 | 1 | 7~8MB</td><td>1~4MB
000000~3FFFFF
CSB0 | A21 | Address
0 | 0 | 1~2MB
0 | 1 | 3~4MB</td><td>5~8MB
400000~7FFFFF
CSB1 | A21 | Address
0 | 0 | 5~6MB
0 | 1 | 7~8MB</td></tr>
</table>

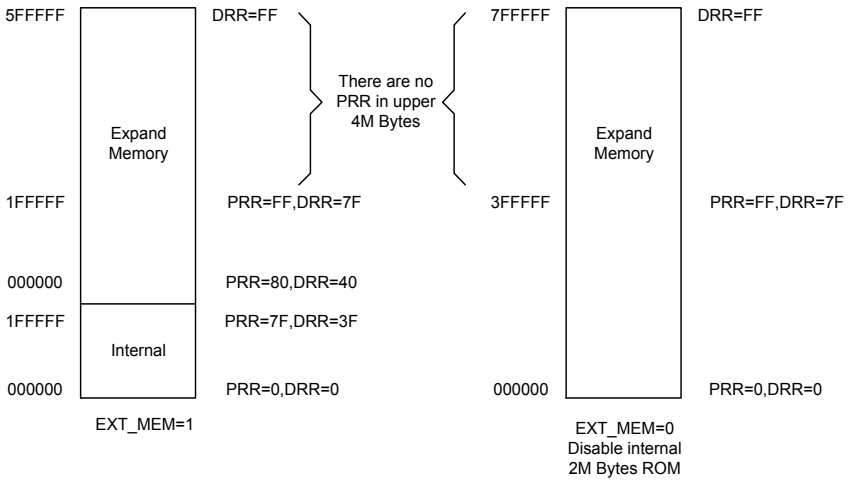


FIGURE 6-1: Bank Registers and Memory Mapping

6.3 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**	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**	R/W	-	PCK[2]	PCK[1]	PCK[0]	PRBS	C1EN	C0EN	DACE=0	-000 0000
		R/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	-	-	-	-	-	000 - - - - -
$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]	???? ????
$029	**DMSH**	R/W	DMS[15]	DMS[14]	DMS[13]	DMS[12]	DMS[11]	DMS[10]	DMS[9]	DMS[8]	???? ????
$02A	**DMDL**	R/W	DMD[7]	DMD[6]	DMD[5]	DMD[4]	DMD[3]	DMD[2]	DMD[1]	DMD[0]	???? ????
$02B	**DMDH**	R/W	DMD[15]	DMD[14]	DMD[13]	DMD[12]	DMD[11]	DMD[10]	DMD[9]	DMD[8]	???? ????
$02C	**DCNTL**	R/W	DCNT[7]	DCNT[6]	DCNT[5]	DCNT[4]	DCNT[3]	DCNT[2]	DCNT[1]	DCNT[0]	???? ????
$02D	**DCNTH**	R/W	-	-	-	DMAM	DCNT[11]	DCNT[10]	DCNT[9]	DCNT[8]	???? ????
$030	**SYS**	R/W	XSEL	OSTP	XSTP	XBAK	WSKP	WAIT	-	-	0000 00- -
$031	**PRR**	R/W	PRR[7]	PRR[6]	PRR[5]	PRR[4]	PRR[3]	PRR[2]	PRR[1]	PRR[0]	???? ????
$032	**DRR**	R/W	DRR[7]	DRR[6]	DRR[5]	DRR[4]	DRR[3]	DRR[2]	DRR[1]	DRR[0]	???? ????
$033	**DMR**	R/W	DMR[7]	DMR[6]	DMR[5]	DMR[4]	DMR[3]	DMR[2]	DMR[1]	DMR[0]	???? ????
$039	**LCFG**	R/W	LCFG[7]	LCFG[6]	LCFG[5]	LCFG[4]	LCFG[3]	LCFG[2]	LCFG[1]	LCFG[0]	???? ????
$03A	**LCTL**	R/W	LPWR	BLANK	REV	-	CTR[3]	CTR[2]	CTR[1]	CTR[0]	0000 0000
$03B	**LCK**	R/W	-	-	-	-	-	LCK[2]	LCK[1]	LCK[0]	- - - - -000
$03C	**IREQ***	R	-	-	IRBT	IRPT	IRT1	IRT0	IRDAC	IRX	- - 00 0000
		W	-	-	CLRBT	CLRPT	CLRT1	CLRT0	CLRDAC	CLRX	- - 00 0000
$03E	**IENA**	*R/W	-	-	IEBT	IEPT	IET1	IET0	IEDAC	IEX	- - 00 0000

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

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. Must initial the PRR, DRR and DMR registers when system reset.

6.3.1 DATA RAM ($0080~$0FFF)

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

6.3.2 STACK RAM ($0100~$01FF)

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

6.3.3 LCD RAM ($1000~$12FF)

Resident LCD-RAM, accessible through write and read instructions, for 112*48 & 128*32 LCD display. Note that this area can also be used as data memory. Refer to section **13.4.1** about the detail usage.

7. INTERRUPTS

7.1 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 ST2012 activity. When a positive edge is detected, there is an initialization sequence lasting six clock cycles. Then the interrupt mask flag is set, the decimal mode is cleared and the program counter will loaded with the restart vector from locations $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 executed, and interrupt mask flag will be cleared. Hardware will push 'PC', 'P ' Register to stack and set interrupt mask flag (I). Program counter will be loaded with the 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 executed, and interrupt mask flag will be cleared. Hardware will push 'PC', 'P ' Register to stack and set interrupt mask flag (I). Program counter will be loaded with the T1 vector from locations $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 7-1:

TABLE 7-1:PREDEFINED VECTORS FOR INTERRUPT

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

7.2 Interrupt request clear

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

TABLE 7-2: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	-	-	IRBT	IRPT	IRT1	IRT0	IRDAC	IRX	- - 00 0000
		W	-	-	CLRBT	CLRPT	CLRT1	CLRT0	CLRDAC	CLRX	- - 00 0000

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

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

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

Bit 2: **IRT0(Read):** Timer0 Interrupt Request bit
1 = Timer0 overflow interrupt occurs
0 = Timer0 overflow interrupt doesn’t occur

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

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

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

TABLE 7-3: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: **IEBT:** Base Timer Interrupt Enable bit
1 = Time base interrupt enable
0 = Time base interrupt disable

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

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

Bit 2: **IET0:** Timer0 Interrupt Enable bit
1 = Timer0 overflow interrupt enable
0 = Timer0 overflow interrupt disable

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

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

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

8. I/O PORTS

8.1 General function

ST2100 has three I/O ports, PORT-A, PORT-B and PORT-C. ST2100 provides for a maximum of 24 I/O pins. For detail pin assignment, please refer to TABLE 8-1:

TABLE 8-1:I/O DESCRIPTION

PORT NAME	PAD NAME	PIN TYPE	FEATURE
PORT A	PA0/INTX	I/O	Programmable input/output pin
	PA1	I/O	
	PA2	I/O	
	PA3	I/O	
	PA4	I/O	
	PA5	I/O	
	PA6	I/O	
	PA7	I/O	
PORT B	PB0	I/O	Programmable input/output pin
	PB1	I/O	
	PB2	I/O	
	PB3	I/O	
	PB4	I/O	
	PB5	I/O	
	PB6	I/O	
	PB7	I/O	
PORT C	PC0	I/O	Programmable input/output pin
	PC1	I/O	
	PC2	I/O	
	PC3	I/O	
	PC4	I/O	
	PC5	I/O	
	PC6	I/O	
	PC7	I/O	

These I/O ports have same feature in control method and their control register refer to TABLE 8-2:

TABLE 8-2:PORT Control register

Address	Register	R/W	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	COMMENT
$00	PA	RW	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	PORT A
$01	PB	RW	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	PORT B
$02	PC	RW	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	PORT C
$08	PCA	RW	PCA7	PCA6	PCA5	PCA4	PCA3	PCA2	PCA1	PCA0	PORT A direction control
$09	PCB	RW	PCB7	PCB6	PCB5	PCB4	PCB3	PCB2	PCB1	PCB0	PORT B direction control
$0A	PCC	RW	PCC7	PCC6	PCC5	PCC4	PCC3	PCC2	PCC1	PCC0	PORT C direction control
$0F	PMCR	RW	PULL	PDBN	INTEG	-	-	-	PSGO	PSGB	PORT mode control

8.2 PORT-A

8.2.1 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 8-3:SUMMARY FOR PORT-A REGISTERS

Address	Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default
$000	**PA**	R/W	PA[7]	PA[6]	PA[5]	PA[4]	PA[3]	PA[2]	PA[1]	PA[0]	1111 1111
$008	**PCA**	R/W	PCA[7]	PCA[6]	PCA[5]	PCA[4]	PCA[3]	PCA[2]	PCA[1]	PCA[0]	0000 0000
$00F	**PMCR**	R/W	PULL	PDBN	INTEG	-	-	-	PSG0	PSGB	100 - - -00
$03C	**IREQ**	R/W	-	-	IRBT	IRPT	IRT1	IRT0	IRDAC	IRX	- - 00 0000
$03E	**IENA**	R/W	-	-	IEBT	IEPT	IET1	IET0	IEDAC	IEX	- - 00 0000

8.2.2 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 8-4: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

8.2.3 PORT A PULL-UP OPTION

PORT A contains pull-up MOS transistors. The PULL control bit (PMCR[7]) controls the on/off of all the pull-up MOS simultaneously. The on/off of every pull-up MOS transistor will be controlled by port data register (PA) and the pull-up MOS will be enabled with data bit "1" and disable with data bit "0". Please refer to the FIGURE 8-1: .

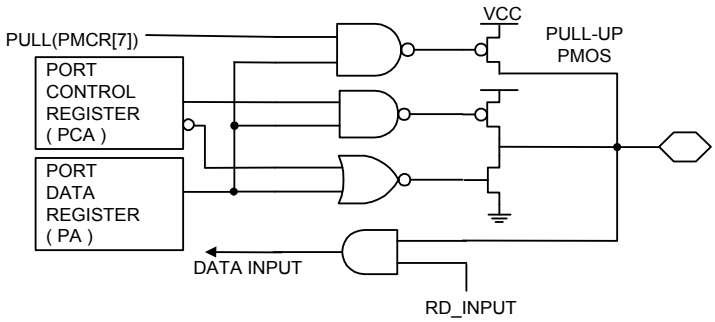


FIGURE 8-1: Port-A Configuration Function Block Diagram

TABLE 8-5:PORT CONDITION CONTROL REGISTER (PMCR)

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

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

Bit 6: **PDBN :** Enable Port-A interrupt 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

8.2.4 Port-A interrupt

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

When programmer enables INTX and PT interrupts, PA0 trigger occur. INTX and PT interrupts will therefore happen sequentially. Please refer to the FIGURE 8-1: .

