# ST2205U

PRELIMINARY

## *8 BIT Integrated Microcontroller with 32K Bytes RAM*

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

## 1. FEATURES

- **Totally static 8-bit CPU**
- **ROM: 16K x 8-bit(OTP)**
- **RAM: 32K x 8-bit**
- **Stack: Up to 128-level deep**
- **Operation voltage: 2.4V ~ 3.6V**
- **Operation frequency:**
  – 4.0Mhz@2.4V(Min.)
  – 6.0Mhz@2.7V(Min.)
- **One 16x8 Signed Multiplier**
- **Low Voltage Reset (LVR)**
  – Two levels of bonding options
- **Low Voltage Detector (LVD)**
  – Programmable 4 levels
  – System power or external battery level can be detected.
- **Flash Memory Interface**
  – On the fly ECC code generation and detection
  – Fast data transfer with dedicated DMA channel
  – Nand and And type Flash supported
- **USB 2.0 Full speed device**
  – Integrate one PLL to produce 48Mhz clock
  – Built-in 3.3V regulator for transceiver
  – Mass storage class supported
  – Double buffering and direct buffer access increase throughput and ease real-time data transfer
- **Direct Memory Access (DMA)**
  – Two channels with special modes for Flash and display
  – Three address generation modes
- **Memory configuration**
  – Four kinds of banks for bios, program, data, interrupt and internal RAM
  – 13-bit bank registers support up to 44M bytes
  – Six programmable chip-selects with 4 modes
  – Maximum single device of 16M bytes
  – Build-in a waiting cycle in external memory I/F's timing can be selected to support low speed device.
- **General-Purpose I/O (GPIO) ports**
  – 56 multiplexed CMOS bit programmable I/Os
  – Hardware de-bounce option for Port-A
  – Bit programmable pull-up/down or open-drain/CMOS
- **Timer/Counter**
  – Four 12-bit and one 8-bit timers
  – Seven fixed time bases
- **Watchdog Timer (WDT)**
  – Two selectable time bases
  – Programmable WDT interrupt or reset
- **Real-time Clock (RTC)**
  – Full clock function, second/ minute/ hour and day, with three counters and interrupts
  – One programmable alarm
- **Three External Interrupt Sources**
- **Three clocking outputs**
  – Clock sources including Timer0, OSCN clock, baud rate generator
- **Prioritized interrupts with dedicated exception vectors**
  – External interrupts (x3) (edge triggered)
  – PortA interrupt (transition triggered)
  – LCD buffer interrupt
  – Base timer interrupt (x8)
  – Timer0~3 interrupts (x4)
  – SPI interrupts (x2)
  – UART interrupts (x2)
  – USB interrupts (x6)
  – PCM interrupt
  – RTC interrupts (x4)
- **Dual clock sources with warm-up timer**
  – Low frequency crystal oscillator (OSCX) ·····32768 Hz
  – High frequency resistor or crystal/resonator oscillator (OSC) selected by pin option ..... 455K~8M Hz
- **LCD Controller (LCDC)**
  – Programmable display size:
    - COM: 512 max. SEG: 1024 max.
    - Max. 160xRGBx120 color STN supported by internal buffer
  – Hardware 4/16 gray levels with 5-bit palette, up to 4096 colors supported
  – Share system memory with display buffer and with no loss of the CPU time
  – Support 1-/4-/8-bit LCD data bus
  – Diverse functions including virtual screen, panning, scrolling, contrast control, alternating signal generator, buffer switching and fast graphic data manipulation
- **Programmable Sound Generator (PSG)**
  – Four channels with three playing modes: 9-bit ADPCM, 8-bit PCM and 8-bit melody
  – One 16-byte buffer and 6-bit volume control per channel
  – Wavetable melody support
  – Two dedicated PWM outputs for direct driving

- **12-bit current DAC with two 4-word buffer**
- **Universal Asynchronous Receiver/Transmitter (UART)**
  - Full-duplex operation
  - Baud rate generator with one digital PLL
  - Standard baud rates of 600 bps to 115.2 kbps
  - Both transmitter and receiver buffers supported
  - Direct glueless support of IrDA physical layer protocol
  - Two sets of I/Os (TX,RX) for two independent devices
- **Serial Peripheral Interface (SPI)**
  - Inter IC sound (IIS) supported
  - Master and slave modes
  - Five serial signals including enable and data-ready
  - Both transmitter and receiver buffers supported
  - Programmable data length from 7-bit to 16-bit
- **Three power down modes**
  - WAI0 mode
  - WAI1 mode
  - STP mode
- **On-chip ICE debug interface**

# 2. GENERAL DESCRIPTION

The ST2205U is a 8-bit integrated microcontroller designed with CMOS silicon gate technology. The true static CPU core, power down modes and dual oscillators design makes the ST2205U suitable for power saving and long battery life designs. The ST2205U integrates various logic to support functions on-chip which are needed by system designers. This is also important for lower system complexity, small board size and, of course, shorter time to market and less cost.

The ST2205U features the capacity of memory access of maximum 44M bytes which is needed by products with large data bases. Six chip selects are equipped for direct connection to external ROM, SRAM, Flash memory or other devices. Maximum one single device of 16M bytes is possible.

Two DMA channels make fast data transfer possible and easy. Both source and destination pointers can refer to the whole memory space with 15-bit pointers and bank registers. Besides normal operation, two special modes are designed for double transfer speed of Nand Flash memory and also fast graphic operation between two display pictures.

Nand Flash is a low cost mass data storage solution for newly design. The ST2205U equips a Nand flash interface to connect both Nand and And Flash memories. Both ECC generating and checking functions are supported. These are very important for Flash data management.

The ST2205U has 56 I/Os grouped into 7 ports, Port-A ~ Port-F and Port-L. Each pin can be programmed to input or output. There are two options: pull-up/down for inputs of Port-C and only pull-up for inputs of the other ports. In case of output, there are open-drain/CMOS options for outputs of PortC and only CMOS for other ports. Port-A is designed for keyboard scan with de-bounce and transition triggered interrupt, while Port-C/D/E/F/L are shared with other system functions. All the properties of I/O pins are still programmable when they are assigned to another function. This enlarges the flexibility of the usage of function signals.

The internal 32K bytes RAM helps to drive large LCD panels up to 160xRGBx120. Together with 16-graylevel support, the ST2205U can rich display information and the diversity of contents as well. This is done with no need of external display RAM because of the special internal memory sharing design. The variable display buffer technique also make large panel size with small internal RAM possible. User may free major internal RAM for temporary computing or access while keeping the display content correct.

The ST2205U equips serial communication ports of one UART and one SPI to perform different communications, ex.: RS-232 and IrDA, with system components or other products such as PC, Notebook, and popular PDA. Two clocking outputs can produce synthesized PWM signals or high frequency carrier for IR remote control. This helps products become more useful in our daily life.

Communication with PC via USB is becoming more and more popular. The ST2205U features one PLL, a 3.3V regulator, and a USB 2.0 Full speed device engine to satisfy the strong demand of fast data transfer from market. Both HID and Mass storage classes are supported as well as the firmware libraries and the Windows drivers.

The built-in four channels PSG and a 12-bit current DAC provide a nice quality voice together with a 4-channel wavetable melody in the background. Both voice and melody functions have buffers to make program easier and well structured, and also a 16x8 multiplier is to control the volumn of each channel. Besides hardware, ADPCM algorithm and a MIDI converter Windows software are also provided to speed up the development. In addition to current DAC, two dedicated pins with large driving capacity can drive a buzzer/speaker directly for minimum cost.

The ST2205 has one Low Voltage Detector (LVD) for power management. The status of internal or external power can be detected and reported to the management software.

Power bouncing during power on is a major problem when designing a reliable system. The ST2205U equips Low Voltage Reset (LVR) function to keep whole system in reset status when power is low. After the power backs to normal, the system may recover its original states and keeps working correctly. Besides LVR, Watch Dog Timer (WDT) is also built-in and is an essential function for a good design.

Power consumption is another big issue for a battery-powered device. The ST2205U has different power down modes and clock switch scheme to make the consuming power as low as possible. The built-in Real Time Clock (RTC) is not only for keeping time correctly but also an alternative of software timer with much lower working power.

The ST2205U equips an ICE debug interface for efficient development flow. Besides hardware emulator, a software simulator is also supported to save programmers setting up the system and makes programming be at anywhere.

With these integrated functions inside, the ST2205U single chip microcontroller is a right solution for PDA, translator, databank and other consumer products.

