nuvoTon

# 8-bit Microcontroller

***Table of Contents-***


1. GENERAL DESCRIPTION ..... 5
2. FEATURES ..... 6
3. PARTS INFORMATION LIST ..... 7
   3.1 Lead Free (RoHS) Parts information list ..... 7
4. PIN CONFIGURATION ..... 8
5. PIN DESCRIPTION ..... 10
   5.1 Port 4 ..... 12
6. MEMORY ORGANIZATION ..... 13
   6.1 Program Memory (on-chip Flash) ..... 13
   6.2 Data Memory ..... 13
   6.3 Auxiliary SRAM ..... 14
   6.4 NVM Data Flash ..... 14
      6.4.1 Operation ..... 19
7. SPECIAL FUNCTION REGISTERS ..... 21
8. INSTRUCTION SET ..... 74
   8.1 Instruction Timing ..... 82
      8.1.1 External Data Memory Access Timing ..... 84
9. POWER MANAGEMENT ..... 87
   9.1 Idle Mode ..... 87
   9.2 Power Down Mode ..... 87
10. RESET CONDITIONS ..... 89
   10.1 Sources of reset ..... 89
      10.1.1 External Reset ..... 89
      10.1.2 Power-On Reset (POR) ..... 89
      10.1.3 Watchdog Timer Reset ..... 89
   10.2 Reset State ..... 90
11. INTERRUPTS ..... 91
   11.1 Interrupt Sources ..... 91
   11.2 Priority Level Structure ..... 91
      11.2.1 Response Time ..... 95
12. PROGRAMMABLE TIMERS/COUNTERS ..... 96
   12.1 Timer/Counters 0 & 1 ..... 96
      12.1.1 Time-Base Selection ..... 96
      12.1.2 Mode 0 ..... 96
      12.1.3 Mode 1 ..... 97
      12.1.4 Mode 2 ..... 97
      12.1.5 Mode 3 ..... 98

12.2 Timer/Counter 2 ..... 98
12.2.1 Capture Mode ..... 99
12.2.2 Auto-reload Mode, Counting up ..... 99
12.2.3 Auto-reload Mode, Counting Up/Down ..... 100
12.2.4 Baud Rate Generator Mode ..... 101
13. WATCHDOG TIMER ..... 102
14. PULSE-WIDTH-MODULATED (PWM) OUTPUTS ..... 105
14.1 PWM Features ..... 105
14.2 PWM Control Registers ..... 106
14.3 PWM Pin Structures ..... 108
14.4 Complementary PWM with Dead-time and Override functions ..... 111
14.5 Dead-Time Insertion ..... 111
14.6 PWM Output Override ..... 113
14.7 Edge Aligned PWM (up-counter) ..... 116
14.8 Center Aligned PWM (up/down counter) ..... 118
14.9 Single Shot (Up-Counter) ..... 121
14.10 Smart Fault Detector ..... 124
14.11 PWM Power-down/Wakeup Procedures ..... 126
15. MOTION FEEDBACK MODULE ..... 128
15.1 Input Capture Module (IC) ..... 128
15.1.1 Compare Mode ..... 135
15.1.2 Reload Mode ..... 136
15.2 Quadrature Encoder Interface (QEI) ..... 136
15.2.1 Free-counting mode ..... 138
15.2.2 Compare-counting mode ..... 138
15.2.3 X2/X4 Counting modes ..... 138
15.2.4 Direction of Count ..... 138
15.2.5 Up-Counting ..... 140
15.2.6 Down-Counting ..... 140
16. SERIAL PORT ..... 141
16.1 Mode 0 ..... 141
16.2 Mode 1 ..... 142
16.3 Mode 2 ..... 143
16.4 Mode 3 ..... 144
16.5 Framing Error Detection ..... 144
16.6 Multiprocessor Communications ..... 146
17. I2C SERIAL PORTS ..... 147
17.1 SIO Port ..... 147
17.2 The I2C Control Registers ..... 147
17.2.1 Slave Address Registers, I2ADDR ..... 148
17.2.2 Data Register, I2DAT ..... 148
17.2.3 Control Register, I2CON ..... 148
17.2.4 Status Register, I2STATUS ..... 149

17.2.5 I2C Clock Baud Rate Control, I2CLK ..... 149
17.2.6 I2C Time-out Counter, I2Timer ..... 149
17.2.7 I2C Maskable Slave Address ..... 149
17.3 Modes of Operation ..... 150
17.3.1 Master Transmitter Mode ..... 150
17.3.2 Master Receiver Mode ..... 150
17.3.3 Slave Receiver Mode ..... 150
17.3.4 Slave Transmitter Mode ..... 151
17.4 Data Transfer Flow in Five Operating Modes ..... 151
17.4.1 Master/Transmitter Mode ..... 152
17.4.2 Figure 17-5: Master Transmitter ModeMaster/Receiver Mode ..... 153
17.4.3 Slave/Transmitter Mode ..... 154
17.4.4 Slave/Receiver Mode ..... 155
17.4.5 GC Mode ..... 156
18. SERIAL PERIPHERAL INTERFACE (SPI) ..... 157
18.1 General descriptions ..... 157
18.2 Block descriptions ..... 157
18.3 Functional descriptions ..... 159
18.3.1 Master mode ..... 159
18.3.2 Slave Mode ..... 162
18.3.3 Slave select ..... 166
18.3.4 /SS output ..... 166
18.3.5 SPI I/O pins mode ..... 167
18.3.6 Programmable serial clock's phase and polarity ..... 168
18.3.7 Receive double buffered data register ..... 169
18.3.8 LSB first enable ..... 170
18.3.9 Write Collision detection ..... 170
18.3.10 Transfer complete interrupt ..... 170
18.3.11 Mode Fault ..... 171
19. ANALOG-TO-DIGITAL CONVERTER ..... 173
19.1 Operation of ADC ..... 173
19.2 ADC Resolution and Analog Supply ..... 174
20. TIMED ACCESS PROTECTION ..... 175
21. PORT 4 STRUCTURE ..... 177
22. IN-SYSTEM PROGRAMMING ..... 180
22.1 The Loader Program Locates at LDFlash Memory ..... 180
22.2 The Loader Program Locates at APFlash Memory ..... 180
23. OPTION BITS ..... 181
23.1 Config0 ..... 181
23.2 Config1 ..... 182
24. ELECTRICAL CHARACTERISTICS ..... 183
24.1 Absolute Maximum Ratings ..... 183
24.2 DC Characteristics ..... 183

24.3 AC Characteristics ..... 186
24.3.1 External Clock Characteristics ..... 186
24.3.2 AC Specification ..... 186
24.3.3 MOVX Characteristics Using Stretch Memory Cycle ..... 187
24.4 The ADC Converter DC ELECTRICAL CHARACTERISTICS ..... 189
24.5 I2C Bus Timing Characteristics ..... 189
24.6 Program Memory Read Cycle ..... 190
24.7 Data Memory Read Cycle ..... 191
24.8 Data Memory Write Cycle ..... 191
25. TYPICAL APPLICATION CIRCUITS ..... 192
25.1 Expanded External Program Memory and Crystal ..... 192
25.2 Expanded External Data Memory and Oscillator ..... 192
26. PACKAGE DIMENSION ..... 193
26.1 44L PLCC ..... 193
26.2 48L LQFP (7x7x1.4mm footprint 2.0mm) ..... 194
27. APPLICATION NOTE ..... 195
28. REVISION HISTORY ..... 201

## 1. GENERAL DESCRIPTION

The W79E22X SERIES is a fast, 8051/52-compatible microcontroller with a redesigned processor core that eliminates wasted clock and memory cycles. Typically, the W79E22X SERIES executes instructions 1.5 to 3 times faster than that of the traditional 8051/52, depending on the type of instruction, and the overall performance is about 2.5 times better at the same crystal speed. As a result, with the fully-static CMOS design, the W79E22X SERIES can accomplish the same throughput with a lower clock speed, reducing power consumption.

The W79E22X SERIES provides 256 bytes of on-chip RAM; 1/2/2-KB of NVM Data Flash EPROM; 1/2/2-KB of auxiliary RAM; four 8-bit, bi-directional and bit-addressable I/O ports; an additional 4-bit port P4 and 2-bit port P5; three 16-bit timer/counters; Motion Feedback Module support; 2 UART serial ports; 1 channels of I2C with master/slave capability; 1 channels of Serial Peripheral Interface (SPI), 8 channels of 12 bit PWM with configurable dead time and 8 channels of 10-bit ADC. These peripherals are all supported by 20 interrupt sources with 4 levels of priority.

The W79E22X SERIES also contains a 16/32/64-KB Flash EPROM whose contents may be updated in-system by a loader program stored in an auxiliary, 4-KB Flash EPROM. Once the contents are confirmed, it can be protected for security.

## 2. FEATURES

- Fully-static-design 8-bit 4T-8051 CMOS microcontroller up to 40MHz.
- 16/32/64-KB of in-system-programmable Flash EPROM (AP Flash EPROM).
- 4-KB of Auxiliary Flash EPROM for the loader program (LD Flash EPROM). User can optionally reboot from LD Flash EPROM by pull low at either P4.3 or P3.6 and P3.7, at external reset.
- 1/2/2-KB auxiliary RAM, software-selectable, accessed by MOVX instruction.
- 1/2/2-KB of NVM Data Flash EPROM for customer data storage used.
- 256 bytes of scratch-pad RAM.
- Four 8-bit bi-directional ports; Port 0 has internal pull-up resisters enabled by software.
- Multipurpose I/O port4 (4 bits for 48L LQFP; 2 bits for 44L PLCC) with Chips select (CS) and boot function.
- Two bits bi-directional port5.
- Three 16-bit timers.
- One 16-bit Timer 3 for Motion Feed-Back Module.
- Motion Feedback Module - QEI decoder and 3 Inputs Capture.
- Eight channels of 12-bit PWM:-
  - Complementary paired output with programmable dead-time insertion.
  - Three modes: Edge aligned, center aligned and single shot.
  - Output override control for BLDC motor application.
- 10-bit ADC with 8-channel inputs.
- Two enhanced full-duplex UART with framing-error detection and automatic address recognition.
- One channel of I2C with master/slave capability.
- One channel of SPI with master/slave capability.
- Software programmable access cycle to external RAM/peripherals.
- 20 interrupt sources with four levels of priority.
- Software reset function.
- Optional H/L state of ALE/PSEN during power down mode.
- Built-in power management.
- Code protection.
- Package:
  - Lead Free (RoHS) PLCC 44: W79E225APG
  - Lead Free (RoHS) LQFP 48: W79E225ALG
  - Lead Free (RoHS) PLCC 44: W79E226APG
  - Lead Free (RoHS) LQFP 48: W79E226ALG
  - Lead Free (RoHS) PLCC 44: W79E227APG
  - Lead Free (RoHS) LQFP 48: W79E227ALG

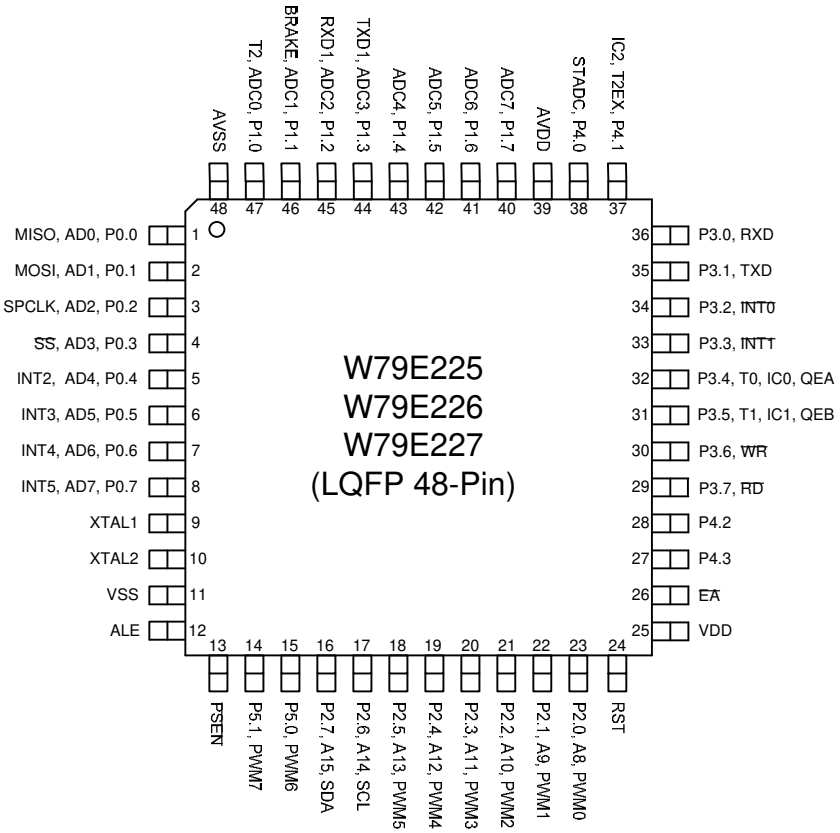
# 3. PARTS INFORMATION LIST

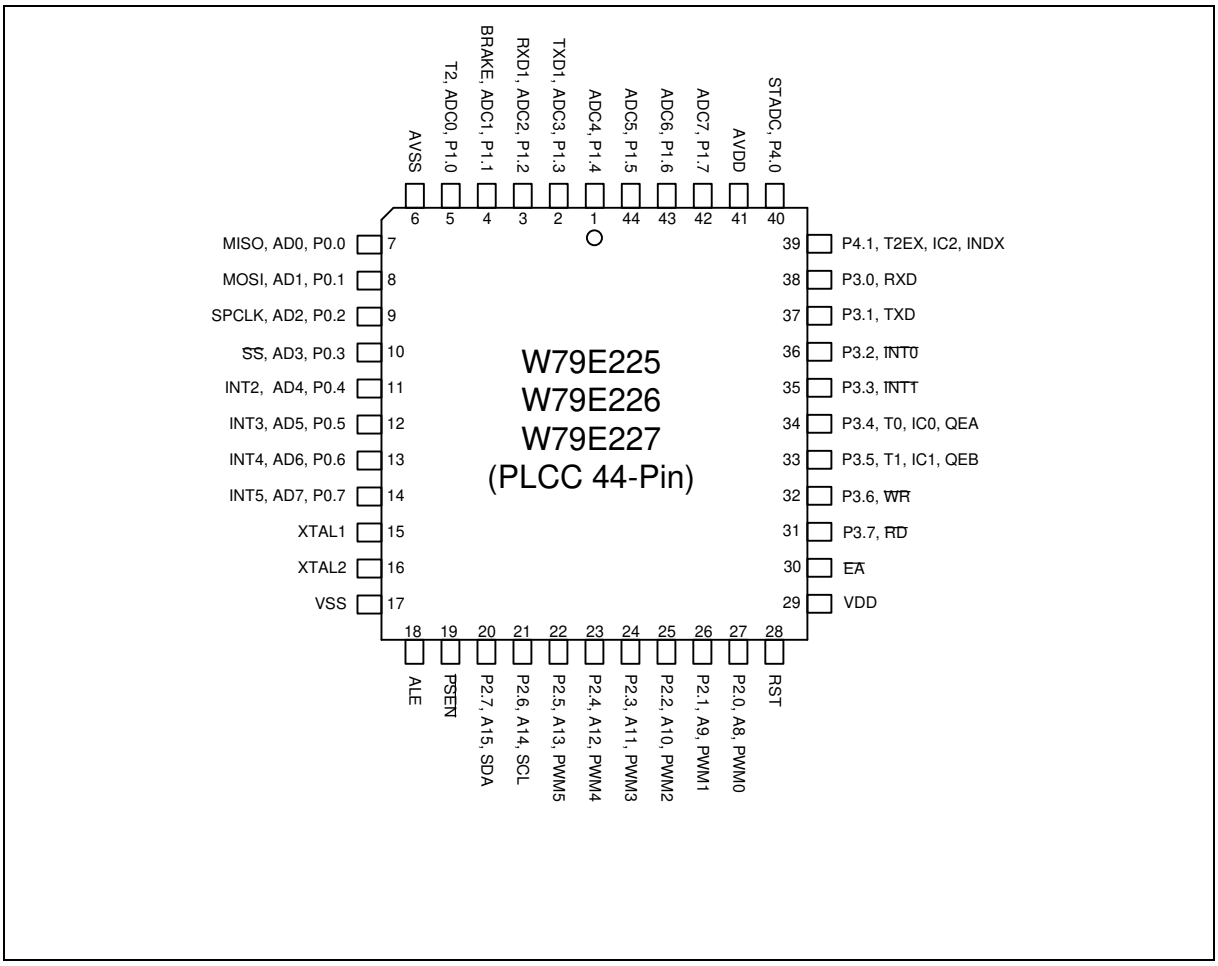
## 3.1 Lead Free (RoHS) Parts information list

| PART NO. | EPROM FLASH SIZE | RAM | OPERATING FREQUENCY | OPERATING VOLTAGE | NVM FLASH EPROM | PACKAGE | REMARK |
|---|---|---|---|---|---|---|---|
| W79E225APG | 16KB | 256B + 1KB | up to 40MHz<br>up to 20MHz | 4.5V ~ 5.5V<br>2.7V[1] ~ 5.5V | 1KB | PLCC-44 Pin | Internal memory |
| | | | up to 24MHz | 4.5V ~ 5.5V | | | External memory |
| W79E225ALG | 16KB | 256B+ 1KB | up to 40MHz<br>up to 20MHz | 4.5V ~ 5.5V<br>2.7V[1] ~ 5.5V | 1KB | LQFP-48 Pin | Internal memory |
| | | | up to 24MHz | 4.5V ~ 5.5V | | | External memory |
| W79E226APG | 32KB | 256B + 2KB | up to 40MHz<br>up to 20MHz | 4.5V ~ 5.5V<br>2.7V[1] ~ 5.5V | 2KB | PLCC-44 Pin | Internal memory |
| | | | up to 24MHz | 4.5V ~ 5.5V | | | External memory |
| W79E226ALG | 32KB | 256B + 2KB | up to 40MHz<br>up to 20MHz | 4.5V ~ 5.5V<br>2.7V[1] ~ 5.5V | 2KB | LQFP-48 Pin | Internal memory |
| | | | up to 24MHz | 4.5V ~ 5.5V | | | External memory |
| W79E227APG | 64KB | 256B + 2KB | up to 40MHz<br>up to 20MHz | 4.5V ~ 5.5V<br>2.7V[1] ~ 5.5V | 2KB | PLCC-44 Pin | Internal memory |
| | | | up to 24MHz | 4.5V ~ 5.5V | | | External memory |
| W79E227ALG | 64KB | 256B + 2KB | up to 40MHz<br>up to 20MHz | 4.5V ~ 5.5V<br>2.7V[1] ~ 5.5V | 2KB | LQFP-48 Pin | Internal memory |
| | | | up to 24MHz | 4.5V ~ 5.5V | | | External memory |

**Note:** 1. Minimum of 3.0V operating voltage for NVM program and erase operations.

## 4. PIN CONFIGURATION




W79E225
W79E226
W79E227
(PLCC 44-Pin)
1 AVSS
2 T2, ADC0, P1.0
3 BRAKE, ADC1, P1.1
4 RXD1, ADC2, P1.2
5 TXD1, ADC3, P1.3
6 ADC4, P1.4
7 ADC5, P1.5
8 ADC6, P1.6
9 ADC7, P1.7
10 AVDD
11 STADC, P4.0
12 MISO, AD0, P0.0
13 MOSI, AD1, P0.1
14 SPCLK, AD2, P0.2
15 SS, AD3, P0.3
16 INT2, AD4, P0.4
17 INT3, AD5, P0.5
18 INT4, AD6, P0.6
19 INT5, AD7, P0.7
20 XTAL1
21 XTAL2
22 VSS
23 ALE
24 PSEN
25 P2.7, A15, SDA
26 P2.6, A14, SCL
27 P2.5, A13, PWM5
28 P2.4, A12, PWM4
29 P2.3, A11, PWM3
30 P2.2, A10, PWM2
31 P2.1, A9, PWM1
32 P2.0, A8, PWM0
33 RST
34 P4.1, T2EX, IC2, INDX
35 P3.0, RXD
36 P3.1, TXD
37 P3.2, INT0
38 P3.3, INT1
39 P3.4, T0, IC0, QEA
40 P3.5, T1, IC1, QEB
41 P3.6, WR
42 P3.7, RD
43 EA
44 VDD

