



Sitronix

ST2032

PRELIMINARY

8 BIT Microcontroller with 32K bytes ROM

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

1. FEATURES

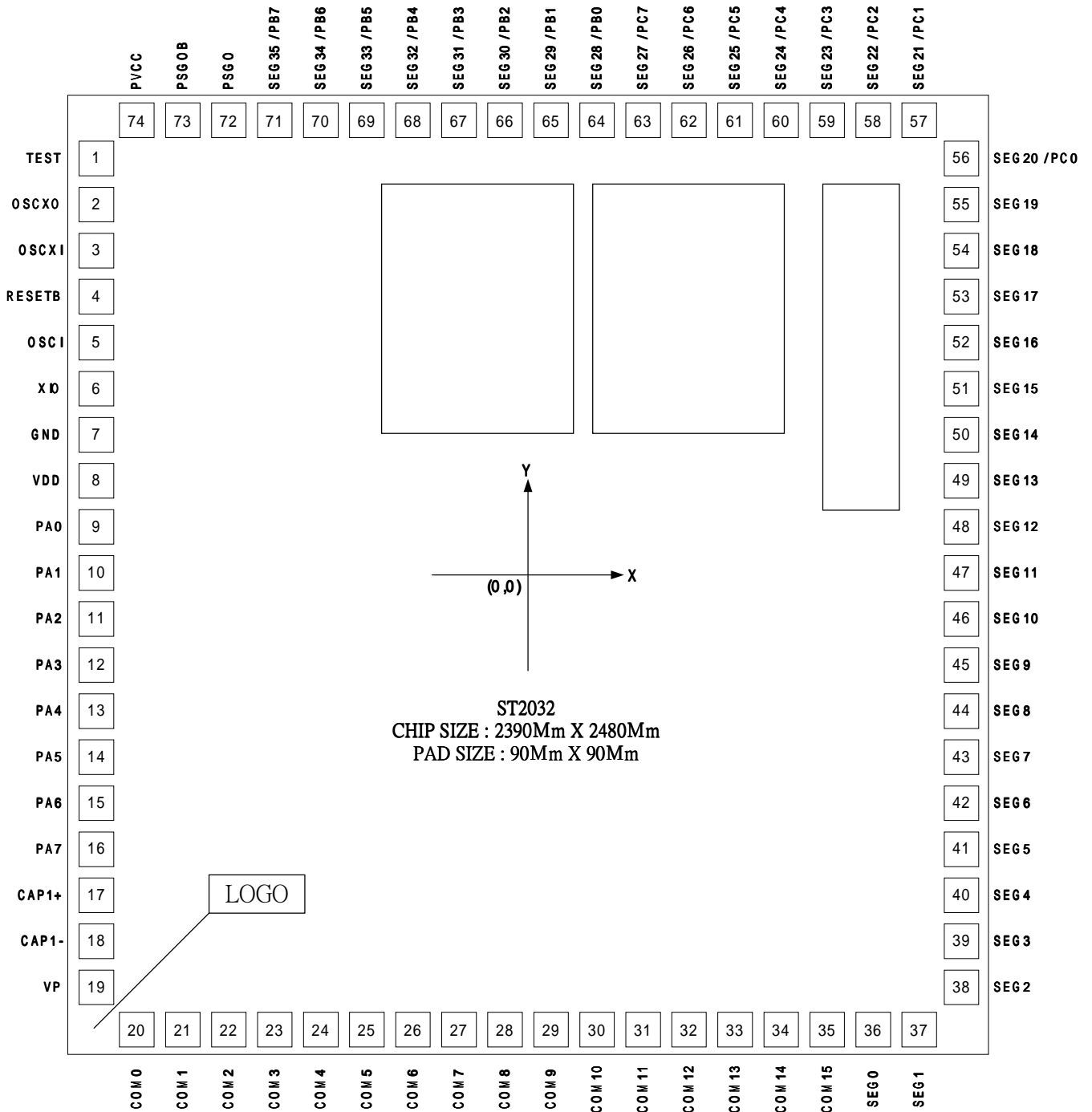
- Totally static 65C02S CPU
- ROM: 32K x 8-bit
- RAM: 1K x 8-bit
- Stack: Up to 128-level deep
- Operation voltage: 2.4V ~ 3.4V
- Built-in double DC-DC voltage converter for LCD driver
- I/O ports
 - 24 CMOS bidirectional bit programmable I/O pins, sixteen (Port-B/C) are shared with LCD drives
 - Bit 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[7~0] 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
- LCD controller/driver
 - Resolution: 20x16 ~ 36x16, maximum 576 dots
 - Two clock source options: RC and resonator oscillator
 - Internal bias resistors (1/5 bias) with 16-level driving strength control
 - Up to 12-level contrast control
 - Keyboard-scan function supported on 16 shared segment drives
- Programmable sound generator (PSG)
 - Two channels with three playing modes
 - Tone/noise generator
 - 16-level volume control
 - Dedicated outputs for directly connection to buzzer
- PWM DAC: Three modes up to 8-bit resolution
- Three power down modes:
 - WAI0 mode
 - WAI1 mode
 - STP mode

2. GENERAL DESCRIPTION

The ST2032 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. PAD CENTER COORDINATES

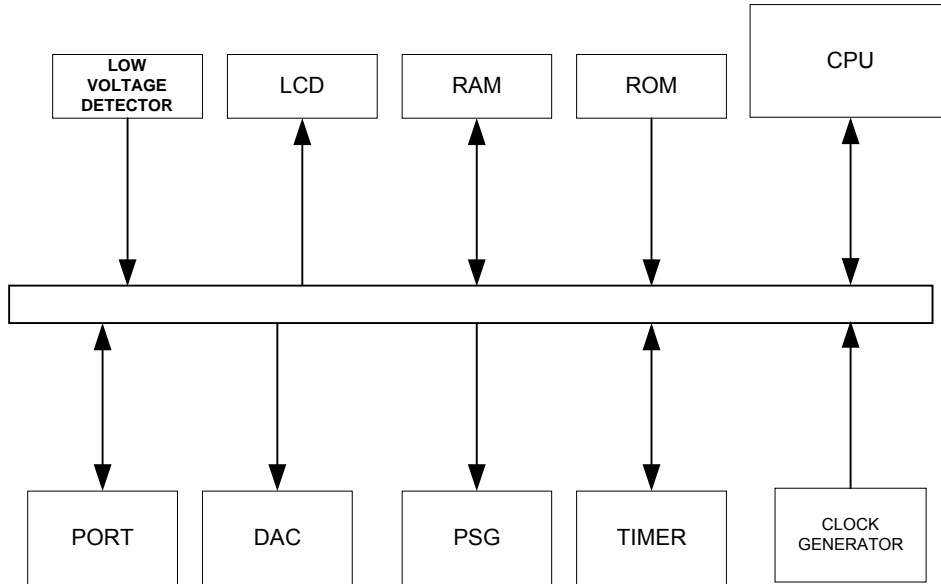
- Chip size: 2390 x 2480 (μm)
- Coordinate: Pad center (μm)
- Origin: Chip center
- Substrate connection: GND

Unit: μm

| PAD# | SIGNAL | Pad Center | | PAD PITCH | |
|------|------------|------------|----------|-----------|--------|
| | | Ml crons | | X | Y |
| | | X | Y | | |
| 1 | TEST | -1134.85 | 1066.95 | 105.1 | 117.95 |
| 2 | OSCX0 | -1134.85 | 946.95 | 0 | 120 |
| 3 | OSCXI | -1134.85 | 826.95 | 0 | 120 |
| 4 | RESETB | -1134.85 | 706.95 | 0 | 120 |
| 5 | OSCI | -1134.85 | 586.95 | 0 | 120 |
| 6 | XI0 | -1134.85 | 466.95 | 0 | 120 |
| 7 | GND | -1134.85 | 346.95 | 0 | 120 |
| 8 | VDD | -1134.85 | 226.95 | 0 | 120 |
| 9 | PA[0] | -1134.85 | 106.95 | 0 | 120 |
| 10 | PA[1] | -1134.85 | -13.05 | 0 | 120 |
| 11 | PA[2] | -1134.85 | -133.05 | 0 | 120 |
| 12 | PA[3] | -1134.85 | -253.05 | 0 | 120 |
| 13 | PA[4] | -1134.85 | -373.05 | 0 | 120 |
| 14 | PA[5] | -1134.85 | -493.05 | 0 | 120 |
| 15 | PA[6] | -1134.85 | -613.05 | 0 | 120 |
| 16 | PA[7] | -1134.85 | -733.05 | 0 | 120 |
| 17 | CAP1+ | -1134.85 | -853.05 | 0 | 120 |
| 18 | CAP1- | -1134.85 | -973.05 | 0 | 120 |
| 19 | VP | -1134.85 | -1093.05 | 0 | 120 |
| 20 | COMPAD[0] | -920.65 | -1173 | -214.2 | 79.95 |
| 21 | COMPAD[1] | -800.65 | -1173 | -120 | 0 |
| 22 | COMPAD[2] | -680.65 | -1173 | -120 | 0 |
| 23 | COMPAD[3] | -560.65 | -1173 | -120 | 0 |
| 24 | COMPAD[4] | -440.65 | -1173 | -120 | 0 |
| 25 | COMPAD[5] | -320.65 | -1173 | -120 | 0 |
| 26 | COMPAD[6] | -200.65 | -1173 | -120 | 0 |
| 27 | COMPAD[7] | -80.65 | -1173 | -120 | 0 |
| 28 | COMPAD[8] | 39.35 | -1173 | -120 | 0 |
| 29 | COMPAD[9] | 159.35 | -1173 | -120 | 0 |
| 30 | COMPAD[10] | 279.35 | -1173 | -120 | 0 |
| 31 | COMPAD[11] | 399.35 | -1173 | -120 | 0 |
| 32 | COMPAD[12] | 519.35 | -1173 | -120 | 0 |
| 33 | COMPAD[13] | 639.35 | -1173 | -120 | 0 |
| 34 | COMPAD[14] | 759.35 | -1173 | -120 | 0 |

| PAD# | SIGNAL | Pad Center | | PAD PITCH | |
|------|------------|------------|--------|-----------|-------|
| | | Mi crons | | X | Y |
| | | X | Y | | |
| 35 | COMPAD[15] | 879.35 | -1173 | -120 | 0 |
| 36 | SEGPAD[0] | 999.35 | -1173 | -120 | 0 |
| 37 | SEGPAD[1] | 1119.35 | -1153 | -120 | -20 |
| 38 | SEGPAD[2] | 1132.6 | -1019 | -13.25 | -134 |
| 39 | SEGPAD[3] | 1132.6 | -899 | 0 | -120 |
| 40 | SEGPAD[4] | 1132.6 | -779 | 0 | -120 |
| 41 | SEGPAD[5] | 1132.6 | -659 | 0 | -120 |
| 42 | SEGPAD[6] | 1132.6 | -539 | 0 | -120 |
| 43 | SEGPAD[7] | 1132.6 | -419 | 0 | -120 |
| 44 | SEGPAD[8] | 1132.6 | -299 | 0 | -120 |
| 45 | SEGPAD[9] | 1132.6 | -179 | 0 | -120 |
| 46 | SEGPAD[10] | 1132.6 | -59 | 0 | -120 |
| 47 | SEGPAD[11] | 1132.6 | 61 | 0 | -120 |
| 48 | SEGPAD[12] | 1132.6 | 181 | 0 | -120 |
| 49 | SEGPAD[13] | 1132.6 | 301 | 0 | -120 |
| 50 | SEGPAD[14] | 1132.6 | 421 | 0 | -120 |
| 51 | SEGPAD[15] | 1132.6 | 541 | 0 | -120 |
| 52 | SEGPAD[16] | 1132.6 | 661 | 0 | -120 |
| 53 | SEGPAD[17] | 1132.6 | 781 | 0 | -120 |
| 54 | SEGPAD[18] | 1132.6 | 901 | 0 | -120 |
| 55 | SEGPAD[19] | 1132.6 | 1021 | 0 | -120 |
| 56 | SEGPAD[20] | 1132.6 | 1141 | 0 | -120 |
| 57 | SEGPAD[21] | 1010.25 | 1184.9 | 122.35 | -43.9 |
| 58 | SEGPAD[22] | 890.25 | 1184.9 | 120 | 0 |
| 59 | SEGPAD[23] | 770.25 | 1184.9 | 120 | 0 |
| 60 | SEGPAD[24] | 650.25 | 1184.9 | 120 | 0 |
| 61 | SEGPAD[25] | 530.25 | 1184.9 | 120 | 0 |
| 62 | SEGPAD[26] | 410.25 | 1184.9 | 120 | 0 |
| 63 | SEGPAD[27] | 290.25 | 1184.9 | 120 | 0 |
| 64 | SEGPAD[28] | 170.25 | 1184.9 | 120 | 0 |
| 65 | SEGPAD[29] | 50.25 | 1184.9 | 120 | 0 |
| 66 | SEGPAD[30] | -69.75 | 1184.9 | 120 | 0 |
| 67 | SEGPAD[31] | -189.75 | 1184.9 | 120 | 0 |
| 68 | SEGPAD[32] | -309.75 | 1184.9 | 120 | 0 |
| 69 | SEGPAD[33] | -429.75 | 1184.9 | 120 | 0 |
| 70 | SEGPAD[34] | -549.75 | 1184.9 | 120 | 0 |
| 71 | SEGPAD[35] | -669.75 | 1184.9 | 120 | 0 |
| 72 | PSGO | -789.75 | 1184.9 | 120 | 0 |
| 73 | PSGOB | -909.75 | 1184.9 | 120 | 0 |
| 74 | PVCC | -1029.75 | 1184.9 | 120 | 0 |