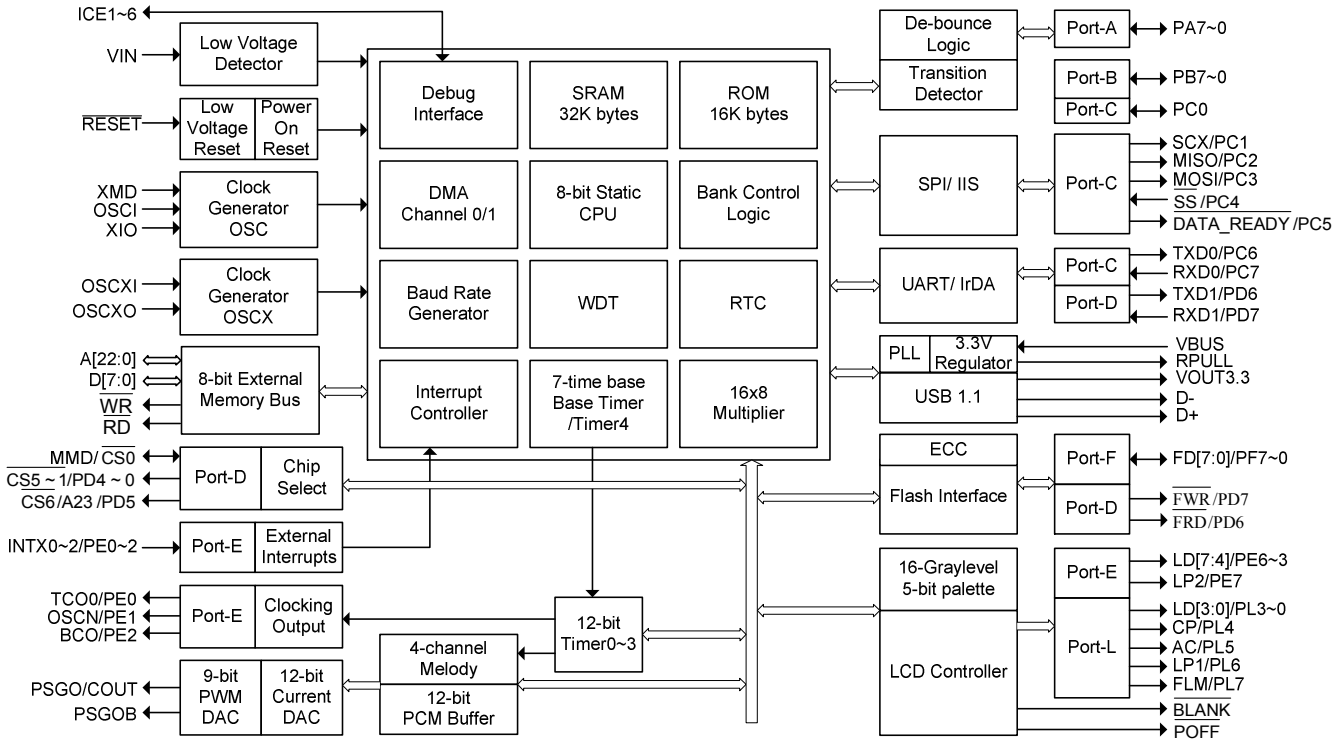



**FIGURE 2-1 ST2205U Block Diagram**

# 3. SIGNAL DESCRIPTIONS

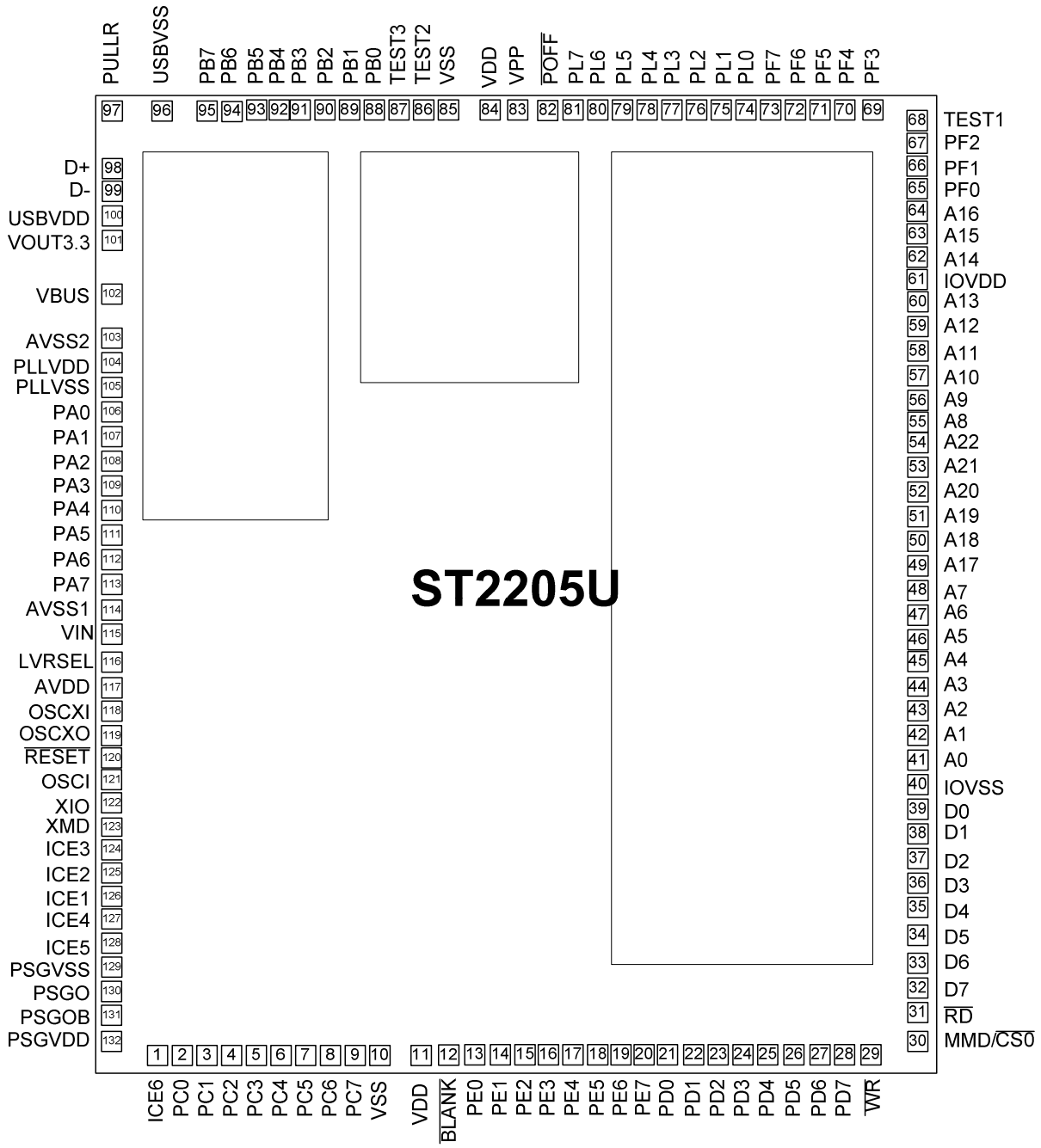
**TABLE 3-1 Signal Function Groups**

| Function Group | Pad No. | Designation | Description |
|---|---|---|---|
| Power | 11,61,83,<br>84,100,<br>104,117<br>132 | VDD, IOVDD,AVDD<br>PSGVDD, USBVDD<br>PLLVDD, VPP | **VDD:** Power supply for internal core<br>**IOVDD:** Power supply for IO<br>**AVDD:** Power supply for analog blocks<br>**PSGVDD:** Power supply for PSGO and PSGOB<br>**USBVDD:** Power supply for USB circuit<br>**PLLVDD:** Power supply for PLL circuit<br>**VPP:** Power supply for programming OTP ROM |
| Ground | 10,40,85,<br>96,103,<br>105,114,<br>129 | VSS,IOVSS,AVSS1<br>AVSS2,PSGVSS,<br>USBVSS,PLLVSS | **VSS:** Power ground for internal core<br>**IOVSS:** Power ground for IO<br>**AVSS:** Power ground for analog blocks<br>**PSGVSS:** Power ground for PSGO and PSGOB<br>**USBVSS:** Power ground for USB circuit<br>**PLLVSS:** Power ground for PLL circuit |
| System control | 1,30,68,86<br>87,115,116<br>120,<br>124~128 | $\overline{\text{RESET}}$ ,<br>TEST1/2/3,<br>ICE1/2/3/4/5/6,<br>MMD/$\overline{\text{CS0}}$ , LVRSEL<br>VIN | $\overline{\textbf{RESET}}$ **:** Active low system reset signal input<br>**TEST1/2/3, ICE1/2/3/4/5/6:** Leave them open when normal operation<br>**MMD/CS0:** Memory modes selection pin<br>**Normal mode:** Enable internal ROM.<br>MMD/$\overline{\text{CS0}}$ connects to GND.<br>**Emulation mode:** Disable internal ROM.<br>MMD/$\overline{\text{CS0}}$ connects to chip-select pin of external ROM. One resistor should be added between VCC and this pin. After reset cycles, MMD/$\overline{\text{CS0}}$ changes to be an output, and outputs signal $\overline{\text{CS0}}$ .<br>**LVRSEL**: LVR active level selection input<br>**Low:** LVR active level is 2.1V<br>**High:** LVR active level is 2.8V<br>**VIN**: Input voltage level for Low Voltage Detection |
| Clock | 118,119,<br>121~123 | XMD,<br>XIO,OSCI<br>OSCXO,OSCXI | **XMD:** High frequency oscillator (OSC) mode selection input<br>**Low:** Crystal mode<br>One crystal or resonator should be connected between OSCI and XIO<br>**High:** Resistor oscillator mode<br>One resistor should be connected between OSCI and VCC<br>**OSCXI, OSCXO:** Connect one 32768Hz crystal between these two pins when using low frequency oscillator |
| External memory bus signals | 29,31 | $\overline{\text{WR}}$ , $\overline{\text{RD}}$ | External memory R/W control signals |
| | 41~60,<br>62~64 | A[22:0] | External memory address bus |
| | 32~39 | D[7:0] | External memory data bus |
| PWM DAC<br>Current DAC | 130,131 | PSGO/COUT,<br>PSGOB | **PSGO/PSGOB:** PSG outputs. Connect to one buzzer or speaker<br>**COUT:** Also 12-bit current DAC output by register control |
| Keyboard scan signal (return line) | 106~113 | PA7~0 | I/O port A |
| GPIO | 2,88~95 | PB7~0<br>PC0 | I/O port B and PC0 |

**TABLE 3-2 Signal Function Groups (continued)**

| Function Group | Pad No. | Designation | Description |
|---|---|---|---|
| Flash Data Bus | 65~67, 69~73 | FD7~0/PF7~0 | Flash data bus |
| Flash read/write signals | 27,28 | RXD1/ $\overline{\text{FWR}}$ /PD7<br>TXD1/ $\overline{\text{FRD}}$ /PD6 | When function bits are set, and I/O direction is output, and FEN=1, PD7/6 are flash control signals |
| Chip selects | 21~26 | $\overline{\text{CS5}}$ ~ 1/PD4~0,<br>$\overline{\text{CS6}}$ /A23/PD5 | I/O port D and chip-select outputs |
| UART | 8,9<br>27,28 | RXD0/PC7,TXD0/PC6, RXD1/ $\overline{\text{FWR}}$ /PD7<br>TXD1/ $\overline{\text{FRD}}$ /PD6 | UART signals and I/Os |
| SPI | 3~7 | $\overline{\text{DATA\_READY}}$ /PC5 ,<br>$\overline{\text{SS}}$ /PC4 , SDO/PC3 ,<br>SDI/PC2 , SCK/PC1 | SPI signals and I/Os |
| Clocking output/ External clock input or interrupt sources | 13~15 | BCO/INTX2/PE2 ,<br>OSCN/INTX1//PE1<br>TCO0/INTX0//PE0 | ■ When function bits are set, and I/O direction is output, these three can be clocking outputs.<br>■ When function bits are set, and I/O direction is input, these three can be external clock inputs or external interrupt sources.<br>■ When function bits are cleared, they are three GPIOs. |
| LCD control signals | 12,16~20, 74~82 | FLM/PL7, LP1/PL6,<br>AC/PL5 , CP/PL4,<br>LD[3:0]/PL3~0,<br>LD[7:4]/PE6~3,<br>LP2/PE7,<br>$\overline{\text{POFF}}$ , $\overline{\text{BLANK}}$ , | LCD control signals |
| USB 2.0 Full speed | 97~99,101 102 | VBUS,<br>RPULL, VOUT3.3,<br>D+, D- | **VBUS:** Connect to USB bus power<br>**D+,D-:** USB differential signal pins<br>**RPULL:** Add a resistor of 1.5KΩ between this pin and D+<br>**VOUT3.3:** 3.3V regulator output. Connect to USBVDD to supply power for the analog transceiver of USB |

# 4. PAD DIAGRAM



# 5. DEVICE INFORMATION

1. **Pad size: 90um x 90um**
2. **Substrate: GND**
3. **Chip size: 3490um x 4070um**

| PAD No. | Symbol | X | Y |
|---|---|---|---|
| 1 | ICE6 | -1465.1 | -1965.0 |
| 2 | PC0 | -1345.1 | -1965.0 |
| 3 | PC1 | -1245.1 | -1965.0 |
| 4 | PC2 | -1145.1 | -1965.0 |
| 5 | PC3 | -1045.1 | -1965.0 |
| 6 | PC4 | -945.1 | -1965.0 |
| 7 | PC5 | -845.1 | -1965.0 |
| 8 | PC6 | -745.1 | -1965.0 |
| 9 | PC7 | -645.1 | -1965.0 |
| 10 | VSS | -545.1 | -1965.0 |
| 11 | VDD | -345.1 | -1965.0 |
| 12 | $\overline{\text{BLANK}}$ | -245.1 | -1965.0 |
| 13 | PE0 | -145.1 | -1965.0 |
| 14 | PE1 | -45.1 | -1965.0 |
| 15 | PE2 | 55.0 | -1965.0 |
| 16 | PE3 | 155.0 | -1965.0 |
| 17 | PE4 | 255.0 | -1965.0 |
| 18 | PE5 | 355.0 | -1965.0 |
| 19 | PE6 | 455.0 | -1965.0 |
| 20 | PE7 | 555.0 | -1965.0 |
| 21 | PD0 | 655.0 | -1965.0 |
| 22 | PD1 | 755.0 | -1965.0 |
| 23 | PD2 | 855.0 | -1965.0 |
| 24 | PD3 | 955.0 | -1965.0 |
| 25 | PD4 | 1055.0 | -1965.0 |
| 26 | PD5 | 1155.0 | -1965.0 |
| 27 | PD6 | 1255.0 | -1965.0 |
| 28 | PD7 | 1355.0 | -1965.0 |
| 29 | $\overline{\text{WR}}$ | 1475.0 | -1965.0 |
| 30 | MMD/$\overline{\text{CS0}}$ | 1675.0 | -1940.0 |
| 31 | $\overline{\text{RD}}$ | 1675.0 | -1820.0 |
| 32 | D7 | 1675.0 | -1700.0 |
| 33 | D6 | 1675.0 | -1600.0 |
| 34 | D5 | 1675.0 | -1500.0 |
| 35 | D4 | 1675.0 | -1400.0 |