## 5. PIN DESCRIPTION

| SYMBOL | TYPE | INITIAL STATE | DESCRIPTIONS |
|---|---|---|---|
| $\overline{EA}$ | I | - | **EXTERNAL ACCESS ENABLE:** This pin forces the processor to execute from external ROM. The ROM address and data are not presented on the bus if the $\overline{EA}$ pin is high.<br>**Note:** This pin has no internal pull-up or pull-down. The pin needs externally pull-up to execute from internal APROM. For executing from external APROM, the pin needs externally pull-down. The pin state is internally latched during all reset. User needs to take note that changes to /EA pin state after reset will not be effective. |
| $\overline{PSEN}$ | O H | High | **PROGRAM STORE ENABLE:** $\overline{PSEN}$ enables the external ROM data in the Port 0 address/data bus. When internal ROM access is performed, $\overline{PSEN}$ strobe signal will not be output from this pin. |
| ALE | O H | High | **ADDRESS LATCH ENABLE:** ALE enables the address latch that separates the address from the data on Port 0. |
| RST | I L | - | **RESET:** Set this pin high for two machine cycles while the oscillator is running to reset the device. |
| XTAL1 | I | - | **CRYSTAL 1:** Crystal oscillator input or external clock input. |
| XTAL2 | O | - | **CRYSTAL 2:** Crystal oscillator output. |
| $V_{SS}$ | I | - | **GROUND**: Ground potential. |
| $V_{DD}$ | I | - | **POWER SUPPLY:** Supply voltage for operation. |
| AVDD | I | - | Analog power supply. |
| AVSS | I | - | Analog ground potential. |
| P0.0–P0.7 | I/O<br>D S H | High-Z | **PORT 0**: Port 0 is an open-drain bi-directional I/O port. This port also provides a multiplexed low byte address/data bus during accesses to external memory. There is an embedded weakly pull-up resistor on each port 0 pin which can be enabled or disabled by setting or clearing of PUP0, bit0 in A2h. The ports have alternate functions which are described below:<br>P0.0, AD0, MISO<br>P0.1, AD1, MOSI<br>P0.2, AD2, SPCLK<br>P0.3, AD3, /SS<br>P0.4, AD4, INT2<br>P0.5, AD5, INT3<br>P0.6, AD6, INT4<br>P0.7, AD7, INT5 |

PIN DESCRIPTION, continued

| SYMBOL | TYPE | INITIAL STATE | DESCRIPTIONS |
|---|---|---|---|
| P1.0–P1.7 | I/O S H | High | **PORT 1**: 8-bit, bi-directional I/O port with internal pull-ups. The ports have alternate functions which are described below:<br>P1.0, ADC0, T2<br>P1.1, ADC1, BRAKE<br>P1.2, ADC2, RXD1<br>P1.3, ADC3, TXD1<br>P1.4, ADC4<br>P1.5, ADC5<br>P1.6, ADC6<br>P1.7, ADC7 |
| P2.0-P2.5 | I/O S | Tri-state | **PORT 2**: 8-bit, bi-directional I/O port. This port also provides the upper address bits for accesses to external memory. P2.6 to P2.7 can be software configured as I2C serial ports. P2.0 to P2.5 also provides PWM0 to PWM5 outputs.<br>P2.0, A8, PWM0<br>P2.1, A9, PWM1<br>P2.2, A10, PWM2<br>P2.3, A11, PWM3<br>P2.4, A12, PWM4<br>P2.5, A13, PWM5<br>P2.6, A14, SCL<br>P2.7, A15, SDA<br>**Note:**<br>P2.6 and P2.7 are permanent open drain pins. When access to external memory beyond 16K region, user requires to add external pull-up registers (up to 2Kohm) on these pins. This will result in slight increase in current consumption. |
| P2.6-P2.7 | I/O D S | High-Z | |
| P3.0-P3.7 | I/O S H | High | **PORT 3**: 8-bit, bi-directional I/O port with internal pull-ups. The ports have alternate functions which are described below:<br>P3.0, RXD<br>P3.1, TXD<br>P3.2, /INT0<br>P3.3, /INT1<br>P3.4, T0, IC0, QEA<br>P3.5, T1, IC1, QEB<br>P3.6, /WR<br>P3.7, /RD |

PIN DESCRIPTION, continued

| SYMBOL | TYPE | INITIAL STATE | DESCRIPTIONS |
|---|---|---|---|
| P4.0-P4.3 | I/O S H | High | **PORT 4**: 4-bit multipurpose programmable I/O port with alternate functions. The Port 4 has four different operation modes.<br>P4.0, STADC<br>P4.1, T2EX, IC2<br>P4.2<br>P4.3<br>**Note:** P4.2 & P4.3 are not supported in PLCC44 pins package. |
| P5.0-P5.1 | I/O S | Tri-state | **PORT 5**: 2-bit, bit-directional I/O port. This port is not bit addressable. The alternate functions are described below:<br>P5.0, PWM6<br>P5.1, PWM7<br>**Note:** P5.0 & P5.1 are not supported in PLCC44 pins package. |

**Note** ：TYPE I: input, O: output, I/O: bi-directional, H: pull-high, L: pull-low, D: open drain S: Schmitt Trigger

## 5.1 Port 4

Port 4, SFR P4 at address A5H, is a 4-bit multipurpose programmable I/O port which functions are I/O and chip-select function. It has four different operation modes:

- Mode 0 - P4.0 ~ P4.3 is 4-bit bi-directional I/O port which is the same as port 1. The default Port 4 is a general I/O function.
- Mode1 - P4.0 ~ P4.3 are read data strobe signals which are synchronized with $\overline{RD}$ signal at specified addresses. These read data strobe signals can be used as chip-select signals for external peripherals.
- Mode2 - P4.0 ~ P4.3 are write data strobe signals which are synchronized with $\overline{WR}$ signal at specified addresses. These write data strobe signals can be used as chip-select signals for external peripherals.
- Mode3 - P4.0 ~ P4.3 are read/write data strobe signals which are synchronized with $\overline{RD}$ or $\overline{WR}$ signal at specified addresses. These read/write data strobe signals can be used as chip-select signals for external peripherals.

When Port 4 is configured with the feature of chip-select signals, the chip-select signal address range depends on the contents of the SFR P4xAH, P4xAL, P4CONA and P4CONB. P4xAH and P4xAL contain the 16-bit base address of P4.x. P4CONA and P4CONB contain the control bits to configure the Port 4 operation mode.

# 6. MEMORY ORGANIZATION

The W79E22X SERIES separates the memory into two sections; Program Memory and Data Memory. Program Memory stores instruction op-codes, while Data Memory stores data or memory-mapped devices.

## 6.1 Program Memory (on-chip Flash)

W79E22X SERIES includes one 16/32/64K bytes of main FLASH EPROM for application program (AP FLASH EPROM) and one 4K bytes of FLASH EPROM for loader program (LD FLASH EPROM) to operate the in-system programming (ISP) feature, and one 1/2/2K bytes of NVM Flash EPROM for data storage. The 16/32/64K bytes Flash EPROM is AP0 bank. The default active bank is AP0.

In normal operation, the microcontroller will execute the code from main FLASH EPROM. By setting program registers, user can force the microcontroller to switch to programming mode which will cause it to execute the code (loader program) from the 4K bytes of auxiliary LD FLASH EPROM to update the contents of the 16/32/64K bytes of main FLASH EPROM. After reset, the microcontroller will executes the new application program in the main FLASH EPROM. This ISP feature makes the job easy and efficient in which the application needs to update firmware frequently without opening the chassis.

## 6.2 Data Memory

W79E22X SERIES can access up to 64Kbytes of external Data Memory. This memory region is accessed by the MOVX instructions. Unlike the 8051 derivatives, W79E22X SERIES contains on-chip 1/2/2 Kbytes of Data Memory, which only can be accessed by MOVX instructions. These 1/2/2 Kbytes of SRAM is between address 0000h and 03FFH/07FFH. Access to the on-chip Data Memory is optional under software control. When enabled by DMEO bit of PMR register, a MOVX instruction that uses this area will go to the on-chip RAM. If MOVX instruction accesses the addresses greater than 03FFH/07FFH CPU will automatically access external memory through Port 0 and 2. When disabled, the 1/2/2 KB memory area is transparent to the system memory map. Any MOVX directed to the space between 0000h and FFFFH goes to the expanded bus on the Port 0 and 2. This is the default condition. In addition, the device has the standard 256 bytes of on-chip RAM. This can be accessed either by direct addressing or by indirect addressing. There are also some Special Function Registers (SFRs), which can only be accessed by direct addressing.

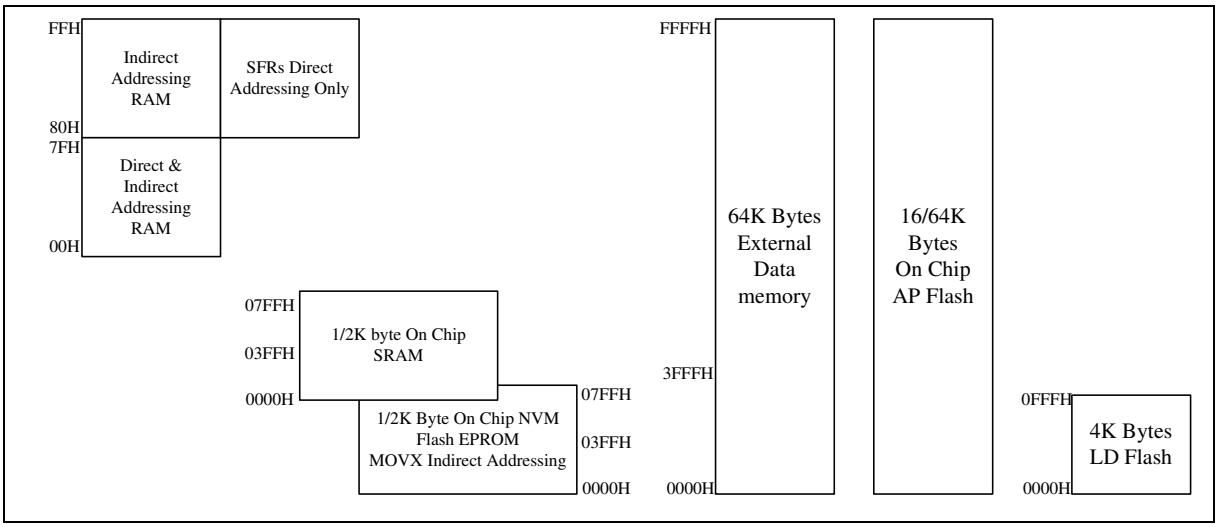



Figure 6-1: W79E22X SERIESA Memory Map

## 6.3 Auxiliary SRAM

W79E22X SERIES has a 1/2/2 KB of data space SRAM which is read/write accessible and is memory mapped. This on-chip SRAM is accessed by the MOVX instruction. There is no conflict or overlap among the 256 bytes scratch-pad memory and the 1/2/2 KB auxiliary sram as they use different addressing modes and instructions. Access to the on-chip Data Memory is optional under software control. Set DMEO bit of PMR SFR to 1 will enable the on-chip 1/2/2 KB MOVX SRAM and at the same time EnNVM bit must be cleared as NVM Data uses the same instruction of MOVX. Refer to.

## 6.4 NVM Data Flash

W79E22X SERIES 1/2/2-KB NVM Data block shown in the diagram on Figure 6-1 shares the same address as AUX-RAM address.

Due to overlapping of AUX-RAM, NVM data memory and external data memory physical address, the following table is defined. EnNVM bit (NVMCON.5) will enable read access to NVM data flash area. DME0 (PMR.0) will enable read access to AUX-RAM.

| ENNVM | DME0 | DATA MEMORY AREA |
|---|---|---|
| 0 | 0 | Enable External RAM read/write access by MOVX |
| 0 | 1 | Enable AUX-RAM read/write access by MOVX |
| 1 | X | Enable NVM data Memory read access by MOVX only. If EER or EWR is set and NVM flash erase or write control is busy, to set this bit read NVM data is invalid. |

Table 6-1: Bits setting for MOVX access to Data Memory Area

| INSTRUCTIONS | | ENNVM = 1<br>NVM SIZE = SRAM (1K) | |
|---|---|---|---|
| | | ADDR ≤ 1K | ADDR > 1K |
| Read access | MOVX A, @DPTR (Read) | NVM[1] | Ext memory[1] |
| | MOVX A, @R0 (Read) | NVM[2] | NOP |
| | MOVX A, @R1 (Read) | NVM[2] | NOP |
| Write access | MOVX @DPTR, A (Write) | NOP | Ext memory[1] |
| | MOVX @R0, A (Write) | NOP | NOP |
| | MOVX @R1, A (Write) | NOP | NOP |

Table 6-2: W79E225 MOVX read/write access destination

| INSTRUCTIONS | | ENNVM = 1<br>NVM SIZE = SRAM (2K) | |
|---|---|---|---|
| | | ADDR ≤ 2K | ADDR > 2K |
| Read access | MOVX A, @DPTR (Read) | NVM[1] | Ext memory[1] |
| | MOVX A, @R0 (Read) | NVM[2] | NOP |
| | MOVX A, @R1 (Read) | NVM[2] | NOP |
| Write access | MOVX @DPTR, A (Write) | NOP | Ext memory[1] |
| | MOVX @R0, A (Write) | NOP | NOP |
| | MOVX @R1, A (Write) | NOP | NOP |

Table 6-3: W79E226/227 MOVX read/write access destination

**Note:**

1. A15~A0=DPTR
2. A15~A8=XRAMAH

It is partition into 16/32/32 pages area and each page has 64 bytes data as below figure. The page 0 is from 0000h ~ 003Fh, page 1 is from 0040h ~ 007Fh until page 31 address located at 07COh ~ 07FFh.