5. BLOCK DIAGRAM



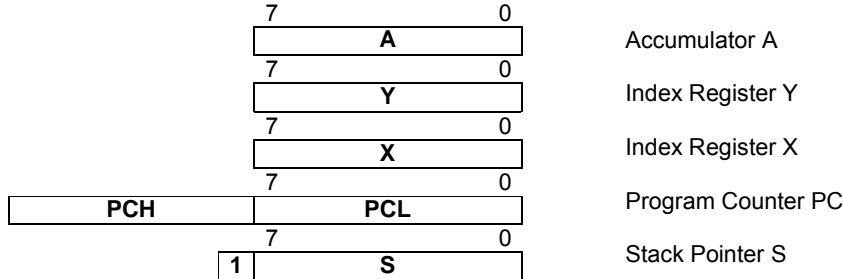
6. PAD DESCRIPTION

| Pin No. | Designation | I/O | Description |
|---------|---------------------------|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 20~35 | COM0~15 | O | LCD common drive output pins, drives 0~15 |
| 36~55 | SEG0~19 | O | LCD segment drive output pins, drives 0~19 |
| 9 | PA0 / INTX | I/O I I I | - Port-A bit programmable I/O - Edge-trigger Interrupt. - Transition-trigger Interrupt - Programmable Timer1 clock source |
| 10~16 | PA1~7 | I/O I | - Port-A bit programmable I/O - Transition-trigger Interrupt |
| 64~71 | SEG28/PB0~ SEG35/PB7 | I/O O | - Port-B bit programmable I/O - LCD segment drives 28~35 |
| 56~63 | SEG20/PC0~ SEG27/PC7 | I/O O | - Port-C bit programmable I/O - LCD segment drives 20~27 |
| 72,73 | PSGO,PSGOB | O | PSG/ PWM DAC Outputs |
| 2,3 | OSC XO, OSC XI | I/O | Low frequency crystal oscillator I/O pins. Connect to external 32768 Hz crystal. |
| 4 | $\overline{\text{RESET}}$ | I | Reset signal input (low active) |
| 5 | OSCI | I I | - RC oscillator input pin. Connected to external resistor - High frequency crystal/resonator oscillator input pin. Connect to external crystal/resonator. |
| 6 | XIO | O | - NC - High frequency crystal/resonator oscillator output pin. Connect to external crystal/resonator. |
| 7 | GND | P | Ground pin |
| 8 | VDD, AVDD | P | Power supply pin, Analogy Power supply pin |
| 17 | CAP1+ | I/O | Connect to booster capacitor positive(+) terminal |
| 18 | CAP1- | I/O | Connect to booster capacitor negative(-) terminal |
| 19 | VP | O | Voltage output of booster circuit |
| 1 | TEST | I | Chip test function. Leave it open. |
| 74 | PVCC | I | PSG power input |

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

7. 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 which 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 step 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 7-1

TABLE 7-1 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 3: D : Decimal mode flag 1 = Decimal mode 0 = Binary mode | | | |
| Bit 6: V : Overflow of signed Arithmetic flag 1 = Negative 0 = Positive | | | | Bit 2: I : Interrupt disable flag 1 = Interrupt disable 0 = Interrupt enable | | | |
| | | | | Bit 1: Z : Zero flag 1 = Zero 0 = Non zero | | | |
| Bit 4: B : BRK interrupt flag 1 = BRK interrupt occur 0 = Non BRK interrupt occur | | | | Bit 0: C : Carry flag 1 = Carry 0 = Non carry | | | |

8. MEMORY CONFIGURATION

8.1 Memory map

ST2032 builds in 32K bytes ROM and 1K bytes RAM. The internal ROM can be used as data memory or program memory. PRR is the Program ROM Bank Register. The logical program ROM address is from \$4000 to \$7FFF (16K bytes) is for logical program ROM address.

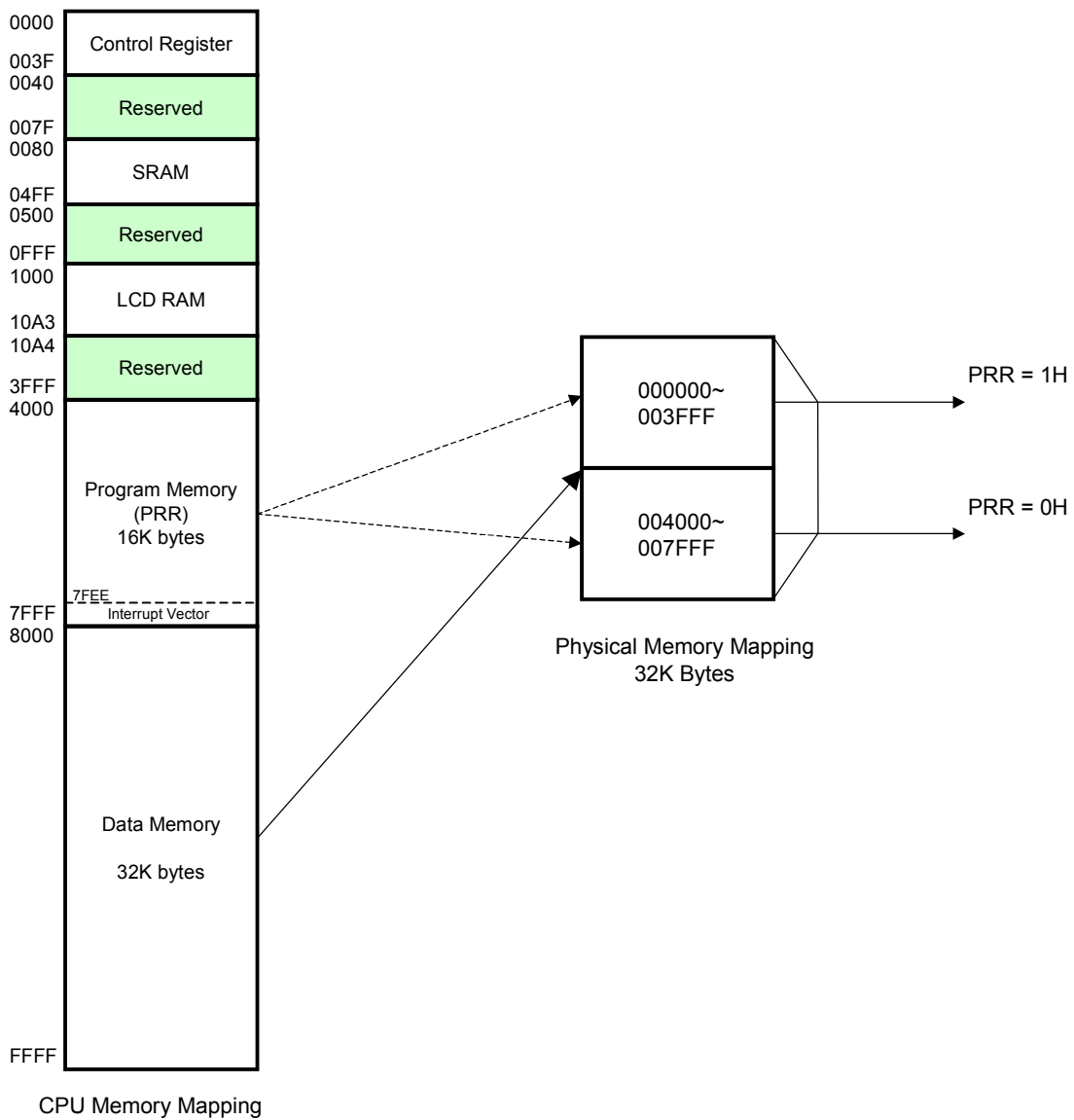


FIGURE 8-1 Memory Mapping of ST2032

8.2 ROM

8.2.1 Bank Description

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

FIGURE 8-2 ROM Bank Selection Registers (\$31~\$32)

| Address | Register | R/W | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---------|----------|-----|------|------|------|------|------|------|------|------|
| PRR | \$31 | RW | - | - | - | - | - | - | - | PRR0 |

8.3 RAM

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

8.3.1 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 | - | - | - | - | - | 100 - - - - |
| \$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 | COEN | 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 |
| \$030 | SYS | R/W | XSEL | OSTP | XSTP | XBAK | WSKP | WAIT | - | LVDET | 0000 00 - 0 |
| \$031 | PRR | R/W | - | - | - | - | - | - | - | PRR[0] | - - - - - - 0 |
| \$039 | LSEL | R/W | - | - | - | LSEL[4] | LSEL[3] | LSEL[2] | LSEL[1] | LSEL[0] | - - - 1 1111 |
| \$03A | LCTL | R/W | LPWR | BLANK | REV | SCAN | CTR[3] | CTR[2] | CTR[1] | CTR[0] | 1000 0000 |
| \$03B | LCK | R/W | DRV[3] | DRV[2] | DRV[1] | DRV[0] | - | LCK[2] | LCK[1] | LCK[0] | 1111 - 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.

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

8.3.2 Data RAM (\$0080~\$04FF)

Data RAM are organized in 1K bytes from \$0080~\$04FF.

8.3.3 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.

8.3.4 LCD Frame Buffer (\$1000~\$10A3)

LCD frame buffer is accessible by both read/write instructions and LCD controller. Note that this area can also be used as data memory. Each pixel of LCD panel is directly mapped into this area. Refer to section 15.4 for the detail mapping.

9. INTERRUPTS

9.1 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 at least of two clock cycles will cease ST2032 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 9-1.

TABLE 9-1 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 |

9.2 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 9-2 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: IRBT : Base Timer Interrupt Request bit 1 = Time base interrupt occurs 0 = Time base interrupt doesn't occur | | | | | | Bit 2: IRT0 : Timer0 Interrupt Request bit 1 = Timer0 overflow interrupt occurs 0 = Timer0 overflow interrupt doesn't occur | | | | | |
| Bit 4: IRPT : Port-A Interrupt Request bit 1 = Port-A transition interrupt occurs 0 = Port-A transition interrupt doesn't occur | | | | | | Bit 1: IRDAC : DAC reload Interrupt Request bit 1 = DAC time out interrupt occurs 0 = DAC time out interrupt doesn't occur | | | | | |
| Bit 3: IRT1 : Timer1 Interrupt Request bit 1 = Timer1 overflow interrupt occurs 0 = Timer1 overflow interrupt doesn't occur | | | | | | Bit 0: IRX : INTX Interrupt Request bit 1 = INTX edge interrupt occurs 0 = INTX edge interrupt doesn't occur | | | | | |

TABLE 9-3 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 2: IET0 : Timer0 Interrupt Enable bit 1 = Timer0 overflow interrupt enable 0 = Timer0 overflow interrupt disable | | | | | |
| Bit 4: IEPT : Port-A Interrupt Enable bit 1 = Port-A transition interrupt enable 0 = Port-A transition interrupt disable | | | | | | Bit 1: IEDAC : DAC reload Interrupt Enable bit 1 = DAC time out interrupt enable 0 = DAC time out interrupt disable | | | | | |
| Bit 3: IET1 : Timer1 Interrupt Enable bit 1 = Timer1 overflow interrupt enable 0 = Timer1 overflow interrupt disable | | | | | | Bit 0: IEX : INTX Interrupt Enable bit 1 = INTX edge interrupt enable 0 = INTX edge interrupt disable | | | | | |

10. I/O PORTS

10.1 Description

ST2032 can supply total 24 GPIOs divided into three I/O ports, Port-A, Port-B, and Port-C. Besides I/O function, Port-B/C can

also be used as LCD segment drives. For detail pin assignment, please refer to TABLE 10-1

TABLE 10-1 I/O Description

| PORT NAME | PAD NAME | PAD NUMBER | PIN TYPE | FEATURE |
|-----------|-----------|------------|----------|-------------------------------|
| Port-A | PA0/INTX | 9 | I/O | Programmable input/output pin |
| | PA1 | 10 | I/O | |
| | PA2 | 11 | I/O | |
| | PA3 | 12 | I/O | |
| | PA4 | 13 | I/O | |
| | PA5 | 14 | I/O | |
| | PA6 | 15 | I/O | |
| | PA7 | 16 | I/O | |
| Port-B | SEG28/PB0 | 64 | I/O | Programmable input/output pin |
| | SEG29/PB1 | 65 | I/O | |
| | SEG30/PB2 | 66 | I/O | |
| | SEG31/PB3 | 67 | I/O | |
| | SEG32/PB4 | 68 | I/O | |
| | SEG33/PB5 | 69 | I/O | |
| | SEG34/PB6 | 70 | I/O | |
| | SEG35/PB7 | 71 | I/O | |
| Port-C | SEG20/PC0 | 56 | I/O | Programmable input/output pin |
| | SEG21/PC1 | 57 | I/O | |
| | SEG22/PC2 | 58 | I/O | |
| | SEG23/PC3 | 59 | I/O | |
| | SEG24/PC4 | 60 | I/O | |
| | SEG25/PC5 | 61 | I/O | |
| | SEG26/PC6 | 62 | I/O | |
| | SEG27/PC7 | 63 | I/O | |

10.2 Port-A

10.2.1 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 resistor 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 10-2 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 | - | - | - | - | - | 100 - - - - - |
| \$03C | IREQ | R/W | - | - | IRBT | IRPT | IRT1 | IRT0 | IRDAC | IRX | - - 00 0000 |
| \$03E | IENA | R/W | - | - | IEBT | IEPT | IET1 | IET0 | IEDAC | IEX | - - 00 0000 |

10.2.2 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 10-3 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 | | | | | | | | | | | |

10.2.3 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 10-1 for the block description.