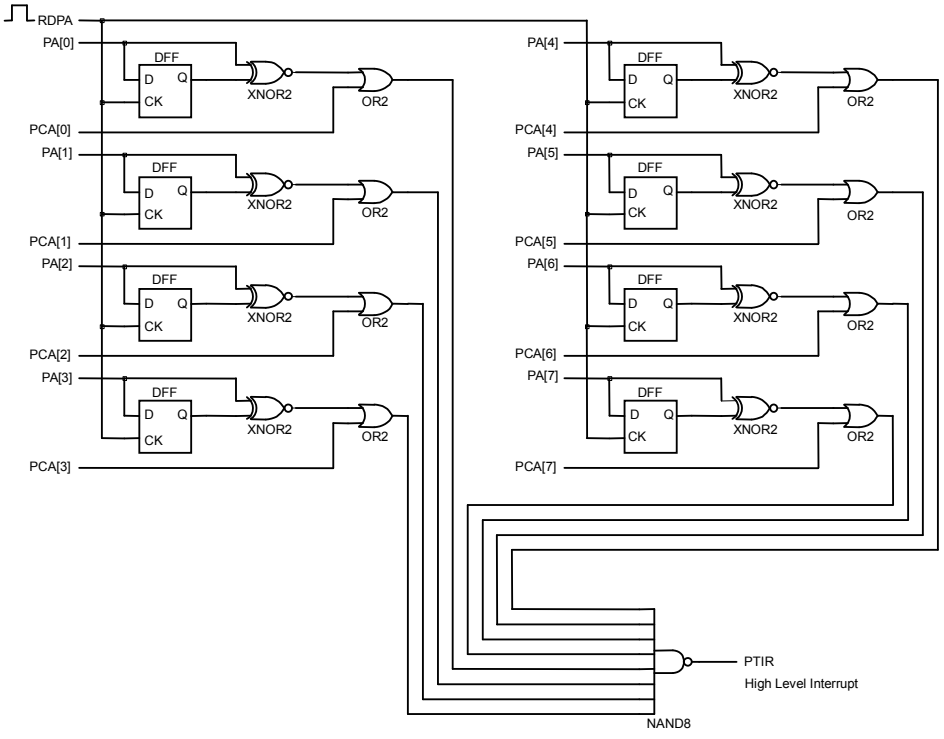
Operating Port-A interrupt step by step :

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

Example :

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

FIGURE 8-2: Port Interrupt Logic Diagram



8.2.5 Port-A interrupt debounce

ST2100ST2100 has hardware debounce option for Port-A interrupt. The debounce will be enabled with “1” and disable with “0” for PDBN. The debounce will active when Port-A transition occurs, PDBN enable and **OSCX enable**.

The debounce time is **OSCX x 512 cycles(about 16 ms).** Refer to the TABLE 9-6.

TABLE 8-6:PORT CONDITION CONTROL REGISTER (PMCR)

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

Bit 6: **PDBN :** Enable Port-A interrupt de-bounce bit
1 = de-bounce for Port-A interrupt
0 = no de-bounce for Port-A interrupt

8.2.6 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 the Figure 9-3.

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

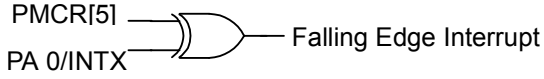
Operating INTX interrupt step by step :

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

Example :

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

FIGURE 8-3: INTX Logic Diagram



8.3 PORT-B and PORT-C

8.3.1 General function

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 8-7:SUMMARY FOR 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

8.3.2 Input/Output control

Direction of Port-B (or Port-C) is controlled by PCB (or PCC). 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 8-8: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: **PCB[7~0] :** Port-B directional bits
1 = Output mode
0 = Input mode

TABLE 8-9: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: **PCC[7~0] :** Port-C directional bits
1 = Output mode
0 = Input mode

8.3.3 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 an input. The PULL control bit of PMCR also controls the ON/OFF of all pull-up MOS simultaneously. Please refer to the 0.

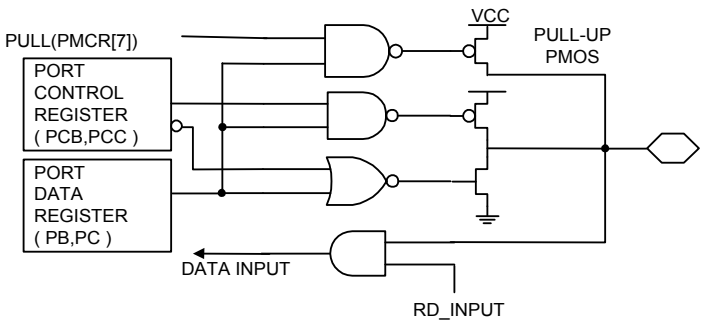


FIGURE 9-4: Port-B and Port-C Configuration Function Block Diagram

TABLE 8-10:PORT CONDITION CONTROL REGISTER (PMCR)

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

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

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

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

9. OSCILLATOR

ST2100 is with dual-clock system. Programmer can choose between OSC(RC) and OSCX(32.768kHz). Or both as clock source through program. The system clock(SYSCK) also can be switched between OSC and OSCX. The OSC will be switch with "0" and OSCX will be switch with "1" for **XSEL**. Whenever system clock be switch, the warm-up cycles are occur at the same time. That is confirm SYSCK really switched when read **XSEL** bit. LCD driver, Timer1, Base Timer and PSG can utilize these two clock sources as well.

TABLE 9-1: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	-	-	0000 00- -

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

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

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

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

Note:

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

2. System warm-up to 16 or 256 oscillation cycles is when system clock (SYSCK) is change or power on reset.

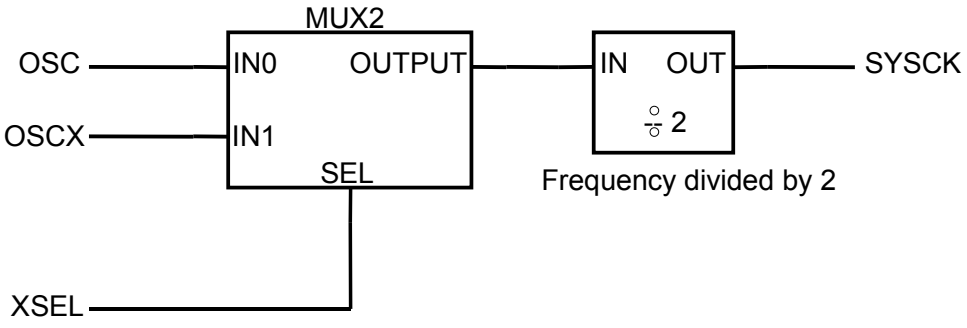


FIGURE 9-1: System Clock Diagram

10. TIMER / EVENT COUNTER

10.1 Prescaller

10.1.1 Function Description

The ST2100 has three timers: Base timer, Timer 0 and Timer 1 with two pre-scaler PRES and PREW. There are the two clock sources for PRES and one clock source(OSCX) for PREW. Refer to FIGURE 10-1:

TABLE 10-1:SUMMARY FOR TIMER REGISTERS

Address	Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default
$021	**BTM**	W	-	-	-	-	BTM[3]	BTM[2]	BTM[1]	BTM[0]	- - - - 0000
$023	**PRS**	R	PRS[7]	PRS[6]	PRS[5]	PRS[4]	PRS[3]	PRS[2]	PRS[1]	PRS[0]	0000 0000
		W	SRES	SENA	SENT	-	-	-	-	-	000 - - - - -
$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
$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

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

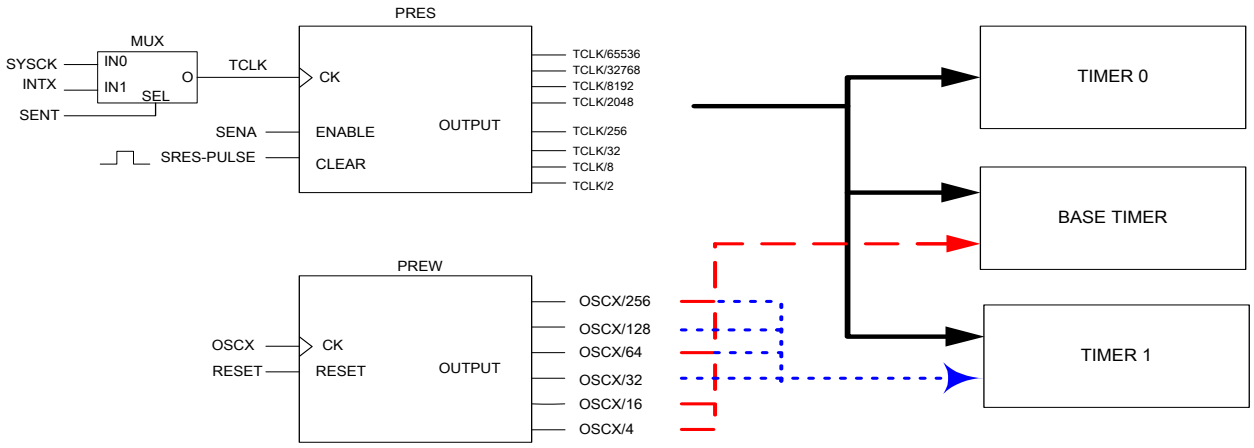


FIGURE 10-1: Prescaler for Timers

10.1.2 PRES

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

TABLE 10-2: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] :** 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"

10.1.3 PREW

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

10.2 Base timer

10.2.1 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 10-2:

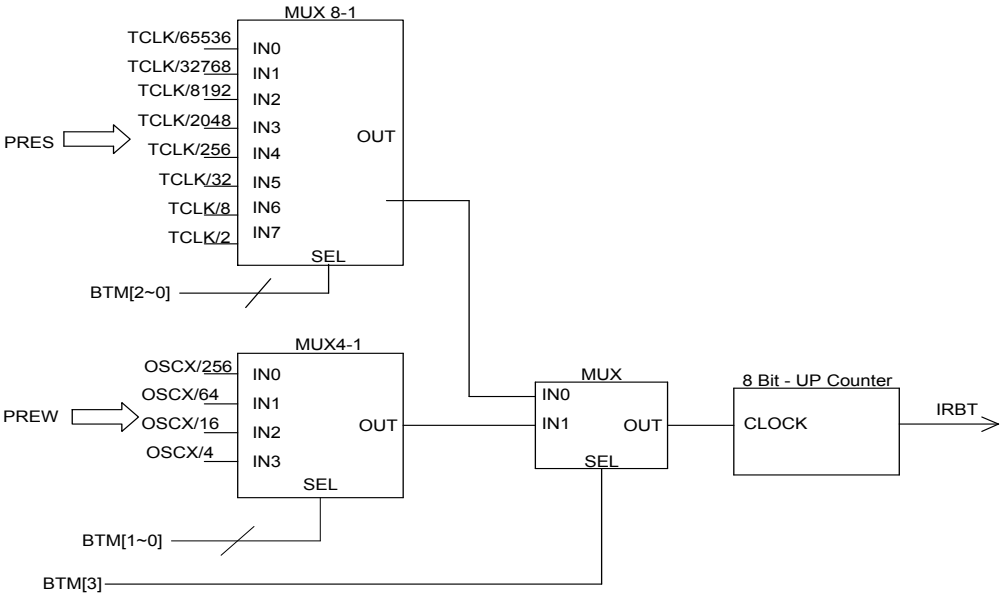


FIGURE 10-2: Structure of Base Timer

10.2.2 Clock source control for Base TimerSeveral clock sources can be selected for Base Timer. Please refer to TABLE 10-3:

TABLE 10-3:CLOCK SOURCE FOR 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

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

10.3 Timer 0

10.3.1 Timer0 descriptionThe Timer0 is an 8-bit up counter. It can be used as a timer or an event counter. T0C($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 10-3: .