| PAD No. | Symbol | X | Y |
|---|---|---|---|
| 36 | D3 | 1675.0 | -1300.0 |
| 37 | D2 | 1675.0 | -1200.0 |
| 38 | D1 | 1675.0 | -1100.0 |
| 39 | D0 | 1675.0 | -1000.0 |
| 40 | IOVSS | 1675.0 | -900.0 |
| 41 | A0 | 1675.0 | -800.0 |
| 42 | A1 | 1675.0 | -700.0 |
| 43 | A2 | 1675.0 | -600.0 |
| 44 | A3 | 1675.0 | -500.0 |
| 45 | A4 | 1675.0 | -400.0 |
| 46 | A5 | 1675.0 | -300.0 |
| 47 | A6 | 1675.0 | -200.0 |
| 48 | A7 | 1675.0 | -100.0 |
| 49 | A17 | 1675.0 | 0.0 |
| 50 | A18 | 1675.0 | 100.0 |
| 51 | A19 | 1675.0 | 200.0 |
| 52 | A20 | 1675.0 | 300.0 |
| 53 | A21 | 1675.0 | 400.0 |
| 54 | A22 | 1675.0 | 500.0 |
| 55 | A8 | 1675.0 | 600.0 |
| 56 | A9 | 1675.0 | 700.0 |
| 57 | A10 | 1675.0 | 800.0 |
| 58 | A11 | 1675.0 | 900.0 |
| 59 | A12 | 1675.0 | 1000.0 |
| 60 | A13 | 1675.0 | 1100.0 |
| 61 | IOVDD | 1675.0 | 1200.0 |
| 62 | A14 | 1675.0 | 1300.0 |
| 63 | A15 | 1675.0 | 1400.0 |
| 64 | A16 | 1675.0 | 1500.0 |
| 65 | PF0 | 1675.0 | 1600.0 |
| 66 | PF1 | 1675.0 | 1700.0 |
| 67 | PF2 | 1675.0 | 1820.0 |
| 68 | TEST1 | 1675.0 | 1940.0 |
| 69 | PF3 | 1475.0 | 1965.0 |
| 70 | PF4 | 1355.0 | 1965.0 |

| PAD No. | Symbol | X | Y |
|---|---|---|---|
| 71 | PF5 | 1255.0 | 1965.0 |
| 72 | PF6 | 1155.0 | 1965.0 |
| 73 | PF7 | 1055.0 | 1965.0 |
| 74 | PL0 | 955.0 | 1965.0 |
| 75 | PL1 | 855.0 | 1965.0 |
| 76 | PL2 | 755.0 | 1965.0 |
| 77 | PL3 | 655.0 | 1965.0 |
| 78 | PL4 | 555.0 | 1965.0 |
| 79 | PL5 | 455.0 | 1965.0 |
| 80 | PL6 | 355.0 | 1965.0 |
| 81 | PL7 | 255.0 | 1965.0 |
| 82 | $\overline{\text{POFF}}$ | 155.0 | 1965.0 |
| 83 | VPP | 55.0 | 1965.0 |
| 84 | VDD | -45.0 | 1965.0 |
| 85 | VSS | -245.0 | 1965.0 |
| 86 | TEST2 | -345.0 | 1965.0 |
| 87 | TEST3 | -445.0 | 1965.0 |
| 88 | PB0 | -545.0 | 1965.0 |
| 89 | PB1 | -645.0 | 1965.0 |
| 90 | PB2 | -745.0 | 1965.0 |
| 91 | PB3 | -845.0 | 1965.0 |
| 92 | PB4 | -945.0 | 1965.0 |
| 93 | PB5 | -1045.0 | 1965.0 |
| 94 | PB6 | -1145.0 | 1965.0 |
| 95 | PB7 | -1245.0 | 1965.0 |
| 96 | USBVSS | -1438.0 | 1965.0 |
| 97 | RPULL | -1656.1 | 1965.0 |
| 98 | D+ | -1675.0 | 1764.7 |
| 99 | D- | -1675.0 | 1644.7 |
| 100 | USBVDD | -1675.0 | 1532.7 |
| 101 | VOUT3.3 | -1675.0 | 1422.7 |
| 102 | VBUS | -1675.0 | 1232.7 |
| 103 | AVSS2 | -1675.0 | 1075.8 |
| 104 | PLLVDD | -1675.0 | 975.8 |
| 105 | PLLVSS | -1675.0 | 875.8 |

| PAD No. | Symbol | X | Y |
|---|---|---|---|
| 106 | PA0 | -1675.0 | 747.1 |
| 107 | PA1 | -1675.0 | 647.1 |
| 108 | PA2 | -1675.0 | 547.1 |
| 109 | PA3 | -1675.0 | 447.1 |
| 110 | PA4 | -1675.0 | 347.1 |
| 111 | PA5 | -1675.0 | 247.1 |
| 112 | PA6 | -1675.0 | 147.1 |
| 113 | PA7 | -1675.0 | 47.1 |
| 114 | AVSS1 | -1675.0 | -53.0 |
| 115 | VIN | -1675.0 | -153.0 |
| 116 | LVRSEL | -1675.0 | -253.0 |
| 117 | AVDD | -1675.0 | -353.0 |
| 118 | OSCXI | -1675.0 | -453.0 |
| 119 | OSCXO | -1675.0 | -553.0 |
| 120 | $\overline{\text{RESET}}$ | -1675.0 | -653.0 |
| 121 | OSCI | -1675.0 | -753.0 |
| 122 | XIO | -1675.0 | -853.0 |
| 123 | XMD | -1675.0 | -953.0 |
| 124 | ICE3 | -1675.0 | -1053.0 |
| 125 | ICE2 | -1675.0 | -1153.0 |
| 126 | ICE1 | -1675.0 | -1253.0 |
| 127 | ICE4 | -1675.0 | -1353.0 |
| 128 | ICE5 | -1675.0 | -1453.0 |
| 129 | PSGVSS | -1675.0 | -1553.0 |
| 130 | PSBO | -1675.0 | -1673.0 |
| 131 | PSGOB | -1675.0 | -1793.0 |
| 132 | PSGVDD | -1675.0 | -1913.0 |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

# 6. INTERRUPT CONTROLLER

The ST2205U supports 16 hardware interrupts as well as one software interrupt Brk. There are 17 exception vectors for these interrupts and another one for reset. All interrupts are controlled by interrupt disable flag "**I**" (bit2 of status register **P**). Hardware interrupts are further controlled by interrupt enable register **IENA**. Setting bits of **IENA** enables respective interrupts.

The interrupt controller owns one priority arbitrator. When more than one interrupts happen at the same time, the one with lower priority number will be executed first. Refer to TABLE 6-1 for priorities of interrupts.

Once an interrupt event was enabled and then happens, the CPU wakes up (if in either wait mode), and associated bit of interrupt request register (**IREQ**) will be set. If "**I**" flag is cleared, the related vector will be fetched and then the interrupt service routine (ISR) will be executed. Interrupt request flag can be cleared by two methods. One is to write "0" to **IREQ**, the other is to initiate related interrupt service routine. Hardware will automatically clear the Interrupt request flag. All interrupt vectors are listed in TABLE 6-1.

**TABLE 6-1 Interrupt Vectors**

| Name | Signal Source | Vector Address | Priority | Description |
|---|---|---|---|---|
| BRK | Internal | $7FFF,$7FFE | 1 | Software BRK operation vector |
| RESET | External | $7FFD,$7FFC | 0 | Reset vector |
| - | - | $7FFB,$7FFA | - | Reserved |
| INTX | External | $7FF9,$7FF8 | 9 | PE0/1/2 edge interrupt |
| T0 | Internal/External | $7FF7,$7FF6 | 10 | Timer0 interrupt |
| T1 | Internal | $7FF5,$7FF4 | 11 | Timer1 interrupt |
| T2 | Internal/External | $7FF3,$7FF2 | 12 | Timer2 interrupt |
| T3 | Internal | $7FF1,$7FF0 | 13 | Timer3 interrupt |
| PT | External | $7FEF,$7FEE | 14 | Port-A transition interrupt |
| BT | Internal | $7FED,$7FEC | 15 | Base Timer interrupt |
| LCD | Internal | $7FEB,$7FEA | 16 | LCD buffer interrupt |
| STX | External | $7FE9,$7FE8 | 1 | SPI transmit buffer empty interrupt |
| SRX | External | $7FE7,$7FE6 | 2 | SPI receive buffer ready interrupt |
| UTX | External | $7FE5,$7FE4 | 3 | UART transmitter interrupt |
| URX | External | $7FE3,$7FE2 | 4 | UART receiver interrupt |
| USB | External | $7FE1,$7FE0 | 5 | USB interrupt |
| Reserved | | $7FDF,$7FDE | 6 | |
| PCM | Internal | $7FDD,$7FDC | 7 | PCM interrupt |
| RTC | Internal | $7FDB,$7FDA | 8 | RTC interrupt |

# 7. GPIO

The ST2205U consists of 48 general-purpose I/O (GPIO) which are divided into six I/O ports: Port-A/B/C/D/E and Port-L.

Each single pin can be programmed to be input or output. This is controlled by port direction control registers **PCx**. Setting bit of **PCx** makes respective pin to output, and clearing this bit for input. There are two options: pull-up/down for inputs of Port-C but only pull-up for inputs of the other ports. In case of output, there are open-drain/CMOS options for outputs of PortC but only CMOS for the other ports. Refer to TABLE 7-1.

**TABLE 7-1 I/O Types Of GPIO Ports**

| I/O Mode | I/O Types | |
|---|---|---|
| | Port-A/B/D/E/L | Port-C |
| **Input** | Pull-up/Pure | Pull-up/Pull-down/Pure |
| **Output** | CMOS | Open-drain/CMOS |

## ■ Input Mode

In case of input function, port data registers **Px** reflect the values on associated pins. Besides read instruction for data of signals input, writing to register **Px** selects I/O types of pins, pull-up or pull-down. Setting bits of all port data register **Px** to select pull-up type. Clearing bits of only **PC** to select pull-down type for pins of Port-C. There are no pull-down resistors for Port-A/B/D/E and Port-L, thereby no pull-down resistors will be enabled if clearing bits of **PA**, **PB**, **PD**, **PE** and **PL**. Pull-up resistors of Port-A/B/D/E/L are also controlled by PULL bit (bit7 of port miscellaneous register **PMCR**), “0” is to disable, while “1” is to enable them. The pull-up/pull-down resistors of Port-C are further controlled by bits of port type select registers **PSC**. They work in the same way with PULL bit of **PMCR** but only on single pin, “0” is to disable, while “1” is to enable.

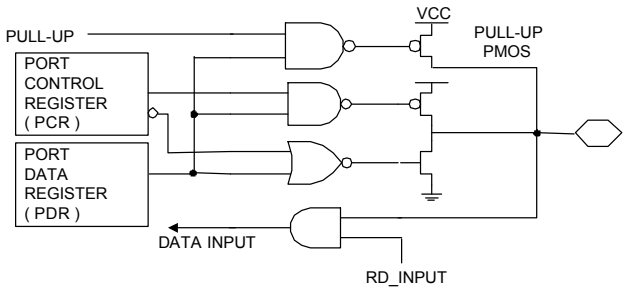



**FIGURE 7-1 Configuration Of Port-A/B/D/E/L**

## ■ Output Mode

In case of output function, Write to port data registers **Px** makes pins to output desired value. This value can also be read back by read instruction. Besides Port-C, the output pins are CMOS type. Port-C have two options of output types: open-drain and CMOS, and is controlled by port type select registers **PSC**. Clearing bits of registers **PSC** is for that disable PMOS of output stage and left only NMOS, while setting bits is for CMOS.

Port-A is designed for keyboard scan with de-bounce and transition triggered interrupt, while Port-C/D/E are multiplexed with other system functions, and are controlled by **PFC**, **PFD**, and **PMCR[2:0]**. Port-L is shared with LCD specific signals of LCDC. Turning off LCDC by setting **LPWR** (**LCTR[7]**) reserves Port-L for GPIO.

Selecting respective pins to be GPIO or signals of system function will not affect original settings of I/O directions and types. This entends the flexibility of the usage of function signals.

Note: All the properties of pins are still programmable and must be ascertained before they are assigned to system functions, especially the direction of pins.