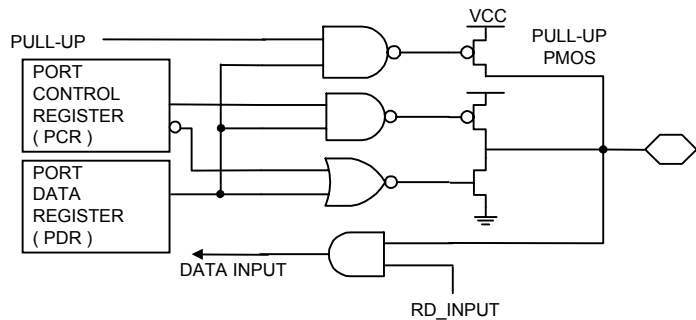


FIGURE 10-1 Port-A Block Diagram

TABLE 10-4 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 | - | - | - | - | - | 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 | | | | | | | | | | | |

10.2.4 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

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.

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 10-2 for the block diagram.

Example:

```

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

```

Interrupt subroutine

```

:
:
LDA <PA ; Keep last state.
RTI

```

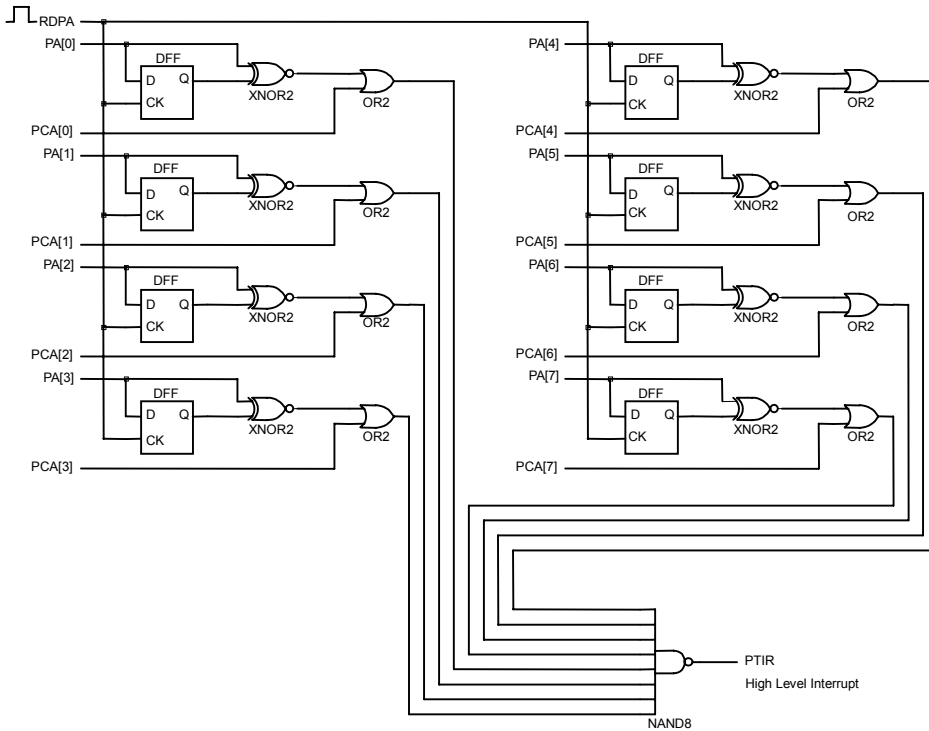


FIGURE 10-2 Port Interrupt Logic Diagram

10.2.5 Port-A Interrupt De-bounce

ST2032 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 OSCX as the sampling clock. The de-bounce time is **OSCX x 512 cycles (about 16 ms)**. Data filtered by de-bounce presents a stable state, then the interrupt can be issued.

TABLE 10-5 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 | - | - | - | - | - | 100 - - - - - |

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

10.2.6 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 10-3. 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
.
.
    