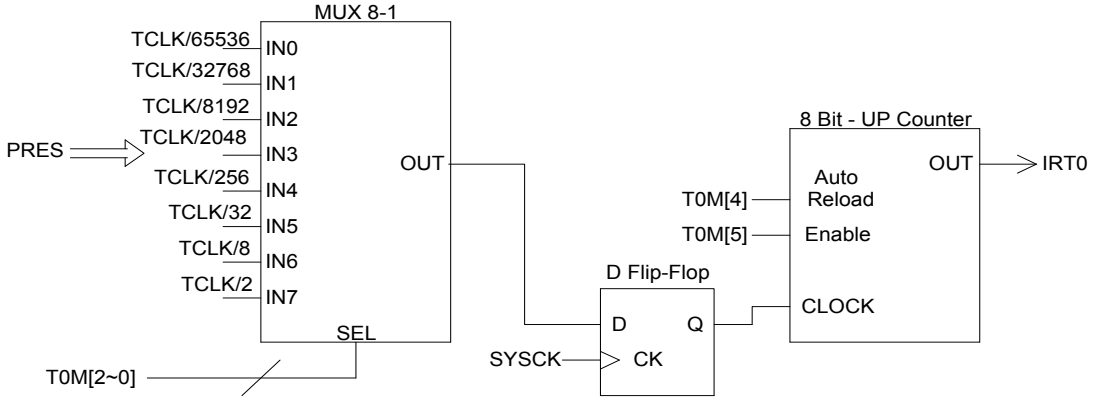


FIGURE 10-3: Timer0 Structure Diagram

10.3.2 Clock source control for Timer0Several clock source can be chosen from for Timer0. It's very important that Timer0 can keep counting as long as SYSCK stays active. Refer to TABLE 10-4:

TABLE 10-4:The clock source can be selected from system sources.

T0M[2]	T0M[1]	T0M[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

T0M[4] : Control automatic reload operation
0 : No auto reload
1 : Auto reload
T0M[5] : Control Timer 0 enable/disable
0 : Disable counting
1 : Enable counting
SENA : Prescaler enable bit
0 : TCLK stop
1 : TCLK counting

TABLE 10-5:TIMER0 COUNTING 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											

10.4 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 10-4: .

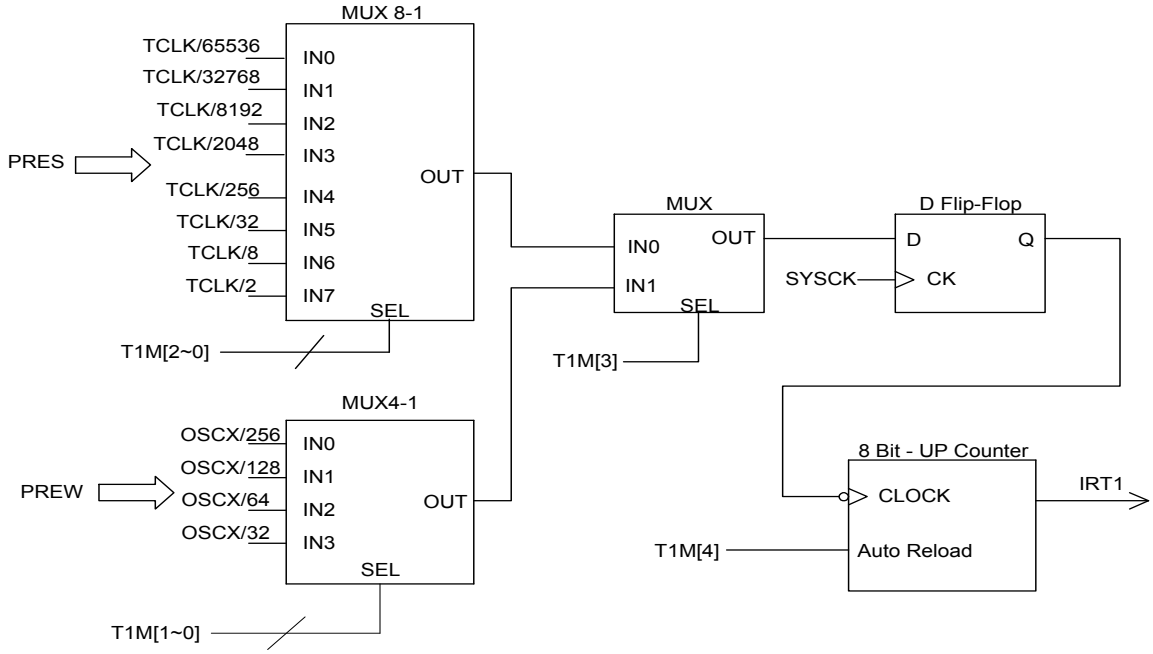


FIGURE 10-4: Timer1 Structure Diagram

TABLE 10-6:TIMER1 COUNTING REGISTER (T1C)

Address	Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default
$027	**T1C**	R/W	T1C[7]	T1C[6]	T1C[5]	T1C[4]	T1C[3]	T1C[2]	T1C[1]	T1C[0]	0000 0000
Bit 7-0: **T1C[7-0] :** Timer1 up counter register											

TABLE 10-7:Timer 1 mode register

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

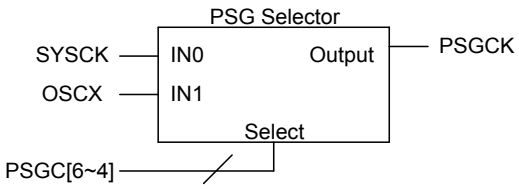
T1M[4]: Control automatic reload operation
0: No auto reload
1: auto reload
SENA : Prescaler enable bit
0 : TCLK stop
1 : TCLK counting

11. PSG

11.1 Function description

The built-in dual channel Programmable Sound Generator (PSG) are 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 11-2: and the PSG clock Source is shown in FIGURE 11-1: . The ST2100 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 11-1: Clock Source for PSG



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

FIGURE 11-2: Program Sound Generator

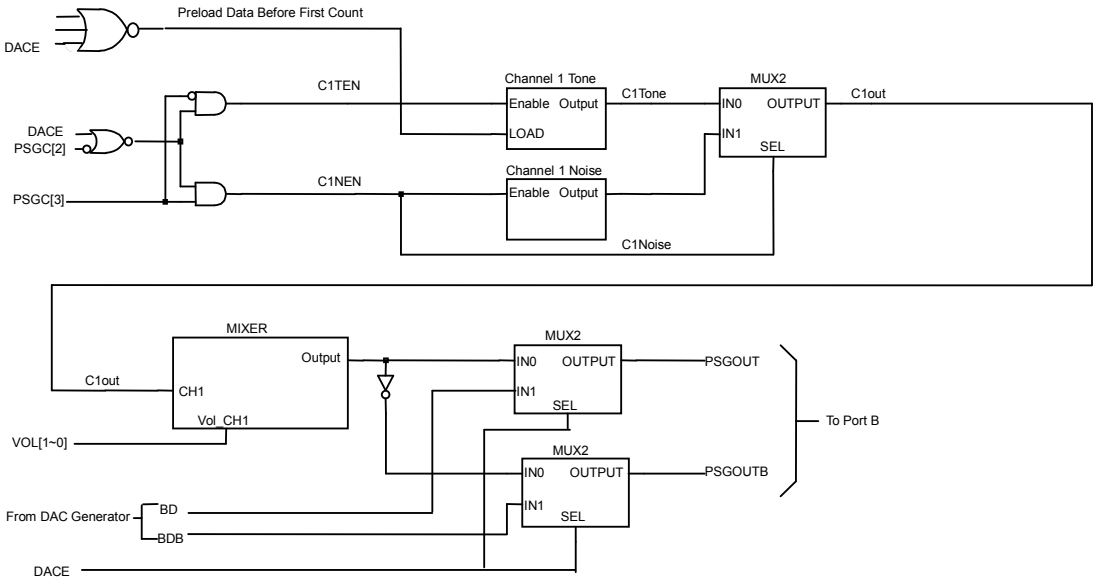


TABLE 11-1:SUMMARY FOR PSG REGISTERS

Address	Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default
$00F	**PMCR**	R/W	PULL	PDBN	INTEG	-	-	-	PSGO	PSGB	100 - - -00
$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 11-2:CONTROL REGISTER FOR PSG OUTPUT (PMCR)

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

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

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

TABLE 11-3:CONTROL REGISTER FOR PSG VOLUME (VOL)

Address	Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default
$017	**VOL**	W	VOL1[3]	VOL1[2]	VOL1[1]	VOL1[0]	VOL0[3]	VOL0[2]	VOL0[1]	VOL0[0]	0000 0000

Bit 3~0: **VOL0[3~0] :** PSG channel 0 volume control bit
0000 = No sound output
0001 = 1/16 volume (PSGCK must >= 320K Hz)
:
0100 = 4/16 volume
:
1000 = 8/16 volume
:
1111 = Maximum volume (PSGCK must >= 20K Hz)

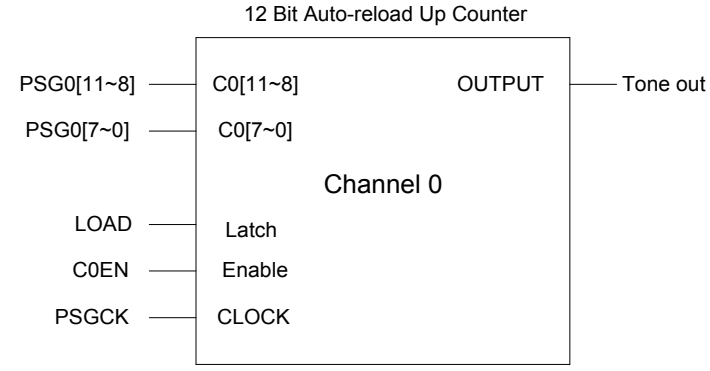
Bit 7~4: **VOL1[3~0] :** PSG channel 1 volume control bit
0000 = No sound output
0001 = 1/16 volume (PSGCK must >= 320K Hz)
:
0100 = 4/16 volume
:
1000 = 8/16 volume
:
1111 = Maximum volume (PSGCK must >= 20K Hz)

If single channel enable then PSG volume can be double level to output.
(16 + 16 = 32 level volume control)

11.2 Tone Generator

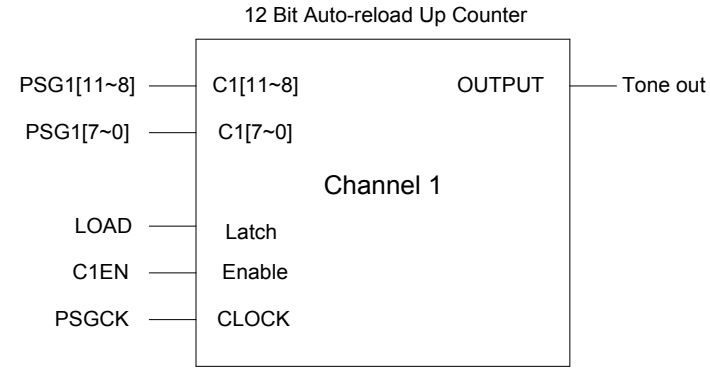
11.2.1 General Description

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



Frequency of Channel 0 Tone = PSGCK/(1000H-PSG0[11~0])/2

FIGURE 11-3: Channel 0 Tone Counter



Frequency of Channel 1 Tone = PSGCK/(1000H-PSG1[11~0])/2

FIGURE 11-4: Channel 1 Tone Counter

11.2.2 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 11-4: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