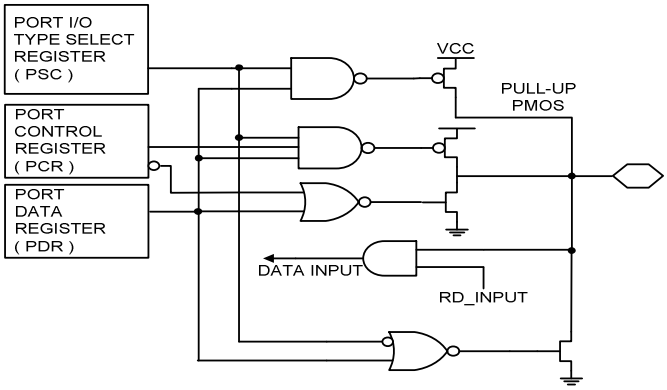


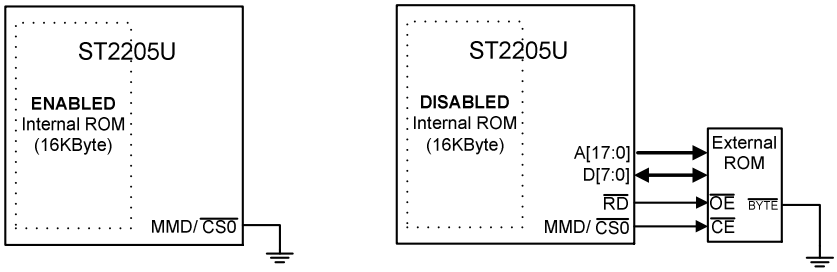

**FIGURE 7-2 Configuration Of Port-C**

# 8. CHIP-SELECT LOGIC (CSL)

The ST2205U builds in one chip-select signal ( $\overline{CS0}$ ) for embedded 16K bytes mask ROM and six chip-select signals multiplexed with PD5~0 of Port-D which are used to select devices on the external bus. There are two options for the first 16K bytes memory which are controlled by MMD pin. Tie MMD to ground to select normal mode and enable internal ROM for the first 16K bytes memory. Connect MMD to chip-select of an external device to select emulation mode and disable internal ROM. After reset cycles, MMD changes to an output and outputs chip-select signal $\overline{CS0}$. Refer to FIGURE 8-1 for two connections of different modes.

Two bits **CSM[1:0]** of port miscellaneous register (**PMCR**) select four modes of CSL which define the memory size of each external chip-select. Chip-select signal $\overline{CS6}$ can change to be address signal A23 to make one single device of 16M bytes at $\overline{CS5}$ possible. The address range of $\overline{CSx}$ of higher number follows the range of previous one of lower number.

Note: Write “1” to bit of port direction control register **PCD**, then to bit of port function-select register **PFD** to activate the designated chip-select signal.




**A. Normal Mode** **B. Emulation Mode**

**FIGURE 8-1 Connections Of MMD/ $\overline{CS0}$**

# 9. TIMER/EVENT COUNTER

## 9.1 Prescaler

### 9.1.1 Function Description

The ST2205U has four 12-bit timers, eight base timers with 7 fixed timer bases and one adjustable. There is a prescaler that generate 6 different clock soure to support the Timers counting to interrupt .

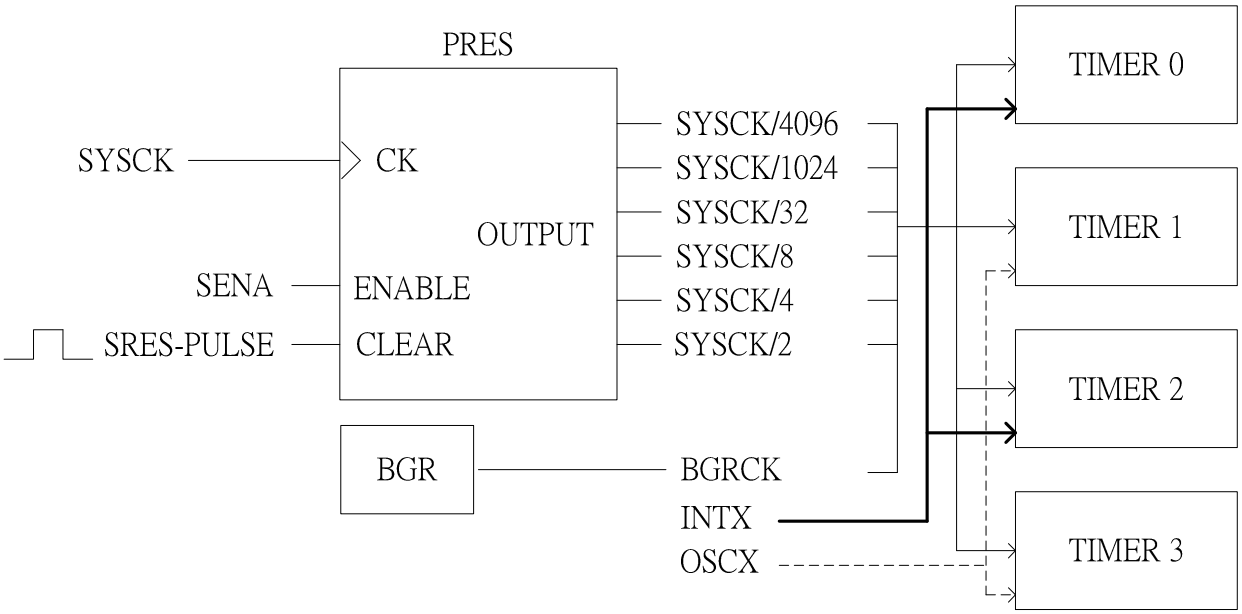



**FIGURE 9-1 Structure Of Two Prescalers**

### 9.1.2 PRES

The prescaler PRES is an 8-bits counter as shown in FIGURE 9-1. Which provides six clock sources for 12bit up counting timer. 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 or enable PRES.

**TABLE 9-1 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $29 | PRS* | R | PRS[7] | PRS[6] | PRS[5] | PRS[4] | PRS[3] | PRS[2] | PRS[1] | PRS[0] | 0000 0000 |
| | | W | SRES | SENA | - | - | - | - | - | - | 00- - - - - - |

READ
Bit 7~0: **PRS[7~0] :** 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

## 9.2 Base Timer

The base timer supports one interrupt, which occurs at seven different fixed rates and one adjustable clock. Applications base on the base timer interrupt can chose an appropriate interrupt rate from eight time bases for their specific needs. These real-time applications may include digitizer sampling, keyboard debouncing, or communication polling. Block diagram of base timer is shown in FIGURE 9-2.

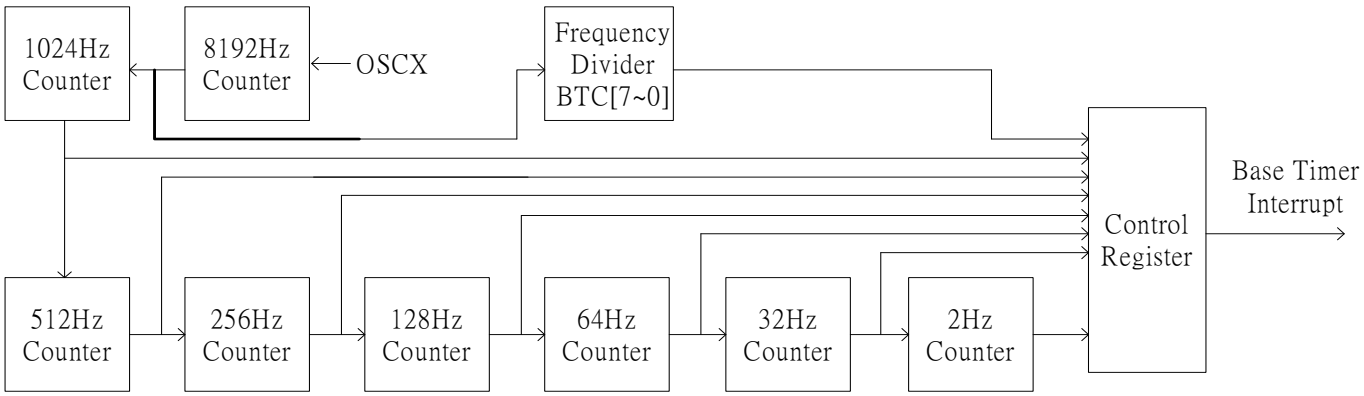



**FIGURE 9-2 Base Timer Block Diagram**

### 9.2.1 Base Timer Operations

The base timer consists of eight sub-counters and one divider to produce eight predefined rates. The connections between overflow signals of these sub-counters and the base timer interrupt are controlled by respective bit fields of base timer enable register (**BTEN**). The enabled overflow signals are ORed to generate the base timer interrupt request. Related bits of base timer status register (**BTSR**) will show which rates of interrupts should be serviced. Write “1” to each bit of the register may clear each bit of the register respectively.

Note: Make sure **BTSR** is cleared after the interrupt was serviced, so that the request can be set next time.

# 10. PSG

## 10.1 Function Description

The built-in four channel Programmable Sound Generator (PSG) is controlled by register file directly. Its flexibility makes it useful in applications such as music synthesis, sound effects generation, audible alarms and tone signaling. In order to generate sound effects while allowing the processor to perform other tasks, the PSG can continue to produce sound after the initial commands have been given by the CPU. The structure of PSG was shown in FIGURE 10-1 and FIGURE 10-2. Each channel of PSG of the ST2205U has three playing type. One for square type tone sound playing. Second for DAC PCM playing. The third sound playing type is DAC ADPCM playing. The three type can be applied in the four channels and mixed to one output signal to make the PSG generates melody and voice at the same time.