```

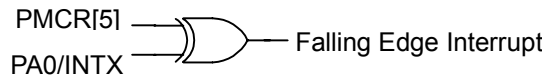


FIGURE 10-3 INTX Logic Diagram

10.3 Port-B and Port-C

10.3.1 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. All of the 16

I/Os can change into LCD segment drives. Control register LSEL specifies which of these I/Os are LCD drives.

TABLE 10-6 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 | - | - | - | - | - | 100 - - - - |
| \$039 | LSEL | R/W | - | - | - | LSEL[4] | LSEL[3] | LSEL[2] | LSEL[1] | LSEL[0] | - - - 1 1111 |

10.3.2 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 10-7 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: PCB[7~0] : Port-B directional bits 1 = Output mode 0 = Input mode | | | | | | | | | | | |

TABLE 10-8 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: PCC[7~0] : Port-C directional bits 1 = Output mode 0 = Input mode | | | | | | | | | | | |

10.3.3 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 10-4 for the block description.

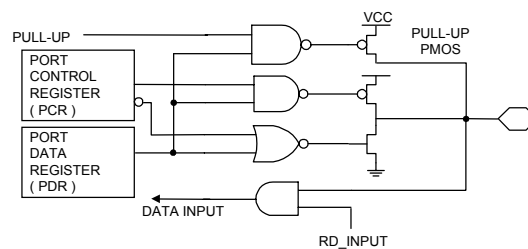


FIGURE 10-4 Port-B and Port-C Block Diagram

TABLE 10-9 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 | - | - | - | - | - | 100 - - - - |
| Bit 7: PULL : Enable all pull-up functions bit 1 = Enable pull-up function 0 = Disable pull-up function | | | | | | | | | | | |

11. OSCILLATOR

ST2032 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 11-1 System Control Register (SYS)

| Address | Name | R/W | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Default |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|-----|-------|-------|-------|-------|-------|-------|-------|-------|-----------|
| \$030 | SYS | R/W | XSEL | OSTP | XSTP | XBAK | WSKP | WAIT | - | LVDET | 0000 00-0 |
| <p>Bit 7: XSEL : System clock (SYSCK) select (write) / confirm (read) bit 1 = OSCX 0 = OSC</p> <p>Bit 6: OSTP : OSC stop control bit 1 = Disable OSC 0 = Enable OSC</p> <p>Bit 5: XSTP : OSCX stop control bit 1 = Disable OSCX 0 = Enable OSCX</p> <p>Bit 4: XBAK : OSCX driver heavy load bit 1 = OSCX normal load 0 = OSCX heavy load</p> | | | | | | | | | | | |

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.

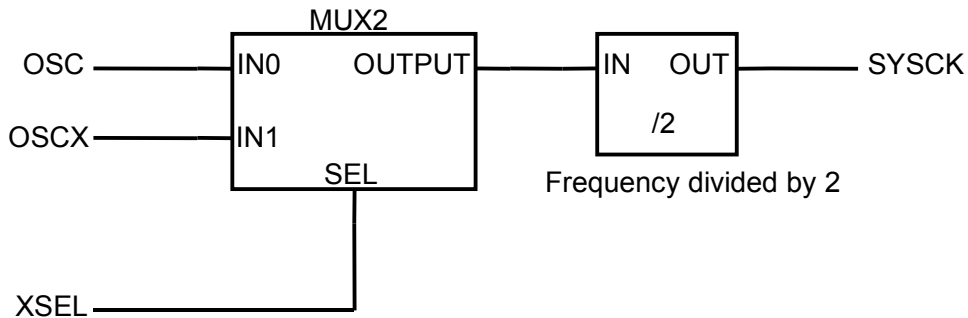


FIGURE 11-1 System Clock Diagram

12. TIMER/EVENT COUNTER

12.1 Prescaler

12.1.1 Function Description

The ST2032 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 12-1

TABLE 12-1 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 | - | - | - | - | - | 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 |

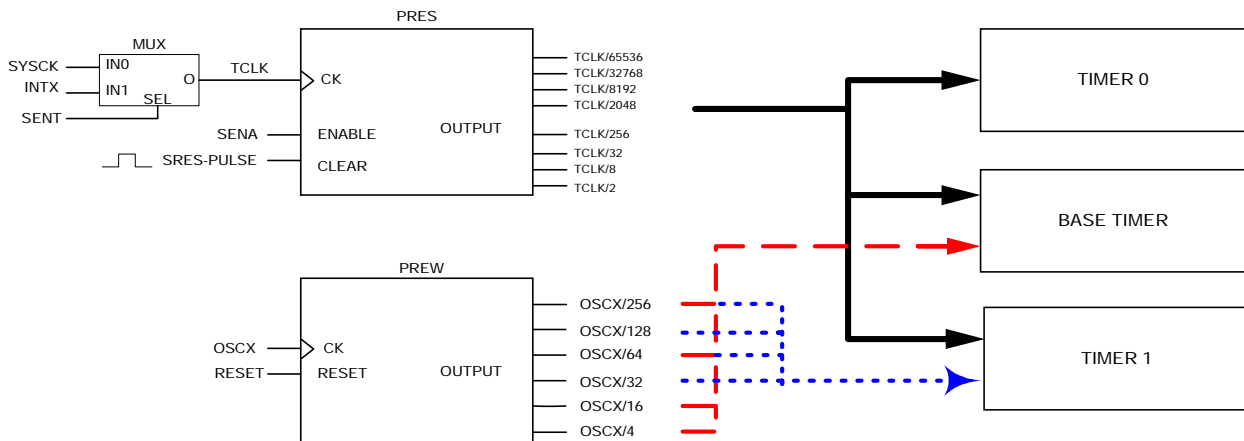


FIGURE 12-1 Structure Of Two Prescalers

12.1.2 PRES

The prescaler PRES is an 8-bits counter as shown in FIGURE 12-1. 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 12-2 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 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"

12.1.3 PREW

The prescaler PREW is an 8-bits counter as shown in FIGURE 12-1. PREW provides four clocks source for base timer and

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

12.2 Base timer

12.2.1 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 12-2

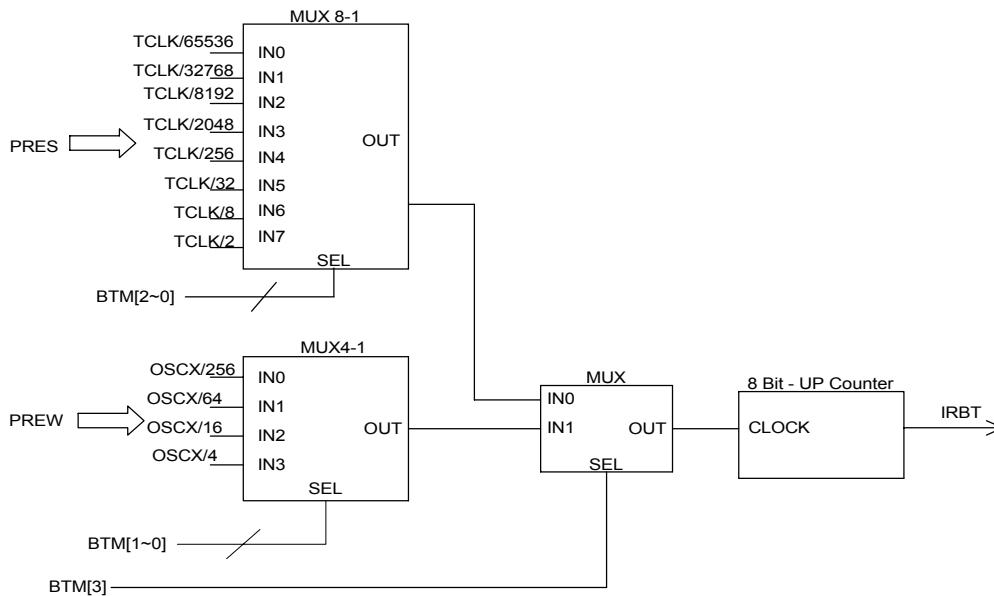


FIGURE 12-2 Structure Of Base Timer

12.2.2 Base Timer Clock Source Control

Several clock sources can be selected for Base Timer. Please refer to TABLE 12-3

TABLE 12-3 Clock Sources Of Base Timer

| * SENA | BTM[3] | BTM[2] | BTM[1] | BTM[0] | Base Timer source clock |
|--------|--------|--------|--------|--------|-------------------------|
| 0 | 0 | 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]).

12.3 Timer 0

12.3.1 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 12-3.

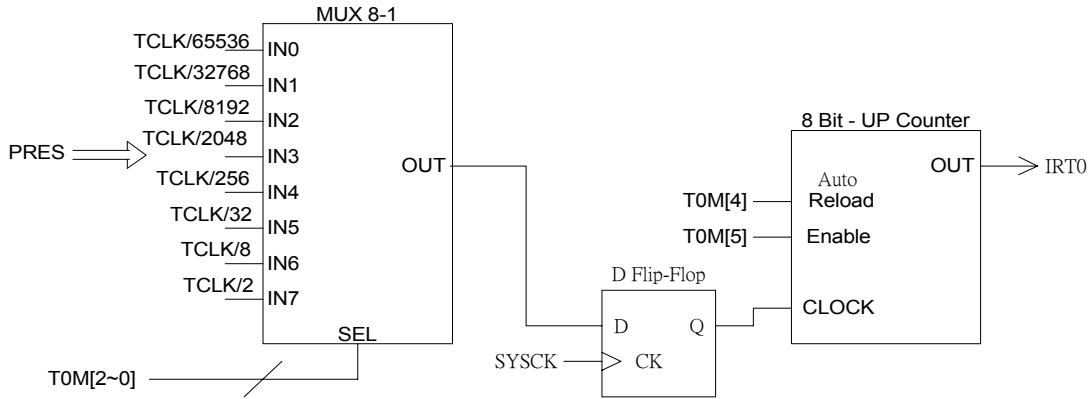


FIGURE 12-3 Timer0 Structure

12.3.2 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 12-4.

TABLE 12-4 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 12-5 Timer0 Register (TOC)

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

12.4 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 12-4.

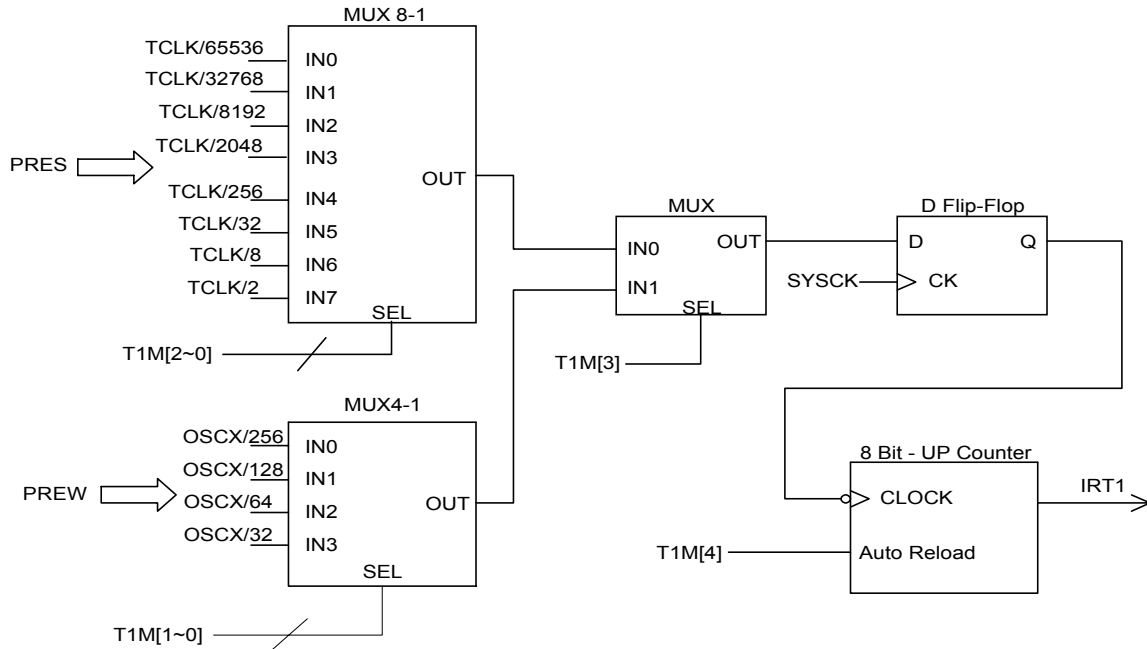


FIGURE 12-4 Timer1 Structure

TABLE 12-6 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 12-7 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

13. PSG

13.1 Function description

The built-in dual channel Programmable Sound Generator (PSG) is controlled by registers. 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 13-2 and the PSG clock source is shown in

FIGURE 13-1. ST2032 has three playing modes. First is that both channel0 (CH0) and channel1 (CH1) output square type tones. Second is CH0 outputs square tone, and CH1 outputs noise. Third mode is PWM DAC mode. Sounds of two channels are mixed into one signal and are outputted in the form of digital waveform from two pins, PSGO/PSGOB. Therefore one AC waveform can be performed.

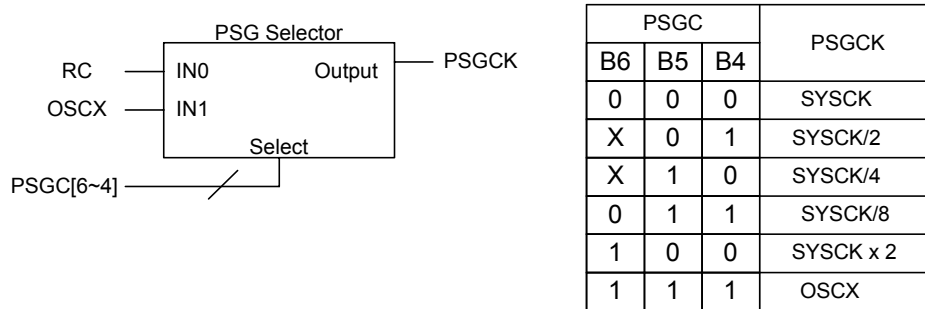


FIGURE 13-1 PSG Clock Source Control

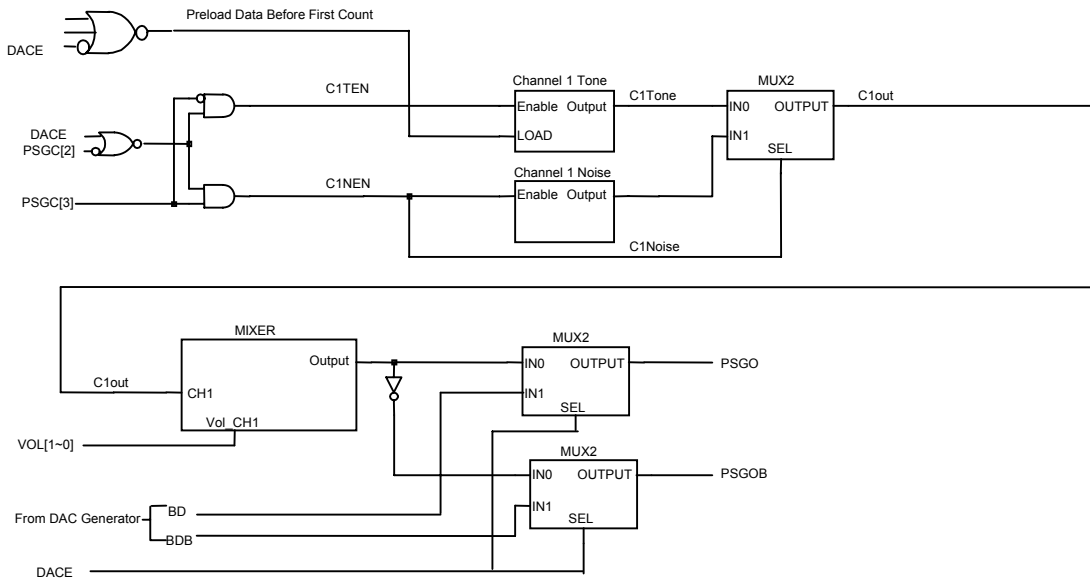


FIGURE 13-2 PSG Block Diagram

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

TABLE 13-2 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 |

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)

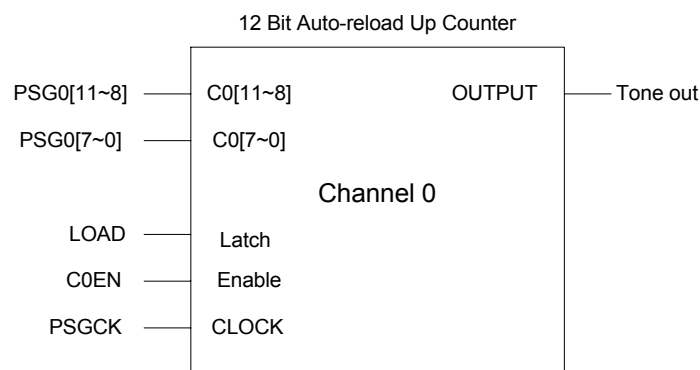
Note: If single channel is enable, then PSG volume control can be double. (16 + 16 = 32 level volume control)

13.2 Tone Generator

13.2.1 General Description

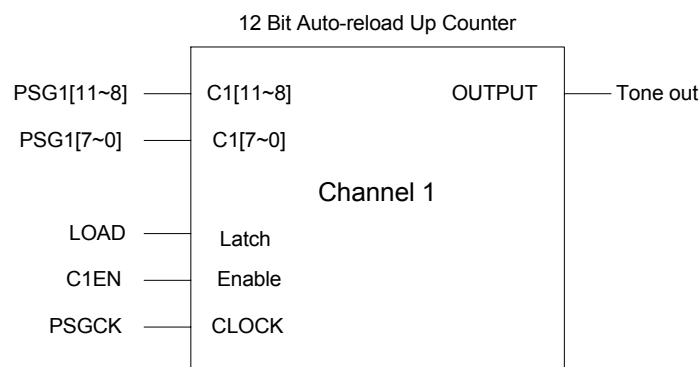
The tone frequency is decided by PSGCK and 12-bit programmable divider (PSG[11~0]). Please refer to

FIGURE 13-3 and FIGURE 13-4.



$$\text{Frequency of Channel 0 Tone} = \text{PSGCK} / (1000\text{H} - \text{PCH0}[11\sim0]) / 2$$

FIGURE 13-3 Tone Generator Channel 0



$$\text{Frequency of Channel 1 Tone} = \text{PSGCK} / (1000\text{H} - \text{PCH1}[11\sim0]) / 2$$

FIGURE 13-4 Tone Generator Channel 1

13.2.2 PSG Tone Programming

Tone or DAC function is defined by register DACE. Write to function. Noise or tone function is selected by PRBS. C1EN will enable tone generator when PSG is in tone

TABLE 13-3 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

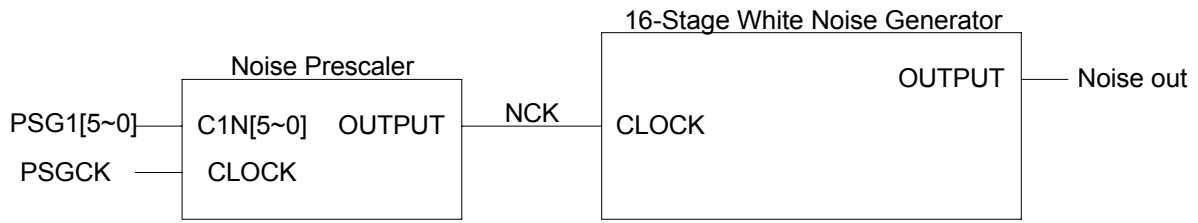
Bit 3: **PRBS** : Tone or Noise generator selection bit
 1 = Noise generator
 0 = Tone generator

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

13.3 Noise Generator Control

13.3.1 General description

Noise generator is shown in FIGURE 13-5, which base frequency is controlled by PSG1[5~0].



$$\text{NCK Frequency} = \text{PSGCK} / (40\text{H-PCH1}[5\sim 0])$$

FIGURE 13-5 Noise Generator

13.3.2 Noise Generator Programming

DACE defines noise or DAC function. Writing a “1” to C1EN will enable noise generator when PSG is in noise mode.

13.4 PSG Application Circuit

Sounds of two channels are modulated by PSGCK and combine together into one AC signal. Then it outputs on

PSGO and PSGOB. Positive part of the AC signal is output from PSGO while the negative part is from PSGOB.

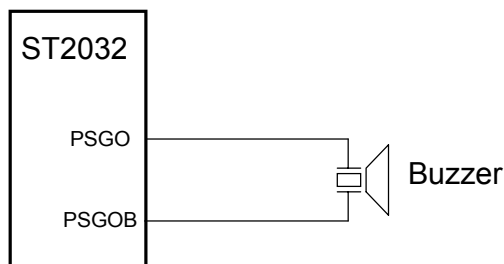


FIGURE 13-6 PSG Application Circuit

14. PWM DAC

14.1 PWM DAC Structure

A built-in PWM DAC is for analog sampling data or voice signals. The structure of DAC is shown in TABLE 14-1. There is an interrupt signal from DAC to CPU whenever

DAC data update is needed and the same signal will decide the sampling rate of voice. In DAC mode, the frequency of RC oscillator can't be less than 2M Hz.

TABLE 14-1 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 |
|---------|-------|-----|---------|---------|---------|---------|----------|----------|---------|---------|--------------|
| \$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 14-2 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 14-3 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.

14.2 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:

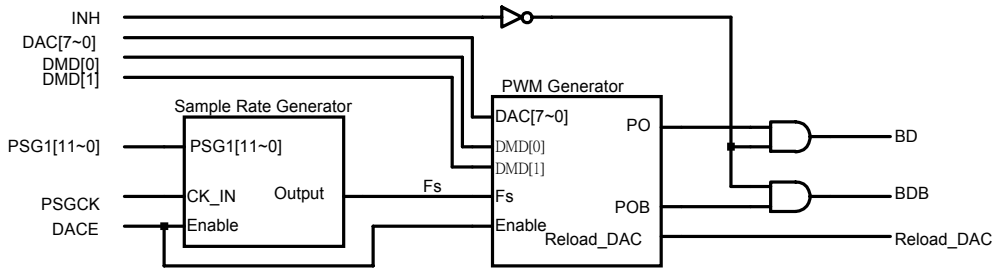


FIGURE 14-1 DAC Diagram

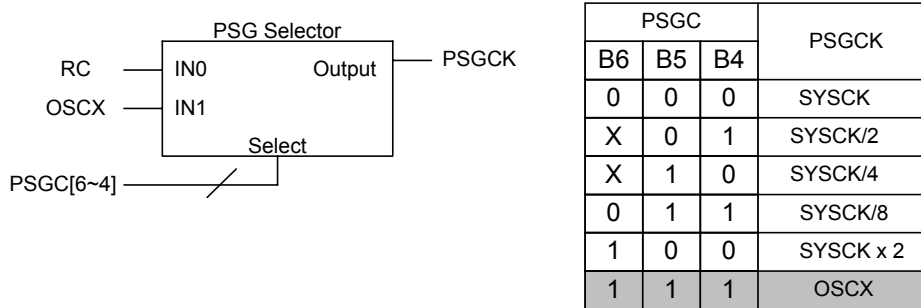


FIGURE 14-2 DAC Clock Source Control

TABLE 14-4 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 |