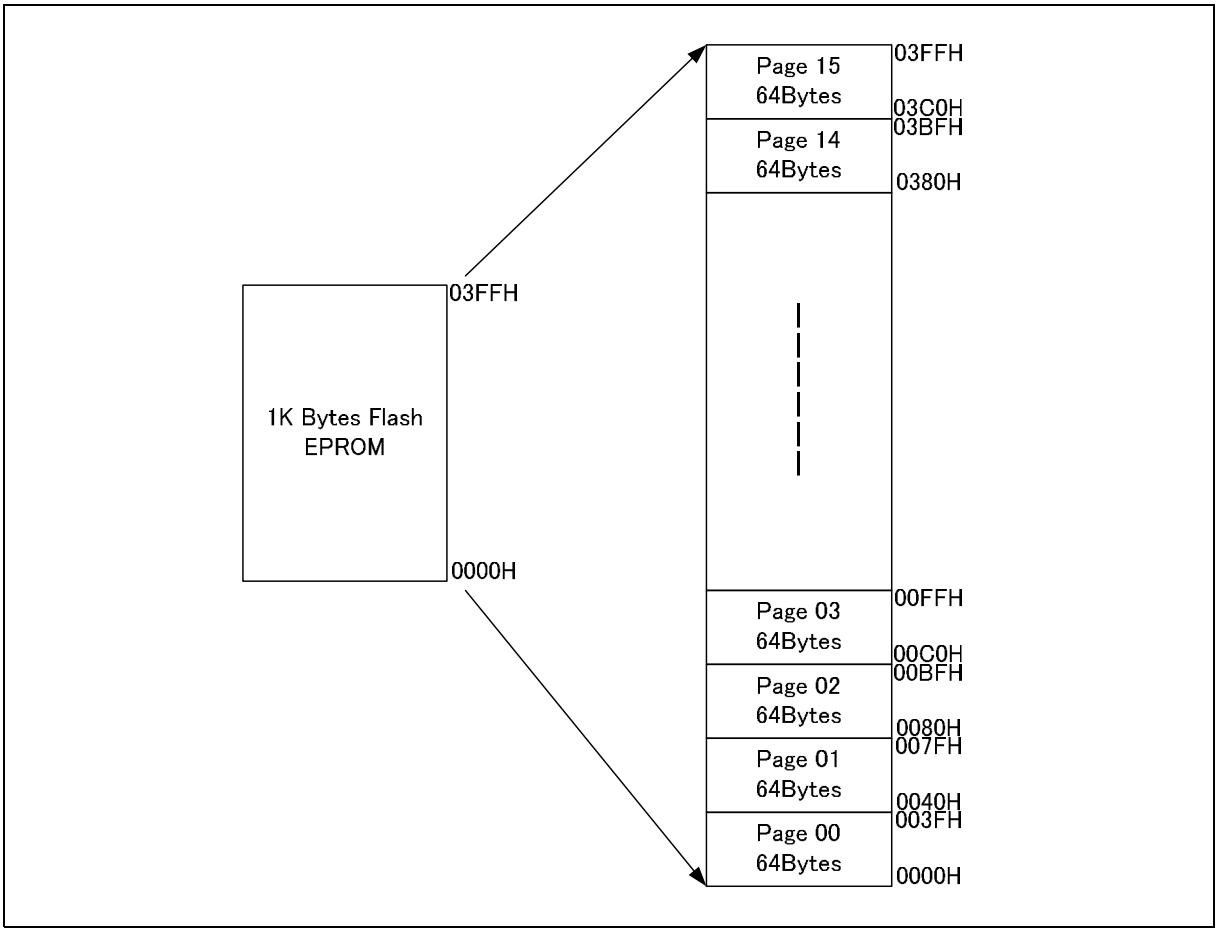



Figure 6-2: W79E225 NVM Data Mapping

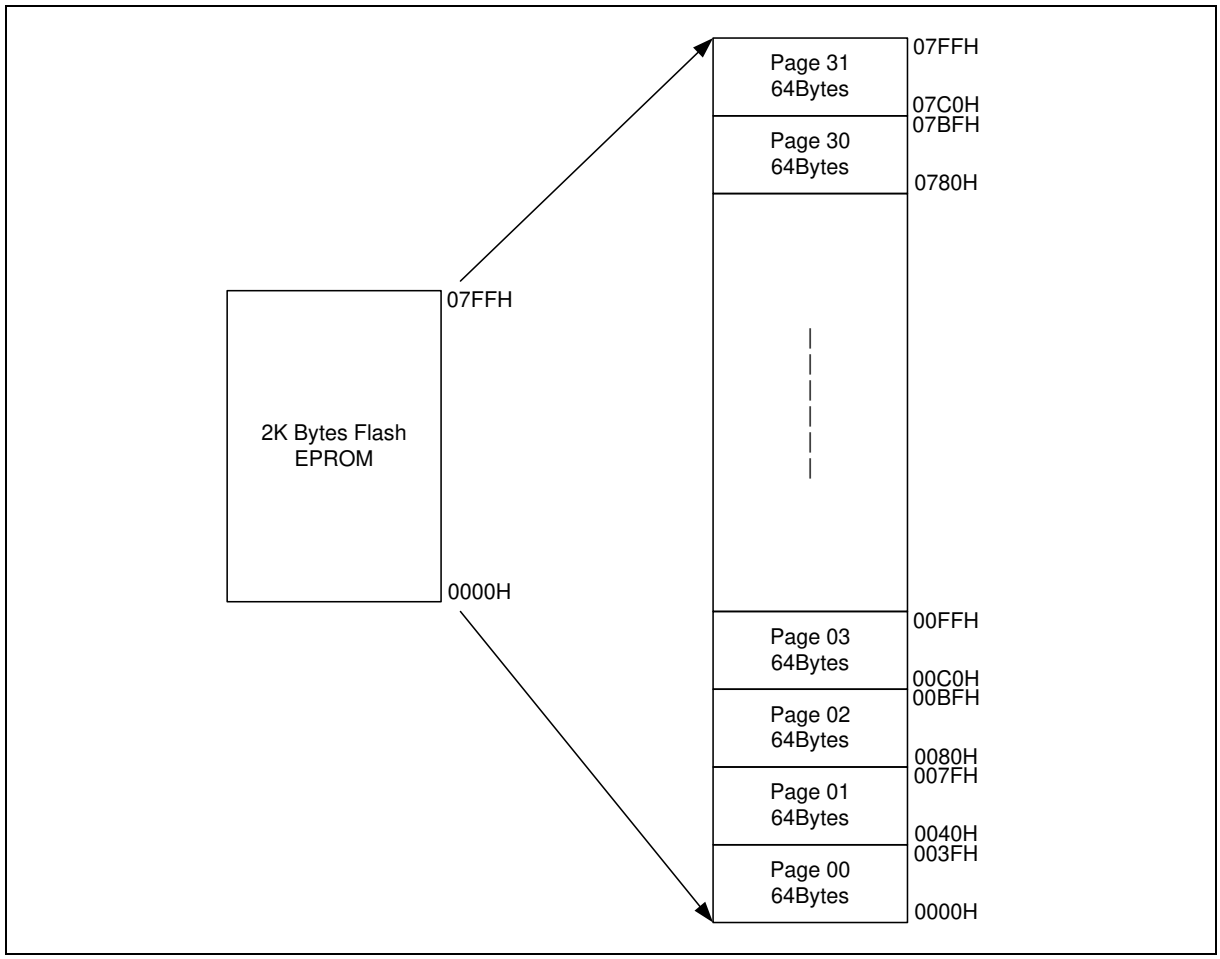



Figure 6-3: W79E226/227 NVM Data Mapping

| PAGE | START ADDRESS | END ADDRESS | PAGE | START ADDRESS | END ADDRESS |
|---|---|---|---|---|---|
| 0 | 0000h | 003Fh | 16 | 0400h | 043Fh |
| 1 | 0040h | 007Fh | 17 | 0440h | 047Fh |
| 2 | 0080h | 00BFh | 18 | 0480h | 04BFh |
| 3 | 00C0h | 00FFh | 19 | 04C0h | 04FFh |
| 4 | 0100h | 013Fh | 20 | 0500h | 053Fh |
| 5 | 0140h | 017Fh | 21 | 0540h | 057Fh |
| 6 | 0180h | 01BFh | 22 | 0580h | 05BFh |
| 7 | 01C0h | 01FFh | 23 | 05C0h | 05FFh |
| 8 | 0200h | 023Fh | 24 | 0600h | 063Fh |
| 9 | 0240h | 027Fh | 25 | 0640h | 067Fh |
| 10 | 0280h | 02BFh | 26 | 0680h | 06BFh |
| 11 | 02C0h | 02FFh | 27 | 06C0h | 06FFh |
| 12 | 0300h | 033Fh | 28 | 0700h | 073Fh |
| 13 | 0340h | 037Fh | 29 | 0740h | 077Fh |
| 14 | 0380h | 03BFh | 30 | 0780h | 07BFh |
| 15 | 03C0h | 03FFh | 31 | 07C0h | 07FFh |

[**Note**: Page 16-31 is for W79E226/227 only]

Table 6-4: W79E22X SERIES NVM page (n) area definition table

It has a dedicated On-Chip RC Oscillator that is fixed at 6MHz +/- 50% frequency to support clock source for the 1/2/2K NVM data Flash. The on chip oscillator is enabled only during program or erase operation, through EWR or EER in NVMCON SFR. EWR or EER bits are cleared by hardware after program or erase operation completed. The program/erase time is automatically controlled by hardware.

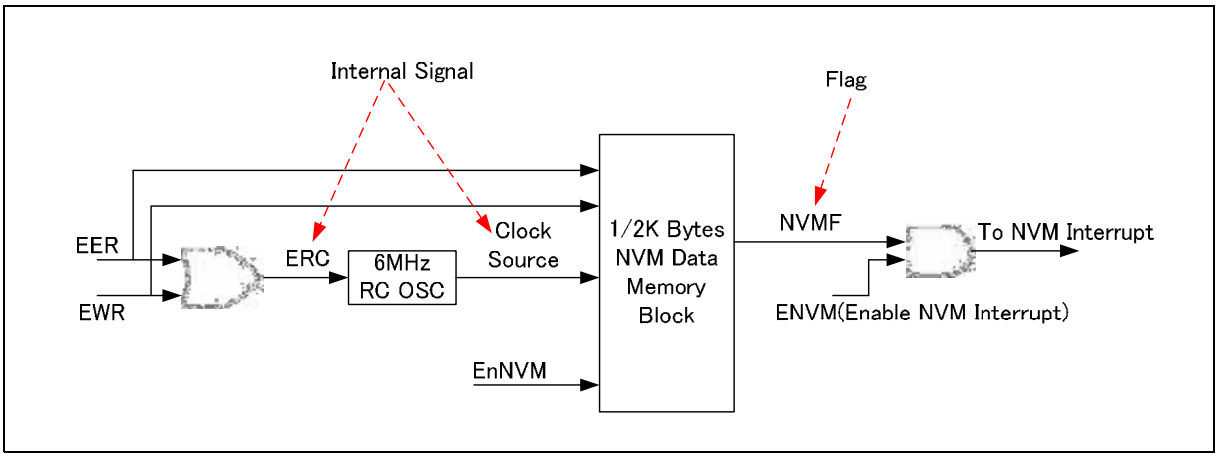



Figure 6-4: NVM control

### 6.4.1 Operation

User is required to enable EnNVM (NVMCON.5) bit for all NVM access (read/write/erase).

Before write data to NVM Data, the page must be erased. A page is erased by setting page address which address will decode and enable page (n) on NVMADDRH and NVMADDRL, then set EER (NVMCON.7) and EnNVM (NVMCON.5). The device will then automatic execute page erase. When completed, NVMF will be set by hardware. NVMF should be cleared by software. Interrupt request will be generated if ENVM (EIE1.5) is enabled. EER bit will be cleared by hardware when erase is completed. The total erase time is about 5ms.

For write, user must set address and data to NVMADDRH/L and NVMDAT, respectively. And then set EWR (NVMCON.6) and EnNVM (NVMCON.5) to enable data write. When completed, the device will set NVMF flag. NVMF flag should be cleared by software. Similarly, interrupt request will be generated if ENVM (EIE1.5) is enabled. The program time is about 50us.

The following shows some examples of NVM operations (using W79E226/227):

**Read NVM data is by MOVX A,@DPTR/R0/R1 instruction:**

A read exceed 2k will read the external address

Example1: DPTR=0x07FF, R0/R1 = 0xFF, XRAMAH=0x07, EnNVM=1

MOVX A,@DPTR → read NVM data at address 0x07FF

MOVX A,@R0 → read NVM data at address 0x07FF

MOVX A,@R1 → read NVM data at address 0x07FF

Example2: DPTR = 0x2000, EnNVM=1, DME0=0

MOVX A,@DPTR → read external RAM data at address 0x2000,

**Erase NVM by SFR register:**

Example1: NVMADDRH = 0x07, NVMADDRL = 0xF0, page 31 will be enabled. After set EER, the page 31 will be erased.

Example2: NVMADDRH = 0x10, NVMADDRL = 0x00, invalid NVM erase instruction (address exceed NVM boundary).

**Write NVM by SFR register:**

Example1: NVMADDRH = 0x07, NVMADDRL = 0xF0

After set EWR, data will be written to the NVM address = 0x07F0 location.

Example2: NVMADDRH = 0x10, NVMADDRL = 0x00, after set EWR, invalid NVM write instruction (address exceed NVM boundary).

During erase, write is invalid. Likewise, during write, erase is invalid. An erase or write is invalid if NVMF is not clear by software. A write to NVMADDRH and NVMADDRL is invalid during Erase or Write, and a write to NVMDAT is invalid only during NVM write access.

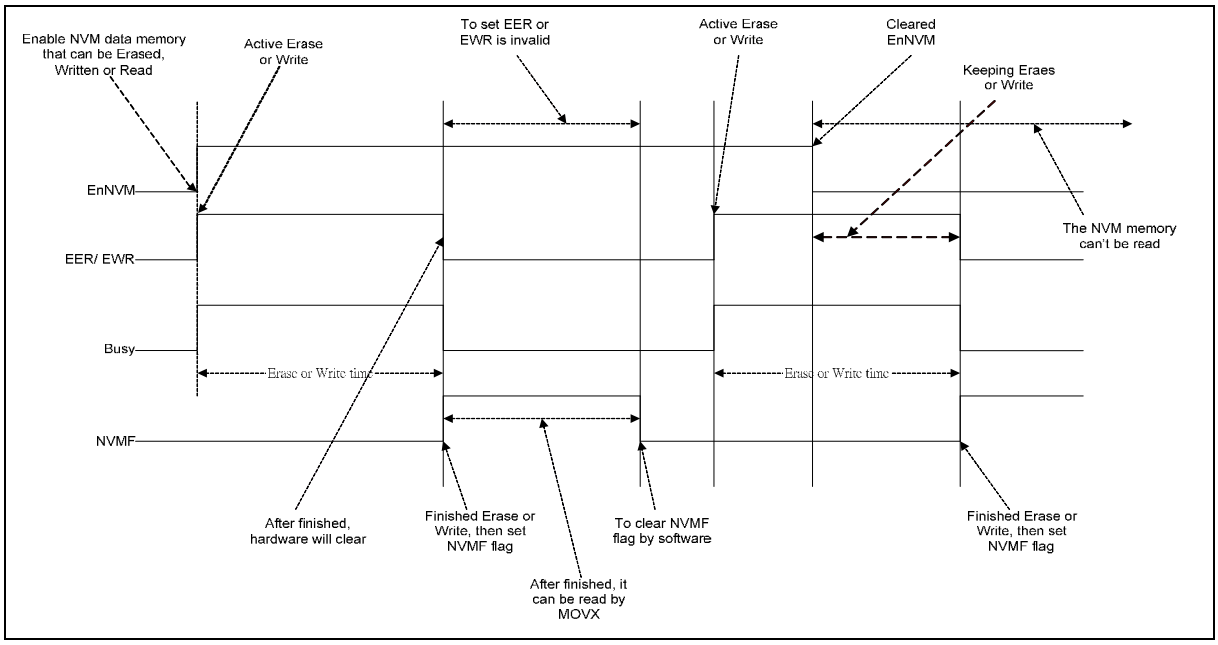



Figure 6-5: NVM data memory control timing

For security purposes, this NVM data flash provides an independent “Lock bit” located in Security bits. It is used to protect the customer’s 1/2/2K bytes of data code. It may be enabled after the external programmer finishes the programming and verifying sequence. Once this bit is set to logic 0, the 1/2/2K bytes of NVM Flash EPROM data can not be accessed again by external device.

**Note:** 1. NVMF can be polled or by h/w interrupt to indicate NVM data memory erase or write operation has completed.

2. While user program is erasing or writing to NVM data memory, the PC counter will continue to fetch for next instruction.

3. When uC is in idle mode and if NVM interrupt and global interrupt are enabled, the completion of either erasing or programming the NVM data memory will exit the idle condition.

## 7. SPECIAL FUNCTION REGISTERS

The W79E22X SERIES uses Special Function Registers (SFR) to control and monitor peripherals. The SFR reside in register locations 80-FFh and are only accessed by direct addressing. The W79E22X SERIES contains all the SFR present in the standard 8051/52, as well as some additional SFR, and, in some cases, unused bits in the standard 8051/52 have new functions. SFR whose addresses end in 0 or 8 (hex) are bit-addressable. The following table of SFR is condensed, with eight locations per row. Empty locations indicate that there are no registers at these addresses.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| F8 | EIP | EIE1 | EIP1 | CCL0 /PCNTL | CCH0 /PCNTH | CCL1 /PLSCNTL | CCH1 /PLSCNTH | INTCTRL |
| F0 | B | | | SPCR | SPSR | SPDR | I2CSADEN | EIPH |
| E8 | EIE | I2CON | I2ADDR | NVMADDRH | I2DAT | I2STATUS | I2CLK | I2TIMER |
| E0 | ACC | ADCCON | ADCH | ADCL | | PDTC1 | PDTC0 | PWMCON4 |
| D8 | WDCON | PWMPL | PWM0L | NVMADDRL | PWMCON1 | PWM2L | PWM6L | PWMCON3 |
| D0 | PSW | PWMPH | PWM0H | NVMDAT | QEICON | PWM2H | PWM6H | WDCON2 |
| C8 | T2CON | T2MOD | RCAP2L | RCAP2H | TL2 | TH2 | PWMCON2 | PWM4L |
| C0 | SCON1 | SBUF1 | T3MOD | T3CON | PMR | FSPLT | ADCPS | TA |
| B8 | IP | SADEN | SADEN1 | POVM | POVD | PIO | PWMEN | PWM4H |
| B0 | P3 | P5 | | | RCAP3L | RCAP3H | EIP1H | IPH |
| A8 | IE | SADDR | SADDR1 | | SFRAL | SFRAH | SFRFD | SFRCN |
| A0 | P2 | XRAMAH | P4CSIN | CAPCON0 | CAPCON1 | P4 | CCL2 /MAXCNTL | CCH2 /MAXCNTH |
| 98 | SCON | SBUF | P42AL | P42AH | P43AL | P43AH | NVMCON | CHPCON |
| 90 | P1 | EXIF | P4CONA | P4CONB | P40AL | P40AH | P41AL | P41AH |
| 88 | TCON | TMOD | TL0 | TL1 | TH0 | TH1 | CKCON | CKCON1 |
| 80 | P0 | SP | DPL | DPH | TL3 | TH3 | | PCON |

Table 7-1: Special Function Register Location Table

| SYMBOL | DEFINITION | ADDRESS | MSB BIT_ADDRESS, LSB | | | | | | | SYMBOL | RESET |
|---|---|---|---|---|---|---|---|---|---|---|---|
| INTCTRL | INTERRUPT CONTROL REGISTER | FFH | - | - | INT5CT1 | INT5CT0 | INT4CT1 | INT4CT0 | INT3CT1 | INT3CT0 | xx00 0000B |
| CCH1 /PLSCNTH | CAPTURE COUNTER HIGH 1 REGISTER | FEH | CCH1.7 /PLSCNTH.7 | CCH1.6 /PLSCNTH.6 | CCH1.5 /PLSCNTH.5 | CCH1.4 /PLSCNTH.4 | CCH1.3 /PLSCNTH.3 | CCH1.2 /PLSCNTH.2 | CCH1.1 /PLSCNTH.1 | CCH1.0 /PLSCNTH.0 | 0000 0000B |
| CCL1 /PLSCNTL | CAPTURE COUNTER LOW 1 REGISTER | FDH | CCL1.7 /PLSCNTL.7 | CCL1.6 /PLSCNTL.6 | CCL1.5 /PLSCNTL.5 | CCL1.4 /PLSCNTL.4 | CCL1.3 /PLSCNTL.3 | CCL1.2 /PLSCNTL.2 | CCL1.1 /PLSCNTL.1 | CCL1.0 /PLSCNTL.0 | 0000 0000B |
| CCH0 /PCNTH | CAPTURE COUNTER HIGH 0 REGISTER | FCH | CCH0.7 /PCNTH.7 | CCH0.6 /PCNTH.6 | CCH0.5 /PCNTH.5 | CCH0.4 /PCNTH.4 | CCH0.3 /PCNTH.3 | CCH0.2 /PCNTH.2 | CCH0.1 /PCNTH.1 | CCH0.0 /PCNTH.0 | 0000 0000B |
| CCL0 /PCNTL | CAPTURE COUNTER LOW 0 REGISTER | FBH | CCL0.7 /PCNTL.7 | CCL0.6 /PCNTL.6 | CCL0.5 /PCNTL.5 | CCL0.4 /PCNTL.4 | CCL0.3 /PCNTL.3 | CCL0.2 /PCNTL.2 | CCL0.1 /PCNTL.1 | CCL0.0 /PCNTL.0 | 0000 0000B |
| EIP1 | EXTENDED INTERRUPT PRIORITY 1 | FAH | - | - | PNVMI | PCPTF | PT3 | PBKF | PPWMF | PSPI | xx00 0000B |
| EIE1 | INTERRUPT ENABLE 1 | F9H | - | - | ENVM | ECPTF | ET3 | EBK | EPWM | ESPI | xx00 0000B |
| EIP | EXTENDED INTERRUPT PRIORITY | F8H | (FF) PS1 | (FE) PX5 | (FD) PX4 | (FC) PWDI | (FB) PX3 | (FA) PX2 | (F9) - | (F8) PI2C | 0000 00x0B |
| EIPH | EXTENDED INTERRUPT HIGH PRIORITY | F7H | PS1H | PX5H | PX4H | PWDIH | PX3H | PX2H | - | PI2CH | 0000 00x0B |
| I2CSADEN | I2C SLAVE ADDRESS MASK | F6H | I2CSADEN.7 | I2CSADEN.6 | I2CSADEN.5 | I2CSADEN.4 | I2CSADEN.3 | I2CSADEN.2 | I2CSADEN.1 | I2CSADEN.0 | 1111 1110B |
| SPDR | SERIAL PERIPHERAL DATA REGISTER | F5H | SPD.7 | SPD.6 | SPD.5 | SPD.4 | SPD.3 | SPD.2 | SPD.1 | SPD.0 | xxxx xxxxB |
| SPSR | SERIAL PERIPHERAL STATUS REGISTER | F4H | SPIF | WCOL | SPIOVF | MODF | DRSS | - | - | - | 0000 0xxxB |
| SPCR | SERIAL PERIPHERAL CONTROL REGISTER | F3H | SSOE | SPE | LSBFE | MSTR | CPOL | CPHA | SPR1 | SPR0 | 0000 0100B |
| B | B REGISTER | F0H | (F7) | (F6) | (F5) | (F4) | (F3) | (F2) | (F1) | (F0) | 0000 0000B |
| I2TIMER | I2C TIMER COUNTER REGISTER | EFH | - | - | - | - | - | ENTI | DIV4 | TIF | xxxx x000B |
| I2CLK | I2C CLOCK RATE | EEH | I2CLK.7 | I2CLK.6 | I2CLK.5 | I2CLK.4 | I2CLK.3 | I2CLK.2 | I2CLK.1 | I2CLK.0 | 0000 0000B |
| I2STATUS | I2C STATUS REGISTER | EDH | | | | | | | | | 1111 1000B |
| I2DAT | I2C DATA | ECH | I2DAT.7 | I2DAT.6 | I2DAT.5 | I2DAT.4 | I2DAT.3 | I2DAT.2 | I2DAT.1 | I2DAT.0 | 0000 0000B |
| NVMADDRH | NVM HIGH BYTE ADDRESS | EBH | - | - | - | - | - | NVMADDRH.10 | NVMADDRH.9 | NVMADDRH.8 | xxxx x000B |
| I2ADDR | I2C SLAVE ADDRESS | EAH | ADDR.7 | ADDR.6 | ADDR.5 | ADDR.4 | ADDR.3 | ADDR.2 | ADDR.1 | GC | 0000 0000B |
| I2CON | I2C CONTROL REGISTER | E9H | - | ENS | STA | STO | SI | AA | I2CIN | - | X000 000xB |
| EIE | EXTENDED INTERRUPT ENABLE | E8H | (EF) ES1 | (EE) EX5 | (ED) EX4 | (EC) EWDI | (EB) EX3 | (EA) EX2 | (E9) | (E8) EI2C | 0000 00x0B |
| PWMCON4 | PWM CONTROL REGISTER 4 | E7H | PWMEOM | PWMOOM | PWM6OM | PWM7OM | - | - | - | BKF | 0000 xxx0B |
| PDTC0 | DEAD TIME CONTROL REGISTER 0 | E6H | PDTC0.7 | PDTC0.6 | PDTC0.5 | PDTC0.4 | PDTC0.3 | PDTC0.2 | PDTC0.1 | PDTC0.0 | 0000 0000B |
| PDTC1 | DEAD TIME CONTROL REGISTER 1 | E5H | PDTC1.7 | PDTC1.6 | PDTC1.5 | PDTC1.4 | PDTC1.3 | PDTC1.2 | PDTC1.1 | PDTC1.0 | 0000 0000B |
| ADCL | ADC CONVERTER RESULT LOW BYTE | E3H | ADCLK1 | ADCLK0 | - | - | - | - | ADC.1 | ADC.0 | 00xx xxxxB |