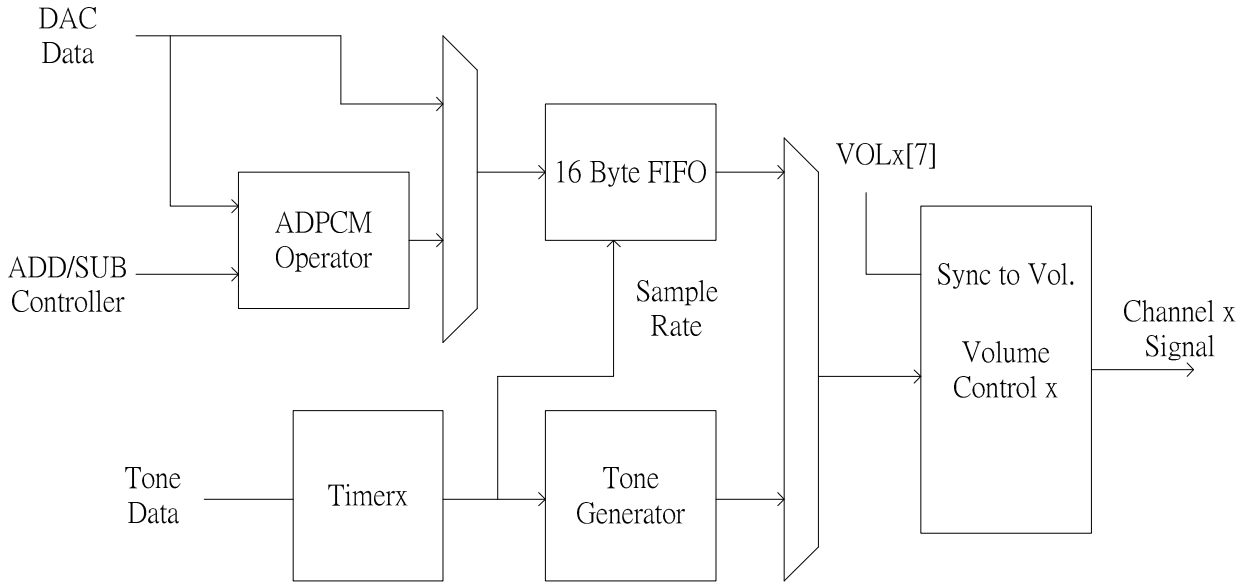



**FIGURE 10-1 PSG One Channel Structure Block**

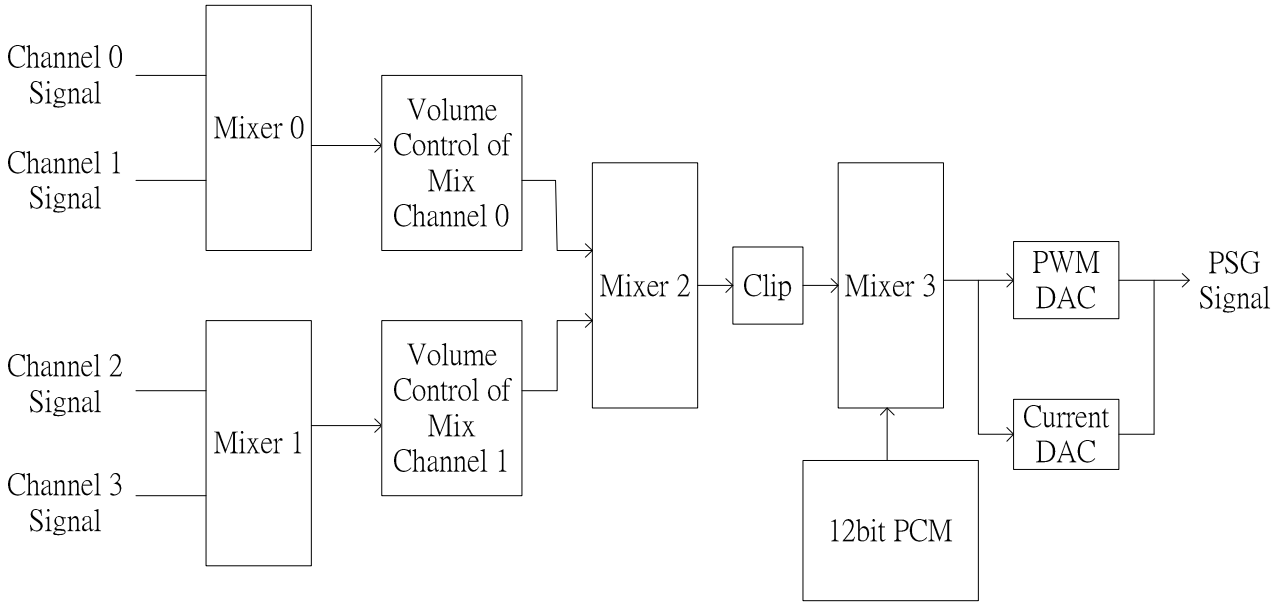



**FIGURE 10-2 PSG Four Channel Mixer Structure Block**

## 10.2 Tone Generator

The tone frequency is decided by Timer and the volume is controlled by DAC data output register (PSGxA).Besides DAC data can be used to adjust volume, the two level volume control(VOLx & VOLMx) are effective, too. So it's very flexible to generate any tone sound which you want.

For example: If the 1KHz tone sound want to be generated on Channel0 and the volume is maximum. First, the Timer0 must be set up 2KHz and write FFH to DAC data (PSGA0). Second, the two level volume control are adjusting to maximum..

## 10.3 PCM DAC

A built-in PWM DAC is for analog sampling data or voice signals. There is an interrupt signal which is controlled by Timer form DAC to CPU whenever DAC data update is needed and the same signal will decide the sampling rate of voice. Each channel has a 16 byte FIFO. When the FIFO empty byte is more than 8, the Timer interrupt will be triggered. Besides, There are two steps volume control to adjust one channel integrate volume and a couple of channels integrate volume. Refer to description of following TABLE.

## 10.4 ADPCM DAC

ADPCM is a kind of encode of voice compression. The compression data usually is an index. It's through the index to get an offset value of the present voice sample data. In ADPCM DAC mode, we just store the offset value to register PSGxA to add to present voice sample data, or store the offset value to register PSGxB to subtract to present voice sample data.

■ **Multiplicator**

ST2205U build-in a 16x8 multiplicator for wave-table operation. We just write twice to "MULH" that first is multiplicand low byte then high byte and "MULL" is multiplier. After the multiplier is written and wait 6 OP cycle, the answer's bit23~8 can be read from "MULH" and "MULL", the bit7~0 is ignored. Besides, the answer was reloaded to multiplicand automatically when the answer has appeared.

## 10.5 PWM DAC Output 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 PSGO[1~0] of register PSGC[2~1]..

### 10.5.1 Single-Pin Mode (8-bit Accuracy)

Single-pin mode is designed for use with a single-transistor amplifier. It has 8 bits of resolution. The duty cycle of the PSGOB 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 127, the duty cycle goes from being high 50% of the time up to 100% high. As the value goes from 0 to -128, the duty cycle decreases from 50% high to 0%. PSGO is inverse of PSGOB's waveform. Figure 13-3 shows the PSGOB waveforms.

# 11. LCD

The LCD controller (LCDC) provides display data and specific signals for external LCD drivers to drive the STN LCD panels. The LCDC fetches display data directly from internal display buffer through one unique memory bus. The special designed internal bus shares almost none of the CPU resources to make both fast display data process and high speed CPU operation possible.

ST2205U support three display modes including black-and-white, 4-gray-level and 16-gray-level and is selected by **GL[3:2]** of control register **LCTR**. Further, it through PWM + FRC technique that selected by **GL[1~0]** to generate 31 gray levels and provides one palette **LPAL($4C)** to choose 16 gray levels which make the 4-gray-level and 16-gray-level more smoothly than only FRC.

The ST2205U builds in 32K bytes SRAM, so the maximum panel size can be 640x400 for B/W, 400x320 for 4-gray-level and 160xRGBx120 for 16-gray-level mode.

LCDCK is for LCDC to generate timings and the pixel clock. The ST2205U supports 1-bit, 4-bit and 8-bit data bus for the compatibility of most popular LCD drivers. The LCD output signals are shared with Port-L, and are controlled by LCD power control bit **LPWR** (**LCTL[7]**) and data bus selection bits **LMOD[1:0]**. In case of 1-bit mode, PL2~1 can still be used for general purpose while only PL0 outputs LCD data.

Note:
- **A.** The LCD signals will be disconnected and Port-L will output values assigned by **PL** after setting **LPWR**.
- **B.** Set **PL**="00h" to make Port-L output zeros when LCDC is off.

Various functions are also supported to rich the display information, including virtual screen, panning, scrolling, contrast control and an alternating signal generator. Control registers used by LCDC are listed below.

## 11.1 LCD Specific Signals

The following signals are generated by LCDC to connect the ST2205U and an LCD module. Two of them are dedicated output pins, while the rest 13 pins are shared with Port-L and Port-E

### ■ FLM (PL7)

The LCD frame marker signal indicates the start of a new display frame. FLM becomes active after the last line pulse of the frame and remains active until the next line pulse, at which point it de-asserts and remains inactive until the next frame.

### ■ LP1 (PL6)

The LCD line pulse signal is used to latch a line of shifted data to the segment drivers' outputs and is also used to shift the line enable signal of common driver. All the driver outputs then control the liquid crystal to form the desired frame on panel.

### ■ AC (PL5)

The LCD alternate signal toggles the polarity of liquid crystal on the panel. This signal can be programmed to toggle for a period of 1 to 31 lines or one frame.

### ■ CP

The LCD shift clock pulse signal is the clock output to which the output data to the LCD panel is synchronized. Data for segment drivers is shifted into the internal line buffer at each falling edge of CP.

### ■ LD7~0 (PE6~3, PL3~0)

The LCD data bus lines transfer pixel data to the LCD panel so that it can be displayed. Three kinds of data busses, 1- , 4- and 8-bit, are supported and are controlled by **LMOD[1:0]** (**LCKR[5:4]**). In case of 1-bit mode, LCDC uses only LD0 to transfer data. LD3~1 can still be programmed to be normal inputs or outputs. The output pixel data can be inverted through programming. Setting **REV** (**LCTR**) will reverse the output data on data bus.

### ■ $\overline{\text{POFF}}$ (Power control)

The LCD power control signal is used to turn on/off the external DC-DC converter, which generates a high voltage for driving liquid crystal. $\overline{\text{POFF}}$ outputs "1" when clearing **LPWR** (**LCTR**), and outputs "0" by setting this bit, which is also the default value.

### ■ $\overline{\text{BLANK}}$ (Contrast control)

The LCD blank signal is used to control the contrast of display by setting contrast level in **LPWM[5:0]** with "00000" (default) represents a maximum level and "11111" is for minimum. The $\overline{\text{BLANK}}$ signal achieves this function by outputting a PWM signal according to the settings of contrast.

Besides contrast control, $\overline{\text{BLANK}}$ signal plays another role of turning display off. This is controlled by register bit **BLNK** (**LCTR[6]**). Setting **BLNK** will make $\overline{\text{BLANK}}$ signal to output "0" to blank the display regardless of contrast control. Setting **BLNK** bit will enable the PWM contrast control and of course the $\overline{\text{BLANK}}$ signal. If **LPWM[5:0]** are all zeros, $\overline{\text{BLANK}}$ signal will stay at high level with no PWM modulation.

### ■ LP2 (PE7)

When PWM gray-level function is enabled by setting **GL[1:0]** (**LCTR[4]**), the PWM line pulse signal will be outputted from this pin. When this function is off , LP2 outputs the identical signal with that of LP1.

## 11.2 Mapping the Display Data

The screen width and height of the LCD panel are programmable through software. Although the maximum screen size can be up to 1024x512, the actual supported resolution is limited by the display buffer size, which is also the internal RAM size, and is 32K bytes. Instead of screen size specified by control registers, larger frame can also be displayed via the Virtual Page Width setting. FIGURE 11-1 illustrates the relationship between the portion of a large graphic to be displayed on the screen and the actual area that can be seen.

Each one or two even four bits in the display memory correspond to a pixel on the LCD panel. TABLE 11-1 shows the mapping of the display data to the pixel on LCD. When clear control bits **GL[3~2]** (**LCTR[3~2]**) and enable B/W mode, every bit of display buffer represents one pixel on the screen. In case of 4-gray-level mode, there needs two bits to present each pixel on the screen. And there needs 4 bits for 16-gray-level mode to display one pixel.

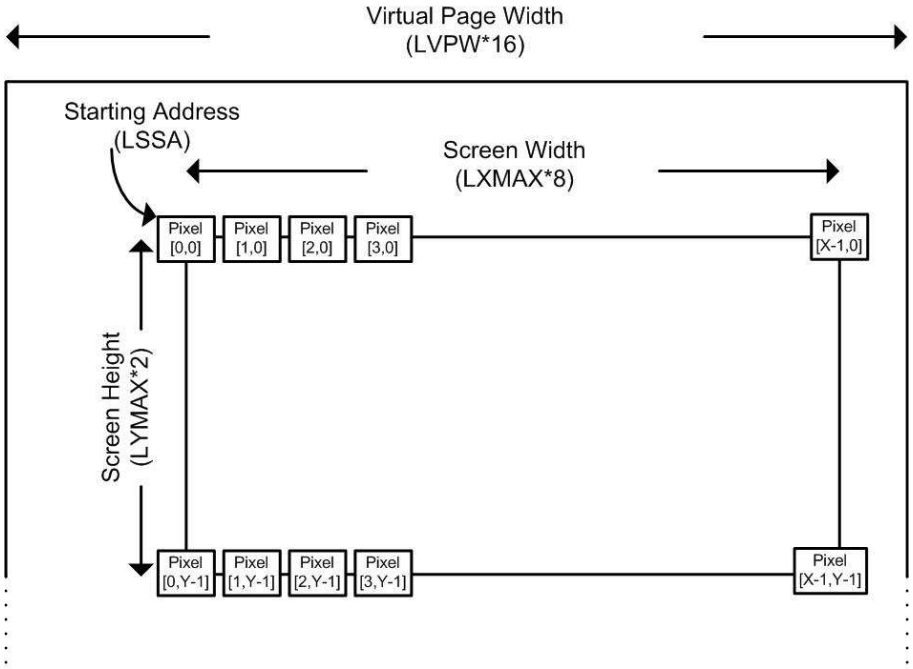



**FIGURE 11-1 LCD Screen Format**

**TABLE 11-1 Mapping Memory Data on the Screen**

**A. 1-bit-per-pixel mode**

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Bit7 | Bit6 | … |
|---|---|---|---|---|---|---|---|---|---|---|
| Pixel [0,0] | Pixel [1,0] | Pixel [2,0] | Pixel [3,0] | Pixel [4,0] | Pixel [5,0] | Pixel [6,0] | Pixel [7,0] | Pixel 8,0] | Pixel [9,0] | … |
| : | : | : | : | : | : | : | : | : | : | … |

**B. 2-bit-per-pixel mode**

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Bit7 | Bit6 | … |
|---|---|---|---|---|---|---|---|---|---|---|
| Pixel [0,0] | | Pixel [1,0] | | Pixel [2,0] | | Pixel [3,0] | | Pixel [4,0] | | … |
| : | | : | | : | | : | | : | | … |

**C. 4-bit-per-pixel mode**

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | … |
|---|---|---|---|---|---|---|---|---|
| Pixel [0,0] | | | | Pixel [1,0] | | | | … |
| : | | | | : | | | | … |

## 11.3 LCD Interface Timing

The LCD controller continuously pumps the pixel data into the LCD panel via the LCD data bus. The bus is timed by the CP, LOAD, and FLM signals. Two kinds of data width, 1- and 4-bit, are supported for most monochrome LCD panels. Refer to FIGURE 11-2 for both 1- and 4-bit interface timing.