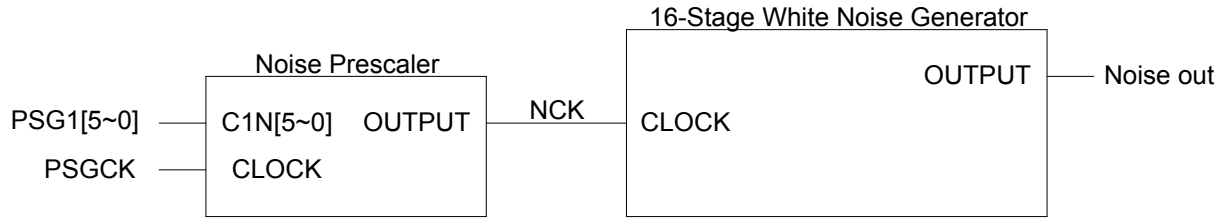
Bit 3: **PRBS :** Tone or Noise generator selection bit
1 = Noise generator
0 = Tone generator

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

11.3 Noise Generator Control

11.3.1 General description Noise generator is shown in FIGURE 11-5: , which base frequency is controlled by PSG1[5~0].

FIGURE 11-5: Noise Generator Diagram



NCK Frequency = PSGCK/(40H-PSG1[5~0])

11.3.2 PSG Noise programming

To program noise generator, PSGO (PMCR[1]) or PSGB (PMCR[0]) should be set to “1” for PB1 or PB0 in order to be in PSG output. Noise or DAC function is defined by DACE. Writing a “1” to C1EN will enable noise generator when PSG is in noise mode.

12. DIGITAL DAC

A built-in digital DAC is for analog sampling data or voice signals. The structure of DAC is shown in Figure 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 OSC can't less 4 M Hz.

TABLE 12-1:SUMMARY FOR DAC REGISTERS

Address	Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default
$00F	**PMCR**	R/W	PULL	PDBN	INTEG	-	-	-	PSG0	PSGB	100 - - -00
$012	**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 00-0
		W	-	PCK[2]	PCK[1]	PCK[0]	DMD[1]	DMD[0]	INH	DACE=1	- 000 0000

TABLE 12-2: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 12-3: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

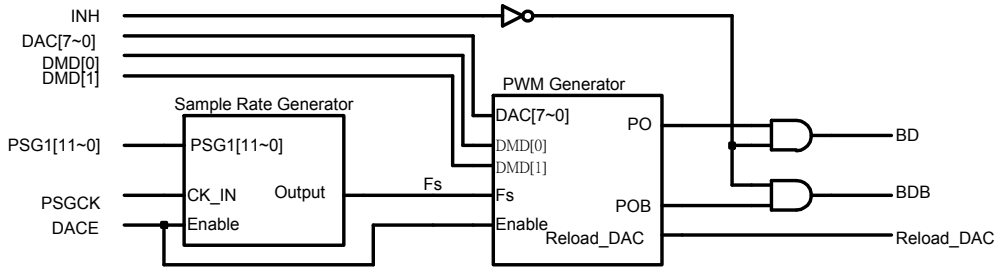
Bit 6~4: **PCK[2~0] :** PSGCK selection for PSG and DAC
000 = SYSCK / 2
X01 = SYSCK / 4
X10 = SYSCK / 8
011 = SYSCK / 16
100 = SYSCK *
111 = OSCX

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

12.2 Sampling Rate Control

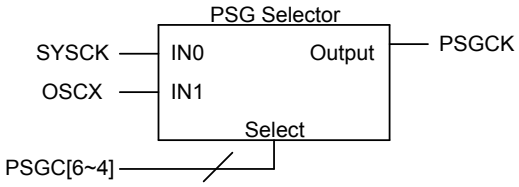
The sample rate is controlled by PSG1L and PSG1H. PSG1[11~7] controls sample rate/post scaling and PSG1[6] must set '0' and PSG1[5~0] must set '1'. The input clock source is controlled by PCK[2~0]. The block diagram is shown as the following:

FIGURE 12-1: DAC Generator



Diagram

FIGURE 12-2: Clock Source for DAC



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

TABLE 12-4:Sample Rate description table

DAC SAMPLE RATE ALGORITHM DESCRIPTION
Sample-Rate = PSGCK / 128 / (20H-PSG1[11~7])
Note: PSG1[6] must set '0 ' and PSG1[5~0] must set '1 ' by DAC mode.

12.3 PWM DAC Mode Select

The PWM DAC generator has three modes, Single-pin mode, Two-pin two ended mode and Two-pin push pull mode. They are depended on the application used. The DAC mode is controlled by DMD[1~0]. (TABLE 13-3)

12.3.1 Single-Pin Mode (Accurate to 7 bits)

Single-pin mode is designed for use with a single-transistor amplifier. It has 7 bits of resolution. The duty cycle of the **PB1** is proportional to the output value. If the output value is 0, the duty cycle is 50%. As the output value increases from 0 to 63, the duty cycle goes from being high 50% of the time up to 100% high. As the value goes from 0 to -64, the duty cycle decreases from 50% high to 0%. **PB0** is inverse of **PB1**'s waveform. Figure 13-3 shows the **PB1** wave-forms.

FIGURE 12-3: Single-Pin PWM DAC Wave-form

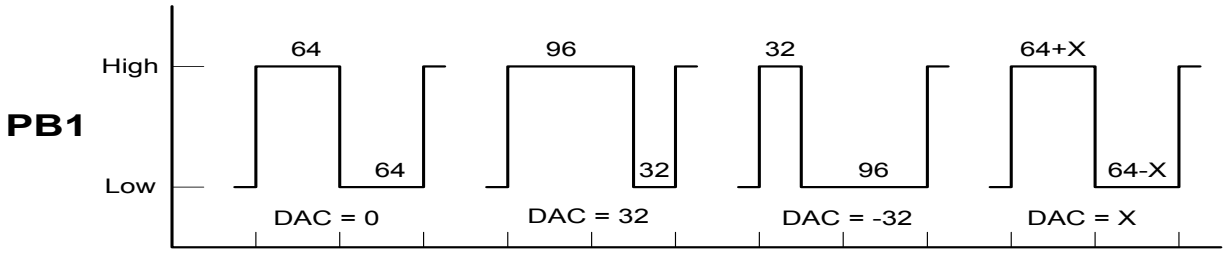
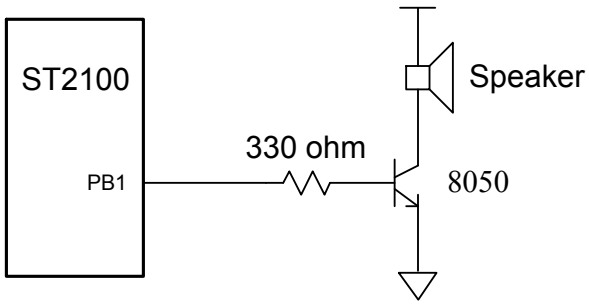


FIGURE 12-4: Single-Pin Application Circuit



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

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

Figure 13-5 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 12-5: Two-Pin Two Ended PWM DAC Wave-form

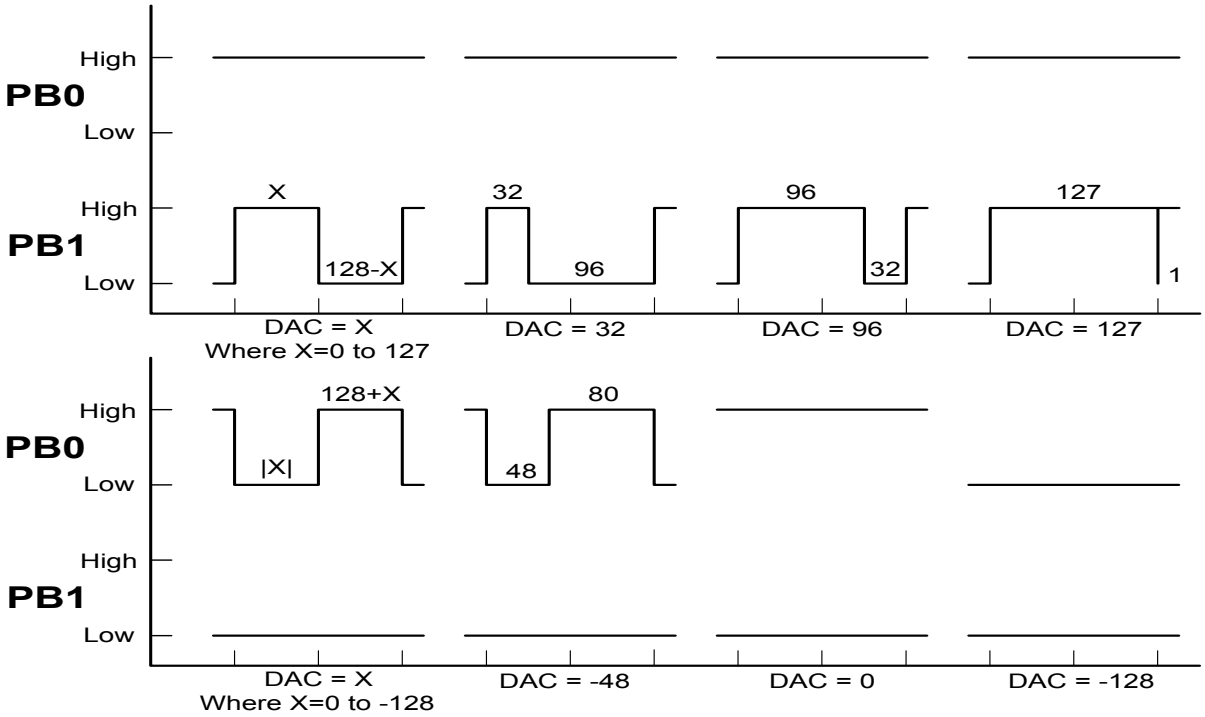
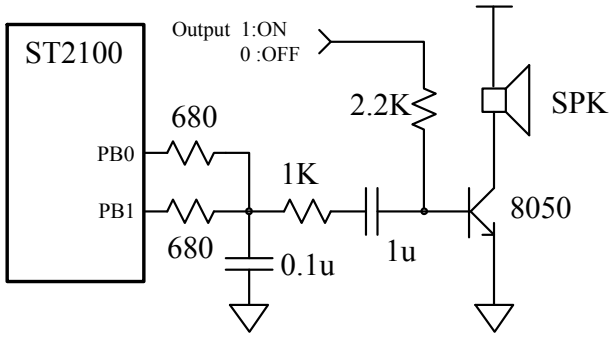