Continued

| SYMBOL | DEFINITION | ADDRESS | MSB BIT_ADDRESS, LSB SYMBOL | | | | | | | | RESET |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ADCH | ADC CONVERTER RESULT HIGH BYTE | E2H | ADC.9 | ADC.8 | ADC.7 | ADC.6 | ADC.5 | ADC.4 | ADC.3 | ADC.2 | xxxx xxxxB |
| ADCCON | ADC CONTROL REGISTER | E1H | ADCEN | - | ADCEX | ADCI | ADCS | AADR2 | AADR1 | AADR0 | 0x00 0000B |
| ACC | ACCUMULATOR | E0H | (E7) | (E6) | (E5) | (E4) | (E3) | (E2) | (E1) | (E0) | 0000 0000B |
| PWMCON3 | PWM CONTROL REGISTER 3 | DFH | PWM7B | PWM6B | PWM5B | PWM4B | PWM3B | PWM2B | PWM1B | PWM0B | 0000 0000B |
| PWM6L | PWM 6 LOW BITS REGISTER | DEH | PWM6.7 | PWM6.6 | PWM6.5 | PWM6.4 | PWM6.3 | PWM6.2 | PWM6.1 | PWM6.0 | 0000 0000B |
| PWM2L | PWM 2 LOW BITS REGISTER | DDH | PWM2.7 | PWM2.6 | PWM2.5 | PWM2.4 | PWM2.3 | PWM2.2 | PWM2.1 | PWM2.0 | 0000 0000B |
| PWMCON1 | PWM CONTROL REGISTER 1 | DCH | PWMRUN | Load | PWMF | CLRPWM | PWM6I | PWM4I | PWM2I | PWM0I | 0000 0000B |
| NVMADDRL | NVM LOW BYTE ADDRESS | DBH | NVMADDRH.7 | NVMADDRH.6 | NVMADDRH.5 | NVMADDRH.4 | NVMADDRH.3 | NVMADDRH.2 | NVMADDRH.1 | NVMADDRH.8 | 0000 0000B |
| PWM0L | PWM 0 LOW BITS REGISTER | DAH | PWM0.7 | PWM0.6 | PWM0.5 | PWM0.4 | PWM0.3 | PWM0.2 | PWM0.1 | PWM0.0 | 0000 0000B |
| PWMPL | PWM COUNTER LOW REGISTER | D9H | PWMP.7 | PWMP.6 | PWMP.5 | PWMP.4 | PWMP.3 | PWMP.2 | PWMP.1 | PWMP.0 | 0000 0000B |
| WDCON | WATCH-DOG CONTROL | D8H | (DF)<br>- | (DE)<br>POR | (DD)<br>- | (DC)<br>- | (DB)<br>WDIF | (DA)<br>WTRF | (D9)<br>EWT | (D8)<br>RWT | 0100 0000B |
| WDCON2 | WATCH-DOG CONTROL2 | D7H | - | - | - | - | - | - | - | STRLD | 0000 0000B |
| PWM6H | PWM 6 HIGH BITS REGISTER | D6H | - | - | - | - | PWM6.11 | PWM6.10 | PWM6.9 | PWM6.8 | xxxx 0000B |
| PWM2H | PWM 2 HIGH BITS REGISTER | D5H | - | - | - | - | PWM2.11 | PWM2.10 | PWM2.9 | PWM2.8 | xxxx 0000B |
| QEICON | QEI CONTROL REGISTER | D4H | - | - | - | DISIDX | DIR | QEIM1 | QEIM0 | QEIEN | xxx0 0000B |
| NVMDAT | NVM DATA | D3H | NVMDAT.7 | NVMDAT.6 | NVMDAT.5 | NVMDAT.4 | NVMDAT.3 | NVMDAT.2 | NVMDAT.1 | NVMDAT.0 | 0000 0000B |
| PWM0H | PWM 0 HIGH BITS REGISTER | D2H | - | - | - | - | PWM0.11 | PWM0.10 | PWM0.9 | PWM0.8 | xxxx 0000B |
| PWMPH | PWM COUNTER HIGH REGISTER | D1H | - | - | - | - | PWMP.11 | PWMP.10 | PWMP.9 | PWMP.8 | xxxx 0000B |
| PSW | PROGRAM STATUS WORD | D0H | (D7)<br>CY | (D6)<br>AC | (D5)<br>F0 | (D4)<br>RS1 | (D3)<br>RS0 | (D2)<br>OV | (D1)<br>F1 | (D0)<br>P | 0000 0000B |
| PWM4L | PWM 4 LOW BITS REGISTER | CFH | PWM4.7 | PWM4.6 | PWM4.5 | PWM4.4 | PWM4.3 | PWM4.2 | PWM4.1 | PWM4.0 | 0000 0000B |
| PWMCON2 | PWM CONTROL REGISTER 2 | CEH | BKCH | BKPS | BPEN | BKEN | FP1 | FP0 | PMOD1 | PMOD0 | 0000 0000B |
| TH2 | T2 REG. HIGH | CDH | TH2.7 | TH2.6 | TH2.5 | TH2.4 | TH2.3 | TH2.2 | TH2.1 | TH2.0 | 0000 0000B |
| TL2 | T2 REG. LOW | CCH | TL2.7 | TL2.6 | TL2.5 | TL2.4 | TL2.3 | TL2.2 | TL2.1 | TL2.0 | 0000 0000B |
| RCAP2H | T2 CAPTURE LOW | CBH | RCAP2H.7 | RCAP2H.6 | RCAP2H.5 | RCAP2H.4 | RCAP2H.3 | RCAP2H.2 | RCAP2H.1 | RCAP2H.0 | 0000 0000B |
| RCAP2L | T2 CAPTURE HIGH | CAH | RCAP2L.7 | RCAP2L.6 | RCAP2L.5 | RCAP2L.4 | RCAP2L.3 | RCAP2L.2 | RCAP2L.1 | RCAP2L.0 | 0000 0000B |
| T2MOD | TIMER 2 MODE | C9H | HC5 | HC4 | HC3 | HC2 | T2CR | - | - | DCEN | 0000 0xx0B |
| T2CON | TIMER 2 CONTROL | C8H | (CF)<br>TF2 | (CE)<br>EXF2 | (CD)<br>RCLK | (CC)<br>TCLK | (CB)<br>EXEN2 | (CA)<br>TR2 | (C9)<br>$C/\overline{T}$ | (C8)<br>$CP/\overline{RL2}$ | 0000 0000B |
| TA | TIME ACCESS REGISTER | C7H | TA.7 | TA.6 | TA.5 | TA.4 | TA.3 | TA.2 | TA.1 | TA.0 | 0000 0000B |
| DDIO | DISABLE DIGITAL I/O | C6H | DDIO.7 | DDIO.6 | DDIO.5 | DDIO.4 | DDIO.3 | DDIO.2 | DDIO.1 | DDIO.0 | 0000 0000B |

Continued

| SYMBOL | DEFINITION | ADDRESS | MSB BIT_ADDRESS, LSB SYMBOL | | | | | | | | RESET |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FSPLT | FAULT SAMPLING TIME REGISTER | C5H | SCMP1 | SCMP0 | SFP1 | SFP0 | SFCEN | SFCST | SFCDIR | LSBD | 0000 0000B |
| PMR | POWER MANAGEMENT REGISTER | C4H | - | - | - | - | - | ALEOFF | - | DME0 | xxxx x0x0B |
| T3CON | TIMER 3 CONTROL | C3H | TF3 | - | - | - | - | TR3 | - | CMP/$\overline{RL3}$ | 0xxx x0x0B |
| T3MOD | TIMER 3 MODE CONTROL | C2H | ENLD | ICEN2 | ICEN1 | ICEN0 | T3CR | - | - | - | 0000 0xxxB |
| SBUF1 | SERIAL BUFFER 1 | C1H | SBUF1.7 | SBUF1.6 | SBUF1.5 | SBUF1.4 | SBUF1.3 | SBUF1.2 | SBUF1.1 | SBUF1.0 | xxxx xxxxB |
| SCON1 | SERIAL CONTROL 1 | C0H | (BF) SM0_1/FE_1 | (BE) SM1_1 | (BD) SM2_1 | (BC) REN_1 | (BB) TB8_1 | (BA) RB8_1 | (B9) TI_1 | (B8) RI_1 | 0000 0000B |
| PWM4H | PWM 4 HIGH BITS REGISTER | BFH | - | - | - | - | PWM4.11 | PWM4.10 | PWM4.9 | PWM4.8 | xxxx 0000B |
| PWMEN | PWM OUTPUT ENABLE REGISTER | BEH | PWM7EN | PWM6EN | PWM5EN | PWM4EN | PWM3EN | PWM2EN | PWM1EN | PWM0EN | 0000 0000B |
| PIO | PWM PIN OUTPUT SOURCE SELECT | BDH | PIO7 | PIO6 | PIO5 | PIO4 | PIO3 | PIO2 | PIO1 | PIO0 | 0000 0000B |
| POVD | PWM OUTPUT STATE REGISTERS | BCH | POVD.7 | POVD.6 | POVD.5 | POVD.4 | POVD.3 | POVD.2 | POVD.1 | POVD.0 | 0000 0000B |
| POVM | PWM OUTPUT OVERRIDE CONTROL REGISTERS | BBH | POVM.7 | POVM.6 | POVM.5 | POVM.4 | POVM.3 | POVM.2 | POVM.1 | POVM.0 | 0000 0000B |
| SADEN1 | SLAVE ADDRESS MASK 1 | BAH | SADEN1.7 | SADEN1.6 | SADEN1.5 | SADEN1.4 | SADEN1.3 | SADEN1.2 | SADEN1.1 | SADEN1.0 | 0000 0000B |
| SADEN | SLAVE ADDRESS MASK | B9H | SADEN.7 | SADEN.6 | SADEN.5 | SADEN.4 | SADEN.3 | SADEN.2 | SADEN.1 | SADEN.0 | 0000 0000B |
| IP | INTERRUPT PRIORITY | B8H | (BF) | (BE) PADC | (BD) PT2 | (BC) PS | (BB) PT1 | (BA) PX1 | (B9) PT0 | (B8) PX0 | 0000 0000B |
| IPH | INTERRUPT HIGH PRIORITY | B7H | - | PADCH | PT2H | PSH | PT1H | PX1H | PT0H | PX0H | x000 0000B |
| EIP1H | EXTENDED INTERRUPT HIGH PRIORITY 1 | B6H | - | - | PNVMIH | PCPTFH | PT3H | PBKFH | PPWMH | PSPIH | xx00 0000B |
| RCAP3H | RELOAD CAPTURE 3 HIGH REGISTER | B5H | RCAP3H.7 | RCAP3H.6 | RCAP3H.5 | RCAP3H.4 | RCAP3H.3 | RCAP3H.2 | RCAP3H.1 | RCAP3H.0 | 0000 0000B |
| RCAP3L | RELOAD CAPTURE 3 LOW REGISTER | B4H | RCAP3L.7 | RCAP3L.6 | RCAP3L.5 | RCAP3L.4 | RCAP3L.3 | RCAP3L.2 | RCAP3L.1 | RCAP3L.0 | 0000 0000B |
| P5 | PORT 5 | B1H | - | - | - | - | - | - | PWM7 | PWM6 | xxxx xx11B |
| P3 | PORT 3 | B0H | (B7) RD | (B6) WR | (B5) T1/ IC1/QEB | (B4) T0/ ICO/QEA | (B3) /INT1 | (B2) /INT0 | (B1) TXD | (B0) RXD | 1111 1111B |
| SFRCN | F/W FLASH CONTROL | AFH | - | WFWIN | NOE | NCE | CTRL3 | CTRL2 | CTRL1 | CTRL0 | x011 1111B |
| SFRFD | F/W FLASH DATA | AEH | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | xxxx xxxxB |
| SFRAH | F/W FLASH HIGH ADDRESS | ADH | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 | 0000 0000B |
| SFRAL | F/W FLASH LOW ADDRESS | ACH | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 | 0000 0000B |
| SADDR1 | SLAVE ADDRESS 1 | AAH | SADDR1.7 | SADDR1.6 | SADDR1.5 | SADDR1.4 | SADDR1.3 | SADDR1.2 | SADDR1.1 | SADDR1.0 | 0000 0000B |
| SADDR | SLAVE ADDRESS | A9H | SADDR.7 | SADDR.6 | SADDR.5 | SADDR.4 | SADDR.3 | SADDR.2 | SADDR.1 | SADDR.0 | 0000 0000B |
| IE | INTERRUPT ENABLE | A8H | (AF) EA | (AE) EADC | (AD) ET2 | (AC) ES | (AB) ET1 | (AA) EX1 | (A9) ET0 | (A8) EX0 | 0000 0000B |

Continued

| SYMBOL | DEFINITION | ADDRESS | MSB BIT_ADDRESS, LSB SYMBOL | | | | | | | | RESET |
|---|---|---|---|---|---|---|---|---|---|---|---|
| CCH2/MAXCNTH | INPUT CAPTURE 2 HIGH REGISTER/ MAXIMUM COUNTER HIGH REGISTER | A7h | CCH2.7 /MAXCNTH.7 | CCH2.6 MAXCNTH.6 | CCH2.5 /MAXCNTH.5 | CCH2.4 /MAXCNTH.4 | CCH2.3 /MAXCNTH.3 | CCH2.2 /MAXCNTH.2 | CCH2.1 /MAXCNTH.1 | CCH2.0 /MAXCNTH.0 | 0000 0000B |
| CCL2/MAXCNTL | INPUT CAPTURE 2 LOW REGISTER/ MAXIMUM COUNTER LOW REGISTER | A6h | CCL2.7 /MAXCNTL.7 | CCL2.6 /MAXCNTL.6 | CCL2.5 /MAXCNTL.5 | CCL2.4 /MAXCNTL.4 | CCL2.3 /MAXCNTL.3 | CCL2.2 /MAXCNTL.2 | CCL2.1 /MAXCNTL.1 | CCL2.0 /MAXCNTL.0 | 0000 0000B |
| P4 | PORT 4 | A5H | - | - | - | - | P4.3 | P4.2 | T2EX/IC2 | STADC | xxxx 1111B |
| CAPCON1 | CAPTURE CONTROL 1 REGISTER | A4H | - | - | ENF2 | ENF1 | ENF0 | CPTF2 | CPTF1/ DIRF | CPTF0/ QEIF | xx00 0000B |
| CAPCON0 | CAPTURE CONTROL 0 REGISTER | A3H | CCT2.1 | CCT2.0 | CCT1.1 | CCT1.0 | CCT0.1 | CCT0.0 | CCLD1 | CCLD0 | 0000 0000B |
| P4CSIN | P4 CS SIGN | A2H | P43INV | P42INV | P41INV | P40INV | - | PWDNH | RMWFP | P0UP | 0000 x000B |
| XRAMAH | RAM HIGH BYTE ADDRESS | A1H | - | - | - | - | - | A10 | A9 | A8 | 0000 0000B |
| P2 | PORT 2 | A0H | (A7) A15/ SDA | (A6) A14/ SCL | (A5) A13/ PWM5 | (A4) A12/ PWM4 | (A3) A11/ PWM3 | (A2) A10/ PWM2 | (A1) A9/ PWM1 | (A0) A8/ PWM0 | 1111 1111B |
| CHPCON | ON CHIP PROGRAMMING CONTROL | 9FH | SWRST/ REBOOT | - | LD/AP | - | - | - | LDSEL | ENP | 0000 0000B |
| NVMCON | NVM CONTROL | 9EH | EER | EWR | EnNVM | - | - | - | - | NVMF | 000x xxx0B |
| P43AH | HI ADDR. COMPARATOR OF P4.3 | 9DH | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 | 0000 0000B |
| P43AL | LO ADDR. COMPARATOR OF P4.3 | 9CH | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 | 0000 0000B |
| P42AH | HI ADDR. COMPARATOR OF P4.2 | 9BH | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 | 0000 0000B |
| P42AL | LO ADDR. COMPARATOR OF P4.2 | 9AH | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 | 0000 0000B |
| SBUF | SERIAL BUFFER | 99H | SBUF.7 | SBUF.6 | SBUF.5 | SBUF.4 | SBUF.3 | SBUF.2 | SBUF.1 | SBUF.0 | xxxx xxxxB |
| SCON | SERIAL CONTROL | 98H | (9F) SM0/FE | (9E) SM1 | (9D) SM2 | (9C) REN | (9B) TB8 | (9A) RB8 | (99) TI | (98) RI | 0000 0000B |
| P41AH | HI ADDR. COMPARATOR OF P4.1 | 97H | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 | 0000 0000B |
| P41AL | LO ADDR. COMPARATOR OF P4.1 | 96H | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 | 0000 0000B |
| P40AH | HI ADDR. COMPARATOR OF P4.0 | 95H | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 | 0000 0000B |
| P40AL | LO ADDR. COMPARATOR OF P4.0 | 94H | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 | 0000 0000B |
| P4CONB | P4 CONTROL REGISTER B | 93H | P43FUN1 | P43FUN0 | P43CMP1 | P43CMP0 | P42FUN1 | P42FUN0 | P42CMP1 | P42CMP0 | 0000 0000B |
| P4CONA | P4 CONTROL REGISTER A | 92H | P41FUN1 | P41FUN0 | P41CMP1 | P41CMP0 | P40FUN1 | P40FUN0 | P40CMP1 | P40CMP0 | 0000 0000B |
| EXIF | EXTERNAL INTERRUPT FLAG | 91H | IE5 | IE4 | IE3 | IE2 | - | - | - | - | 0000 xxxxB |
| P1 | PORT 1 | 90H | (97) ADC7 | (96) ADC6 | (95) ADC5 | (94) ADC4 | (93) TXD1/ ADC3 | (92) RXD1/ ADC2 | (91) ADC1/ Brake | (90) T2/ ADC0 | 1111 1111B |
| CKCON1 | CLOCK CONTROL 1 | 8FH | - | - | - | - | - | - | CCDIV1 | CCDIV0 | 0000 0000B |
| CKCON | CLOCK CONTROL | 8EH | WD1 | WD0 | T2M | T1M | T0M | MD2 | MD1 | MD0 | 0000 0001B |
| TH1 | TIMER HIGH 1 | 8DH | TH1.7 | TH1.6 | TH1.5 | TH1.4 | TH1.3 | TH1.2 | TH1.1 | TH1.0 | 0000 0000B |

Continued

| SYMBOL | DEFINITION | ADDRESS | MSB LSB | BIT_ADDRESS, SYMBOL | | | | | | | RESET |
|---|---|---|---|---|---|---|---|---|---|---|---|
| TH0 | TIMER HIGH 0 | 8CH | TH0.7 | TH0.6 | TH0.5 | TH0.4 | TH0.3 | TH0.2 | TH0.1 | TH0.0 | 0000 0000B |
| TL1 | TIMER LOW 1 | 8BH | TL1.7 | TL1.6 | TL1.5 | TL1.4 | TL1.3 | TL1.2 | TL1.1 | TL1.0 | 0000 0000B |
| TL0 | TIMER LOW 0 | 8AH | TL0.7 | TL0.6 | TL0.5 | TL0.4 | TL0.3 | TL0.2 | TL0.1 | TL0.0 | 0000 0000B |
| TMOD | TIMER MODE | 89H | GATE | $C/\overline{T}$ | M1 | M0 | GATE | $C/\overline{T}$ | M1 | M0 | 0000 0000B |
| TCON | TIMER CONTROL | 88H | (8F) TF1 | (8E) TR1 | (8D) TF0 | (8C) TR0 | (8B) IE1 | (8A) IT1 | (89) IE0 | (88) IT0 | 0000 0000B |
| PCON | POWER CONTROL | 87H | SMOD | SMOD0 | - | - | GF1 | GF0 | PD | IDL | 00xx 0000B |
| TH3 | TIMER HIGH 3 | 85H | TH3.7 | TH3.6 | TH3.5 | TH3.4 | TH3.3 | TH3.2 | TH3.1 | TH3.0 | 0000 0000B |
| TL3 | TIMER LOW 3 | 84H | TL3.7 | TL3.6 | TL3.5 | TL3.4 | TL3.3 | TL3.2 | TL3.1 | TL3.0 | 0000 0000B |
| DPH | DATA POINTER HIGH | 83H | DPH.7 | DPH.6 | DPH.5 | DPH.4 | DPH.3 | DPH.2 | DPH.1 | DPH.0 | 0000 0000B |
| DPL | DATA POINTER LOW | 82H | DPL.7 | DPL.6 | DPL.5 | DPL.4 | DPL.3 | DPL.2 | DPL.1 | DPL.0 | 0000 0000B |
| SP | STACK POINTER | 81H | SP.7 | SP.6 | SP.5 | SP.4 | SP.3 | SP.2 | SP.1 | SP.0 | 0000 0111B |
| P0 | PORT 0 | 80H | (87) INT5 | (86) INT4 | (85) INT3 | (84) INT2 | (83) /SS | (82) SPCLK | (81) MOSI | (80) MISO | 1111 1111B |

Table 7-2: Special Function Registers

**PORT 0**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | P0.7 | P0.6 | P0.5 | P0.4 | P0.3 | P0.2 | P0.1 | P0.0 |

Mnemonic: P0 Address: 80h

Port 0 is an open-drain 8-bit bi-directional I/O port. As an alternate function Port 0 can function as the multiplexed address/data bus to access off-chip memory. During the time when ALE is high, the LSB of a memory address is presented. When ALE is low, the port transits to a bi-directional data bus. This bus is used for reading external ROM and for reading or writing external RAM memory or peripherals. When used as a memory bus, the port provides active high drivers. The reset condition of Port 0 is tri-state. Pull-up resistors are required when using Port 0 as an I/O port.

| BIT | NAME | FUNCTION |
|---|---|---|
| 0 | P0.0 | MISO: SPI Master In Slave Out. |
| 1 | P0.1 | MOSI: SPI Master Out Slave In. |
| 2 | P0.2 | SPCLK: SPI Clock. |
| 3 | P0.3 | /SS: Slave Select. |
| 4 | P0.4 | INT2: External Interrupt 2. |
| 5 | P0.5 | INT3: External Interrupt 3. |
| 6 | P0.6 | INT4: External Interrupt 4. |
| 7 | P0.7 | INT5: External Interrupt 5. |

**STACK POINTER**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SP.7 | SP.6 | SP.5 | SP.4 | SP.3 | SP.2 | SP.1 | SP.0 |

Mnemonic: SP Address: 81h

The Stack Pointer stores the Scratch-pad RAM address where the stack begins. In other words it always points to the top of the stack.