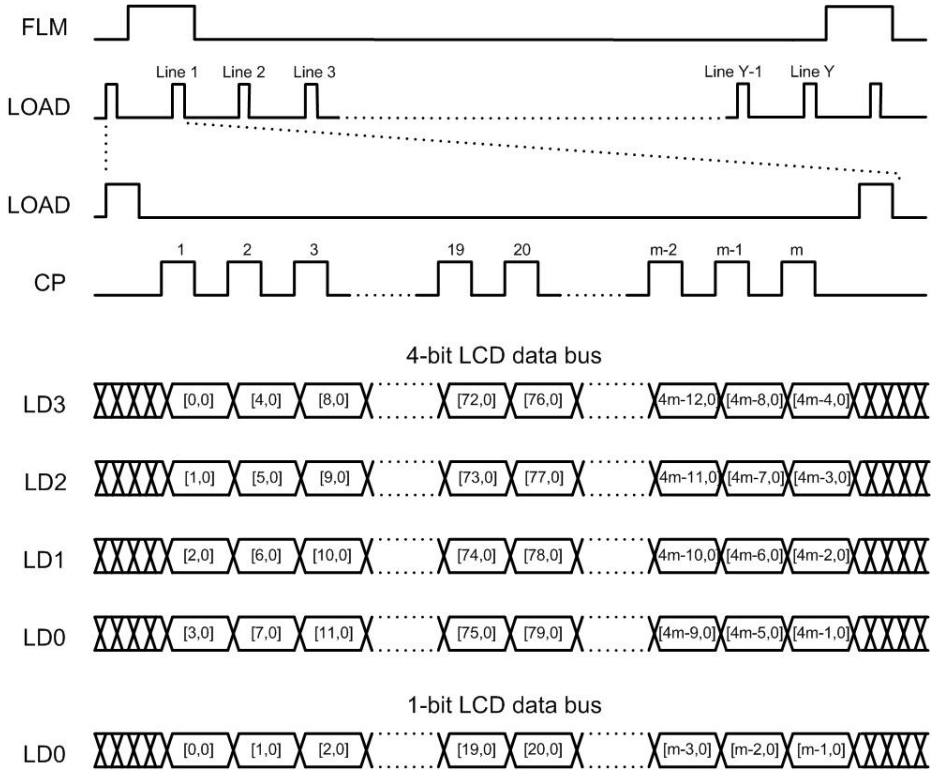



**FIGURE 11-2 LCD Interface Timing for 1-/4-Bit Data**

# 12. SERIAL PERIPHERAL INTERFACE

The ST2205U contains one serial peripheral interface (SPI) module to interface with external devices, such as Flash memory, analog-to-digital converter, and other peripherals, including another ST2205U. The SPI consists of a master- or slave-configurable interface so that connections of both master and slave devices are allowable. Five signals multiplexed with Port-C are used by SPI. With equipped $\overline{\text{DATA\_READY}}$ and $\overline{\text{SS}}$ (slave-select) control signals and transmit/receive buffers, faster data exchange with fewer software interrupts is easy to be made. Data length is widely supported from 7-bit up to 16-bit to satisfy various applications. One clock generator is provided for the synchronous communication clock SCK, which is sourced from OSCK. FIGURE 12-1 illustrates the block diagram of SPI.

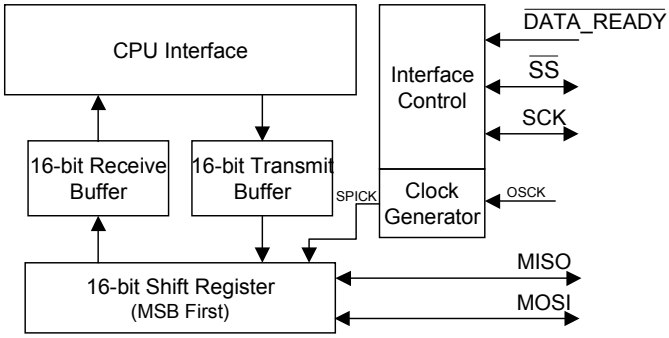



**FIGURE 12-1 SPI Block Diagram**

## 12.1 SPI Operations

The SPI contains one 16-bit shift register and two 16-bit buffers for transmission and receiving respectively. Data with variable length from 7-bit to 16-bit can be exchanged with external devices through two data lines. Data length is controlled by bit count register **BC[3:0]** (bit3~0 of SPI clock control register **SCKR)**. The current exchange will be over while the exchanged bit number reaches bit count setting.

The synchronous communication clock SCK is used to synchronize two devices and transfer data in and out of the shift register. Data is clocked by SCK with a programmable data rate, which is assigned by **SCK[2:0]** (bit6~4 of SPI clock control register **SCKR**). The SPI block is controlled by **SPIEN** (**SCTR[7]**). Setting **SPIEN** will enable SPI function and the clock divider. Then the internal states of SPI will be reset to initial values. After that, write data to **SDATAL** will initiate an exchange. While exchanging, the busy flag will be set and is reported in **SBZ** (bit 4 of SPI status register **SSR**).

A slave select signal $\overline{\text{SS}}$ (multiplexed with PC4) is used to identify individual selection of a slave SPI device. Slave devices that are not selected do not interfere with SPI bus activities. For a master SPI device, $\overline{\text{SS}}$ can be used to indicate a multiple-master bus contention which can be reported in mode fault bit **MDERR** (bit3 of SPI status register **SSR**).

# 13. UNIVERSAL ASYNCHRONOUS RECEIVER/TRANSMITTE

The ST2205U integrates one universal asynchronous receiver/transmitter (UART), which can be used to communicate with external serial devices. Serial data is transmitted and received at standard bit rates using the internal baud rate generator (BGR), which is controlled by BGR control register **BCTR**. Settings of clock output of BGR (BGRCK) can be found in section 9. FIGURE 13-1 shows the block diagram of UART. Summary of UART control registers is listed in FIGURE 13-1

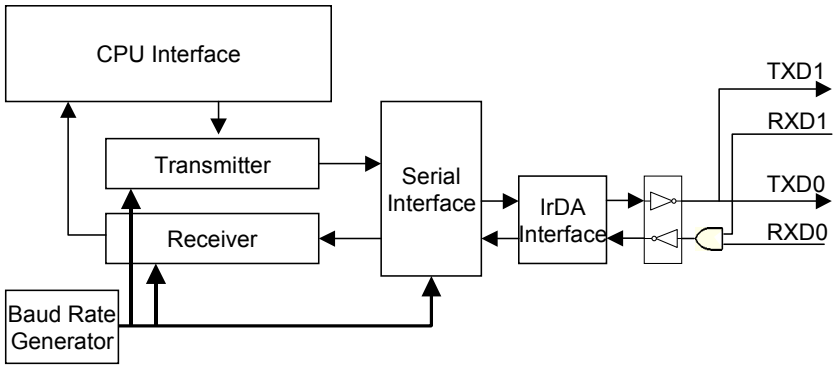



**FIGURE 13-1 UART Block Diagram**

## 13.2 UART Operations

The UART has two modes of operation, NRZ and IrDA, which represent data in different ways for serial communication protocols, RS-232 and IrDA.

### 13.2.1 NRZ mode

The non-return to zero (NRZ) mode is primarily associated with RS-232. Each character is transmitted as a frame delimited by a start bit at the beginning and a stop bit at the end. Data bits are transmitted least significant bit (LSB) first, and each bit occupies a period of time equal to 1 full bit. If parity is used, the parity bit is transmitted after the most significant bit. Data settings including data length, stop bit number and parity are controlled by bit fields in **UCTR**. FIGURE 13-2 illustrates a character “S” in NRZ mode.

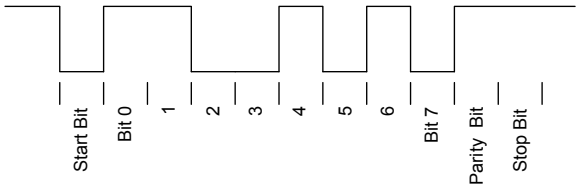



**FIGURE 13-2 NRZ ASCII “S” with Odd Parity**

### 13.2.2 IrDA mode

IrDA mode uses character frames as NRZ mode does, but, instead of driving ones and zeros for a full bit-time period, zeros are transmitted as three-sixteenth (or less) bit-time pulses (which is selected by **PW[1:0]** (**IRCTR[2:1]**), and ones remain low. The polarity of transmitted pulses and expected receive pulses can be inverted so that a direct connection can be made to external IrDA transceiver modules that use active low pulses. This is controlled by **RXINV** and **TXINV** (**IRCTR[7:6]**). IrDA mode is enabled by control bit **IREN** (**IRCTR[0]**). FIGURE 13-3 illustrates a character “S’ in IrDA

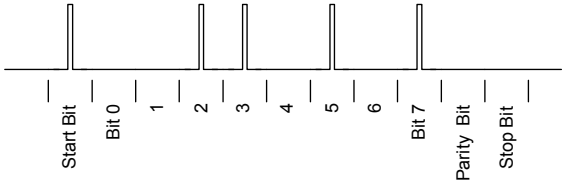



**FIGURE 13-3 IrDA ASCII “S” with Odd Parity**

# 14. UNIVERSAL SERIAL BUS (USB)

The ST2205U incorporates one PLL, a 3.3V regulator, and a USB 2.0 Full speed device engine to satisfy the strong demand of fast data transfer from market. Both HID and Mass storage classes are supported as well as the firmware libraries and the Windows 98 driver. Whole USB function is controlled by setting **USBEN** (**USBCON[7]**). After connects to a USB host port, 6 interrupts which share the same interrupt vector play the main role of USB communication. Proper routines responding to every host command should be executed to generate the right answer into the endpoint buffers to be transferred back.

Three endpoints are supported including control endpoint (EP0), bulk-in endpoint (BKI) and bulk-out endpoint (BKO). EP0 has a buffer of 8 bytes long while BKI and BKO each has a 64 bytes buffer which three range from $200 to $28F. Refer to TABLE 14-1 for the memory mapping. Write “1” to **BUFEN** (**USBIEN[7]**) to enable these buffers. There are still total 144 bytes of user RAM to use when USB buffer is hidden by clearing **BUFEN**.

Double buffer scheme is applied to both BKI and BKO buffers to increase throughput and eases real-time data transfer.

**TABLE 14-1 Summary of USB Buffers**

| Buffer | Address |
|---|---|
| BKO | $200~$23F |
| BKI | $240~$27F |
| EP0OUT | $280~$287 |
| EP0OIN | $288~$28F |

# 15. DIRECT MEMORY ACCESS (DMA)

To speed up the data transfer, DMA works efficiently without CPU involved and moves one byte of data in only two SYSCK cycles. After a write to **DCNTH**, CPU pauses and then DMA starts. Meanwhile the address and data bus is freed for DMA job. In each transfer, up to 32KB data can be moved. Only single instruction is needed for a repeated transfer. It can the one of three as below: **a.** STZ zp (3 cycles) **b.** SMB7 zp (5 cycles) **c.** RMB7 zp (5 cycles)

DMA works only on the logical address of $8000~$FFFF, combines with source and destination bank registers, all physical memory can be accessed including whole 32KB internal RAM if bit16 of bank register is set.