FIGURE 12-6: Two-Pin Two Ended mode Application Circuit



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

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

Figure 13-7 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 12-7: Two-Pin Push Pull PWM DAC Wave-form

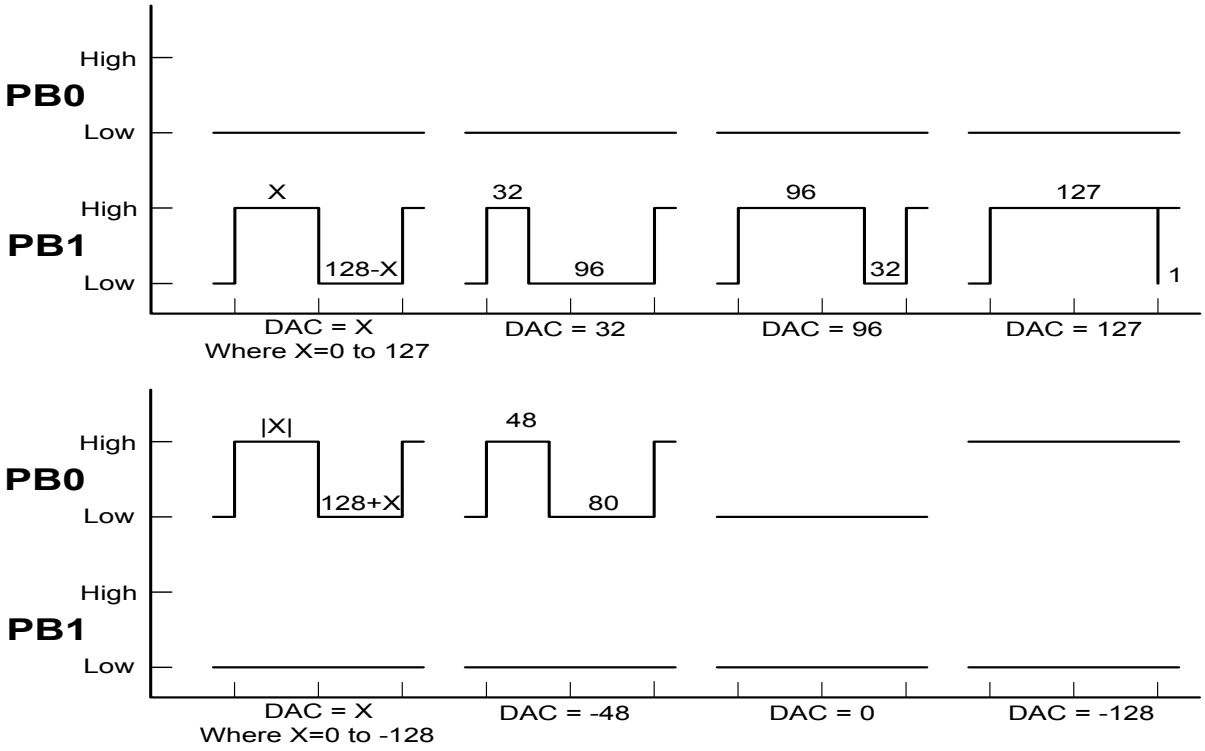
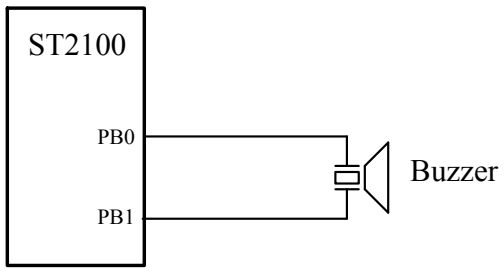


FIGURE 12-8: Two-Pin Push Pull Application Circuit



13. LCD

Full-graphic modes is supported on this chip. The graphic memory is constructed by 48 COM * 112 SEG or 32 COM *128 SEG

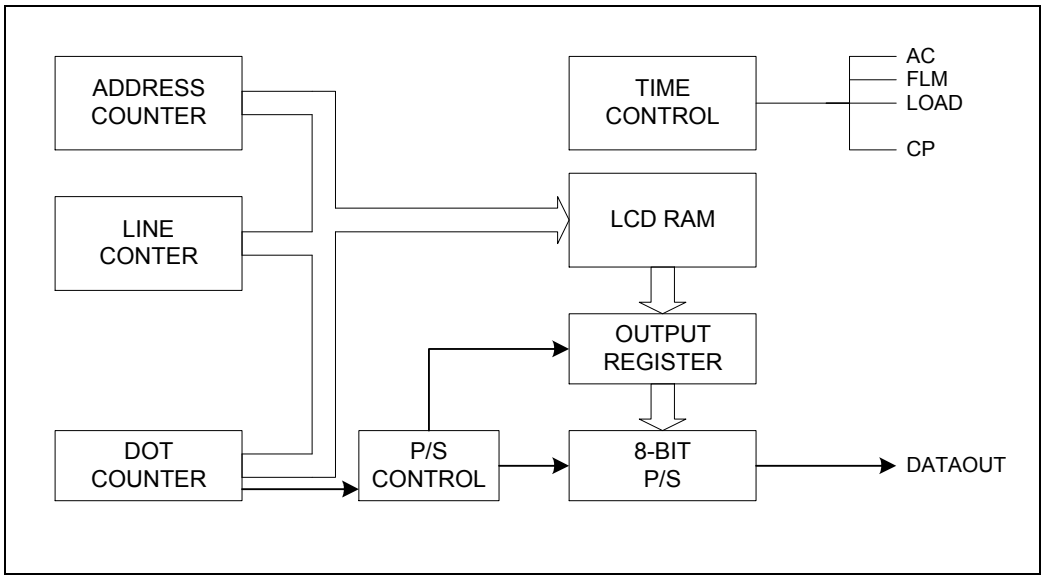
13.1 Features

Address: Straight binary
Attributes:

- Display inversion
- Display blank

Applicable LCD duties is 1/48 or 1/32.

13.2 LCD Block Diagram

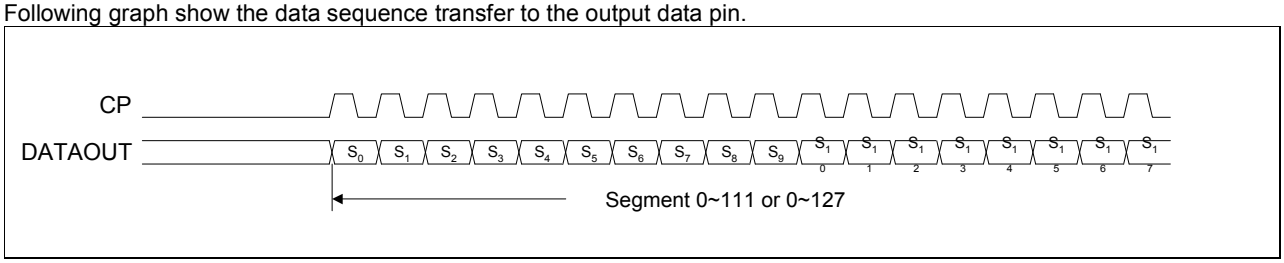


13.3 Pin Description

PAD	Description
AC	LCD alternating signal
FLM	First line mark for COM signal
LOAD	Load data into LCD driver's data latch
CP	Shift clock pulse for LCD driver
DATAOUT	Output serial data for LCD driver

13.4 Display Data to LCD Drivers

Following graph show the data sequence transfer to the output data pin.



13.4.1 Display RAM

There are two display mode for ST2100 controlled by register LCTL[4] : 48*112 or 32*128

Mode: 48 common * 112 segment (LCDM=0):
Address from $1000 to $12EF

	SEG0	SEG1	SEG2	SEG3	SEG4	SEG5		SEG111
Address	**1000H**	**1001H**	**1002H**	**1003H**	**1004H**	**1005H**	.....	**106FH**
COM0	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7		Bit7
COM1	Bit6	Bit6	Bit6	Bit6	Bit6	Bit6		Bit6
COM2	Bit5	Bit5	Bit5	Bit5	Bit5	Bit5		Bit5
COM3	Bit4	Bit4	Bit4	Bit4	Bit4	Bit4	.....	Bit4
COM4	Bit3	Bit3	Bit3	Bit3	Bit3	Bit3		Bit3
COM5	Bit2	Bit2	Bit2	Bit2	Bit2	Bit2		Bit2
COM6	Bit1	Bit1	Bit1	Bit1	Bit1	Bit1	.....	Bit1
COM7	Bit0	Bit0	Bit0	Bit0	Bit0	Bit0		Bit0
Address	**1080H**	**1081H**	**1082H**	**1083H**	**1084H**	**1085H**		**10EFH**
COM8	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7		Bit7
COM9	Bit6	Bit6	Bit6	Bit6	Bit6	Bit6	.....	Bit6
COM10	Bit5	Bit5	Bit5	Bit5	Bit5	Bit5		Bit5
COM11	Bit4	Bit4	Bit4	Bit4	Bit4	Bit4		Bit4
COM12	Bit3	Bit3	Bit3	Bit3	Bit3	Bit3	.....	Bit3
COM13	Bit2	Bit2	Bit2	Bit2	Bit2	Bit2		Bit2
COM14	Bit1	Bit1	Bit1	Bit1	Bit1	Bit1		Bit1
COM15	Bit0	Bit0	Bit0	Bit0	Bit0	Bit0	.....	Bit0
Address	**1100H**	**1101H**	**1102H**	**1103H**	**1104H**	**1105H**	.....	**116FH**
: : :	: :	: :	:					
Address	**1180H**	**1181H**	**1182H**	**1183H**	**1184H**	**1185H**	.....	**11EFH**
: : :	: :	: :	:					
Address	**1200H**	**1201H**	**1202H**	**1203H**	**1204H**	**1205H**	.....	**126FH**
: : :	: :	: :	:					
Address	**1280H**	**1281H**	**1282H**	**1283H**	**1284H**	**1285H**		**12EFH**
COM40	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7		Bit7
COM41	Bit6	Bit6	Bit6	Bit6	Bit6	Bit6	.....	Bit6
COM42	Bit5	Bit5	Bit5	Bit5	Bit5	Bit5		Bit5
COM43	Bit4	Bit4	Bit4	Bit4	Bit4	Bit4		Bit4
COM44	Bit3	Bit3	Bit3	Bit3	Bit3	Bit3	.....	Bit3
COM45	Bit2	Bit2	Bit2	Bit2	Bit2	Bit2		Bit2
COM46	Bit1	Bit1	Bit1	Bit1	Bit1	Bit1		Bit1
COM47	Bit0	Bit0	Bit0	Bit0	Bit0	Bit0	.....	Bit0

Notice: CAN use undefined RAM area ($1070~$107F, $10F0~$10FF, $1170~$117F, $11F0~$11FF) under different display mode.