14.3 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)

14.3.1 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 PSGO 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%. PSGOB is inverse of PSGO's waveform. Figure 13-3 shows the PSGO waveforms.

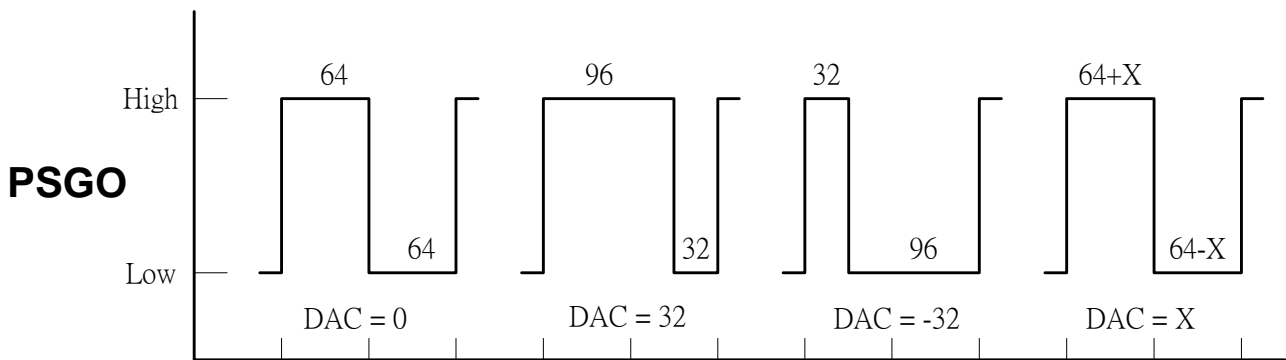


FIGURE 14-3 Single-Pin Mode Wave Form

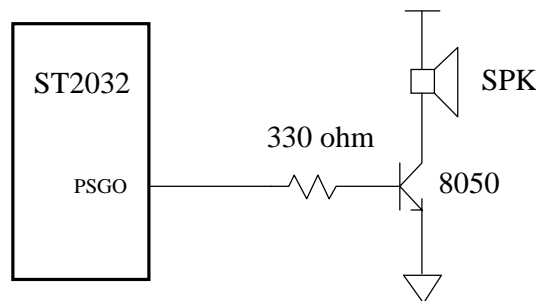


FIGURE 14-4 Single-Pin Mode Application Circuit

14.3.2 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 pins that PSGO and PSGOB. When the DAC value is positive, PSGO goes high with a duty cycle proportional to the output value, while PSGOB stays high. When the DAC value is negative, PSGOB goes low with a duty cycle proportional to the output value, while PSGO 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, PSGO goes high for X segments while PSGOB stays high. For a negative output value $x=0$ to -127, PSGOB goes low for $|X|$ segments while PSGO stays low.

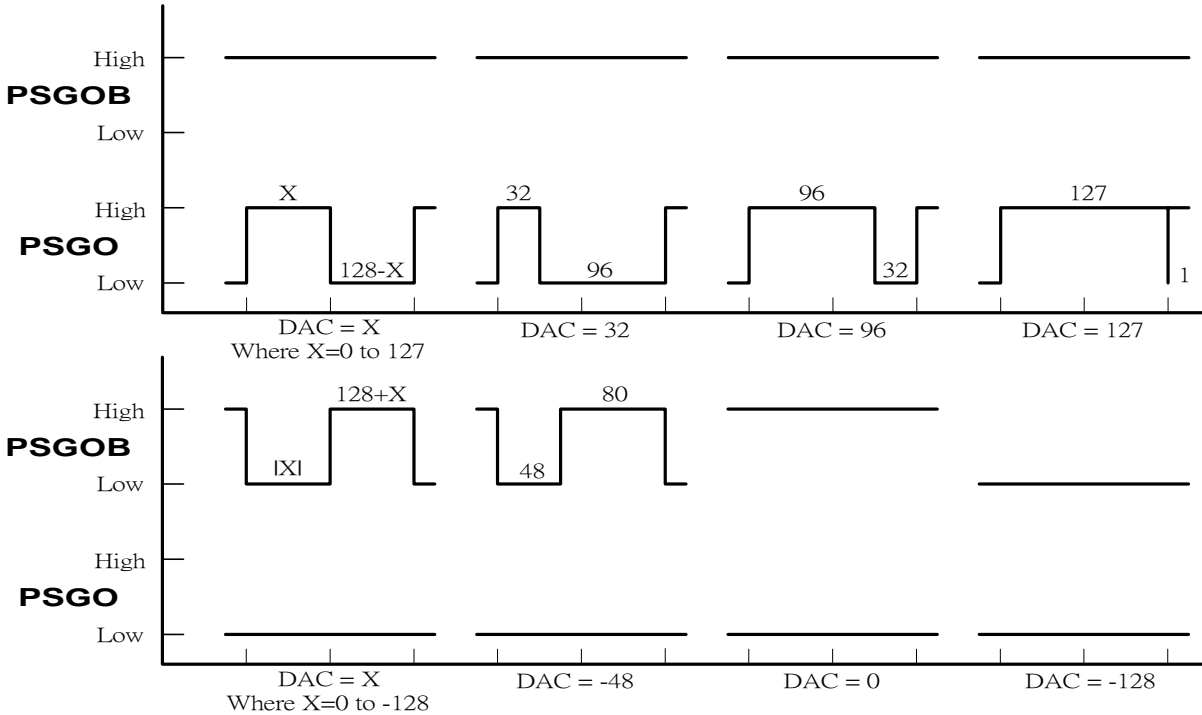


FIGURE 14-5 Two-Pin Two Ended Mode Wave-Form

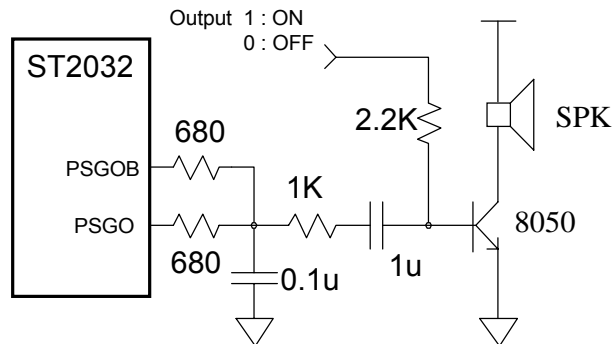


FIGURE 14-6 Two-Pin Two Ended Mode Application Circuit

14.3.3 Two-Pin Push Pull Mode (8-bit Accuracy)

Two-Pin Push Pull mode is designed for buzzer. It requires two pin that PSGO and PSGOB. When the DAC value is 0, both pins are low. When the DAC value is positive, PSGO goes high with a duty cycle proportional to the output value, while PSGOB stays low. When the DAC value is negative, PSGOB goes high with a duty cycle proportional to the output value, while PSGO 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, PSGO goes high for X segments while PSGOB stays low. For a negative output value $x=0$ to -127, PSGOB goes high for $|X|$ segments while PSGO stays low.

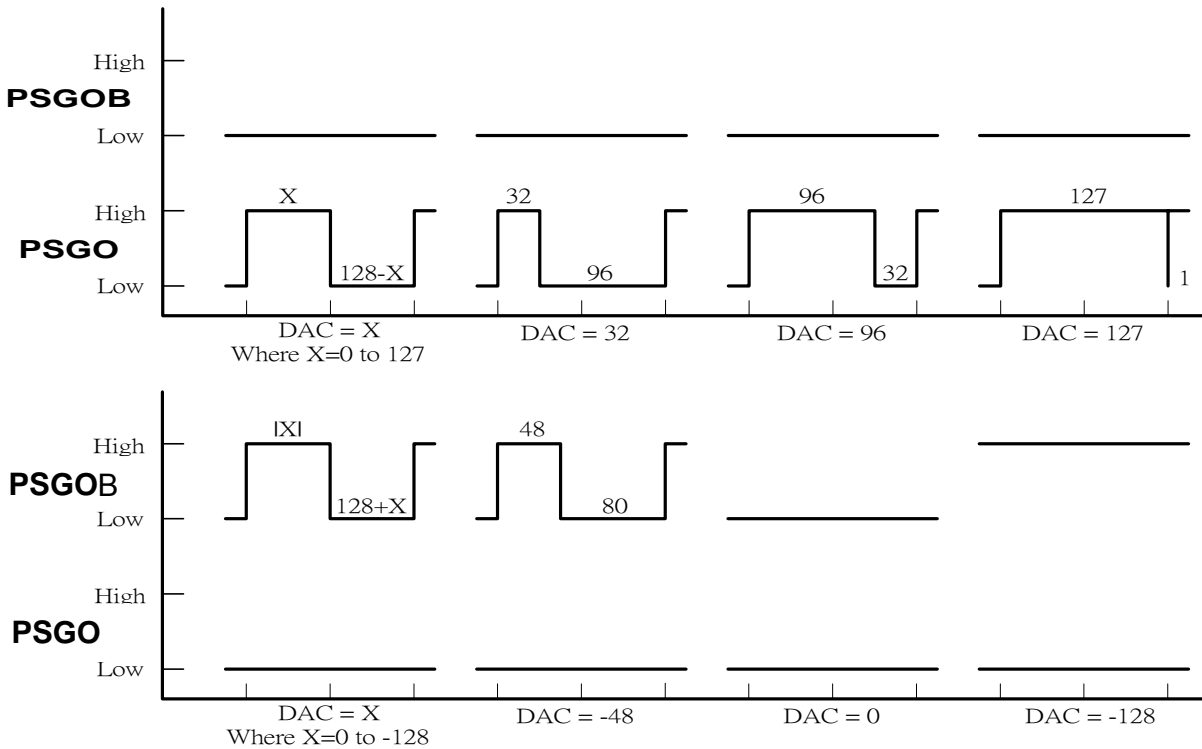


FIGURE 14-7 Two-Pin Push Pull Mode Wave Form

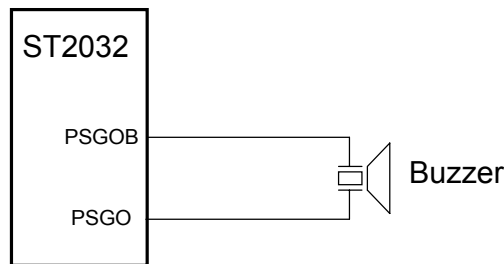


FIGURE 14-8 Two-Pin Push Pull Mode Application Circuit

15. LCD

ST2032 is capable of driving one 1/16 duty, 1/5 bias LCD panel of segment number from 20 to 36(up to 576 dots). LCD block includes display frame buffer (\$1000~ \$10A3) for storing display data, 16 common and 20 segment dedicated drives. The rest 16 segment drives are shared with two I/O ports, Port-B/C. Data in frame buffer is undefined after power on, so correct frame data should be filled in before turn on display. One double DC-DC converter is equipped for higher LCD voltage, and is

controlled by LPWR (LCTL[7]) for on/off. The LCD power should be turned on before setting display on, and should be turned off after setting display off. Both SYSCK and OSCX can be chose as LCD clock source, therefore the display can still works after power down. There are two frame rate options, 64Hz and 85Hz, for each different clock sources. In case of 64Hz frame rate, 8-level driving strength and 12-level contrast are adjustable by software for different panel size and LC voltage.

15.1 LCD Waveform

LCD driving waveform is based on the display data and the alternation signal, which toggles every one frame. The related output voltage levels are shown below.

FIGURE 15-1/ FIGURE 15-2 shows the segment and common waveforms.

TABLE 15-1 Driver Output Levels

| Driver | Mode | Alternation | Display data output level |
|---------|--------------|-------------|---------------------------|
| Common | Selected | H | VP |
| | | L | GND(V5) |
| | Non-selected | H | V1 |
| | | L | V4 |
| Segment | Selected | H | VP |
| | | L | GND(V5) |
| | Non-selected | H | V2 |
| | | L | V3 |

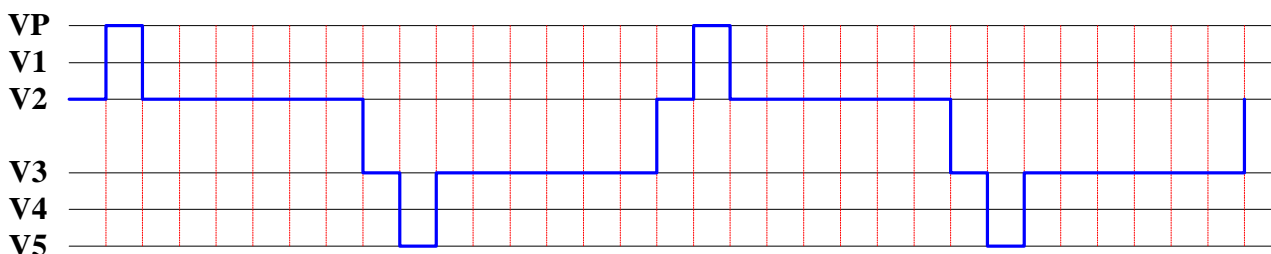


FIGURE 15-1 LCD Segment Waveform

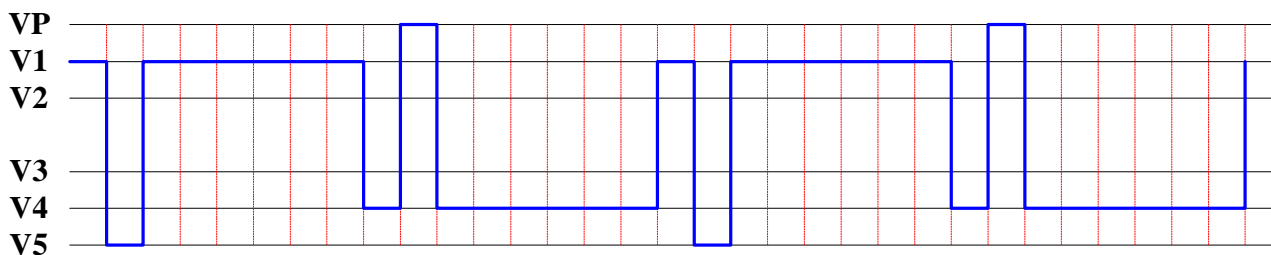


FIGURE 15-2 LCD Common Waveform

15.2 LCD Control Register

TABLE 15-2 LCD Segment Number Selection Register (LSEL)

| Address | Name | R/W | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Default | | | | | |
|---------------------------------------------------|----------------|-------|-------|-------|-------|---------|---------|---------|---------|---------|-----------|-------|-------|-------|-------|-------|
| \$039 | LSEL | R/W | - | - | - | LSEL[4] | LSEL[3] | LSEL[2] | LSEL[1] | LSEL[0] | ---1 1111 | | | | | |
| Bit 4~0: LSEL[4:0] : LCD segment number selection | | | | | | | | | | | | | | | | |
| LSEL[4:0] | Pad Definition | | | | | | | | | | | | | | | |
| | PAD | PAD | PAD | PAD | PAD | PAD | PAD | PAD | PAD | PAD | PAD | PAD | PAD | PAD | PAD | PAD |
| 0 xxxx | PC0 | PC1 | PC2 | PC3 | PC4 | PC5 | PC6 | PC7 | PB0 | PB1 | PB2 | PB3 | PB4 | PB5 | PB6 | PB7 |
| 1 0000 | SEG20 | PC1 | PC2 | PC3 | PC4 | PC5 | PC6 | PC7 | PB0 | PB1 | PB2 | PB3 | PB4 | PB5 | PB6 | PB7 |
| 1 0001 | SEG20 | SEG21 | PC2 | PC3 | PC4 | PC5 | PC6 | PC7 | PB0 | PB1 | PB2 | PB3 | PB4 | PB5 | PB6 | PB7 |
| 1 0010 | SEG20 | SEG21 | SEG22 | PC3 | PC4 | PC5 | PC6 | PC7 | PB0 | PB1 | PB2 | PB3 | PB4 | PB5 | PB6 | PB7 |
| 1 0011 | SEG20 | SEG21 | SEG22 | SEG23 | PC4 | PC5 | PC6 | PC7 | PB0 | PB1 | PB2 | PB3 | PB4 | PB5 | PB6 | PB7 |
| 1 0100 | SEG20 | SEG21 | SEG22 | SEG23 | SEG24 | PC5 | PC6 | PC7 | PB0 | PB1 | PB2 | PB3 | PB4 | PB5 | PB6 | PB7 |
| 1 0101 | SEG20 | SEG21 | SEG22 | SEG23 | SEG24 | SEG25 | PC6 | PC7 | PB0 | PB1 | PB2 | PB3 | PB4 | PB5 | PB6 | PB7 |
| 1 0110 | SEG20 | SEG21 | SEG22 | SEG23 | SEG24 | SEG25 | SEG26 | PC7 | PB0 | PB1 | PB2 | PB3 | PB4 | PB5 | PB6 | PB7 |
| 1 0111 | SEG20 | SEG21 | SEG22 | SEG23 | SEG24 | SEG25 | SEG26 | SEG27 | PB0 | PB1 | PB2 | PB3 | PB4 | PB5 | PB6 | PB7 |
| 1 1000 | SEG20 | SEG21 | SEG22 | SEG23 | SEG24 | SEG25 | SEG26 | SEG27 | SEG28 | PB1 | PB2 | PB3 | PB4 | PB5 | PB6 | PB7 |
| 1 1001 | SEG20 | SEG21 | SEG22 | SEG23 | SEG24 | SEG25 | SEG26 | SEG27 | SEG28 | SEG29 | PB2 | PB3 | PB4 | PB5 | PB6 | PB7 |
| 1 1010 | SEG20 | SEG21 | SEG22 | SEG23 | SEG24 | SEG25 | SEG26 | SEG27 | SEG28 | SEG29 | SEG30 | PB3 | PB4 | PB5 | PB6 | PB7 |
| 1 1011 | SEG20 | SEG21 | SEG22 | SEG23 | SEG24 | SEG25 | SEG26 | SEG27 | SEG28 | SEG29 | SEG30 | SEG31 | PB4 | PB5 | PB6 | PB7 |
| 1 1100 | SEG20 | SEG21 | SEG22 | SEG23 | SEG24 | SEG25 | SEG26 | SEG27 | SEG28 | SEG29 | SEG30 | SEG31 | SEG32 | PB5 | PB6 | PB7 |
| 1 1101 | SEG20 | SEG21 | SEG22 | SEG23 | SEG24 | SEG25 | SEG26 | SEG27 | SEG28 | SEG29 | SEG30 | SEG31 | SEG32 | SEG33 | PB6 | PB7 |
| 1 1110 | SEG20 | SEG21 | SEG22 | SEG23 | SEG24 | SEG25 | SEG26 | SEG27 | SEG28 | SEG29 | SEG30 | SEG31 | SEG32 | SEG33 | SEG34 | PB7 |
| 1 1111 | SEG20 | SEG21 | SEG22 | SEG23 | SEG24 | SEG25 | SEG26 | SEG27 | SEG28 | SEG29 | SEG30 | SEG31 | SEG32 | SEG33 | SEG34 | SEG35 |

TABLE 15-3 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 | R/W | LPWR | BLANK | REV | SCAN | CTR[3] | CTR[2] | CTR[1] | CTR[0] | 1000 0000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <p>Bit 7: LPWR : LCD power ON/OFF bit 1 = LCD power OFF 0 = LCD power ON</p> <p>Bit 6: BLANK : LCD display ON/OFF bit 1 = Disable LCD display (Common line is still scanning) 0 = Enable LCD display</p> <p>Bit 5: REV : LCD display reverse 1 = Reverse display 0 = Normal display</p> <p>Bit 4: SCAN : LCD segment keyboard-scan function 1 = Enable keyboard-scan signal in LCD waveforms 0 = Disable keyboard-scan signal</p> <p>Bit 3~0: CTR[3~0] : LCD contrast control</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2">Frame Rate = 64Hz</th> <th colspan="2">Frame Rate = 85Hz</th> </tr> </thead> <tbody> <tr> <td>00xx</td> <td>= contrast level 12</td> <td>0xxx</td> <td>= contrast level 8</td> </tr> <tr> <td>0100</td> <td>= contrast level 12 (maximum)</td> <td>1000</td> <td>= contrast level 8 (maximum)</td> </tr> <tr> <td>0101</td> <td>= contrast level 11</td> <td>1001</td> <td>= contrast level 7</td> </tr> <tr> <td>0110</td> <td>= contrast level 10</td> <td>1010</td> <td>= contrast level 6</td> </tr> <tr> <td>0111</td> <td>= contrast level 9</td> <td>1011</td> <td>= contrast level 5</td> </tr> <tr> <td>1000</td> <td>= contrast level 