**DATA POINTER LOW**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DPL.7 | DPL.6 | DPL.5 | DPL.4 | DPL.3 | DPL.2 | DPL.1 | DPL.0 |

Mnemonic: DPL Address: 82h

This is the low byte of the standard 8032 16-bit data pointer.

**DATA POINTER HIGH**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DPH.7 | DPH.6 | DPH.5 | DPH.4 | DPH.3 | DPH.2 | DPH.1 | DPH.0 |

Mnemonic: DPH Address: 83h

This is the high byte of the standard 8032 16-bit data pointer.

**TIMER 3 LSB**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TL3.7 | TL3.6 | TL3.5 | TL3.4 | TL3.3 | TL3.2 | TL3.1 | TL3.0 |

Mnemonic: TL3 Address: 84h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7-0 | Timer 3 LSB | LSB of Timer3 |

**TIMER 3 MSB**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TH3.7 | TH3.6 | TH3.5 | TH3.4 | TH3.3 | TH3.2 | TH3.1 | TH3.0 |

Mnemonic: TH3 Address: 85h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7-0 | Timer 3 MSB | MSB of Timer3 |

**POWER CONTROL**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SMOD | SMOD0 | - | - | GF1 | GF0 | PD | IDL |

Mnemonic: PCON Address: 87h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7 | SMOD | This bit doubles the serial port baud rate in mode 1, 2, and 3 when set to 1. |
| 6 | SMOD0 | Framing Error Detection Enable. When SMOD0 is set to 1, then SCON.7 (SCON1.7) now indicates a Frame Error and acts as the FE (FE_1) flag. When SMOD0 is 0, then SCON.7 (SCON1.7) acts as per the standard 8032 function. |
| 5-4 | - | Reserved. |
| 3-2 | GF1-0 | These two bits are general purpose user flags. |
| 1 | PD | Setting this bit causes the device to go into the POWERDOWN mode. In this mode all the clocks are stopped and program execution is frozen. |
| 0 | IDL | Setting this bit causes the device to go into the IDLE mode. In this mode the clock to the CPU is stopped, so program execution is frozen, but the clock to the serial ports, timer, PWM, ADC, SPI and interrupt blocks is not stopped, and these blocks continue operating unhindered. |

**TIMER CONTROL**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TF1 | TR1 | TF0 | TR0 | IE1 | IT1 | IE0 | IT0 |

Mnemonic: TCON Address: 88h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7 | TF1 | Timer 1 Overflow Flag. This bit is set when Timer 1 overflows. It is cleared automatically when the program does a timer 1 interrupt service routine. Software can also set or clear this bit. |
| 6 | TR1 | Timer 1 Run Control. This bit is set or cleared by software to turn timer/counter on or off. |
| 5 | TF0 | Timer 0 Overflow Flag. This bit is set when Timer 0 overflows. It is cleared automatically when the program does a timer 0 interrupt service routine. Software can also set or clear this bit. |
| 4 | TR0 | Timer 0 Run Control. This bit is set or cleared by software to turn timer/counter on or off. |
| 3 | IE1 | Interrupt 1 Edge Detect Flag: Set by hardware when an edge/level is detected on INT1. This bit is cleared by hardware when the service routine is vectored to only if the interrupt was edge triggered. Otherwise it follows the inverse of the pin. |
| 2 | IT1 | Interrupt 1 Type Control. Set/cleared by software to specify falling edge/ low level triggered external inputs. |
| 1 | IE0 | Interrupt 0 Edge Detect Flag. Set by hardware when an edge/level is detected on INT0. This bit is cleared by hardware when the service routine is vectored to only if the interrupt was edge triggered. Otherwise it follows the inverse of the pin. |
| 0 | IT0 | Interrupt 0 Type Control: Set/cleared by software to specify falling edge/ low level triggered external inputs. |

**TIMER MODE CONTROL**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | GATE | C/$\overline{T}$ | M1 | M0 | GATE | C/$\overline{T}$ | M1 | M0 |
| | TIMER1 | | | | TIMER0 | | | |

Mnemonic: TMOD Address: 89h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7 | GATE | Gating control: When this bit is set, Timer 1 is enabled only while the $\overline{INT1}$ pin is high and the TR1 control bit is set. When cleared, the $\overline{INT1}$ pin has no effect, and Timer 1 is enabled whenever TR1 is set. |
| 6 | C/$\overline{T}$ | Timer or Counter Select: When clear, Timer 1 is incremented by the internal clock. When set, the timer counts falling edges on the T1 pin. |
| 5 | M1 | Timer 1 mode select bit 1. See table below. |
| 4 | M0 | Timer 1 mode select bit 0. See table below. |
| 3 | GATE | Gating control: When this bit is set, Timer 0 is enabled only while the $\overline{INT0}$ pin is high and the TR0 control bit is set. When cleared, the $\overline{INT0}$ pin has no effect, and Timer 0 is enabled whenever TR0 is set. |
| 2 | C/$\overline{T}$ | Timer or Counter Select: When clear, Timer 0 is incremented by the internal clock. When set, the timer counts falling edges on the T0 pin. |
| 1 | M1 | Timer 0 mode select bit 1. See table below. |
| 0 | M0 | Timer 0 mode select bit 0. See table below. |

M1, M0: Mode Select bits:

| M1 | M0 | Mode |
|---|---|---|
| 0 | 0 | Mode 0: 8-bit timer/counter TLx serves as 5-bit pre-scale. |
| 0 | 1 | Mode 1: 16-bit timer/counter, no pre-scale. |
| 1 | 0 | Mode 2: 8-bit timer/counter with auto-reload from THx |
| 1 | 1 | Mode 3:<br>(Timer 0) TL0 is an 8-bit timer/counter controlled by the standard Timer-0 control bits. TH0 is an 8-bit timer only controlled by Timer-1 control bits.<br>(Timer 1) Timer/Counter 1 is stopped. |

**TIMER 0 LSB**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TL0.7 | TL0.6 | TL0.5 | TL0.4 | TL0.3 | TL0.2 | TL0.1 | TL0.0 |

Mnemonic: TL0 Address: 8Ah

TL0.7-0 Timer 0 LSB

**TIMER 1 LSB**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TL1.7 | TL1.6 | TL1.5 | TL1.4 | TL1.3 | TL1.2 | TL1.1 | TL1.0 |

Mnemonic: TL1 Address: 8Bh

TL1.7-0 Timer 1 LSB

**TIMER 0 MSB**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TH0.7 | TH0.6 | TH0.5 | TH0.4 | TH0.3 | TH0.2 | TH0.1 | TH0.0 |

Mnemonic: TH0 Address: 8Ch

TH0.7-0 Timer 0 MSB

**TIMER 1 MSB**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TH1.7 | TH1.6 | TH1.5 | TH1.4 | TH1.3 | TH1.2 | TH1.1 | TH1.0 |

Mnemonic: TH1 Address: 8Dh

TH1.7-0 Timer 1 MSB

**CLOCK CONTROL**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | WD1 | WD0 | T2M | T1M | T0M | MD2 | MD1 | MD0 |

Mnemonic: CKCON Address: 8Eh

| BIT | NAME | FUNCTION |
|---|---|---|
| 7 | WD1 | Watchdog Timer mode select bit 1. See table below. |
| 6 | WD0 | Watchdog Timer mode select bit 0. See table below. |
| 5 | T2M | Timer 2 clock select:<br>1: divide-by-4 clock.<br>0: divide-by-12 clock. |
| 4 | T1M | Timer 1 clock select:<br>1: divide-by-4 clock.<br>0: divide-by-12 clock. |
| 3 | T0M | Timer 0 clock select:<br>1: divide-by-4 clock.<br>0: divide-by-12 clock. |
| 2 | MD2 | Stretch MOVX select bit 2:<br>MD2, MD1, and MD0 select the stretch value for the MOVX instruction. The $\overline{RD}$ or $\overline{WR}$ strobe is stretched by the selected interval, which enables the device to access faster or slower external memory devices or peripherals without the need for external circuits. By default, the stretch value is one. See table below.<br>(**Note**: When accessing on-chip SRAM, these bits have no effect, and the MOVX instruction always takes two machine cycles.) |
| 1 | MD1 | Stretch MOVX select bit 1. See MD2. |
| 0 | MD0 | Stretch MOVX select bit 0. See MD2. |

**WD1, WD0: Mode Select bits:**

These bits determine the time-out periods for the Watchdog Timer. The reset time-out period is 512 clocks more than the interrupt time-out period.

| WD1 | WD0 | INTERRUPT TIME-OUT | RESET TIME-OUT |
|---|---|---|---|
| 0 | 0 | $2^{17}$ | $2^{17}$ + 512 |
| 0 | 1 | $2^{20}$ | $2^{20}$ + 512 |
| 1 | 0 | $2^{23}$ | $2^{23}$ + 512 |
| 1 | 1 | $2^{26}$ | $2^{26}$ + 512 |

**MD2, MD1, MD0: Stretch MOVX select bits:**

| MD2 | MD1 | MD0 | STRETCH VALUE | MOVX DURATION |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 2 machine cycles |
| 0 | 0 | 1 | 1 | 3 machine cycles *(Default)* |
| 0 | 1 | 0 | 2 | 4 machine cycles |
| 0 | 1 | 1 | 3 | 5 machine cycles |
| 1 | 0 | 0 | 4 | 6 machine cycles |
| 1 | 0 | 1 | 5 | 7 machine cycles |
| 1 | 1 | 0 | 6 | 8 machine cycles |
| 1 | 1 | 1 | 7 | 9 machine cycles |

**CLOCK CONTROL 1**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | - | CCDIV1 | CCDIV0 |

Mnemonic: CKCON1 Address: 8Fh

| BIT | NAME | FUNCTION |
|---|---|---|
| 7-2 | - | Reserved. |
| 1-0 | CCDIV | Timer 3 clock select. |

| CCDIV1 | CCDIV0 | Timer 3 clock |
|---|---|---|
| 0 | 0 | Fosc |
| 0 | 1 | Fosc/4 |
| 1 | 0 | Fosc/16 |
| 1 | 1 | Fosc/32 |

**PORT 1**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | P1.7 | P1.6 | P1.5 | P1.4 | P1.3 | P1.2 | P1.1 | P1.0 |

Mnemonic: P1 Address: 90h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7-0 | P1 | General purpose I/O port. Most instructions will read the port pins in case of a port read access, however in case of read-modify-write instructions, the port latch is read. Some pins also have alternate input or output functions. The alternate functions are described below. |

| | ALTERNATE FUNCTION1 | ALTERNATE FUNCTION2 |
|---|---|---|
| P1.0 | T2: External I/O for Timer/Counter 2 | ADC0: Analog input0 |
| P1.1 | PWM Brake | ADC1: Analog input1 |
| P1.2 | RXD1 | ADC2: Analog input2 |
| P1.3 | TXD1 | ADC3: Analog input3 |
| P1.4 | | ADC4: Analog input4 |
| P1.5 | | ADC5: Analog input5 |
| P1.6 | | ADC6: Analog input6 |
| P1.7 | | ADC7: Analog input7 |

**EXTERNAL INTERRUPT FLAG**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | IE5 | IE4 | IE3 | IE2 | - | - | - | - |

Mnemonic: EXIF Address: 91h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7 | IE5 | External Interrupt 5 flag. Set by hardware when a rising/falling/both edges is detected onINT5 pin. |
| 6 | IE4 | External Interrupt 4 flag. Set by hardware when a rising/falling/both edges is detected on INT4 pin. |
| 5 | IE3 | External Interrupt 3 flag. Set by hardware when a rising/falling/both edges is detected on INT3 pin. |
| 4 | IE2 | External Interrupt 2 flag. Set by hardware when a rising edge is detected on INT2 pin. |
| 3-0 | - | Reserved. |

**PORT 4 CONTROL REGISTER A**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | P41FUN1 | P41FUN0 | P41CMP1 | P41CMP0 | P40FUN1 | P40FUN0 | P40CMP1 | P40CMP0 |

Mnemonic: P4CONA Address: 92h

### PORT 4 CONTROL REGISTER B

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | P43FUN1 | P43FUN0 | P43CMP1 | P43CMP0 | P42FUN1 | P42FUN0 | P42CMP1 | P42CMP0 |

Mnemonic: P4CONB Address: 93h

| BIT NAME | FUNCTION |
|---|---|
| P4xFUN1, P4xFUN0 | Port 4 alternate modes.<br>=00: Mode 0. P4.x is a general purpose I/O port which is the same as Port 1.<br>=01: Mode 1. P4.x is a Read Strobe signal for chip select purpose. The address range depends on the SFR P4xAH, P4xAL and bits P4xCMP1, P4xCMP0.<br>=10: Mode 2. P4.x is a Write Strobe signal for chip select purpose. The address range depends on the SFR P4xAH, P4xAL and bits P4xCMP1, P4xCMP0.<br>=11: Mode 3. P4.x is a Read/Write Strobe signal for chip select purpose. The address range depends on the SFR P4xAH, P4xAL and bits P4xCMP1, P4xCMP0. |
| P4xCMP1, P4xCMP0 | Port 4 Chip-select Mode address comparison:<br>=00: Compare the full address (16 bits length) with the base address registers P4xAH and P4xAL.<br>=01: Compare the 15 high bits (A15-A1) of address bus with the base address registers P4xAH and P4xAL.<br>=10: Compare the 14 high bits (A15-A2) of address bus with the base address registers P4xAH and P4xAL.<br>=11: Compare the 8 high bits (A15-A8) of address bus with the base address registers P4xAH and P4xAL. |

### P4.0 BASE ADDRESS LOW BYTE REGISTER

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |

Mnemonic: P40AL Address: 94h

### P4.0 BASE ADDRESS HIGH BYTE REGISTER

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 |

Mnemonic: P40AH Address: 95h

### P4.1 BASE ADDRESS LOW BYTE REGISTER

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |

Mnemonic: P41AL Address: 96h

### P4.1 BASE ADDRESS HIGH BYTE REGISTER

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 |

Mnemonic: P41AH Address: 97h

### SERIAL PORT CONTROL

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SM0/FE | SM1 | SM2 | REN | TB8 | RB8 | TI | RI |

Mnemonic: SCON Address: 98h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7 | SM0/FE | Serial Port mode select bit 0 or Framing Error Flag: This bit is controlled by the SMOD0 bit in the PCON register.<br>(SM0) See table below.<br>(FE) This bit indicates an invalid stop bit. It must be manually cleared by software. |
| 6 | SM1 | Serial Port mode select bit 1. See table below. |
| 5 | SM2 | Serial Port Clock or Multi-Processor Communication.<br>(Mode 0) This bit controls the serial port clock. If set to zero, the serial port runs at a divide-by-12 clock of the oscillator. This is compatible with the standard 8051/52. If set to one, the serial clock is a divide-by-4 clock of the oscillator.<br>(Mode 1) If SM2 is set to one, RI is not activated if a valid stop bit is not received.<br>(Modes 2 / 3) This bit enables multi-processor communication. If SM2 is set to one, RI is not activated if RB8, the ninth data bit, is zero. |
| 4 | REN | Receive enable:<br>1: Enable serial reception.<br>0: Disable serial reception. |
| 3 | TB8 | (Modes 2 / 3) This is the 9th bit to transmit. This bit is set by software. |
| 2 | RB8 | (Mode 0) No function.<br>(Mode 1) If SM2 = 0, RB8 is the stop bit that was received.<br>(Modes 2 / 3) This is the 9th bit that was received. |
| 1 | TI | Transmit interrupt flag: This flag is set by the hardware at the end of the 8th bit in mode 0 or at the beginning of the stop bit in the other modes during serial transmission. This bit must be cleared by software. |
| 0 | RI | Receive interrupt flag: This flag is set by the hardware at the end of the 8th bit in mode 0 or halfway through the stop bits in the other modes during serial reception. However, SM2 can restrict this behavior. This bit can only be cleared by software. |

**SM1, SM0: Mode Select bits:**

| SM0 | SM1 | MODE | DESCRIPTION | LENGTH | BAUD RATE |
|---|---|---|---|---|---|
| 0 | 0 | 0 | Synchronous | 8 | Tclk divided by 4 or 12 |
| 0 | 1 | 1 | Asynchronous | 10 | Variable |
| 1 | 0 | 2 | Asynchronous | 11 | Tclk divided by 32 or 64 |
| 1 | 1 | 3 | Asynchronous | 11 | Variable |

**ERIAL DATA BUFFER**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SBUF.7 | SBUF.6 | SBUF.5 | SBUF.4 | SBUF.3 | SBUF.2 | SBUF.1 | SBUF.0 |

Mnemonic: SBUF Address: 99h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7-0 | SBUF | Serial data is read from or written to this location. It consists of two separate 8 bit registers. One is the receive buffer, and the other is the transmit buffer. Any read access gets data from the receive data buffer, while write access is to the transmit data buffer. |