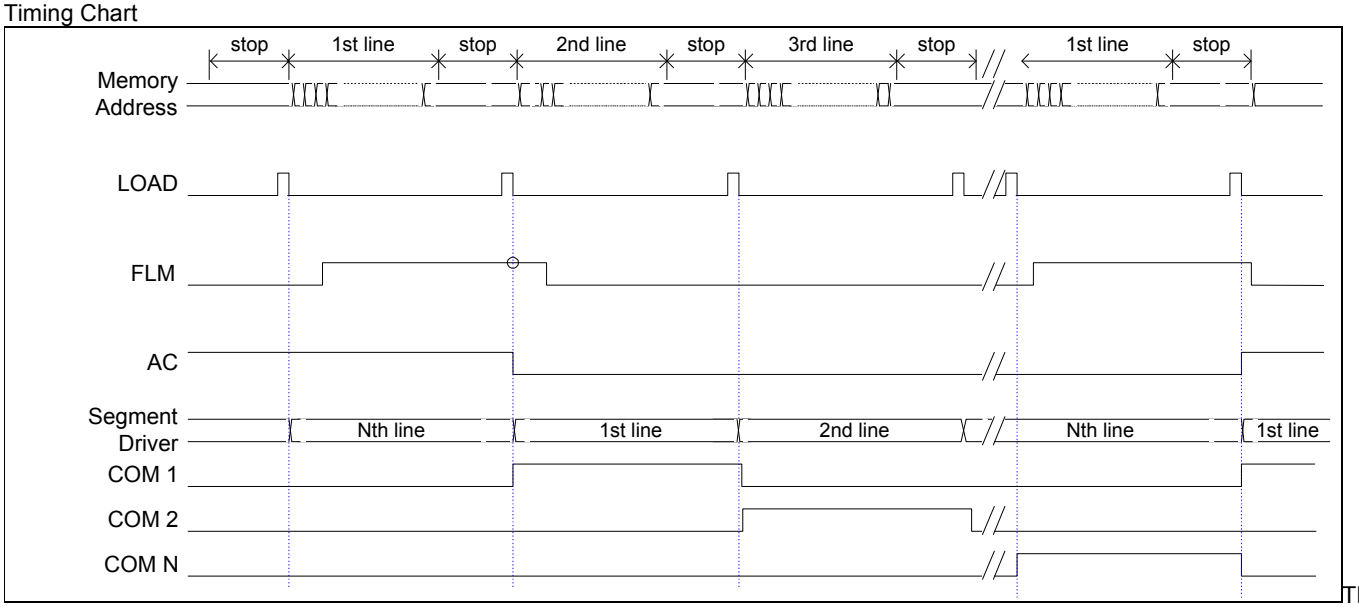
Mode: 32 common * 128 segment (LCDM=1):
Address from $1000 to $11FF

	SEG0	SEG1	SEG2	SEG3	SEG4	SEG5		SEG127
Address	**1000H**	**1001H**	**1002H**	**1003H**	**1004H**	**1005H**		**107FH**
COM0	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7	.....	Bit7
COM1	Bit6	Bit6	Bit6	Bit6	Bit6	Bit6		Bit6
COM2	Bit5	Bit5	Bit5	Bit5	Bit5	Bit5		Bit5
COM3	Bit4	Bit4	Bit4	Bit4	Bit4	Bit4	.....	Bit4
COM4	Bit3	Bit3	Bit3	Bit3	Bit3	Bit3		Bit3
COM5	Bit2	Bit2	Bit2	Bit2	Bit2	Bit2		Bit2
COM6	Bit1	Bit1	Bit1	Bit1	Bit1	Bit1	.....	Bit1
COM7	Bit0	Bit0	Bit0	Bit0	Bit0	Bit0		Bit0
Address	**1080H**	**1081H**	**1082H**	**1083H**	**1084H**	**1085H**		**10FFH**
COM8	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7	.....	Bit7
COM9	Bit6	Bit6	Bit6	Bit6	Bit6	Bit6		Bit6
COM10	Bit5	Bit5	Bit5	Bit5	Bit5	Bit5		Bit5
COM11	Bit4	Bit4	Bit4	Bit4	Bit4	Bit4	.....	Bit4
COM12	Bit3	Bit3	Bit3	Bit3	Bit3	Bit3		Bit3
COM13	Bit2	Bit2	Bit2	Bit2	Bit2	Bit2		Bit2
COM14	Bit1	Bit1	Bit1	Bit1	Bit1	Bit1	.....	Bit1
COM15	Bit0	Bit0	Bit0	Bit0	Bit0	Bit0		Bit0
Address	**1100H**	**1101H**	**1102H**	**1103H**	**1104H**	**1105H**	.....	**117FH**
: : :	: :	: :	:					
Address	**1180H**	**1181H**	**1182H**	**1183H**	**1184H**	**1185H**		**11FFH**
COM24	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7	.....	Bit7
COM25	Bit6	Bit6	Bit6	Bit6	Bit6	Bit6		Bit6
COM26	Bit5	Bit5	Bit5	Bit5	Bit5	Bit5		Bit5
COM27	Bit4	Bit4	Bit4	Bit4	Bit4	Bit4	.....	Bit4
COM28	Bit3	Bit3	Bit3	Bit3	Bit3	Bit3		Bit3
COM29	Bit2	Bit2	Bit2	Bit2	Bit2	Bit2		Bit2
COM30	Bit1	Bit1	Bit1	Bit1	Bit1	Bit1	.....	Bit1
COM31	Bit0	Bit0	Bit0	Bit0	Bit0	Bit0		Bit0

Notice: CAN use undefined RAM area ($1200~$126F, $1280~$12EF) under different display mode.

13.4.2 Frame Pulse, Frame, Load

Timing Chart



The most suitable FLM frequency is 50 to 70 Hz. (For 50Hz is about 20ms.)

f_{FLM} ≅ 50 ~ 70 Hz

f_{CP} = 112 * 48 * f_{FLM} = 268K ~ 376K

13.5 LCD control register

Address	Register	R/W	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	COMMENT
$3A	LCTL	W	LPWR	BLANK	REV	-	CTR[3]	CTR[2]	CTR[1]	CTR[0]	LCD Register

LPWR (LCTL[7])
Control LCD Driver power On/OFF
0: ON (Default)
1: OFF

BLANK (LCTL[6])
Control LCD panel display On/OFF
0: Display On
1: Display Off

REV (LCTL[5])
LCD Display Reverse
0: Normal display
1: Reverse display

CTR[3~0] (LCTL[3~0])
LCD contrast control

When DUTY = 1/48
0000: contrast maximum (level 13) (default)
0001: contrast maximum (level 13)
0010: contrast maximum (level 13)
0011: contrast maximum (level 13)
0100: contrast level 12
0101: contrast level 11
:
:
1111: contrast minimum (level 1)

When DUTY = 1/32
0000: contrast maximum (level 16) (default)
0001: contrast level 15
0010: contrast level 14
0011: contrast level 13
0100: contrast level 12
0101: contrast level 11
:
:
1111: contrast minimum (level 1)

13.6 LCD clock 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 clock control

		DUTY = 1/48	DUTY = 1/32	
000 :	RC / 32	: 23.25 frames/sec	30.5 frames/sec	(For RC = 4MHz)
001 :	RC / 16	: 46.5 frames/sec	61 frames/sec	(For RC = 4MHz)
010 :	RC / 8	: 93 frames/sec	122 frames/sec	(For RC = 4MHz)
011 :	RC / 4	: 186 frames/sec	244 frames/sec	(For RC = 4MHz)
100 :	RC / 2	: 372 frames/sec	488 frames/sec	(For RC = 4MHz)
101 :	X			
110 :	X			
111 :	X			

13.7 LCD configuration register

Address	Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default
$039	**LCFG**	W	LCFG[7]	LCFG[6]	LCFG[5]	LCFG[4]	LCFG[3]	LCFG[2]	LCFG[1]	LCFG[0]	???? ????

Bit 7~0 : LCFG[7~0] : LCD configuration control (write only, no preset value when power on)

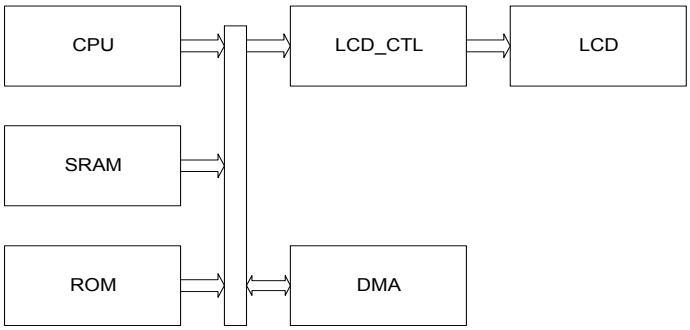
Address $39 register used to set LCD display dimension, write SEG-1 first, then COM-1 next.
Since it is a word control, always write two bytes into the address.
<Example>

```
LCD_SEG   EQU   111          ; Segment = 112
LCD_COM   EQU   47           ; Common = 48
          LDA   #LCD_SEG
          STA   $39
          LDA   #LCD_COM
          STA   $39
```

14. 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.**

14.1 Block Diagram



14.2 Control register

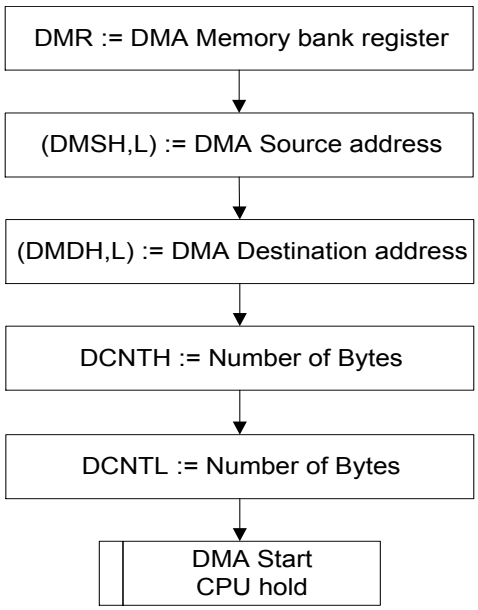
The control register is shown as followed:

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	-	-	-	DMAM	DCNT11	DCNT10	DCNT9	DCNT8	DMA Counter high byte
$033	DMR	R/W	DMR7	DMR6	DMR5	DMR4	DMR3	DMR2	DMR1	DMR0	DMA memory bank register

DCNT H[4]: DMAM 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. The maximum number of bytes DMA can move are 4K bytes.
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.