8</td> <td>1100</td> <td>= contrast level 4</td> </tr> <tr> <td>1001</td> <td>= contrast level 7</td> <td>1101</td> <td>= contrast level 3</td> </tr> <tr> <td>1010</td> <td>= contrast level 6</td> <td>1110</td> <td>= contrast level 2</td> </tr> <tr> <td>1011</td> <td>= contrast level 5</td> <td>1111</td> <td>= contrast level 1 (minimum)</td> </tr> <tr> <td>1100</td> <td>= contrast level 4</td> <td></td> <td></td> </tr> <tr> <td>1101</td> <td>= contrast level 3</td> <td></td> <td></td> </tr> <tr> <td>1110</td> <td>= contrast level 2</td> <td></td> <td></td> </tr> <tr> <td>1111</td> <td>= contrast level 1 (minimum)</td> <td></td> <td></td> </tr> </tbody> </table> | | | | | | | | | | | | Frame Rate = 64Hz | | Frame Rate = 85Hz | | 00xx | = contrast level 12 | 0xxx | = contrast level 8 | 0100 | = contrast level 12 (maximum) | 1000 | = contrast level 8 (maximum) | 0101 | = contrast level 11 | 1001 | = contrast level 7 | 0110 | = contrast level 10 | 1010 | = contrast level 6 | 0111 | = contrast level 9 | 1011 | = contrast level 5 | 1000 | = contrast level 8 | 1100 | = contrast level 4 | 1001 | = contrast level 7 | 1101 | = contrast level 3 | 1010 | = contrast level 6 | 1110 | = contrast level 2 | 1011 | = contrast level 5 | 1111 | = contrast level 1 (minimum) | 1100 | = contrast level 4 | | | 1101 | = contrast level 3 | | | 1110 | = contrast level 2 | | | 1111 | = contrast level 1 (minimum) | | |
| Frame Rate = 64Hz | | Frame Rate = 85Hz | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 00xx | = contrast level 12 | 0xxx | = contrast level 8 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0100 | = contrast level 12 (maximum) | 1000 | = contrast level 8 (maximum) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0101 | = contrast level 11 | 1001 | = contrast level 7 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0110 | = contrast level 10 | 1010 | = contrast level 6 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0111 | = contrast level 9 | 1011 | = contrast level 5 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1000 | = contrast level 8 | 1100 | = contrast level 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1001 | = contrast level 7 | 1101 | = contrast level 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1010 | = contrast level 6 | 1110 | = contrast level 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1011 | = contrast level 5 | 1111 | = contrast level 1 (minimum) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1100 | = contrast level 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1101 | = contrast level 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1110 | = contrast level 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1111 | = contrast level 1 (minimum) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

TABLE 15-4 LCD Clock Source and Driving Strength 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 | R/W | DRV[3] | DRV[2] | DRV[1] | DRV[0] | - | LCK[2] | LCK[1] | LCK[0] | 111- -000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <p>Bit 7~4: DRV[3:0] : LCD driving strength control</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2">Frame Rate = 64Hz</th> <th colspan="2">Frame Rate = 85Hz</th> </tr> </thead> <tbody> <tr> <td>0000</td> <td>= driving level 16/16 (maximum)</td> <td>00XX</td> <td>= driving level 12/12</td> </tr> <tr> <td>0001</td> <td>= driving level 15/16</td> <td>0100</td> <td>= driving level 12/12(maximum)</td> </tr> <tr> <td>0010</td> <td>= driving level 14/16</td> <td>0101</td> <td>= driving level 11/12</td> </tr> <tr> <td></td> <td>⋮</td> <td></td> <td>⋮</td> </tr> <tr> <td></td> <td>⋮</td> <td></td> <td>⋮</td> </tr> <tr> <td></td> <td>⋮</td> <td></td> <td>⋮</td> </tr> <tr> <td>1110</td> <td>= driving level 3/16</td> <td>1101</td> <td>= driving level 3/12</td> </tr> <tr> <td>1110</td> <td>= driving level 2/16</td> <td>1110</td> <td>= driving level 2/12</td> </tr> <tr> <td>1111</td> <td>= driving level 1/16 (minimum)</td> <td>1111</td> <td>= driving level 1/12 (minimum)</td> </tr> </tbody> </table> <p>Bit 2~0: LCK[2:0] : LCD frame rate control</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>LCK[2:0]</th> <th>Clock Source</th> <th>Frame Rate</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>OSCX (32768Hz)</td> <td>64 Hz</td> </tr> <tr> <td>001</td> <td>OSCX (32768Hz)</td> <td>85 Hz</td> </tr> <tr> <td>010</td> <td>OSC (2MHz)</td> <td>64 Hz</td> </tr> <tr> <td>011</td> <td>OSC (2MHz)</td> <td>85 Hz</td> </tr> <tr> <td>100</td> <td>OSC (4MHz)</td> <td>64 Hz</td> </tr> <tr> <td>101</td> <td>OSC (4MHz)</td> <td>85 Hz</td> </tr> <tr> <td>110</td> <td>OSC (8MHz)</td> <td>64 Hz</td> </tr> <tr> <td>111</td> <td>OSC (8MHz)</td> <td>85 Hz</td> </tr> </tbody> </table> | | | | | | | | | | | | Frame Rate = 64Hz | | Frame Rate = 85Hz | | 0000 | = driving level 16/16 (maximum) | 00XX | = driving level 12/12 | 0001 | = driving level 15/16 | 0100 | = driving level 12/12(maximum) | 0010 | = driving level 14/16 | 0101 | = driving level 11/12 | | ⋮ | | ⋮ | | ⋮ | | ⋮ | | ⋮ | | ⋮ | 1110 | = driving level 3/16 | 1101 | = driving level 3/12 | 1110 | = driving level 2/16 | 1110 | = driving level 2/12 | 1111 | = driving level 1/16 (minimum) | 1111 | = driving level 1/12 (minimum) | LCK[2:0] | Clock Source | Frame Rate | 000 | OSCX (32768Hz) | 64 Hz | 001 | OSCX (32768Hz) | 85 Hz | 010 | OSC (2MHz) | 64 Hz | 011 | OSC (2MHz) | 85 Hz | 100 | OSC (4MHz) | 64 Hz | 101 | OSC (4MHz) | 85 Hz | 110 | OSC (8MHz) | 64 Hz | 111 | OSC (8MHz) | 85 Hz |
| Frame Rate = 64Hz | | Frame Rate = 85Hz | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0000 | = driving level 16/16 (maximum) | 00XX | = driving level 12/12 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0001 | = driving level 15/16 | 0100 | = driving level 12/12(maximum) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0010 | = driving level 14/16 | 0101 | = driving level 11/12 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | ⋮ | | ⋮ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | ⋮ | | ⋮ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | ⋮ | | ⋮ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1110 | = driving level 3/16 | 1101 | = driving level 3/12 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1110 | = driving level 2/16 | 1110 | = driving level 2/12 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1111 | = driving level 1/16 (minimum) | 1111 | = driving level 1/12 (minimum) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| LCK[2:0] | Clock Source | Frame Rate | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 000 | OSCX (32768Hz) | 64 Hz | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 001 | OSCX (32768Hz) | 85 Hz | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 010 | OSC (2MHz) | 64 Hz | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 011 | OSC (2MHz) | 85 Hz | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 100 | OSC (4MHz) | 64 Hz | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 101 | OSC (4MHz) | 85 Hz | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 110 | OSC (8MHz) | 64 Hz | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 111 | OSC (8MHz) | 85 Hz | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

15.3 Keyboard-scan Function on LCD drives

LCD keyboard awaking pulses are combined with LCD waveform. The purpose is to trigger Port-A interrupt to wake up the system.

Note: LCD keyboard awaking pulses can only be turned on under 3V operating voltage.

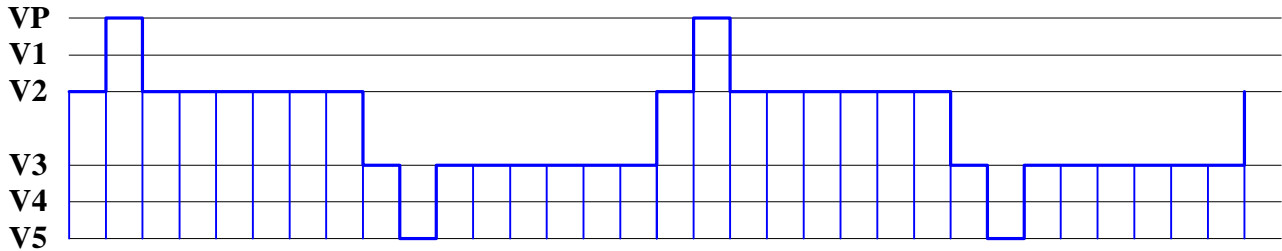


FIGURE 15-3 LCD Segment Waveform (With Keyboard Awaking Pulses)

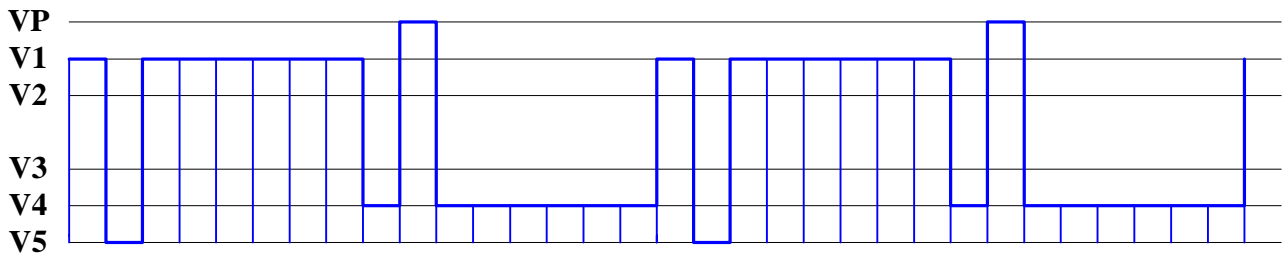


FIGURE 15-4 LCD Common Waveform (With Keyboard Awaking Pulses)

15.3.2 Keyboard-scan Function Example:

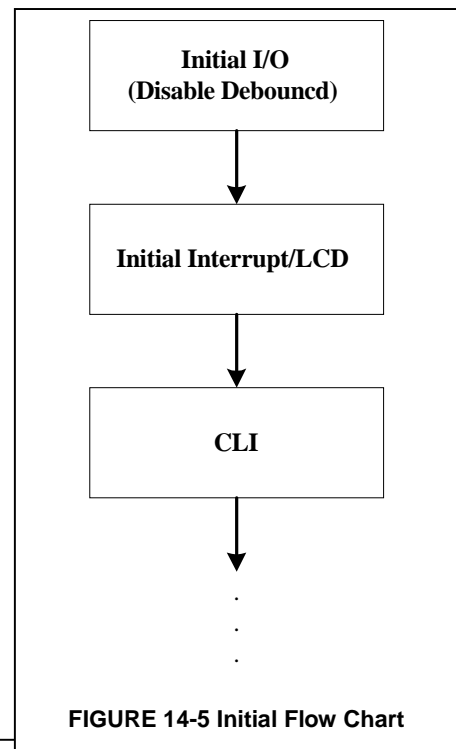
- a. Keyboard : 64Keys (8x8)
- b. Return Lines : Port-A
- c. Scan Lines : Port-B

INITIAL_Port_And_LCD