**P4.2 BASE ADDRESS LOW BYTE REGISTER**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |

Mnemonic: P42AL Address: 9Ah

**P4.2 BASE ADDRESS HIGH BYTE REGISTER**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 |

Mnemonic: P42AH Address: 9Bh

**P4.3 BASE ADDRESS LOW BYTE REGISTER**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |

Mnemonic: P43AL Address: 9Ch

**P4.3 BASE ADDRESS HIGH BYTE REGISTER**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 |

Mnemonic: P43AH Address: 9Dh

**NVM CONTROL**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | EER | EWR | EnNVM | - | - | - | - | NVMF |

Mnemonic: NVMCON Address: 9Eh

| BIT | NAME | FUNCTION |
|---|---|---|
| 7 | EER | Set this bit to erase NVM data of page (n) to FFH.<br>The NVM has 32 pages that each page has 64 bytes data memory. By select NVMADDRH and NVMADDRL of NVM address registers that will automatic enable page area. If set this bit, the page will be page erased, after finished, the NVMF flag will be set to “1”, then this bit will be cleared. If NVMF flag is set, the erase and write NVM data memory are invalid. |

Continued

| BIT | NAME | FUNCTION |
|---|---|---|
| 6 | EWR | Set this bit is write data to NVM data memory by NVMADDRH and NVMADDRL to decode NVM data memory. If finished, NVMF flag will be set to "1", and then this bit will be cleared. If NVMF flag is set, the erase and write NVM are invalid. |
| 5 | EnNVM | To enable read NVM data memory area, refer as below table.<br>0: To disable the MOVX instruction to read NVM data memory.<br>1: To enable the MOVX instruction to read NVM data memory, the External RAM or AUX-RAM will be disabled. |
| 4~1 | - | Reserved. |
| 0 | NVMF | NVM data memory erases or writes finished flag.<br>If NVM data memory is finished by erase or write, it will be set to "1" by hardware and clear by software. And it will be interrupted when NVM erase/write interrupt is enabled. |

**ISP CONTROL REGISTER**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SWRST/ HWB | - | LD/AP | - | - | - | LDSEL | ENP |

Mnemonic: CHPCON Address: 9Fh

| BIT | NAME | FUNCTION |
|---|---|---|
| 7 | W:SWRST<br>R:HWB | Write access to this bit is different from read access.<br>Write this bit to 1 to force the microcontroller to reset to the initial condition, just like power-on reset. This action re-boots the microcontroller and starts normal operation. This bit will be cleared during the reset.<br>Read this bit to determine whether or not a hardware reboot is in progress. If CPU is rebooted by P3.6 & P3.7 or P4.3, this bit is set to 1 after the hardware reboot.<br>**Note**: P4.3 pin is available in 48L LQFP package only. |
| 6 | - | Reserved. |
| 5 | LD/AP<br>(read-only) | 0: CPU is executing AP Flash EPROM<br>1: CPU is executing LD Flash EPROM |
| 4-2 | - | Reserved. |
| 1 | LDSEL<br>(write-only) | Loader Program Location Selection. This bit should be set before entering ISP mode.<br>0: The executing program is in the 64-KB AP Flash EPROM. The 4-KB LD Flash EPROM is the destination for re-programming.<br>1: The executing program is in the 4-KB memory bank. The 64-KB AP Flash EPROM is the destination for re-programming. |
| 0 | ENP | FLASH EPROM Programming Enable.<br>1: Enable in-system programming mode. In this mode, erase, program and read operations are achieved.<br>0: Disable in-system programming mode. The on-chip flash memory is read-only. |

The way to enter ISP mode is to set ENP to 1 and write LDSEL properly then force CPU in IDLE mode, after IDLE mode is released CPU will restart from AP or LD ROM according the value of LDSEL.

**PORT 2**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | P2.7 | P2.6 | P2.5 | P2.4 | P2.3 | P2.2 | P2.1 | P2.0 |

Mnemonic: P2 Address: A0h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7-0 | P2 | This port functions as an address bus during external memory access, and as a general-purpose I/O port on devices that incorporate internal program memory. When P2 functions a non-multiplexed address bus A15-A8 the port latch cannot be used for general I/O purposes but exists to support the MOVX instructions. Port 2 data will only be brought out on the P2.7-0 pins during indirect MOVX instructions. |

| | ALTERNATE FUNCTION |
|---|---|
| P2.0 | PWM0 output. |
| P2.1 | PWM1 output. |
| P2.2 | PWM2 output. |
| P2.3 | PWM3 output. |
| P2.4 | PWM4 output. |
| P2.5 | PWM5 output. |
| P2.6 | SCL, I2C serial clock. |
| P2.7 | SDA, I2C serial data. |

**XRAMAH**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | A10 | A9 | A8 |

Mnemonic: XRAMAH Address: A1h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7-3 | - | Reserved. |
| 2-0 | A10-8 | XRAMAH is used for high byte address memory access through A15-8, when CPU executes MOVX with R0 (or R1) instructions. Depending EnNVM and DME0 setting, and address, the memory accessed may differs. Table below shows the memory access destination.<br><br>This device has on-chip sram at 1/2/2K bytes. |

**Note: User should take care when accessing the memory with this instruction. Access to invalid regions may cause undesirable results.**

### PORT 4 CHIP-SELECT POLARITY

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | P43INV | P42INV | P41INV | P40INV | - | PWDNH | RMWFP | PUP0 |

Mnemonic: P4CSIN Address: A2h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7-4 | P4xINV | The Active Polarity of P4.x when it is set as a chip-select strobe output. High = Active High. Low = Active Low. **Note:** x = 3,2,1,0. |
| 3 | - | Reserved. |
| 2 | PWDNH | Set PWDNH to logic 1 then ALE and PSEN will keep high state, clear this bit to logic 0 then ALE and PSEN will output low during power down mode. |
| 1 | RMWFP | Control Read Path of Instruction "Read-Modify-Write". When this bit is set, the read path of executing "read-modify-write" instruction is from port pin otherwise from SFR. |
| 0 | PUP0 | Enable Port 0 weak pull up. |

### CAPTURE CONTROL 0 REGISTER

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CCT2.1 | CCT2.0 | CCT1.1 | CCT1.0 | CCT0.1 | CCT0.0 | CCLD1 | CCLD0 |

Mnemonic: CAPCON0 Address: A3h

<table>
<tr><th>BIT</th><th>NAME</th><th>FUNCTION</th></tr>
<tr><td>7-6</td><td>CCT2.1-0</td><td>Capture 2 edge select.
<table>
<tr><th>CCT2.1</th><th>CCT2.0</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>Rising edge trigger</td></tr>
<tr><td>0</td><td>1</td><td>Falling edge trigger</td></tr>
<tr><td>1</td><td>0</td><td>Rising and falling edge trigger</td></tr>
<tr><td>1</td><td>1</td><td>Reserved.</td></tr>
</table></td></tr>
<tr><td>5-4</td><td>CCT1.1-0</td><td>Capture 1 edge select.
<table>
<tr><th>CCT1.1</th><th>CCT1.0</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>Rising edge trigger</td></tr>
<tr><td>0</td><td>1</td><td>Falling edge trigger</td></tr>
<tr><td>1</td><td>0</td><td>Rising and falling edge trigger</td></tr>
<tr><td>1</td><td>1</td><td>Reserved.</td></tr>
</table></td></tr>
<tr><td>3-2</td><td>CCT0.1-0</td><td>Capture 0 edge select.
<table>
<tr><th>CCT0.1</th><th>CCT0.0</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>Rising edge trigger</td></tr>
<tr><td>0</td><td>1</td><td>Falling edge trigger</td></tr>
<tr><td>1</td><td>0</td><td>Rising and falling edge trigger</td></tr>
<tr><td>1</td><td>1</td><td>Reserved.</td></tr>
</table></td></tr>
</table>

Continued

<table>
<tr><th>BIT</th><th>NAME</th><th>FUNCTION</th></tr>
<tr><td>1-0</td><td>CCLD.1-0</td><td>Reload trigger select.
<table>
<tr><th>CCLD1</th><th>CCLD0</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>Timer 3 overflow (default)</td></tr>
<tr><td>0</td><td>1</td><td>Reload by capture 0 block</td></tr>
<tr><td>1</td><td>0</td><td>Reload by capture 1 block</td></tr>
<tr><td>1</td><td>1</td><td>Reload by capture 2 block</td></tr>
</table></td></tr>
</table>

**CAPTURE CONTROL 1 REGISTER**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | - | ENF2 | ENF1 | ENF0 | CPTF2 | CPTF1/ | CPTF0 |

Mnemonic: CAPCON1 Address: A4h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7-6 | - | Reserved. |
| 5 | ENF2 | Enable filter for capture input 2. |
| 4 | ENF1 | Enable filter for capture input 1. |
| 3 | ENF0 | Enable filter for capture input 0. |
| 2 | CPTF2 | Input capture/reload 2 interrupt flag. |
| 1 | CPTF1/DIRF | Input Capture 2 flag share the same bit with DIRF flag.<br>**IC mode** - Input capture/reload 1 interrupt flag.<br>**QEI mode** - Direction changed interrupt flag. Bit is set by hardware when direction index (DIR) changes state and direction change interrupt is requested if it is enabled. DIRF is cleared by software. |
| 0 | CPTF0/QEIF | Input Capture 0 flag share the same bit with QEI flag.<br>**IC mode** – Input capture/reload 0 interrupt flag.<br>**QEI mode** - QEI interrupt flag.<br>1. In free-counting mode, if Pulse Counter overflows or underflows.<br>2. In compare-counting mode, if Pulse Counter overflows from Maximum Count to zero or underflows from zero to Maximum Count. |

**PORT 4**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | P4.3 | P4.2 | P4.1 | P4.0 |

Mnemonic: P4 Address: A5h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7-4 | - | Reserved. |
| 3-2 | P4 | GPIO. |
| 1 | P4 | GPIO. Alternate function T2EX/IC2 for Timer 2 external trigger/Input Capture 2 respectively. |
| 0 | P4 | GPIO. Alternate function STADC. External start ADC trigger input. |

### INPUT CAPTURE 2/MAXIMUM COUNTER LOW REGISTER

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CCL2.7/ MAXCNT L.7 | CCL2.6/ MAXCNT L.6 | CCL2.5/ MAXCNT L.5 | CCL2.4/ MAXCNT L.4 | CCL2.3/ MAXCNT L.3 | CCL2.2/ MAXCNT L.2 | CCL2.1/ MAXCNT L.1 | CCL2.0/ MAXCNT L.0 |

Mnemonic: CCL2/MAXCNTL Address: A6h

### INPUT CAPTURE 2/MAXIMUM COUNTER HIGH REGISTER

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CCH2.7/ MAXCNT H.7 | CCH2.6/ MAXCNT H.6 | CCH2.5/ MAXCNT H.5 | CCH2.4/ MAXCNT H.4 | CCH2.3/ MAXCNT H.3 | CCH2.2/ MAXCNT H.2 | CCH2.1/ MAXCNT H.1 | CCH2.0/ MAXCNT H.0 |

Mnemonic: CCH2/MAXCNTH Address: A7h

### INTERRUPT ENABLE

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | EA | EADC | ET2 | ES | ET1 | EX1 | ET0 | EX0 |

Mnemonic: IE Address: A8h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7 | EA | Global enable: Enable/disable all interrupts. |
| 6 | EADC | Enable ADC interrupt. |
| 5 | ET2 | Enable Timer 2 interrupt. |
| 4 | ES | Enable Serial Port 0 interrupts. |
| 3 | ET1 | Enable Timer 1 interrupt. |
| 2 | EX1 | Enable external interrupt 1. |
| 1 | ET0 | Enable Timer 0 interrupt. |
| 0 | EX0 | Enable external interrupt 0. |

### SLAVE ADDRESS

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SADDR.7 | SADDR.6 | SADDR.5 | SADDR.4 | SADDR.3 | SADDR.2 | SADDR.1 | SADDR.0 |

Mnemonic: SADDR Address: A9h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7-0 | SADDR | The SADDR should be programmed to the given or broadcast address for serial port to which the slave processor is designated. |

### SLAVE ADDRESS 1

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SADDR1.7 | SADDR1.6 | SADDR1.5 | SADDR1.4 | SADDR1.3 | SADDR1.2 | SADDR1.1 | SADDR1.0 |

Mnemonic: SADDR1 Address: AAh

| BIT | NAME | FUNCTION |
|---|---|---|
| 7-0 | SADDR1 | The SADDR1 should be programmed to the given or broadcast address for serial port 1 to which the slave processor is designated. |

**ISP ADDRESS LOW BYTE**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |

Mnemonic: SFRAL Address: ACh

Low byte destination address for In System Programming operations.

**ISP ADDRESS HIGH BYTE**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 |

Mnemonic: SFRAH Address: ADh

Low byte destination address for In System Programming operations. (SFRAH, SFRAL) represents the address of the ROM byte that will be erased, programmed or read.

**ISP DATA BUFFER**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |

Mnemonic: SFRFD Address: AEh

In ISP mode, read/write a specific byte ROM content must go through SFRFD register.

**ISP OPERATION MODES**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | WFWIN | NOE | NCE | CTRL3 | CTRL2 | CTRL1 | CTRL0 |

Mnemonic: SFRCN Address: AFh

| BIT | NAME | FUNCTION |
|---|---|---|
| 7 | - | Reserved. |
| 6 | WFWIN | On-chip FLASH EPROM bank select for in-system programming.<br>0= AP FLASH EPROM bank is selected as destination for re-programming.<br>1= LD FLASH EPROM bank is selected as destination for re-programming. |
| 5 | NOE | Flash EPROM output enable. |
| 4 | NCE | Flash EPROM chip enable. |
| 3-0 | CTRL | The Flash Control Signals. |

| ISP MODE | WFWIN | NOE | NCE | CTRL[3:0] | SFRAH, SFRAL | SFRFD |
|---|---|---|---|---|---|---|
| Erase 4KB LD Flash | 1 | 1 | 0 | 0010 | X | X |
| Erase 16/32/64K AP Flash | 0 | 1 | 0 | 0010 | X | X |
| Program 4KB LD Flash | 1 | 1 | 0 | 0001 | Address in | Data in |
| Program 16/32/64KB AP Flash | 0 | 1 | 0 | 0001 | Address in | Data in |
| Read 4KB LD Flash | 1 | 0 | 0 | 0000 | Address in | Data out |
| Read 16/32/64KB AP Flash | 0 | 0 | 0 | 0000 | Address in | Data out |

**PORT 3**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | P3.7 | P3.6 | P3.5 | P3.4 | P3.3 | P3.2 | P3.1 | P3.0 |

Mnemonic: P3 Address: B0h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7-0 | P3 | General purpose I/O port. Each pin also has an alternate input or output function that is controlled by other SFRs. The alternate function is enabled if the corresponding port latch bit is set to 1. |

| | ALTERNATE FUNCTION |
|---|---|
| P3.7 | RD Strobe for read from external RAM. |
| P3.6 | WR Strobe for write to external RAM. |
| P3.5 | T1/IC1/QEB; Timer/counter 1 external count input/Input Capture 1/QEI input B. |
| P3.4 | T0/IC0/QEA; Timer/counter 0 external count input/Input Capture 0/QEI input A. |
| P3.3 | /INT0 External interrupt 1. |
| P3.2 | /INT1 External interrupt 0. |
| P3.1 | TxD Serial port output. |
| P3.0 | RxD Serial port input. |

**PORT 5**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | - | P5.1 | P5.0 |

Mnemonic: P5 Address: B1h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7-2 | - | Reserved. |
| 1-0 | P5 | General purpose I/O port. Each pin also has an alternate input or output function. This port can not support bit addressable. |

| | ALTERNATE FUNCTION |
|---|---|
| P5.1 | PWM7 output function |
| P5.0 | PWM6 output function |

**TIMER 3 RELOAD LSB**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RCAP3L.7 | RCAP3L.6 | RCAP3L.5 | RCAP3L.4 | RCAP3L.3 | RCAP3L.2 | RCAP3L.1 | RCAP3L.0 |

Mnemonic: RCAP3L Address: B4h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7-0 | RCAP3L | Timer 3 Reload LSB: This register is LSB of a 16 bit reload value when timer 3 is configured in reload mode. It served also as a compare register when timer 3 is configured as compare mode (see CMP/RL3 bit). |

**TIMER 3 RELOAD MSB**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RCAP3H.7 | RCAP3H.6 | RCAP3H.5 | RCAP3H.4 | RCAP3H.3 | RCAP3H.2 | RCAP3H.1 | RCAP3H.0 |

Mnemonic: RCAP3H Address: B5h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7-0 | RCAP3H | Timer 3 Reload MSB: This register is MSB of a 16 bit reload value when timer 3 is configured in reload mode. It served also as a compare register when timer 3 is configured as compare mode (see CMP/RL3 bit). |

**EXTENDED INTERRUPT HIGH PRIORITY 1**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | - | PNVMIH | PCPTFH | PT3H | PBKFH | PPWMFH | PSPIH |

Mnemonic: EIP1 Address: B6h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7-6 | - | Reserved. |
| 5 | PNVMIH | NVM interrupt High priority. PNVMIH = 1 sets it to highest priority level. |
| 4 | PCPTFH | Capture/reload Interrupt High priority. PCPTFH = 1 sets it to highest priority level. |
| 3 | PT3H | Timer 3 Interrupt High priority. PT3H = 1 sets it to highest priority level. |
| 2 | PBKFH | PWM Brake Interrupt High priority. PBKFH = 1 sets it to highest priority level. |
| 1 | PPWMFH | PWM period Interrupt High priority. PPWMFH = 1 sets it to highest priority level. |
| 0 | PSPIH | SPI Interrupt High Priority. PSPIH = 1 sets it to highest priority level. |

**INTERRUPT HIGH PRIORITY**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | PADCH | PT2H | PSHH | PT1H | PX1H | PT0H | PX0H |

Mnemonic: IPH Address: B7h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7 | - | Reserved. |
| 6 | PADCH | This bit defines the ADC interrupt High priority. PADCH = 1 sets it to highest priority level. |
| 5 | PT2H | This bit defines the Timer 2 interrupt High priority. PT2H = 1 sets it to highest priority level. |
| 4 | PSH | This bit defines the Serial port 0 interrupt High priority. PSH = 1 sets it to highest priority level. |
| 3 | PT1H | This bit defines the Timer 1 interrupt High priority. PT1H = 1 sets it to highest priority level. |
| 2 | PX1H | This bit defines the External interrupt 1 High priority. PX1H = 1 sets it to highest priority level. |
| 1 | PT0H | This bit defines the Timer 0 interrupt High priority. PT0H = 1 sets it to highest priority level. |
| 0 | PX0H | This bit defines the External interrupt 0 High priority. PX0H = 1 sets it to highest priority level. |

**INTERRUPT PRIORITY**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | PADC | PT2 | PS | PT1 | PX1 | PT0 | PX0 |

Mnemonic: IP Address: B8h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7 | - | Reserved. |
| 6 | PADC | This bit defines the ADC interrupt priority. PADC = 1 sets it to higher priority level. |
| 5 | PT2 | This bit defines the Timer 2 interrupt priority. PT2 = 1 sets it to higher priority level. |
| 4 | PS | This bit defines the Serial port 0 interrupt priority. PS = 1 sets it to higher priority level. |
| 3 | PT1 | This bit defines the Timer 1 interrupt priority. PT1 = 1 sets it to higher priority level. |
| 2 | PX1 | This bit defines the External interrupt 1 priority. PX1 = 1 sets it to higher priority level. |
| 1 | PT0 | This bit defines the Timer 0 interrupt priority. PT0 = 1 sets it to higher priority level. |
| 0 | PX0 | This bit defines the External interrupt 0 priority. PX0 = 1 sets it to higher priority level. |

**SLAVE ADDRESS MASK ENABLE**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SADEN.7 | SADEN.6 | SADEN.5 | SADEN.4 | SADEN.3 | SADEN.2 | SADEN.1 | SADEN.0 |

Mnemonic: SADEN Address: B9h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7-0 | SADEN | This register enables the Automatic Address Recognition feature of the Serial port. When a bit in the SADEN is set to 1, the same bit location in SADDR will be compared with the incoming serial port data. When SADEN.n is 0, then the bit becomes don't care in the comparison. This register enables the Automatic Address Recognition feature of the Serial port. When all the bits of SADEN are 0, interrupt will occur for any incoming address. |