14.3 Programming Flowchart



14.4 Application program 1:

This program is fill “00” to $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
  :
  :
```

14.5 Application program 2:

This program is $1080~$12FF move 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
  :
  :
```

14.6 Application program 3:

This program is $8000~$803F move to $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
  :
  :
```

15. POWER DOWN MODE

The ST2100 has three power down modes: WAI-0, WAI-1 and STP. The instruction WAI will enable mode WAI-0 or WAI-1, which are controlled by WAIT(SYS[2]). The instruction WAI (WAI-0 and WAI-1 modes) can be wake-up by interrupt. However, the instruction of STP can only be wake-up by hardware reset.

15.1 WAI-0 Mode:

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

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

15.2 WAI-1 Mode:

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

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

15.3 STP Mode:

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

```
.
.
STP
.
.
```

TABLE 15-1:STATUS UNDER POWER DOWN MODE

(SYSCK source from 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 from 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

16. ELECTRICAL CHARACTERISTICS

16.1 Absolute Maximum Ratings*

DC Supply Voltage ---------- -0.3V to +7.5V

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.

16.2 DC Electrical Characteristics

Standard operation conditions: V_{DD} = 3.0V, GND = 0V, T_A = 25°C, OSCX = 32768Hz, OSC=4Mhz, unless otherwise specified

Parameter	Symbol	Min.	Typ.	Max.	Unit	Condition
Operating Voltage	V_{DD}	2.4	3	5.5	V	
Operating Current	I_{OP}		800		μA	All output pins unload, execute NOP instruction
Standby Current	I_{SB1}		2	*4*	μA	All output pins unload (WAIT mode)
Standby Current	I_{SB2}			1	μA	All output pins unload (STOP mode), LCD off
Input High Voltage	V_{IH}	0.7V_{DD}		V_{DD} + 0.3	V	PORT A, PORT B, PORT C
		0.85V_{DD}			V	$\overline{RESET}$, $\overline{INT}$
Input Low Voltage	V_{IL}	GND -0.3		0.3*V_{DD}	V	PORT A, PORT B, PORT C
				0.15*V_{DD}	V	$\overline{RESET}$, $\overline{INT}$
Pull-up resistance	R_{OH}	60	80	100	KΩ	PORTA, PORTB, PORTC (IOH = -37uA, VOH=0).
Output high voltage	V_{OH1}	0.7VDD			V	PORTA, PORTB, PORTC (IOH = -2.5mA).
Output low voltage	V_{OL1}			0.8	V	PORTA, PORTB, PORTC (IOL= 6mA).
Output high voltage	V_{OH2}	0.7 VDD			V	PSG, IOH = -2.5mA.
Output low voltage	V_{OL2}			0.8	V	PSG, IOL= 6mA.
Output high voltage	V_{OH3}	2.8			V	SEGx, Ioh = -800uA, C=50P,rise time < 200ns
Output low voltage	V_{OL3}			0.2	V	SEGx, Iol = 800uA
Output high voltage	V_{OH4}	0.7VDD			V	SEG0 – 3 to be output port, Ioh = -1mA.
Output low voltage	V_{OL4}			0.8	V	SEG0 – 3 to be output port, Iol = 2.5mA.
Output high voltage	V_{OH5}	0.7VDD			V	COM4 – 7 to be output port, Ioh = -1mA.
Output low voltage	V_{OL5}			0.8	V	COM4 – 7 to be output port, Iol = 2.5mA.
LCD Lighting	I_{LCD}		4	6	μA	Normal size LCD
Oscillation start time	T_{STT}		1	3	s	
Frequency stability	Δ F / F			1	PPM	[F(3.0)-F(2.5)]/F(3.0)(crystal oscillator)
Frequency variation	Δ F / F	-10		10	PPM	C1= 5 - 25P.

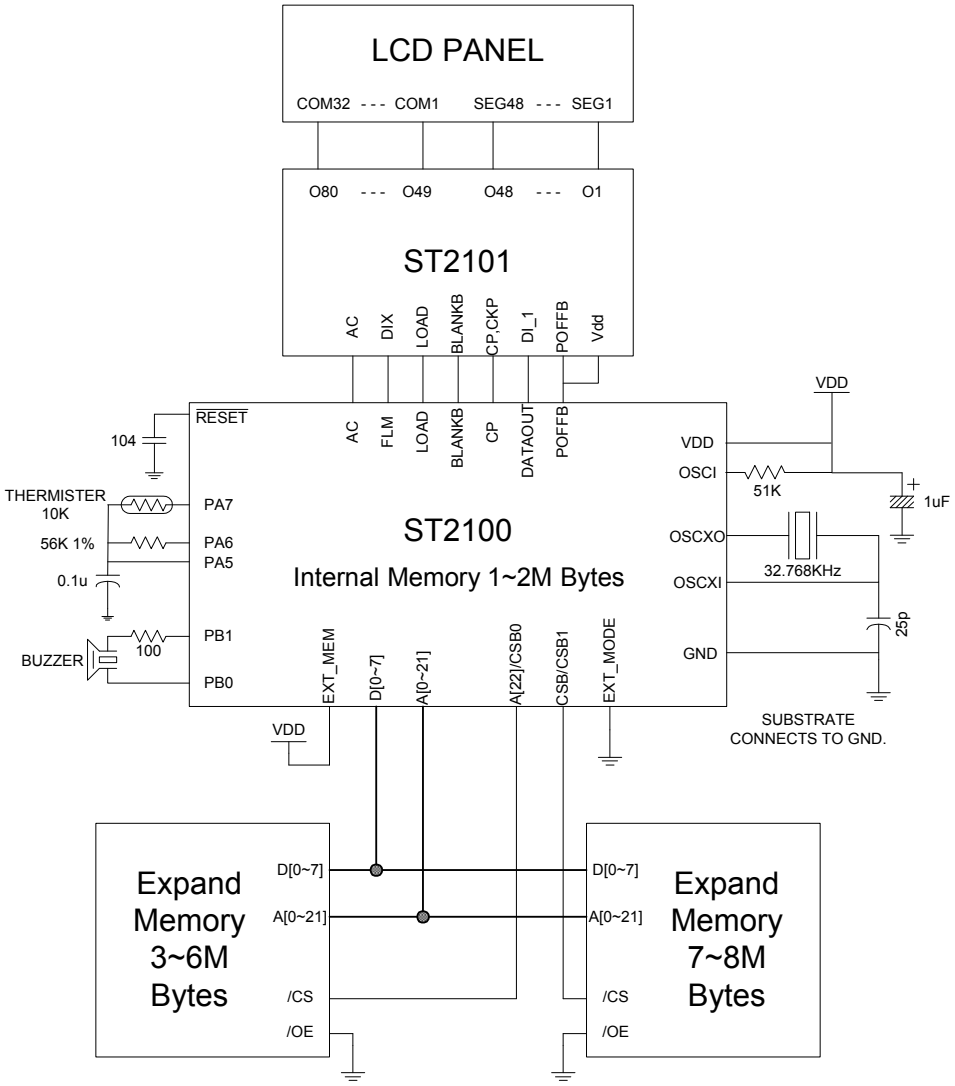
* The relationship between resistance and frequency (OSCI) To Be reference.

Resistance	Frequency
120K	2.0 MHz
51K	4.0 MHz

17. APPLICATION CIRCUITS

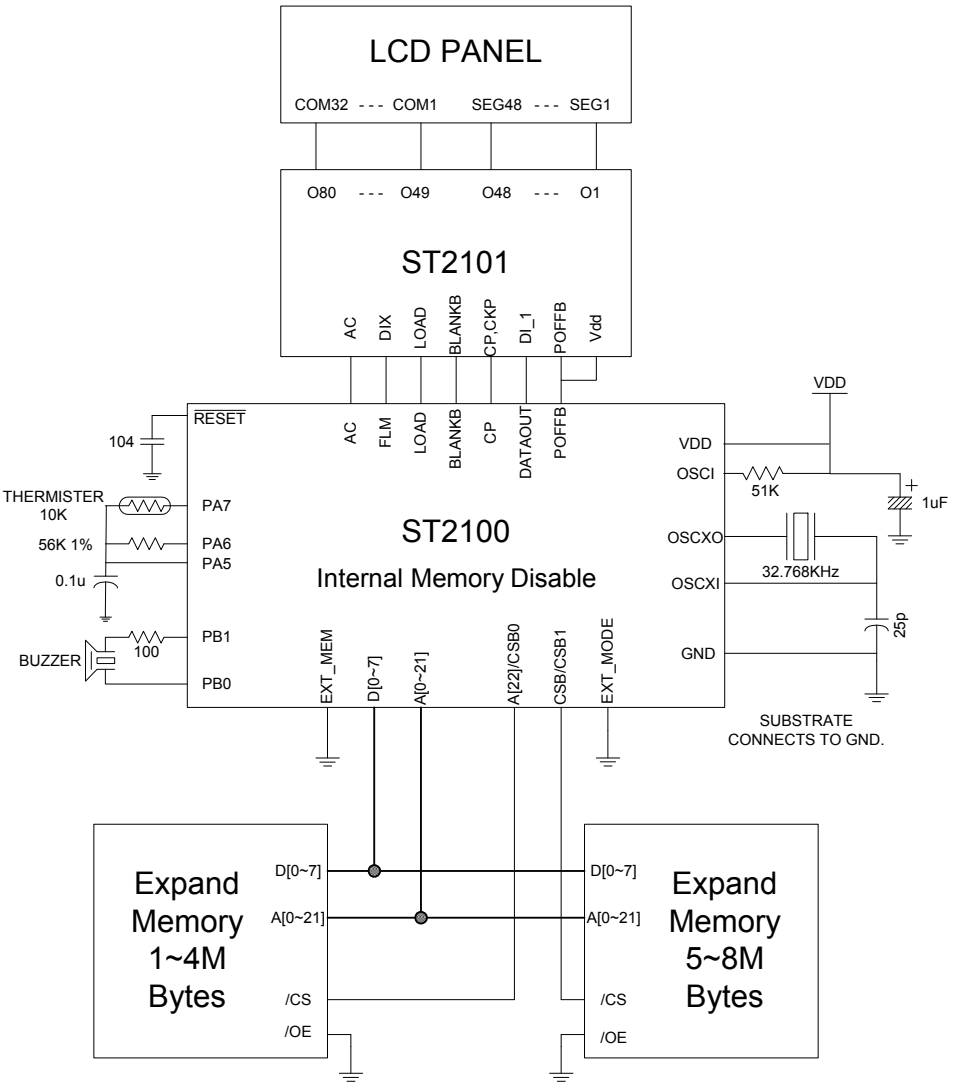
17.1 Application 1:

Internal memory = 1~2M bytes, Expand memory = 3~8M bytes.



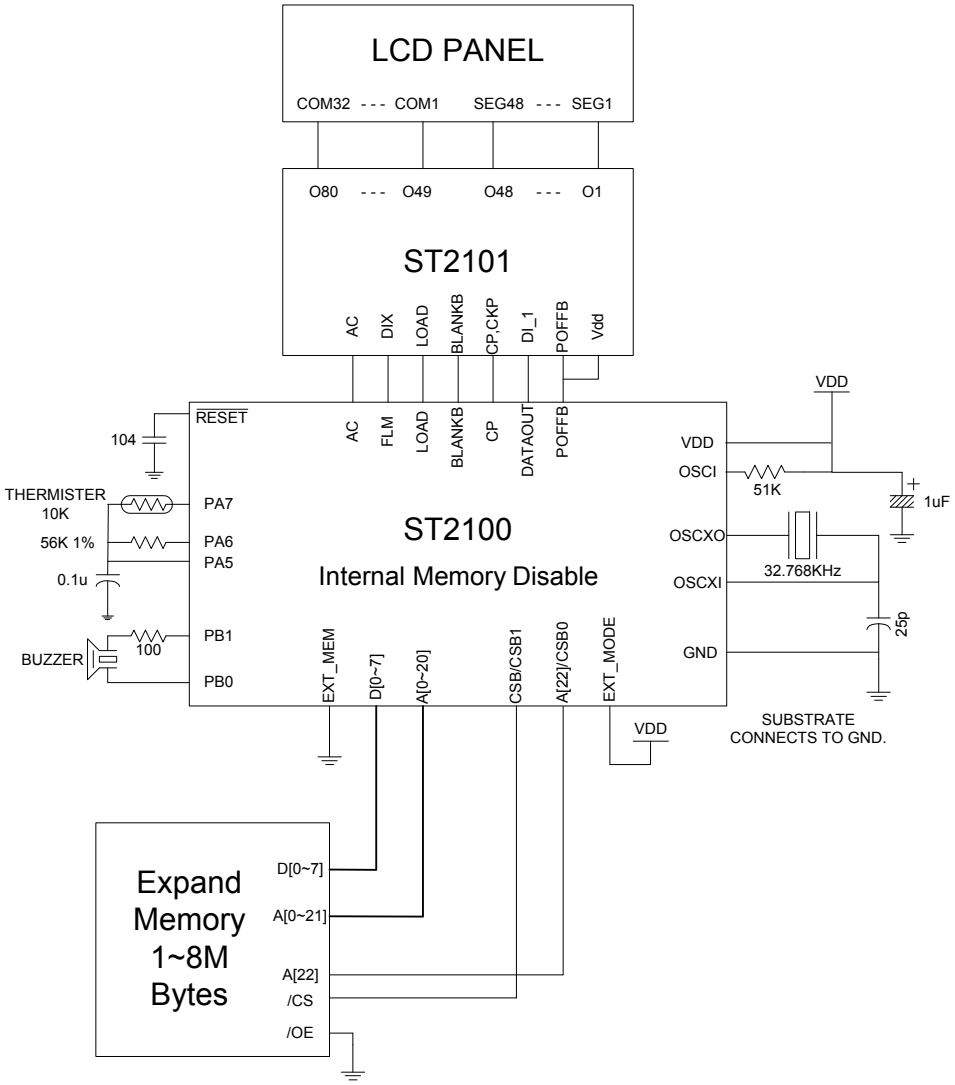
17.2 Application 2:

Internal memory = Disable, Expand memory = 1~8M bytes.



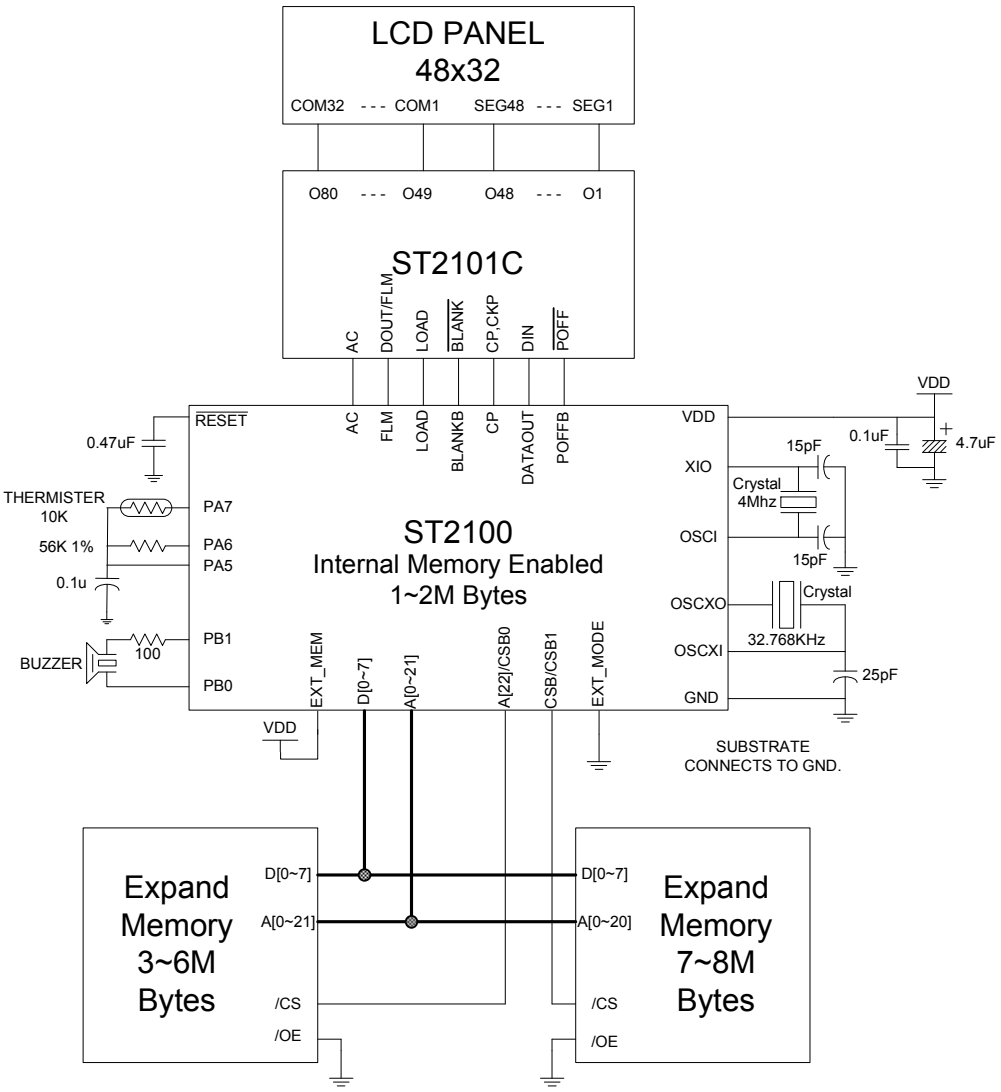
17.3 Application 3:

Internal memory = Disable, Expand memory = 1~8M bytes.



17.4 Application 5:

Internal memory = Enabled, Expand memory = 6M bytes.