```

SMB4 <LCTL           ;;Enable Keyboard Awaking Waveform
LDA #00011111B       ;;Port-B/C Shared With SEGs
STA <LSEL

STZ <PCA             ;;Set Port-A as Inputs for Return Line
LDA #FFH
STA <PA              ;;Port-A Pull-High
STA <PCB             ;;Set Port-B as Outputs for Scan Line
LDA #10000000B
STA <PMCR            ;;Enable Pull up,Disable Debounce
LDA #00010000B
STA <IENA            ;;Enable Port-A Interrupt
LDA <PA
STZ <IREQ            ;;Reset Interrupt Request Register
.
.
.
CLI
    
```



Interrupt-Subroutine

```

Port_ISR
  PHA
  PHX

  LDX    #00010111B    ;;X: Value for LSEL when activate segments
  LDA    #11111110B    ;;Set Working zero for Port-B
  STA    <ScanValue

?Scan_PB
  STA    <PB
  STZ    <LSEL          ;;Disable segment waveforms
  nop
  nop          ;;Wait for return line to be stable
  nop
  nop
  LDA    <PA
  STX    <LSEL          ;;Enable segment waveforms
  JSR    Store-Key-Data
  SEC
  ROL    <Scanvalue     ;;Shift working zero to left
  LDA    <ScanValue
  BCS    ?Scan_PB       ;;Keep on scanning until ScanValue = FF
  .
  .
  PLX
  PLA
  RTI
    