**SLAVE ADDRESS MASK ENABLE 1**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SADEN1.7 | SADEN1.6 | SADEN1.5 | SADEN1.4 | SADEN1.3 | SADEN1.2 | SADEN1.1 | SADEN1.0 |

Mnemonic: SADEN1 Address: BAh

| BIT | NAME | FUNCTION |
|---|---|---|
| 7-0 | SADEN1 | This register enables the Automatic Address Recognition feature of the Serial port 1. When a bit in the SADEN1 is set to 1, the same bit location in SADDR1 will be compared with the incoming serial port data. When SADEN1.n is 0, then the bit becomes don't care in the comparison. This register enables the Automatic Address Recognition feature of the Serial port. When all the bits of SADEN1 are 0, interrupt will occur for any incoming address. |

**PWM OUTPUT OVERRIDE CONTROL REGISTERS**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | POVM.7 | POVM.6 | POVM.5 | POVM.4 | POVM.3 | POVM.2 | POVM.1 | POVM.0 |

Mnemonic: POVM Address: BBh

| BIT | NAME | FUNCTION |
|---|---|---|
| 7-0 | POVM | PWM Override Mode enable bits;<br>0: The PWM output follows the corresponding PWM generator.<br>1: The PWM output is equal to corresponding bit in POVD. |

**PWM OUTPUT STATE REGISTERS**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | POVD.7 | POVD.6 | POVD.5 | POVD.4 | POVD.3 | POVD.2 | POVD.1 | POVD.0 |

Mnemonic: POVD Address: BCh

| BIT | NAME | FUNCTION |
|---|---|---|
| 7-0 | POVD | PWM Override Data represents the value of PWM[7:0] respectively in override mode.<br>1 = Output on PWM I/O pin is ACTIVE when the corresponding PWM output override bit is cleared.<br>0 = Output on PWM I/O pin is INACTIVE when the corresponding PWM output override bit is cleared. |

**PWM PIN OUTPUT SOURCE SELECT**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PIO7 | PIO6 | PIO5 | PIO4 | PIO3 | PIO2 | PIO1 | PIO0 |

Mnemonic: PIO Address: BDh

| BIT | NAME | FUNCTION |
|---|---|---|
| 7-0 | PIO.x | Select pin output source from PWM or I/O register; x=0~7; PIOn is effective only when option bit PWMOE/PWMEE/PWM6E/PWM7E is in enabled status. Reset value=0;<br>1 = Correspondent I/O pin with high source/sink current.<br>0 = PWMn output; n=0~7 with high source/sink current. |

**PWM OUTPUT ENABLE REGISTER**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PWM7EN | PWM6EN | PWM5EN | PWM4EN | PWM3EN | PWM2EN | PWM1EN | PWM0EN |

Mnemonic: PWMEN Address: BEh

| BIT | NAME | FUNCTION |
|---|---|---|
| 6,4,2,0 | PWMeEN | Set high to enable even PWM output; e = 0,2,4,6; Reset value=0;<br>1 = Enable PWM output.<br>0 = Disable PWM output. |
| 7,5,3,1 | PWMoEN | Set high to enable odd PWM output; o = 1,3,5,7; Reset value=0;<br>1 = Enable PWM output.<br>0 = Disable PWM output. |

**PWM 4 HIGH BITS REGISTER**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | PWM4.11 | PWM4.10 | PWM4.9 | PWM4.8 |

Mnemonic: PWM4H Address: BFh

| BIT | NAME | FUNCTION |
|---|---|---|
| 7~4 | - | Reserved |
| 3~0 | PWM4.11 ~PWM4.8 | The PWM 4 Register bit 11~8. |

**SERIAL PORT CONTROL 1**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SM0_1/FE_1 | SM1_1 | SM2_1 | REN_1 | TB8_1 | RB8_1 | TI_1 | RI_1 |

Mnemonic: SCON1 Address: C0h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7 | SM0_1/ FE_1 | Serial Port 1 mode select bit 0 or Framing Error Flag: This bit is controlled by the SMOD0 bit in the PCON register.<br>(SM0) See table below.<br>(FE) This bit indicates an invalid stop bit. It must be manually cleared by software. |
| 6 | SM1_1 | Serial Port 1 mode select bit 1. See table below. |
| 5 | SM2_1 | Serial Port Clock or Multi-Processor Communication.<br>(Mode 0) This bit controls the serial port clock. If set to zero, the serial port runs at a divide-by-12 clock of the oscillator. This is compatible with the standard 8051/52. If set to one, the serial clock is a divide-by-4 clock of the oscillator.<br>(Mode 1) If SM2_1 is set to one, RI_1 is not activated if a valid stop bit is not received.<br>(Modes 2 / 3) This bit enables multi-processor communication. If SM2_1 is set to one, RI_1 is not activated if RB8_1, the ninth data bit, is zero. |
| 4 | REN_1 | Receive enable:<br>1: Enable serial reception.<br>0: Disable serial reception. |
| 3 | TB8_1 | (Modes 2 / 3) This is the 9th bit to transmit. This bit is set by software. |
| 2 | RB8_1 | (Mode 0) No function.<br>(Mode 1) If SM2_1 = 0, RB8_1 is the stop bit that was received.<br>(Modes 2 / 3) This is the 9th bit that was received. |
| 1 | TI_1 | Transmit interrupt flag: This flag is set by the hardware at the end of the 8th bit in mode 0 or at the beginning of the stop bit in the other modes during serial transmission. This bit must be cleared by software. |
| 0 | RI_1 | Receive interrupt flag: This flag is set by the hardware at the end of the 8th bit in mode 0 or halfway through the stop bits in the other modes during serial reception. However, SM2_1 can restrict this behavior. This bit can only be cleared by software. |

**SM1_1, SM0_1: Mode Select bits:**

| SM0_1 | SM1_1 | MODE | DESCRIPTION | LENGTH | BAUD RATE |
|---|---|---|---|---|---|
| 0 | 0 | 0 | Synchronous | 8 | Tclk divided by 4 or 12 |
| 0 | 1 | 1 | Asynchronous | 10 | Variable |
| 1 | 0 | 2 | Asynchronous | 11 | Tclk divided by 32 or 64 |
| 1 | 1 | 3 | Asynchronous | 11 | Variable |

**SERIAL DATA BUFFER 1**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SBUF_1.7 | SBUF_1.6 | SBUF_1.5 | SBUF_1.4 | SBUF_1.3 | SBUF_1.2 | SBUF_1.1 | SBUF_1.0 |

Mnemonic: SBUF1 Address: C1h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7-0 | SBUF_1 | For Serial Port 1. Serial data is read from or written to this location. It actually consists of two separate 8 bit registers. One is the receive buffer, and the other is the transmit buffer. Any read access gets data from the receive data buffer, while write access is to the transmit data buffer. |

**TIMER 3 MODE CONTROL**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ENLD | ICEN2 | ICEN1 | ICEN0 | T3CR | - | - | - |

Mnemonic: T3MOD Address: C2h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7 | ENLD | Enable reloads from RCAP3 registers to timer 3 counters. |
| 6 | ICEN2 | Capture 2 External Enable. This bit enables the capture/reload function on the IC2 pin. An edge trigger (programmable by CAPCON0.CCT2[1:0] bits) detected on the IC2 pin will result in capture from free running timer 3 counters to input capture 2 registers, or reload from RCAP3 registers to timer 3 counters.<br>1 = Enable.<br>0 = Disable. |
| 5 | ICEN1 | Capture 1 External Enable. This bit enables the capture/reload function on the IC1 pin. An edge trigger (programmable by CAPCON0.CCT1[1:0] bits) detected on the IC1 pin will result in capture from free running timer 3 counters to input capture 1 registers, or reload from RCAP3 registers to timer 3 counters.<br>1 = Enable.<br>0 = Disable. |
| 4 | ICEN0 | Capture 0 External Enable. This bit enables the capture/reload function on the IC0 pin. An edge trigger (programmable by CAPCON0.CCT0[1:0] bits) detected on the IC0 pin will result input capture from free running timer 3 counters to input capture 0 registers, or reload from RCAP3 registers to timer 3 counters.<br>1 = Enable.<br>0 = Disable. |
| 3 | T3CR | Timer 3 Capture Reset. In the Timer 3 Capture Mode this bit enables/disables hardware automatically reset timer 3 while the value in TL3 and TH3 have been transferred into the input capture register (CCLx, CCHx). Priority is given to T3CR to reset counter after capture. |
| 2-0 | - | Reserved. |

### TIMER 3 CONTROL

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TF3 | - | - | - | - | TR3 | - | CMP/$\overline{RL3}$ |

Mnemonic: T3CON Address: C3h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7 | TF3 | Timer 3 overflows flag. This bit is set when Timer 3 overflows. It is cleared only by software and set by hardware. |
| 6-3 | - | Reserved. |
| 2 | TR3 | Timer 3 Run Control. This bit enables/disables the operation of timer 3. Halting this will preserve the current count in TH3, TL3. |
| 1 | - | Reserved. |
| 0 | CMP/$\overline{RL3}$ | Compare/Reload Select. This bit determines whether the Timer 3 will be use for compare or reload function.<br>0 = Timer 3 as reload mode, TF3 indicates the overflow flag<br>1 = Timer 3 as compare mode, TF3 indicates the compare match flag. |

### POWER MANAGEMENT REGISTER

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | ALEOFF | - | DME0 |

Mnemonic: PMR Address: C4h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7-3 | - | Reserved. |
| 2 | ALEOFF | This bit disables the expression of the ALE signal on the device pin during all on board program and data memory accesses. External memory accesses will automatically enable ALE independent of ALEOFF. ALEOFF=0: ALE expression is enabled. ALEOFF=1: ALE expression is disabled. |
| 1 | - | Reserved. |
| 0 | DME0 | This bit determines the on chip MOVX SRAM to be enabled or disabled. Set this bit to 1 will enable the on chip 2 KB MOVX SRAM. |

### FAULT SAMPLING TIME REGISTER

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SCMP1 | SCMP0 | SFP1 | SFP0 | SFCEN | SFCST | SFCDIR | LSBD |

Mnemonic: FSPLT Address: C5h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7-6 | SCMP [1:0] | Smart fault compare value selector (read/write):<br>00 = 4<br>01 = 16<br>10 = 64<br>11 = 128 |
| 5-4 | SFP[1:0] | Smart fault sampling frequency selector (read/write):<br>00 = FOSC/4<br>01 = FOSC/8<br>10 = FOSC/16<br>11 = FOSC/128 |
| 3 | SFCEN | Smart fault/brake counter enable (read/write):<br>0 = Disable, and clear internal smart fault counter.<br>1 = Enable smart fault detector. |
| 2 | SFCST | Smart fault/brake counter status (read only):<br>0 = Counter is non-active.<br>1 = Counter is active. |
| 1 | SFCDIR | Smart fault/brake counters direction status (read only):<br>0 = Down counting.<br>1 = Up counting. |
| 0 | LSBD | Low level smart brake detector:<br>0 = Disable low level smart brake detector.<br>1 = Enable low level smart brake detector. It will be cleared by software. |

**ADC PIN SELECT**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ADCPS.7 | ADCPS.6 | ADCPS.5 | ADCPS.4 | ADCPS.3 | ADCPS.2 | ADCPS.1 | ADCPS.0 |

Mnemonic: ADCPS Address: C6h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7-0 | ADCPS | ADC input pin select. There are 8 ADC input pins shared with P1.0~P1.7. Its' functions are controlled by the bit value in ADCPS. Set the bit to switch the corresponding pin to ADC input port; clear the bit to disable the pin to perform ADC input port. The reset value is 00H. |

| BIT | CORRESPONDING PIN | BIT | CORRESPONDING PIN |
|---|---|---|---|
| ADCPS.0 | P1.0 | ADCPS.4 | P1.4 |
| ADCPS.1 | P1.1 | ADCPS.5 | P1.5 |
| ADCPS.2 | P1.2 | ADCPS.6 | P1.6 |
| ADCPS.3 | P1.3 | ADCPS.7 | P1.7 |

**TIMED ACCESS**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TA.7 | TA.6 | TA.5 | TA.4 | TA.3 | TA.2 | TA.1 | TA.0 |

Mnemonic: TA Address: C7h

| BIT | NAME | FUNCTION |
|---|---|---|

| | | |
|---|---|---|
| 7-0 | TA | The Timed Access register controls the access to protected bits. To access protected bits, the user must first write AAh to TA. This must be immediately followed by a write of 55h to TA. Now a window is opened in the protected bits for three machine cycles, during which the user can write to these bits. For detail data, please refer "TIMED ACCESS PROTECTION" section. |

**TIMER 2 CONTROL**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TF2 | EXF2 | RCLK | TCLK | EXEN2 | TR2 | C/$\overline{T2}$ | CP/$\overline{RL2}$ |

Mnemonic: T2CON Address: C8h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7 | TF2 | Timer 2 overflows flag. This bit is set when Timer 2 overflows. It is also set when the count is equal to the capture register in down count mode. It can be set only if RCLK and TCLK are both 0. It is cleared only by software. Software can also set this bit. |
| 6 | EXF2 | Timer 2 External Flag. A negative transition on the T2EX pin (P4.1) or timer 2 underflow/overflow will cause this flag to set based on $CP/\overline{RL2}$, EXEN2 and DCEN bits. If EXF2 is set by a negative transition, this flag must be cleared by software. Setting this bit in software or detection of a negative transition on T2EX pin will force a timer interrupt if enabled. |
| 5 | RCLK | Receive clock Flag. This bit determines the serial port time-base when receiving data in serial modes 1 or 3. If it is 0, then timer 1 overflow is used for baud rate generation, else timer 2 overflow is used. Setting this bit forces timer 2 in baud rate generator mode. |
| 4 | TCLK | Transmit clock Flag: This bit determines the serial port time-base when transmitting data in mode 1 and 3. If it is set to 0, the timer 1 overflow is used to generate the baud rate clock; else timer 2 overflow is used. Setting this bit forces timer 2 in baud rate generator mode. |
| 3 | EXEN2 | Timer 2 External Enable. This bit enables the capture/reload function on the T2EX pin if Timer 2 is not generating baud clocks for the serial port. If this bit is 0, then the T2EX pin will be ignored, else a negative transition detected on the T2EX pin will result in capture or reload. |
| 2 | TR2 | Timer 2 Run Control. This bit enables/disables the operation of timer 2. Halting this will preserve the current count in TH2, TL2. |
| 1 | C/$\overline{T}$ | Counter/Timer select. This bit determines whether timer 2 will function as a timer or a counter. Independent of this bit, the timer will run at 2 clocks per tick when used in baud rate generator mode. If it is set to 0, then timer 2 operates as a timer at a speed depending on T2M bit (CKCON.5), else, it will count negative edges on T2 pin. |
| 0 | CP/$\overline{RL2}$ | Capture/Reload Select. This bit determines whether the capture or reload function will be used for timer 2. If either RCLK or TCLK is set, this bit will not function and the timer will function in an auto-reload mode following each overflow. If the bit is 0 then auto-reload will occur when timer 2 overflows or a falling edge is detected on T2EX if EXEN2 =1. If this bit is 1, then timer 2 captures will occur when a falling edge is detected on T2EX if EXEN2=1. |

**TIMER 2 MODE CONTROL**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | HC5 | HC4 | HC3 | HC2 | T2CR | - | - | DCEN |

Mnemonic: T2MOD Address: C9h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7 | HC5 | Hardware clears INT5 flag. Setting this bit allows the flag of External Interrupt 5 to be automatically cleared by hardware while entering the interrupt service routine. |
| 6 | HC4 | Hardware clears INT4 flag. Setting this bit allows the flag of External Interrupt 4 to be automatically cleared by hardware while entering the interrupt service routine. |
| 5 | HC3 | Hardware clears INT3 flag. Setting this bit allows the flag of External Interrupt 3 to be automatically cleared by hardware while entering the interrupt service routine. |
| 4 | HC2 | Hardware clears INT2 flag. Setting this bit allows the flag of External Interrupt 2 to be automatically cleared by hardware while entering the interrupt service routine. |
| 3 | T2CR | Timer 2 Capture Reset. In the Timer 2 Capture Mode this bit enables/disables hardware automatically reset timer 2 while the value in TL2 and TH2 have been transferred into the capture register. |
| 2-1 | - | Reserved. |
| 0 | DCEN | Down Count Enable. This bit, in conjunction with the T2EX pin, controls the up/down direction that timer 2 counts in 16-bit auto-reload mode. |

**TIMER 2 CAPTURE LSB**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RCAP2L.7 | RCAP2L.6 | RCAP2L.5 | RCAP2L.4 | RCAP2L.3 | RCAP2L.2 | RCAP2L.1 | RCAP2L.0 |

Mnemonic: RCAP2L Address: CAh

| BIT | NAME | FUNCTION |
|---|---|---|
| 7-0 | RCAP2L | Timer 2 Capture LSB: This register is used to capture the TL2 value when a timer 2 is configured in capture mode. RCAP2L is also used as the LSB of a 16 bit reload value when timer 2 is configured in auto reload mode. |

**TIMER 2 CAPTURE MSB**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RCAP2H.7 | RCAP2H.6 | RCAP2H.5 | RCAP2H.4 | RCAP2H.3 | RCAP2H.2 | RCAP2H.1 | RCAP2H.0 |

Mnemonic: RCAP2H Address: CBh

| BIT | NAME | FUNCTION |
|---|---|---|
| 7-0 | RCAP2H | Timer 2 Capture HSB: This register is used to capture the TH2 value when a timer 2 is configured in capture mode. RCAP2H is also used as the HSB of a 16 bit reload value when timer 2 is configured in auto reload mode. |

**TIMER 2 LSB**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TL2.7 | TL2.6 | TL2.5 | TL2.4 | TL2.3 | TL2.2 | TL2.1 | TL2.0 |

Mnemonic: TL2 Address: CCh

TL2 Timer 2 LSB

**TIMER 2 MSB**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TH2.7 | TH2.6 | TH2.5 | TH2.4 | TH2.3 | TH2.2 | TH2.1 | TH2.0 |

Mnemonic: TH2 Address: CDh

TH2 Timer 2 MSB

**PWM CONTROL REGISTER 2**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | BKCH | BKPS | BPEN | BKEN | FP1 | FP0 | PMOD1 | PMOD0 |

Mnemonic: PWMCON2 Address: CEh

<table>
<tr><th>BIT</th><th>NAME</th><th>FUNCTION</th></tr>
<tr><td>7</td><td>BKCH</td><td>See table below for BKCH settings.</td></tr>
<tr><td>6</td><td>BKPS</td><td>Select which brake condition triggers brake flag. LSBD bit is described in SFR FSPLT.<br><table><tr><th>BKPS</th><th>LSBD</th><th>Description</th></tr><tr><td>0</td><td>0</td><td>0 = Brake is asserted if P1.1 is low.</td></tr><tr><td>1</td><td>0</td><td>1 = Brake is asserted if P1.1 is high</td></tr><tr><td>x</td><td>1</td><td>Low level smart brake detector.</td></tr></table></td></tr>
<tr><td>5</td><td>BPEN</td><td>See table below for BPEN settings.</td></tr>
<tr><td>4</td><td>BKEN</td><td>0 = The Brake is never asserted.<br>1 = The Brake is enabled.</td></tr>
<tr><th>BIT</th><th>NAME</th><th>FUNCTION</th></tr>
<tr><td>3-2</td><td>FP[1:0]</td><td>Select PWM frequency prescaler select bits. The clock source of prescaler, Fpwm is in phase with Fosc if PWMRUN=1.<br><table><tr><th>FP[1:0]</th><th>Fpwm</th></tr><tr><td>00</td><td>$F_{OSC}$</td></tr><tr><td>01</td><td>$F_{OSC}$ /2</td></tr><tr><td>10</td><td>$F_{OSC}$ /4</td></tr><tr><td>11</td><td>$F_{OSC}$ /16</td></tr></table></td></tr>
</table>

Continued

| BIT | NAME | FUNCTION |
|---|---|---|
| 1-0 | PMOD[1:0] | PWM Mode selects bits:<br>PMOD[1:0] – Description<br>00 – Edge-aligned mode. (up counter)<br>01 – Single-shot mode. (up counter)<br>10 – Center aligned mode (up-down counter)<br>11 – Reserved |

**Brake Condition Table**

| BPEN | BKCH | BRAKE CONDITION |
|---|---|---|
| 0 | 0 | Brake On, (Software brake and keeping brake).<br>Software brake condition. When active (BPEN=BKCH=0, and BKEN=1), PWM output follows PWMnB setting. This brake has no effect on PWMRUN bit; therefore, internal PWM generator continues to run. When the brake is released, the state of PWM output depends on the current state of PWM generator output during the release. |
| 0 | 1 | Brake On, when PWM is not running (PWMRUN=0), the PWM output condition is follow PWMnB setting. When the brake is released (by disabling BKEN = 0), the PWM output resumes to the state when PWM generator stop running prior to enabling the brake.<br>Brake Off, when PWM is running (PWMRUN=1). |
| 1 | 0 | Brake On, when Brake Pin asserted, PWM output follows PWMnB setting. The PWMRUN will be clear.<br>External pin brake condition. When active (by external pin), PWM output follows PWMnB setting. PWMRUN will be cleared by hardware. BKF flag will be set. When the brake is released (by de-asserting the external pin + disabling BKEN = 0), the PWM output resumes to the state of the PWM generator output prior to the brake. |
| 1 | 1 | This brake condition (by Brake Pin) causes BKF to be set, but PWM generator continues to run. The PWM output does not follow PWMnB, instead it output continuously as per normal without affected by the brake. |

**PWM 4 LOW BITS REGISTER**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PWM4.7 | PWM4.6 | PWM4.5 | PWM4.4 | PWM4.3 | PWM4.2 | PWM4.1 | PWM4.0 |

Mnemonic: PWM4L Address: CFh

PWM4.7-0 PWM4 Low Bits Register.