Note:
If bit16 of bank register is set, $8000~$807F will refer to control registers

There are two DMA channels and are selected by **DMSEL[1]**(**DCTR[1]**). After selecting a channel, source or destination registers are then chose by **DMSEL[0]**(**DCTR[0]**) to make further register access correct.

- 15-bit source pointer: **DPTR** (**DMSEL[0]**=0)
- 15-bit destination pointer: **DPTR** (**DMSEL[0]**=1)
- 11-bit source bank register: **DBKR** (**DMSEL[0]**=0)
- 11-bit destination bank register: **DBKR** (**DMSEL[0]**=1)
- 15-bit data length register: **DCNT**

There are three modes for manipulation of both pointers: **a.** Continue, **b.** Reload, and **c.** Fixed. Pointer increases one after each transfer in continue mode, and becomes $8000 after $FFFF is reached. At this time, **DBKR** also increases one to map to the next bank. Reload mode acts like continue mode except pointer and bank registers will back to their original values when each transfer stops. In case of Fixed mode, pointer keeps the same value always.

Excepting normal operation, there is one special function for each channel, and is controlled by **FUNC[1:0]**(**DMOD[5:4]**). DMA channel0 can help image data operations. AND, OR and XOR logic operations can be done between source and destination data being moved. Regarding channel1, double data transfer speed is possible while moving data from/to Nand Flash via port-F.

# 16. NAND FLASH INTERFACE

The ST2205U has a simplified Nand Flash(Flash for short in the following) interface for both And and Nand types which only 9 or 10 specific signals are needed. Combine other GPIOs, this serial interface carries commands and data between MCU and Flash memory by CPU read/write instructions or by DMA channel1.
Data moved by DMA channel1 may has ECC codes generated at the same time. When data write to Flash is performed, ECC codes will be ready at the end of transmission, then they are to be written to Flash and stored in the redundant area. In case of data read, ECC codes calculated by MCU are to be compared with those in redundant area and check if there is any bit error, even correct this error.

## 16.1 Nand Flash Interface, Port-F

Flash memory is a serial accessed memory. Typical interface signals for And and Nand types are listed in FIGURE 16-1 as well as the connection with ST2205U. If **FEN**(**FCTR[7]**) is set, port-F will be the 8-bit serial data bus and PD7/$\overline{\text{FWR}}$, PD6/$\overline{\text{FRD}}$ will play write/read signals, while other control signals are controlled by $\overline{\text{CSx}}$, INTXx and GPIOs. The And type flash interface needs only PD7 and further saves PD6 for GPIO. Since Nand Flash interface has higher priority, PD7/6 will be $\overline{\text{FWR}}$/$\overline{\text{FRD}}$ signals if **FEN**=1, regardless of settings of **PFD**. Port-F works the same way, I/O directions will not be controlled by PCF but by read/write access of data when Flash interface enabled. It is floating when not being accessed, output when write to port-F and is input when read from port-F.

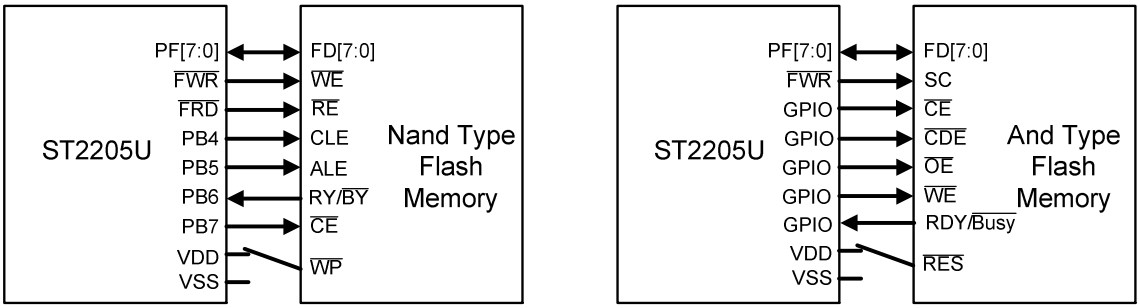



**FIGURE 16-1 Connecting Nand and And Flash Memories**

## 16.2 Error Correction Code (ECC)

ECC code consists of 3 bytes per 256 bytes of data. The XORed result of new and old ECC codes shows if there is a bit error between two 256 bytes of data, even the location of the error bit. Two sets of ECC codes, **ECC0** and **ECC1**, are supported and are selected by **ECCSEL**. So results of up to 512 bytes can be processed and stored. Three bytes of each can be accessed at three registers **ECCL/M/H**.

There are two ways to trigger ECC calculation. First is execute read/write to PF when **ECCEN**=1 and **PFECC**=1. Second is moving Flash data via DMA channel1. ECC of first 256 bytes will be calculated first in **ECC0**, and then changes to **ECC1** automatically for those after 256. The calculation stops after 512 bytes are reached even there are still more being moved.

Before Flash data transfer, clear ECC codes and the counter by writing “1” to **ECCCLR**. After write of 512 bytes is performed, control **ECCSEL** and get the results from **ECC0** and **ECC1**. In case of read transfer, after reading 512 bytes, retrieve two 3-byte ECC codes in the redundant area and write them into **ECC0/1** respectively. Each write to **ECCL/M/H** will make a XOR operation between the original data and the byte written into. After **ECCH** is wrote a byte, ECC checking starts. The result will be reported at FSR[1:0] in one SYSCK cycle. Meanwhile **ECCL/M/H** also report the error bit position if there is one.

# 17. POWER DOWN MODES

ST2205U 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.