```

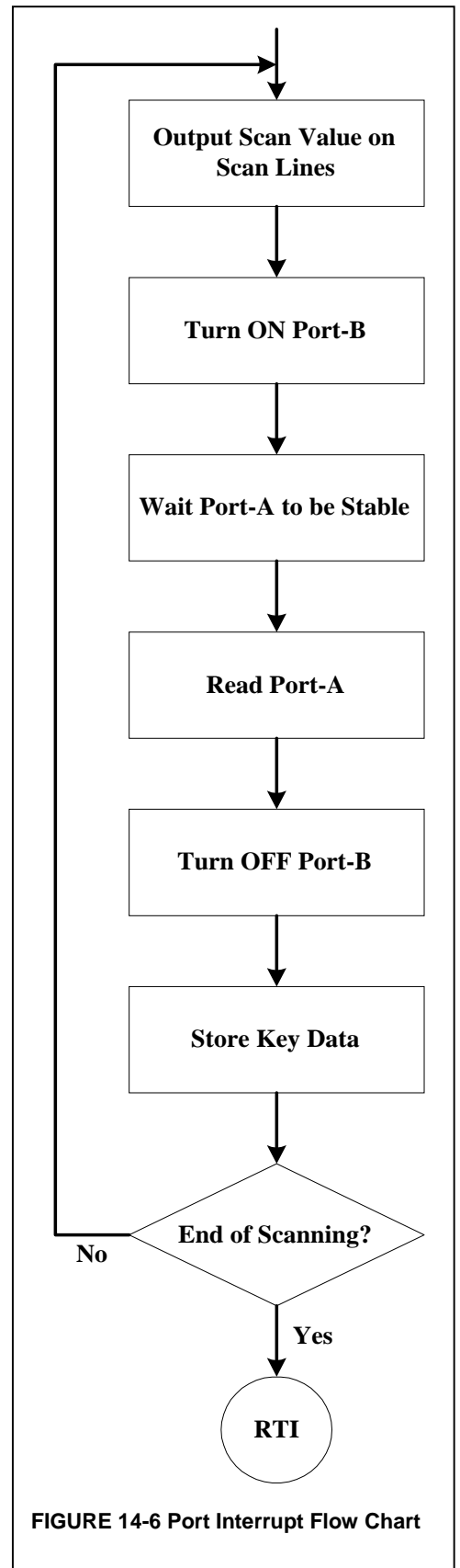


FIGURE 14-6 Port Interrupt Flow Chart

15.4 LCD Frame Buffer

Each pixel of LCD panel is directly mapped into LCD frame buffer. If some segments are not used, the corresponding

RAM can still be accessed for data memory. Refer to TABLE 15-5 for detail mapping.

TABLE 15-5 LCD Frame Buffer Memory Mapping

| | SEG0 | SEG1 | SEG2 | SEG3 | SEG4 | SEG5 | | SEG35 |
|----------------|--------------|--------------|--------------|--------------|--------------|--------------|-------|--------------|
| Address | 1000H | 1001H | 1002H | 1003H | 1004H | 1005H | | 1023H |
| 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 | | 10A3H |
| 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 |

Note: Undefined RAM area, \$1024~\$107F and \$10A4~\$10FF, is not accessible.

16. POWER DOWN MODES

ST2032 has three power down modes: WAI-0, WAI-1 and STP. The instruction WAI will enable either WAI-0 or WAI-1, which is controlled by WAIT(SYS[2]). And the instruction

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

TABLE 16-1 System Control Register (SYS)

| Address | Name | R/W | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Default |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|-----|-------|-------|-------|-------|-------|-------|-------|-------|-----------|
| \$030 | SYS | R/W | XSEL | OSTP | XSTP | XBAK | WSKP | WAIT | - | LVDET | 0000 00-0 |
| <p>Bit 3: WSKP : System warm-up control bit 1 = Warm-up to 16 oscillation cycles 0 = Warm-up to 256 oscillation cycles</p> <p>Bit 2: WAIT : WAI-0 / WAI-1 mode 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.

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 | | | Retain | | | Reset, Any interrupt |
| STP | Stop | Stop | Stop | Retain | | | 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 | | | Retain | | | | Reset, Any interrupt |
| STP | Stop | Stop | Retain | | | Retain | | | | Reset |

17. LOW VOLTAGE DETECTOR

ST2032 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 μ 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 | XBAK | WSKP | WAIT | - | LVDET | 0000 00-0 |
| Bit 0: LVDET : Low voltage detect 1 = Enable detector (write) / Low voltage (read) 0 = Disable detector (write) / Normal voltage (read) | | | | | | | | | | | |

18. ELECTRICAL CHARACTERISTICS

| | | |
|-------------------------------|-------|-----------------|
| DC Supply Voltage | ----- | -0.3V to +4.5V |
| Operating Ambient Temperature | ----- | -10°C to +60°C |
| Storage Temperature | ----- | -10°C to +125°C |

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

18.1 DC Electrical Characteristics

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

| Parameter | Symbol | Min. | Typ. | Max. | Unit | Condition |
|------------------------------|--------------------|-------------------------------------------|------|----------------------|------|-------------------------------------------------------------------------|
| Operating Voltage | VCC | 2.4 | | 5.5 | V | Logic |
| | | | | 3.4 | | Built-in double DC-DC voltage converter for LCD driver: |
| Operating Current | I _{OP} | 573 | 451 | | μA | All I/O ports are input and pull-up, LCD driving strength is maximum. |
| Standby Current | I _{SB0} | | 1.3 | | μA | All I/O ports are input and pull-up, OSCX on, LCD off (WAIT1/STOP mode) |
| Standby Current | I _{SB1} | | 65 | | μA | All I/O ports are input and pull-up, OSCX on, LCD off (WAIT0 mode) |
| LCD consumption | I _{LCD} | | 12.5 | | μA | LCD Clock source=OSCX Driving strength=1/16 Condition: WAIT1 mode. |
| LCD consumption | I _{LCD} | | 55.6 | | μA | LCD Clock source=OSCX Driving strength=16/16 Condition: WAIT1 mode. |
| Input High Voltage | V _{IH} | 0.7V _{CC} 0.85V _{CC} | | V _{CC} +0.3 | V | PORT A, PORT B, PORT C |
| | | | | | | RESET, INT |
| Input Low Voltage | V _{IL} | GND-0.3 | | 0.3V _{CC} | V | PORT A, PORT B, PORT C |
| | | | | 0.15V _{CC} | | RESET, INT |
| Pull-up resistance | R _{IH} | | 142 | | KΩ | PORTA (Voltage difference=0.9V) |
| Pull-up resistance | R _{IH} | | 48 | | KΩ | PORTA (Voltage difference=0.9V). Operation voltage=5V |
| Pull-up resistance | R _{IH} | | 150 | | KΩ | PORTB, PORT C (Voltage difference=0.9V) |
| Output high voltage | V _{OH1} | 0.7V _{CC} | | | V | PORTA (IOH=-3.5mA) |
| Output high voltage | V _{OH1} | 0.7V _{CC} | | | V | PORTB, PORTC (IOH=-2.5mA) |
| Output low voltage | V _{OL1} | | | 0.3V _{CC} | V | PORTA (IOL=7.5mA) |
| Output low voltage | V _{OL1} | | | 0.3V _{CC} | V | PORTB, PORT C (IOL=4.5mA) |
| Output high voltage | V _{OH2} | 0.7V _{CC} | | | V | PB0/1 as PSG/DAC, IOH = -25mA. |
| Output low voltage | V _{OL2} | | | 0.3V _{CC} | V | PB0/1 as PSG/DAC, IOL= 53mA. |
| Oscillation start time | T _{STT} | | 1 | 2 | s | |
| Low voltage detector current | I _{lvdet} | | 114 | | μA | No detector voltage adjustment |

FIGURE 18-1 Relation between operation voltage & frequency

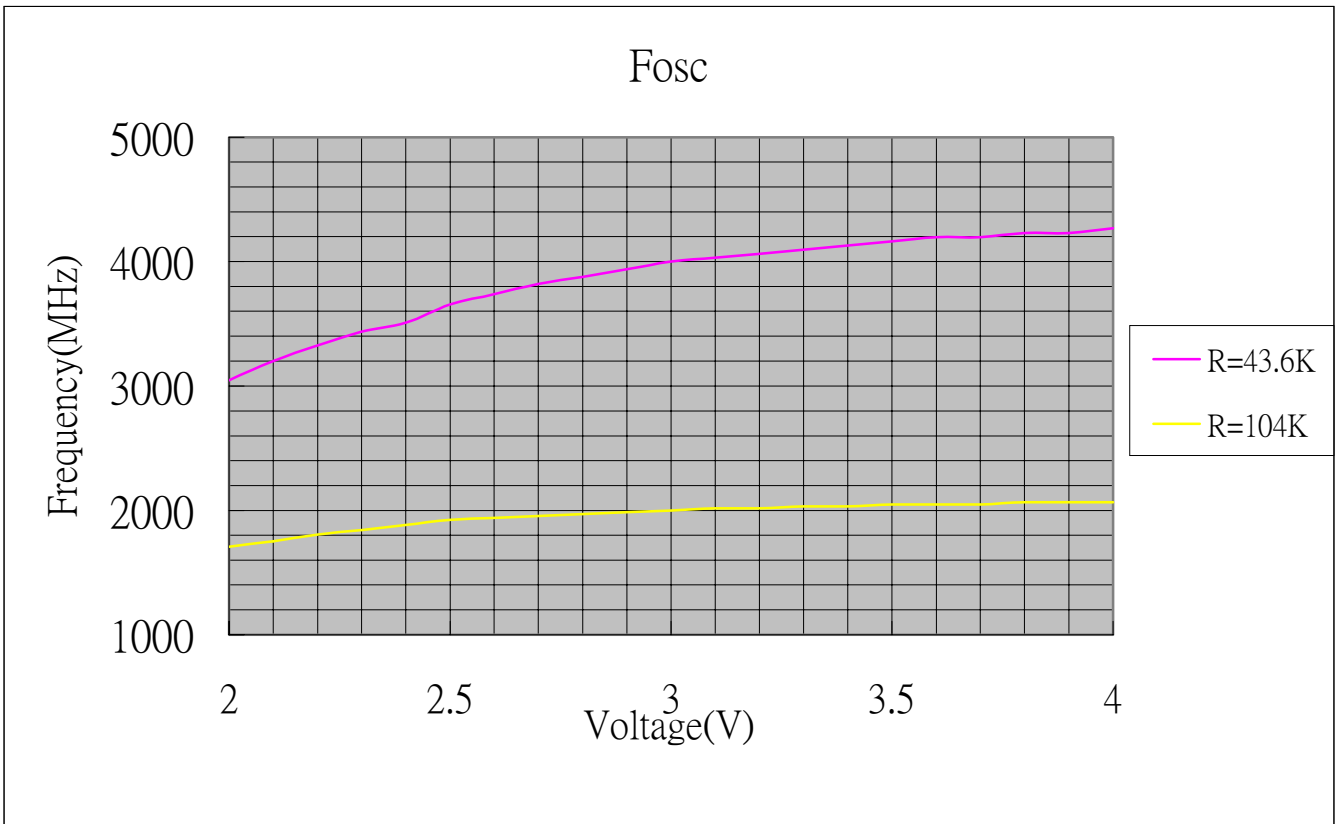


TABLE 18-1 R-Oscillator V.S. Frequency

| | |
|---------|---------|
| Voltage | 3V |
| Freq. | |
| 4MHz | 47Kohm |
| 2MHz | 100Kohm |

19. APPLICATION CIRCUIT

VDD : 3V
 Clock : 32768Hz crystal and 2.0MHz RC oscillator
 LCD : 1/16 duty
 I/O : PORT-A
 ALARM : PSG0, PSG1

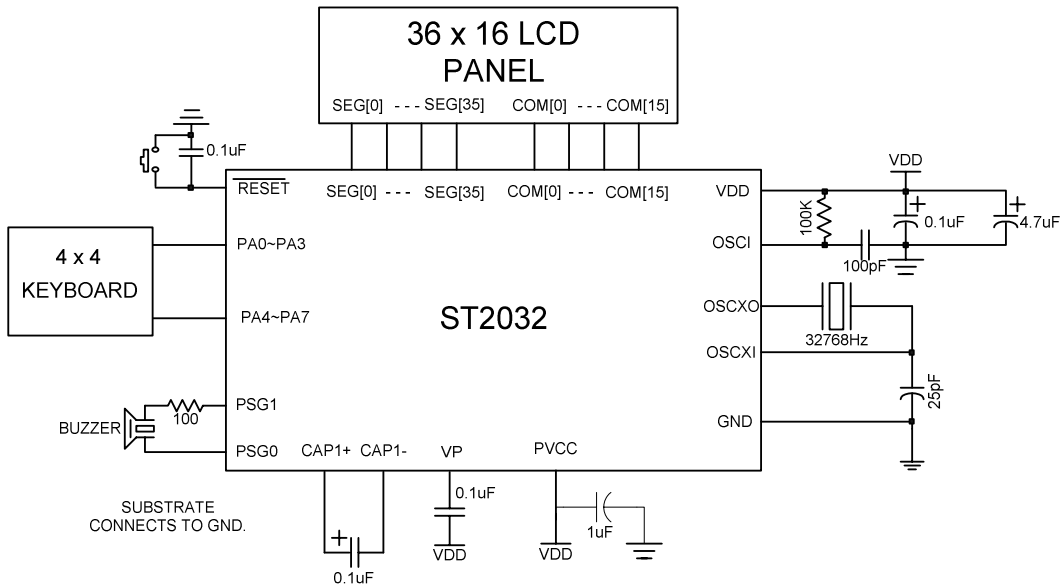


FIGURE 19-1 Application Circuit without LCD keyboard awaking pulse

VDD : 3V
 Clock : 32768Hz crystal and 2.0MHz RC oscillator
 LCD : 1/16 duty
 I/O : PORT-A/B/C
 ALARM : PSG0, PSG1

Note: Because the COMs and SEGs output VDD level while the LCD is turned off. There is no keyboard awaking pulse to wake up the system. So the ON/OFF key must connect between GND and Port-A.

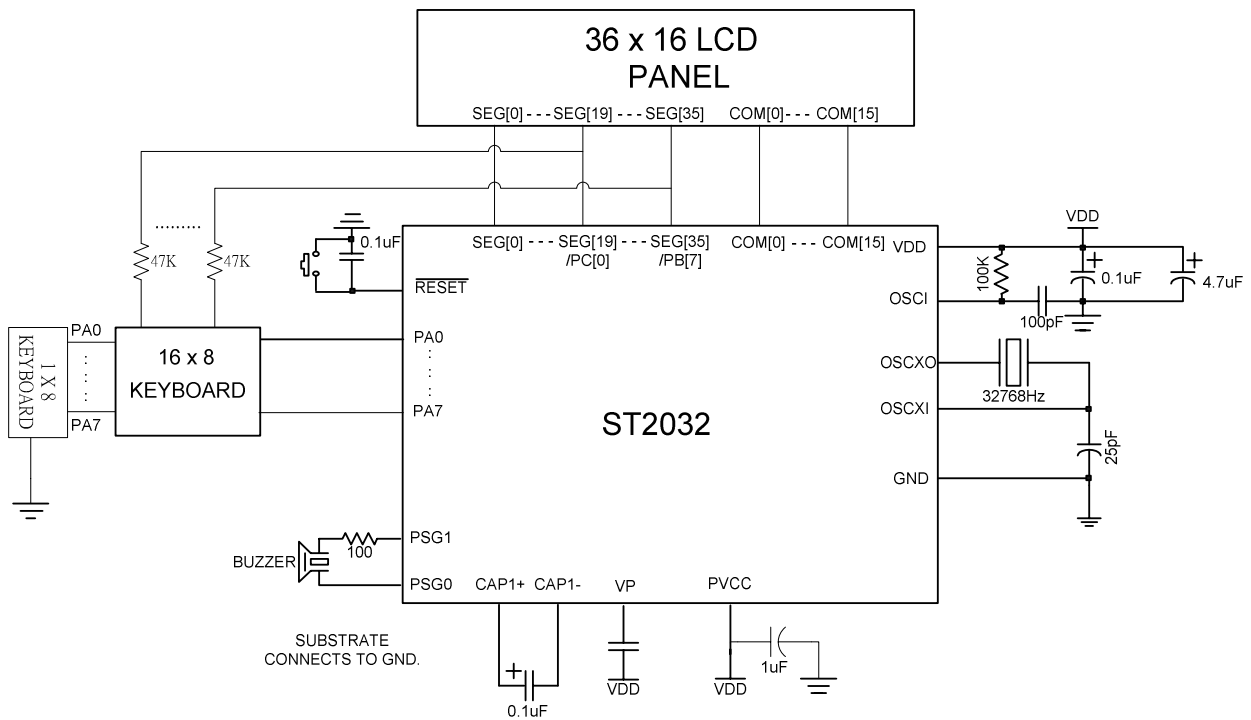


FIGURE 19-2 Application Circuit with LCD keyboard awaking pulse

VDD : 5V
 Clock : 32768Hz crystal and 2.0MHz RC oscillator
 LCD : 1/16 duty
 I/O : PORT-A/B/C
 ALARM : PSG0, PSG1

Note: LCD keyboard awaking pulses should be turned off under 5V operating voltage.

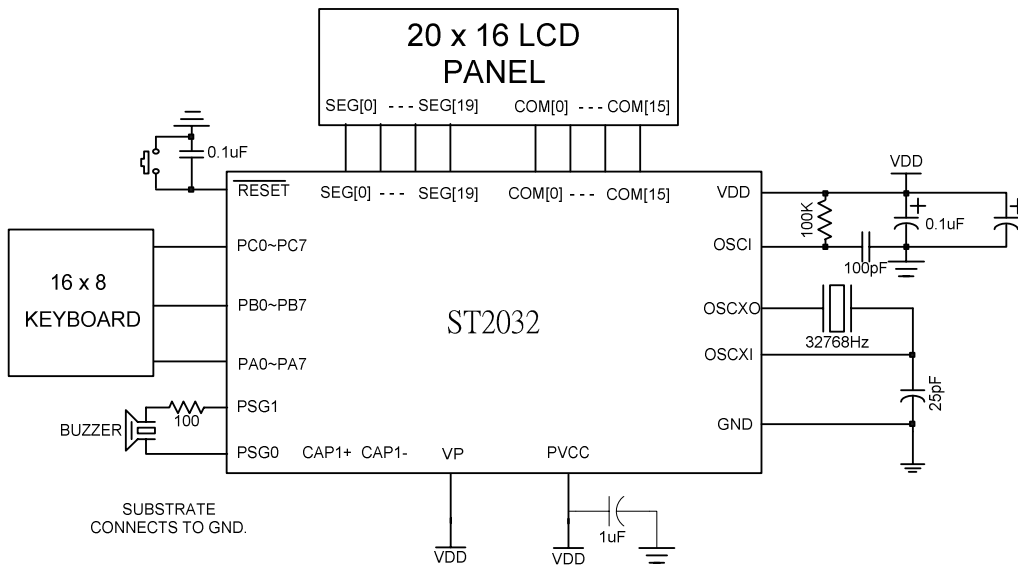


FIGURE 19-3 Application circuit without DC-DC converter

20. REVISIONS

| | | |
|--------------------------------------|-------------------------------------------------------------------------------------|------------|
| Version 1.2 | | |
| Page 42 | Adding FIGURE 18-1 and TABLE 18-1..... | 2003/10/14 |
| Version 1.1 | | |
| Page 8 | Modify the range of User RAM from 200H~3FFH to 200H~4FFH. | |
| Page 20 | Modify Basetimer "STOP" setting on TABLE 12-3..... | 2003/9/30 |
| Version 1.0 | | |
| Page 2/44 | Exchange the pad location of PSGO & PSGOB..... | 2003/6/26 |
| Version 0.5 | | |
| Page 39 | Added Port-A pull-up resistor value under 5V operating voltage. | |
| Page 33/42 | LCD keyboard awaking pulses can't use on 5V operating voltage. | |
| Page 26 | Modify the value of TABLE 14-4..... | 2002/11/11 |
| Version 0.4 | | |
| Page 40/41/42 | Refresh application circuit. | |
| Page 39 | Fill the DC electrical characteristic table..... | 2002/10/14 |
| Version 0.3..... | | 2002/6/10 |
| Page 6 | Modify the memory map of SRAM. | |
| Page 2/4/21/22/23/24/27/28/29/40/42: | Rename pad PSG0/1 to PSGOB/O. | |
| Page 21: | Added explain of PSG | |
| Page 24: | Added PSG Application Circuit. | |
| Page 4/42: | Rename VCC2 to PVCC | |
| Page 33: | Fix Keyboard-scan Function Example to LDA #00011111B ;;Port-B/C Shared With SEGs | |
| Page 40: | Fix Application Circuit. | |
| Page 41/42: | Added PAD Center Coordinates. | |
| Version 0.2: | | 2002/4/17 |
| Page 2: | Added PAD Diagram. | |
| Version 0.1: | | |
| First release | | |