Code option: Use resonator 4MHz



18. BONDING INFORMATION

unit: um

Pin No.	Name	X	Y	Pin No.	Name	X	Y
1	FLM	2341.1	2199.3	40	PC6	-2341.1	1625.8
2	LOAD	2341.1	2309.5	41	PC7	-2341.1	1515.6
3	POFFB	2341.1	2419.7	42	EXT_MODE	-2341.1	1405.4
4	TEST	2150.0	2561.8	43	EXT_MEM	-2341.1	1295.2
5	DATOUT	2039.8	2561.8	44	R/W	-2341.1	1185.0
6	CP	1929.6	2561.8	45	D7	-2341.1	1074.8
7	BLANKB	1819.4	2561.8	46	D6	-2341.1	964.6
8	AC	1709.2	2561.8	47	D5	-2341.1	854.4
9	XIO	1599.0	2561.8	48	D4	-2341.1	744.2
10	OSCI	1488.8	2561.8	49	D3	-2341.1	634.0
11	VDD	1378.6	2561.8	50	D2	-2341.1	523.8
12	VDD	1268.6	2561.8	51	D1	-2341.1	413.6
13	RESETB	1158.2	2561.8	52	D0	-2341.1	303.4
14	OSCXO	1048.0	2561.8	53	GND	-2341.1	193.2
15	OSCXI	937.8	2561.8	54	CSB/CSB1	-2341.1	83.0
16	GND	827.6	2561.8	55	A1	-2341.1	-27.2
17	PA0	717.4	2561.8	56	A2	-2341.1	-137.4
18	PA1	607.2	2561.8	57	A3	-2341.1	-247.6
19	PA2	497.0	2561.8	58	A4	-2341.1	-357.8
20	PA3	386.8	2561.8	59	A5	-2341.1	-468.0
21	PA4	276.6	2561.8	60	A6	-2341.1	-578.2
22	PA5	166.4	2561.8	61	A7	-2341.1	-688.4
23	PA6	56.2	2561.8	62	A8	-2341.1	-798.6
24	PA7	-54.1	2561.8	63	A18	-2341.1	-908.8
25	PB0	-164.3	2561.8	64	A19	-2341.1	-1019.0
26	PB1	-274.5	2561.8	65	A22/CSB0	-2341.1	-1129.2
27	PB2	-384.7	2561.8	66	GND	-2341.1	-1239.4
28	PB3	-494.9	2561.8	67	A21	-2341.1	-1349.6
29	GND	-605.1	2561.8	68	A20	-2341.1	-1459.8
30	PB4	-715.3	2561.8	69	A9	-2341.1	-1570.0
31	PB5	-825.5	2561.8	70	A10	-2341.1	-1680.2
32	PB6	-935.7	2561.8	71	A11	-2341.1	-1790.4
33	PB7	-1045.9	2561.8	72	A12	-2341.1	-1900.6
34	PC0	-1156.1	2561.8	73	A13	-2341.1	-2010.8
35	PC1	-1266.3	2561.8	74	A14	-2341.1	-2121.0
36	PC2	-1376.5	2561.8	75	A15	-2341.1	-2231.2
37	PC3	-1486.7	2561.8	76	A16	-2341.1	-2341.4
38	PC4	-2341.1	1846.2	77	A17	-2341.1	-2451.6
39	PC5	-2341.1	1736.0	78	A0	-2341.1	-2561.8

chip size = 4890um X 5330um
substrate connect to ground

19. REVISIONS

Version 1.72:
Page7 section 7.1 8M bytes ROM space
Page8 section 7.2.2 EXT_MEM mode

Version 1.7 - Page 20 Timer registers.
Page 37 LCD control register.

Version 1.6 - Page 48 Application circuits.

Version 1.5 - Page 1 Features.
- Page 2 Pad diagram.
- Page 4 Pad description.
- Page 7 Memory map.
- Page 44~47 Application circuits.

Version 1.4 - Page 1 Modify 4K RAM.
- Page 2 Pad diagram.
- Page 3 Pad description.
- Page 6 Modify 4K RAM.
- Page 8 Addition $39 LCD configuration register & modify LCTL only write & Modify 4K RAM.
- Page 9 Modify BRK interrupt priority => 8.
- Page 12 TABLE9-1:Pad number.
- Page 13 9.1.2~9.1.5 move to 9.2.
- Page 18 Modify oscillator & TABLE10-1:bit7 description.
- Page 28 Modify digital DAC & TABLE13-3 description.
- Page 36 Modify LCTL[3~0]:contrast level define.
- Page 37 Addition $39 LCD configuration register description.
- Page 38 Modify DMA description.
- Page 41 Modify power down mode & TABLE16-1 description.
- Page 42 Modify DC electrical characteristics.
- Page 43 Modify application circuits.

Version 1.3 ..... Page 8,38 7.3 & 15.2: DMSL(H),DMDL(H),DCNTL(H) only write, PRR,DRR,DMR can R/W. Before Read/Write you have to initial the PRR, DRR, DMR register when system reset.
Version 1.73 ..... modify pad description of DF, FLM
Version 1.8..... Add bonding information
Version 1.9..... Change pin 12 as Vdd (improve noise immunity)
Version 2.0..... Modify application circuit, connect Vdd of ST2101 to POFFB of ST2100
Version 2.1..... change LCD alternating signal pin name 'DF' into 'AC', correct AC,FLM description
Version 2.2..... split appendix section and pad allocation section
Version 2.3..... correct A22/CSB0, type error page2 and application circuit
Version 2.4..... Modify LCTL and LCK Description.
Version 2.6..... Page18 Modify PRES counter description.
Version 2.6c..... Modify memory mapping modes in page8