## 17.1 SWAI-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
```

## 17.2 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 occur when waking from WAI-1. Sample program is shown as following:

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

## 17.3 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 occur at the same time.

**FIGURE 17-1 Status Under Power Down Modes**

**SYSCK source is OSC:**

| Mode | Timer0,1 | SYSCK | LCD | OSC | OSCX | Base Timer | RAM | REG. | I/O | Wake-up condition |
|---|---|---|---|---|---|---|---|---|---|---|
| **WAI-0** | Retain | | | | | | | | | Reset, Any interrupt |
| **WAI-1** | Stop | Stop | Stop | Stop | Retain | | | | | Reset, Any interrupt |
| **STP** | Stop | Stop | Stop | Stop | Retain | | | | | Reset |

**SYSCK source is OSCX:**

| Mode | Timer0,1 | SYSCK | OSC | OSCX | Base Timer | RAM | REG. | I/O | LCD | Wake-up condition |
|---|---|---|---|---|---|---|---|---|---|---|
| **WAI-0** | Retain | | | | | | | | Wrong Frame | Reset, Any interrupt |
| **WAI-1** | Stop | Stop | Retain | | | | | | Stop | Reset, Any interrupt |
| **STP** | Stop | Stop | Retain | | | | | | Stop | Reset |

# 18. WATCHDOG TIMER

The watchdog timer (WDT) is an added check that a program is running and sequencing properly. When the application software is running, it is responsible for keeping the 2- or 8-second watchdog timer from timing out. If the watchdog timer times out, it is an indication that the software is no longer being executed in the intended sequence. At this time the watchdog timer generates a reset signal to the system.

## 18.1 WDT Operations

The WDT is enabled by setting the WDT enable flag **WDTEN** (**MISC[3]**). Two time settings, 2 and 8 seconds, are selectable with selection bit **WDTPS** (**MISC[2]**).WDT is clocked by the 2Hz clock from the base timer and therefore has 0.5-second resolution. It is recommended that the watchdog timer be periodically cleared by software once it is enabled. Otherwise, software reset will be generated when the timer reached a binary value of 4 or 16.

Note:The WDT can be reset by writing any value to **MISC** register.

After a system reset, **WDTEN** is cleared. Then the WDT returns to be idle.

**TABLE 18-1 System Miscellaneous Register (MISC)**

| Address | Name | R/W | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Default |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $038 | MISC | R | Test | - | - | - | WDTEN | WDTPS | TEST | TEST | - - - - 1100 |
| | | W | Reset WDT | | | | | | | | |

**WDTPS:** WDT period selection bit
0: Timer period is 72ms
1: Timer period is 2s

**WDTEN:** WDT enable bit
(W) 0: Disable WDT
(W) 1: Enable WDT
(R) 0: WDT reset did not occur
(R) 1: WDT reset occurred

Bit 7: **TEST :** These two bits should be both zero in normal operation
Bit 1~0: **TEST :** These two bits should be both zero in normal operation

# 19. REAL TIME CLOCK

# 20. LOW VOLTAGE DETECTOR (LVD)

ST2205U has a built-in low voltage detector for power management. Two voltage signals can be selected by the control bit **LVDS** (**LVCTR[1]**). First is the power applied to ST2205U and has four detection levels can be selected by **LVD[1:0]**(**LVCTR[3:2**]). Second is the signal applied to input pin VIN, and has four detection levels can be selected, too. When **LVDEN** (**LVCTR[0]**) is set, LVD is enabled and the detection result will be outputted at the same bit after 30us. Using read instruction twice can get this result: first read will enable initial stableness control. Second read equal '0' represents 'low voltage'. Once LVD is enabled, it keeps on consuming power. So it is important to write “0” to **LVDEN** and disable the detector after detection is completed. In FIGURE 20-1 shows an application circuit for detecting battery voltage applied to VIN(**LVDS=1**). Note that the DC current of two external resistors can be cut off by setting PC0 to open. Also add one capacitor to VIN to minimize noise and narrow the low voltage detection range. In FIGURE 20-2 shows another application circuit. It will consume a constant current but save the delay time for VIN to be stable. If **LVDS=0** and detecting VDD, please leave VIN pin open.

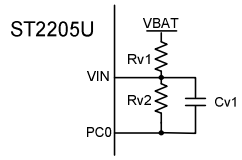



**FIGURE 20-1 Application of LVD (1)**

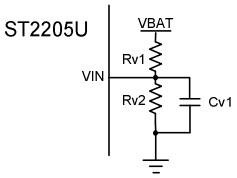



**FIGURE 20-2 Application of LVD (2)**

# 21. LOW VOLTAGE RESET (LVR)

Power bouncing during power on is a major problem when designing a reliable system. The ST2205U equips Low Voltage Reset function to keep whole system in reset status when power is not stable. Once low voltage status is detected, an active low pulse will be output from pin $\overline{\text{RESET}}$ to perform this protection. After the power backs to normal, will output high and the system may recover its original states and keeps working correctly.

The LVR circuit always works and it consumes very few current.

# 22. ELECTRICAL CHARACTERISTICS

## 22.1 Absolute Maximum Rations

DC Supply Voltage ---------- -0.3V to +4.5V
Operating Ambient Temperature ---------- -10°C to +60°C
Storage Temperature ---------- -55°C to +125°C

***Note:** 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.

## 22.2 DC Electrical Characteristics

Standard operation conditions: VCC = 3.0V, GND = 0V, $T_A$ = 25°C, OSC = 8MHz (CPU clock=4MHz), unless otherwise specified

| Parameter | Symbol | Min. | Typ. | Max. | Unit | Condition |
|---|---|---|---|---|---|---|
| Operating Voltage | VCC | 2.4 | | 3.6 | V | |
| Operating Frequency(OSC) | $F_1$ | | | 12 | MHz | VCC = 2.4V ~ 3.6V (CPU clock=6MHz) |
| Operating Frequency(OSC) | $F_2$ | | | 16 | MHz | VCC = 3.0 ~ 3.6V (CPU clock=8MHz) |
| Operating Current | $I_{OP}$ | | 6.1 | | mA | All I/O port are input and pull-up, execute NOP instruction, LCDC on |
| Standby Current | $I_{SB0}$ | | 1200 | | μA | All I/O port are input and pull-up, OSCX on, LCDC on (WAIT0 mode) SEG=240, CP=SYS, LFRA=30 |
| Standby Current | $I_{SB1}$ | | 28 | 38 | μA | All I/O port are input and pull-up, OSCX on, heavy load, LCDC off (WAIT1 mode) LVR=2.8V |
| | | | 16 | 21 | μA | All I/O port are input and pull-up, OSCX on, heavy load, LCDC off (WAIT1 mode) LVR=2.1V |
| Standby Current | $I_{SB2}$ | | 19 | 25 | μA | All I/O port are input and pull-up, OSCX on, normal load, LCDC off (WAIT1 mode) LVR=2.8V |
| | | | 7 | 10 | μA | All I/O port are input and pull-up, OSCX on, normal load, LCDC off (WAIT1 mode) LVR=2.1V |
| Standby Current | $I_{SB3}$ | | 15 | 20 | μA | All I/O port are input and pull-up, OSCX off, LCDC off (WAIT1 mode) LVR=2.8V |
| | | | 3 | 5 | μA | All I/O port are input and pull-up, OSCX off, LCDC off (WAIT1 mode) LVR=2.1V |
| Input High Voltage | $V_{IH}$ | 0.7Vcc | | | V | Port-A/B/C/D/E/L |
| Input Low Voltage | $V_{IL}$ | | | 0.3Vcc | V | Port-A/B/C/D/E/L |
| Pull-up resistance | $R_{IH}$ | | 90 | | KΩ | Port-A/B/C/D/E/L (input Voltage=0.7VCC) |
| Output high voltage | $V_{OH1}$ | 0.7Vcc | | | V | Port-A/B/C/D/L ($I_{OH}$ =-4.5mA) |
| Output low voltage | $V_{OL1}$ | | | 0.3Vcc | V | Port-A/B/C/D/E/L ($I_{OL}$ =6.5mA) |
| Output high voltage | $V_{OH2}$ | 0.7Vcc | | | V | PSG/DAC, $I_{OH}$ = -40mA. |
| Output low voltage | $V_{OL2}$ | | | 0.3Vcc | V | PSG/DAC, $I_{OL}$ = 40mA. |

| | | | | | | |
|---|---|---|---|---|---|---|
| Current DAC ouput | Iout | | 3 | | mA | 4095th step |
| Low Voltage Reset level | $V_{LVR1}$ | 1.8 | 1.9 | 2 | V | Pin option LVRSEL=0 |
| Low Voltage Reset level | $V_{LVR1}$ | 2.55 | 2.65 | 2.75 | V | Pin option LVRSEL=1 |
| Low Voltage Detect current | $I_{LVR}$ | | 38 | 60 | μA | Total LVD circuit current consumption |
| Low Voltage Detect level | $V_{LVR1}$ | 2.2 | 2.4 | 2.6 | | Internal mode LVDS[1:0](**LVCTR[3:2]**)=00 |
| Low Voltage Detect level | $V_{LVR2}$ | 2.4 | 2.6 | 2.8 | | Internal mode LVDS[1:0](**LVCTR[3:2]**)=01 |
| Low Voltage Detect level | $V_{LVR3}$ | 2.6 | 2.8 | 3.0 | | Internal mode LVDS[1:0](**LVCTR[3:2]**)=10 |
| Low Voltage Detect level | $V_{LVR4}$ | 2.8 | 3.0 | 3.2 | | Internal mode LVDS[1:0](**LVCTR[3:2]**)=11 |
| Low Voltage Detect level | $V_{LVR5}$ | 1.1 | 1.2 | 1.3 | | External mode LVDS[1:0](**LVCTR[3:2]**)=00 |
| Low Voltage Detect level | $V_{LVR6}$ | 1.2 | 1.3 | 1.4 | | External mode LVDS[1:0](**LVCTR[3:2]**)=01 |
| Low Voltage Detect level | $V_{LVR7}$ | 1.3 | 1.4 | 1.5 | | External mode LVDS[1:0](**LVCTR[3:2]**)=10 |
| Low Voltage Detect level | $V_{LVR8}$ | 1.4 | 1.5 | 1.6 | | External mode LVDS[1:0](**LVCTR[3:2]**)=11 |
| Warm up time | $T_{WM1}$ | | 0.3 | | S | 32768 Crystal Heavy mode. |
| | | | 3 | | S | 32768 Crystal Normal mode. |
| Warm up time | $T_{WM2}$ | | 8 | | mS | Main frequency crystal 8192 warm-up cycle |
| | | | 12 | | mS | Main frequency crystal 32768 warm-up cycle |
| Warm up time | $T_{WM3}$ | | 20 | | uS | Main frequency R-OSC 16 warm-up cycle |
| | | | 80 | | uS | Main frequency R-OSC 256 warm-up cycle |

# AC Electrical Characteristics

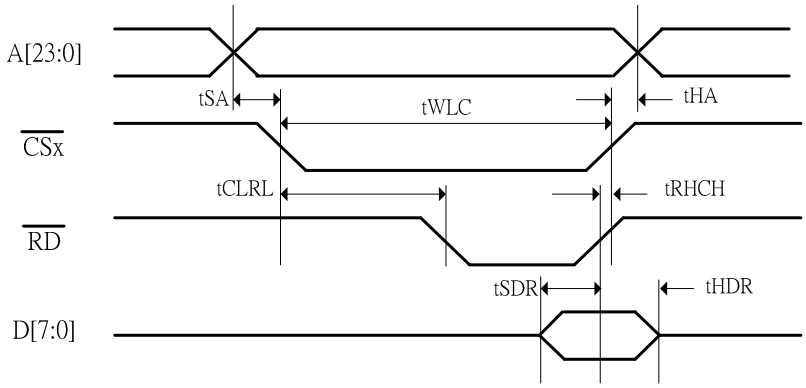



**FIGURE 22-1 External Read Timing Diagram**

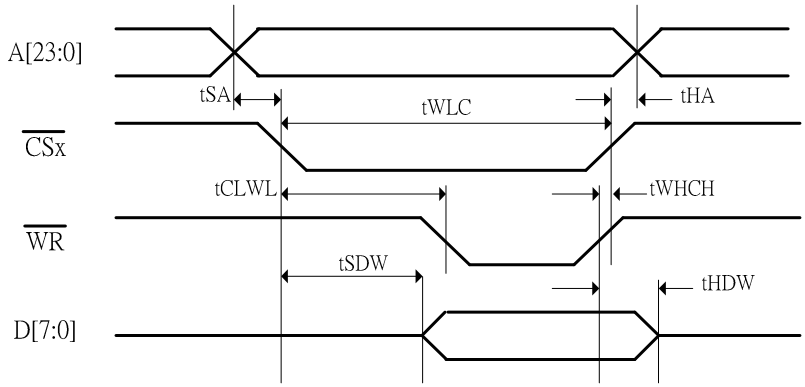


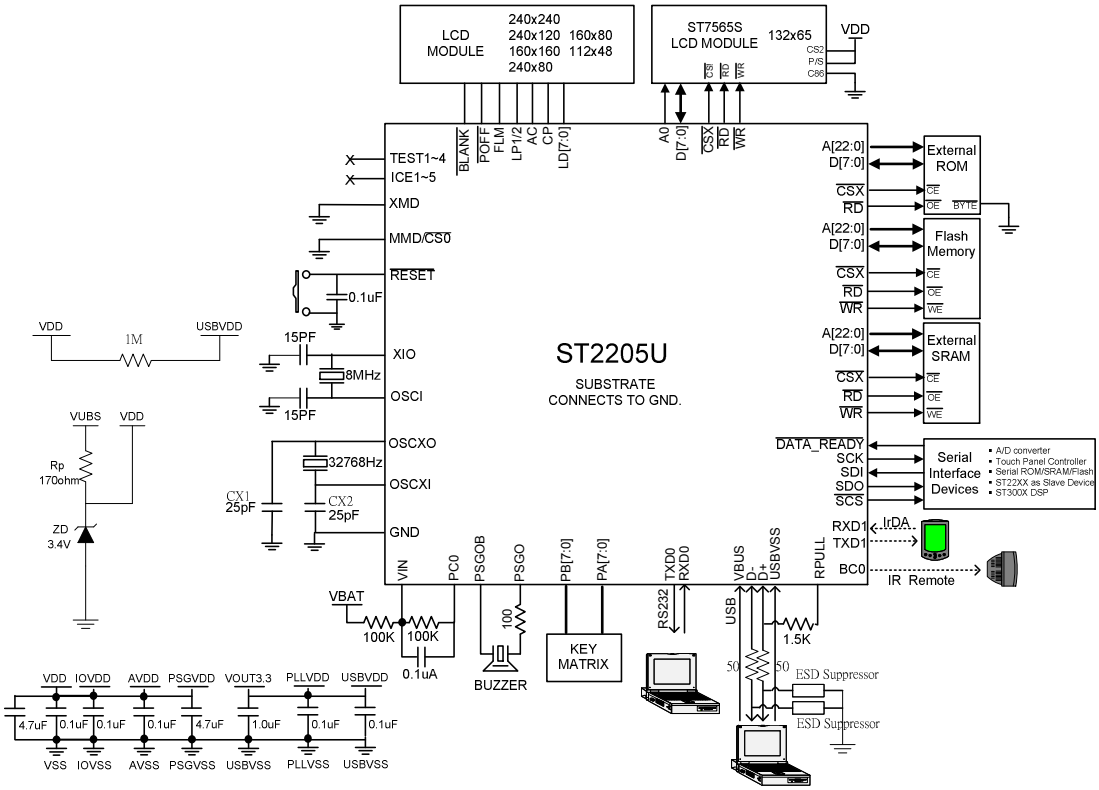

**FIGURE 22-2 External Write Timing Diagram**

**TABLE 22-1 Timing parameters for FIGURE 22-1 and FIGURE 22-2**

Standard operation conditions: VCC = 3.0V, GND = 0V, $T_A$ = 25°C

| Symbol | Characteristic | Rating | | | Unit |
|---|---|---|---|---|---|
| | | Min. | Typ. | Max. | |
| tSA | Address setup time | — | — | 10 | ns |
| tHA | Address hold time | 0 | — | — | ns |
| tWLC | CS "L" pulse width | 166 | — | — | ns |
| tCLWL | CS asserted to $\overline{WR}$ asserted | — | 1/2 tWLC | — | ns |
| tWHCH | CS negated after $\overline{WR}$ is negated | 10 | — | — | ns |
| tSDW | CS asserted to data-out is valid | — | 1/2 tWLC | — | ns |
| tHDW | Data-out hold time after $\overline{WR}$ is negated | 20 | — | | ns |
| tCLRL | CS asserted to $\overline{RD}$ asserted | — | 1/2 tWLC | — | ns |
| tRHCH | CS negated after $\overline{RD}$ is negated | 10 | — | — | ns |
| tSDR | Data-in valid before $\overline{RD}$ is negated | 30 | — | — | ns |
| tHDR | Data-in hold time after $\overline{RD}$ is negated | 10 | — | — | ns |
| tR | Signal rise time | — | 20 | — | ns |
| tF | Signal fall time | — | 10 | — | ns |

# 23. APPLICATION CIRCUITS




Note: 1. Keep the trace between oscillation resistor and the PCB pad as close as possible for a more stable clock.
2. The OSCX can still work if remove CX1 and increase CX2 to 47pF.
3. The capacitors that connect to VOUT3.3, PLLVDD, USBVDD must as close as possible to reduce noises.
4. Resister Rp and zenor diode ZD provide a solution for using host power when USB cable plugged in.

# 24. OTP ROM PROGRAMMING INTERFACE

## 24.1 INTERFACE DESCRIPTION

In order to program OTP ROM, several pins have to be reserved on the PCB which is bounding with ST2205U. These totals are 34 pins that include following list TABLE 24-1. It just be used to connect writer to program OTP ROM.

**TABLE 24-1 PIN ASSIGNMENT OF INTERFACE**

| Pad Name | | Pin Type | Description |
|---|---|---|---|
| VPP | | Power | High Voltage Power Supply<br>1) OTP Program, Program Verify, Test modes. 9V<br>2) OTP Read: VPP need connect to VDD |
| VDD | | Power | |
| VSS | | Power | |
| RESETB | | Input | |
| TEST2 | | Input | |
| PL6 | | Input | |
| PL5 | | Input | |
| PL4 | | Input | |
| PL3 | | Input | |
| PL[2:0] | | Input | |
| Data[7:0] | | I/O | |
| | | I/O | |
| Address[13:0] | | Input | |

# 25. REVISIONS

| REVISION | DESCRIPTION | PAGE | DATE |
|---|---|---|---|
| 1.0 | Add waiting cycle feature. | 1 | 2005/9/26 |
| | Modify USB 1.1 to USB 2.0 full speek | 1,2,4,20 | |
| 0.1 | First release | | 2005/3/16 |