**PROGRAM STATUS WORD**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CY | AC | F0 | RS1 | RS0 | OV | F1 | P |

Mnemonic: PSW Address: D0h

| BIT | NAME | FUNCTION |
|---|---|---|

| | | |
|---|---|---|
| 7 | CY | Carry flag. Set for an arithmetic operation which results in a carry being generated from the ALU. It is also used as the accumulator for the bit operations. |
| 6 | AC | Auxiliary carry: Set when the previous operation resulted in a carry (during addition) or a borrow (during subtraction) from the high order nibble. |
| 5 | F0 | User flag 0. A general purpose flag that can be set or cleared by the by software. |
| 4-3 | RS.1-0 | Register bank selects bits:<br>RS1 / RS2 / Register Bank / Address<br>0 / 0 / 0 / 00-07h<br>0 / 1 / 1 / 08-0Fh<br>1 / 0 / 2 / 10-17h<br>1 / 1 / 3 / 18-1Fh |
| 2 | OV | Overflow flag. Set when a carry was generated from the seventh bit but not from the 8th bit as a result of the previous operation or vice-versa. |
| 1 | F1 | User Flag 1. General purpose flag that can be set or cleared by the user by software. |
| 0 | P | Parity flag. Set/cleared by hardware to indicate odd/even number of 1's in the accumulator. |

**PWMP COUNTER HIGH BITS REGISTER**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | PWMP.11 | PWMP.10 | PWMP.9 | PWMP.8 |

Mnemonic: PWMPH Address: D1h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7-4 | - | Reserved. |
| 3-0 | PWMP.11~PWMP.8 | PWM Counter Register bits 11~8. |

**PWM 0 HIGH BITS REGISTER**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | PWM0.11 | PWM0.10 | PWM0.9 | PWM0.8 |

Mnemonic: PWM0H Address: D2h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7~4 | - | Reserved. |
| 3~0 | PWM0.11 ~PWM0.8 | The PWM 0 Register bit 11~8. |

**NVM DATA**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | NVMDAT.7 | NVMDAT.6 | NVMDAT.5 | NVMDAT.4 | NVMDAT.3 | NVMDAT.2 | NVMDAT.1 | NVMDAT.0 |

Mnemonic: NVMDAT Address: D3h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7~0 | NVMDAT.7~NVMDAT.0 | The NVM data write register. The read NVM data is by MOVX instruction. |

**QEI CONTROL REGISTER**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | - | - | DISIDX | DIR | QEIM1 | QEIM0 | QEIEN |

Mnemonic: QEICON Address: D4h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7-5 | - | Reserved. |
| 4 | DISIDX | Disable Input Capture 2 edge detection function:<br>0 = Enable IC2 edge detection function (default).<br>1 = Disable IC2 edge detection function.<br>This bit is effective when QEIEN=1. |
| 3 | DIR | Direction index of motion detection bit:<br>1 = Forward (Up-counting).<br>0 = Backward (Down-counting).<br>This bit is writable and readable. |
| 2-1 | QEIM[1:0] | QEI mode select bits:<br>QEIM1 / QEIM0 / Descriptions<br>0 / 0 / X4 free-counting mode<br>0 / 1 / X2 free-counting mode<br>1 / 0 / X4 compare-counting mode<br>1 / 1 / X2 compare-counting mode |
| 0 | QEIEN | Input module mode select bit:<br>0 = Input module performs Input Capture Functions. (Default value).<br>1 = Input module works as QEI. |

**PWM 2 HIGH BITS REGISTER**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | PWM2.11 | PWM2.10 | PWM2.9 | PWM2.8 |

Mnemonic: PWM2H Address: D5h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7~4 | - | Reserved |
| 3~0 | PWM2.11 ~PWM2.8 | PWM 2 Register bit 11~8. |

**PWM 6 HIGH BITS REGISTER**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | PWM6.11 | PWM6.10 | PWM6.9 | PWM6.8 |

Mnemonic: PWM6H Address: D6h

| BIT | NAME | FUNCTION |
|---|---|---|

| | | |
|---|---|---|
| 7~4 | - | Reserved |
| 3~0 | PWM6.11 ~PWM6.8 | PWM 6 Register bit 11~8. |

**WATCHDOG CONTROL 2**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | - | - | STRLD |

Mnemonic: WDCON2 Address: D7h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7-6 | - | Reserved. |
| 0 | STRLD | Set this bit, CPU will restart from LD Flash EPROM after watchdog reset. Clear this bit, CPU will restart from AP Flash EPROM after watchdog reset. This register is protected by timer access (TA) register. |

**WATCHDOG CONTROL**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | POR | - | - | WDIF | WTRF | EWT | RWT |

Mnemonic: WDCON Address: D8h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7 | - | Reserved. |
| 6 | POR | Power-on Reset Flag. Hardware will set this flag on a power up condition. This flag can be read or written by software. A write by software is the only way to clear this bit once it is set. |
| 5-4 | - | Reserved. |
| 3 | WDIF | Watchdog Timer Interrupt Flag. This bit is set by hardware to indicate that the time-out period has elapsed and invoke watch dog timer interrupt if enabled (EWDI=1). This bit must be clear by software. |
| 2 | WTRF | Watchdog Timer Reset Flag. Hardware will set this bit when the watchdog timer causes a reset if EWT= 1. Software can read it but must clear it manually. A power-on reset will also clear the bit. This bit helps software in determining the cause of a reset |
| 1 | EWT | Enable Watchdog timer Reset. Setting this bit will enable the Watchdog timer Reset function after 512 clocks delay from time out and setting WTRF flag. |
| 0 | RWT | Reset Watchdog Timer. This bit restarts the watchdog timer and helps in putting the watchdog timer into a know state. It also helps in resetting the watchdog timer before a time-out occurs. If EWDI (EIE.4) is set, an interrupt will occur when time-out. If EWT is set, 512 clocks after the time-out, a system reset will occur and CPU starts from 0000H. This bit is self-clearing. The WDCON SFR is set to a 0x0x0xx0b on an external reset. WTRF is set to a 1 on a Watchdog timer reset, but to a 0 on power on resets. WTRF is not altered by an external reset. POR is set to 1 by a power-on reset. EWT is cleared to 0 on a Power-on reset and unaffected by other resets. |

The WDCON SFR is set to x0xx 0000b on an external reset. WTRF is set to a 1 on a Watchdog timer reset, but to a 0 on power on resets. POR is set to 1 by a power-on reset. EWT is cleared to 0 on a Power-on reset, reset pin reset, Watch Dog Timer reset and ISP reset.

All the bits in this SFR have unrestricted read access. The bits of POR, WDIF, EWT and RWT require Timed Access (TA) procedure to write. The remaining bits have unrestricted write accesses. Please refer TA register description.

**PWMP COUNTER LOW BITS REGISTER**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PWMP.7 | PWMP.6 | PWMP.5 | PWMP.4 | PWMP.3 | PWMP.2 | PWMP.1 | PWMP.0 |

Mnemonic: PWMPL Address: D9h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7~0 | PWMP.7 ~PWMP.0 | PWM Counter Low Bits Register. |

**PWM0 LOW BITS REGISTER**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PWM0.7 | PWM0.6 | PWM0.5 | PWM0.4 | PWM0.3 | PWM0.2 | PWM0.1 | PWM0.0 |

Mnemonic: PWM0L Address: DAh

| BIT | NAME | FUNCTION |
|---|---|---|
| 7~0 | PWM0.7 ~PWM0.0 | PWM 0 Low Bits Register. |

**NVM LOW BYTE ADDRESS**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | NVMADDRL.7 | NVMADDRL.6 | NVMADDRL.5 | NVMADDRL.4 | NVMADDRL.3 | NVMADDRL.2 | NVMADDRL.1 | NVMADDRL.0 |

Mnemonic: NVMADDRL Address: DBh

| BIT | NAME | FUNCTION |
|---|---|---|
| 7~0 | NVMADDRL.7~ NVMADDRL.0 | NVM low byte address. |

**PWM CONTROL REGISTER 1**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PWMRUN | Load | PWMF | CLRPWM | PWM6I | PWM4I | PWM2I | PWM0I |

Mnemonic: PWMCON1 Address: DCh

| BIT | NAME | FUNCTION |
|---|---|---|
| 7 | PWMRUN | 0 = The PWM is not running.<br>1 = The PWM counter is running. |
| 6 | Load | This bit is auto cleared by hardware after the PWMP and PWMn are transferred to counter and compare register:<br>0 = The registers value of PWMP and PWMn is never loaded to counter and compare registers.<br>1 = The PWMP and PWMn registers load value to counter and compare registers at the counter underflow/match. |
| 5 | PWMF | 12 bit counter overflow flag:<br>0 = The 12-bit counter is not underflow/match.<br>1 = The 12-bit counter is underflow/match. It will be set by hardware and cleared by software. |
| 4 | CLRPWM | 1 = Clear 12-bit PWM counter to 000H. It will be automatically clear by hardware. |
| 3-0 | PWMxI | 0 = PWM0 output is non-inverted.<br>1 = PWM0 output is inverted.<br>**Note**: x = 0,2,4,6. |

**PWM 2 LOW BITS REGISTER**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PWM2.7 | PWM2.6 | PWM2.5 | PWM2.4 | PWM2.3 | PWM2.2 | PWM2.1 | PWM2.0 |

Mnemonic: PWM2L Address: DDh

| BIT | NAME | FUNCTION |
|---|---|---|
| 7~0 | PWM2.7 ~PWM2.0 | PWM 2 Low Bits Register. |

**PWM 6 LOW BITS REGISTER**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PWM6.7 | PWM6.6 | PWM6.5 | PWM6.4 | PWM6.3 | PWM6.2 | PWM6.1 | PWM6.0 |

Mnemonic: PWM6L Address: DEh

| BIT | NAME | FUNCTION |
|---|---|---|
| 7~0 | PWM6.7 ~PWM6.0 | PWM 6 Low Bits Register. |

**PWM CONTROL REGISTER 3**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PWM7B | PWM6B | PWM5B | PWM4B | PWM3B | PWM2B | PWM1B | PWM0B |

Mnemonic: PWMCON3 Address: DFh

| BIT | NAME | FUNCTION |
|---|---|---|
| 7-0 | PWMxB | 0 = The PWM0 output is low, when Brake is asserted.<br>1 = The PWM0 output is high, when Brake is asserted.<br>**Note**: x = 0~7 |

**ACCUMULATOR**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ACC.7 | ACC.6 | ACC.5 | ACC.4 | ACC.3 | ACC.2 | ACC.1 | ACC.0 |

Mnemonic: ACC Address: E0h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7-0 | ACC | The A or ACC register is the standard 8032 accumulator |

**ADC CONTROL REGISTER**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ADCEN | - | ADCEX | ADCI | ADCS | AADR.2 | AADR.1 | AADR.0 |

Mnemonic: ADCCON Address: E1h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7 | ADCEN | Enable A/D Converter Function. Set ADCEN to logic high to enable ADC block. |
| 6 | - | Reserved. |
| 5 | ADCEX | Enable external start control of ADC conversion by a rising edge from P4.0. ADCEX=0: Disable external start. ADCEX=1: Enable external start control. |
| 4 | ADCI | A/D Converting Complete/Interrupt Flag. This flag is set when ADC conversion is completed. The ADC interrupt is requested if the interrupt is enabled. ADCI is set by hardware and cleared by software only. |
| 3 | ADCS | ADC Start and Status: Set this bit to start an A/D conversion. It may also be set by STADC if ADCEX is 1. This signal remains high while the ADC is busy and is reset right after ADCI is set.<br>**Notes**:<br>1. It is recommended to clear ADCI before ADCS is set. However, if ADCI is cleared and ADCS is set at the same time, a new A/D conversion may start on the same channel.<br>2. Software clearing of ADCS will abort conversion in progress.<br>3. ADC cannot start a new conversion while ADCS or ADCI is high. |
| 2-0 | AADR | Select and enable analog input channel from ADC0 to ADC7. |

| AADR[2:0] | ADC selected input | AADR[2:0] | ADC selected input |
|---|---|---|---|
| 000 | ADCCH0 (P1.0) | 100 | ADCCH4 (P1.4) |
| 001 | ADCCH1 (P1.1) | 101 | ADCCH5 (P1.5) |
| 010 | ADCCH2 (P1.2) | 110 | ADCCH6 (P1.6) |
| 011 | ADCCH3 (P1.3) | 111 | ADCCH7 (P1.7) |

The ADCI and ADCS control the ADC conversion as below:

| ADCI | ADCS | ADC STATUS |
|---|---|---|
| 0 | 0 | ADC not busy; A conversion can be started. |
| 0 | 1 | ADC busy; Start of a new conversion is blocked. |
| 1 | 0 | Conversion completed; Start of a new conversion requires ADCI = 0. |
| 1 | 1 | This is an internal temporary state that user can ignore it. |

**ADC CONVERTER RESULT HIGH REGISTER**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ADC.9 | ADC.8 | ADC.7 | ADC.6 | ADC.5 | ADC.4 | ADC.3 | ADC.2 |

Mnemonic: ADCH Address: E2h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7-0 | ADC[9:2] | 8 MSB of 10 bit A/D conversion result. ADCH is a read only register. |

**ADC CONVERTER RESULT LOW REGISTER**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ADCLK.1 | ADCLK.0 | - | - | - | - | ADC.1 | ADC.0 |

Mnemonic: ADCL Address: E3h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7-6 | ADCLK | ADC Clock Frequency Select. The 10 bit ADC needs a clock to drive the converting that the clock frequency may not over 4MHz. ADCLK[1:0] controls the frequency of the clock to ADC block:<br>ADCLK.1 / ADCLK.0 / ADC Clock Frequency<br>0 / 0 / Crystal clock / 4 (Default)<br>0 / 1 / Crystal clock / 8<br>1 / 0 / Crystal clock / 16<br>1 / 1 / Reserved |
| 1-0 | ADC | 2 LSB of 10-bit A/D conversion result. Both bits are read only. |

**PWM DEAD-TIME CONTROL REGISTER 1**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PDTC1.7 | PDTC1.6 | PDTC1.5 | PDTC1.4 | PDTC1.3 | PDTC1.2 | PDTC1.1 | PDTC1.0 |

Mnemonic: PDTC1 Address: E5h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7-6 | PDTC1 | Dead-time clock frequency (FDT) prescaler select bits.<br>PDTC1.7 / PDTC1.6 / FDT<br>0 / 0 / FOSC/2<br>0 / 1 / FOSC/4<br>1 / 0 / FOSC/8<br>1 / 1 / FOSC/16 |
| 5-0 | PDTC1 | Dead time counter. Unsigned 6 bit dead time value bits for Dead Time Unit.<br>**Dead-time = FDT * (PDTC1 [5:0]+1)** |

**PWM DEAD-TIME CONTROL REGISTER 0**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PDTC0.7 | PDTC0.6 | PDTC0.5 | PDTC0.4 | PDTC0.3 | PDTC0.2 | PDTC0.1 | PDTC0.0 |

Mnemonic: PDTC0 Address: E6h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7-4 | PDTC0 | Control complementary PWM to delay a dead-time at every rising edge or falling edge. Reset value = 0.<br>1 = Dead-time is inserted at falling edge.<br>0 = Dead-time is inserted at rising edge.<br>PDTC0.4 - controls the pair of (PWM0, PWM1).<br>PDTC0.5 - controls the pair of (PWM2, PWM3).<br>PDTC0.6 - controls the pair of (PWM4, PWM5).<br>PDTC0.7 - controls the pair of (PWM6, PWM7). |
| 3-0 | PDTC0 | Enable dead-time insertion; Dead-time insertion is only active when the pair of complementary PWM is enabled. Reset value=0. If dead-time insertion is inactive, the outputs of pin pair are complementary without any delay.<br>1 = Programmable dead-time is inserted into the pair signals of comparator output to delay the pair signals change from low to high.<br>0 = Disable dead-time insertion.<br>PDTC0.0 - enables the dead-time insertion on the pin pair (PWM0, PWM1).<br>PDTC0.1 - enables the dead-time insertion on the pin pair (PWM2, PWM3).<br>PDTC0.2 - enables the dead-time insertion on the pin pair (PWM4, PWM5).<br>PDTC0.3 - enables the dead-time insertion on the pin pair (PWM6, PWM7). |

**PWM CONTROL REGISTER 4**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PWMEOM | PWMOOM | PWM6OM | PWM7OM | - | - | - | BKF |

Mnemonic: PWMCON4 Address: E7h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7 | PWMEOM | PWM Channel 0, 2 and 4 Output Mode.<br>0 = Disable PWM channels 0, 2 and 4 to pwm output pins.<br>1 = Enable PWM channels 0, 2 and 4 to pwm output pins. |
| 6 | PWMOOM | PWM Channel 1, 3 and 5 Output Mode.<br>0 = Disable PWM channels 1, 3 and 5 to pwm output pins.<br>1 = Enable PWM channels 1, 3 and 5 to pwm output pins. |
| 5 | PWM6OM | PWM Channel 6 Output Mode.<br>0 = Disable PWM channel 6 to pwm output pin.<br>1 = Enable PWM channel 6 to pwm output pin. |
| 4 | PWM7OM | PWM Channel 7 Output Mode.<br>0 = Disable PWM channel 7 to pwm output pin.<br>1 = Enable PWM channel 7 to pwm output pin. |
| 3-1 | - | Reserved. |
| 0 | BKF | The External Brake Pin Flag.<br>0 = The PWM is not brake.<br>1 = The PWM is brake by external brake pin. It will be cleared by software. |

Together with option bits (PWMEE and PWMOE), PWMEOM, PWMOOM, PWM6OM and PWM7OM control the PWM pin structure, as follow;

| PWMEE/PWMOE (OPTION BITS) | PWMEOM/PWMOOM /PWM6OM/PWM7OM | PIO.X (X = 0-7) | PIN STRUCTURES |
|---|---|---|---|
| X | 0 | X | Tri-state |
| 1 (Disable) | 1 | X | Quasi (I/O output) |
| 0 (Enable) | 1 | 0 | Push Pull (PWM output) |
| 0 (Enable) | 1 | 1 | Push Pull (I/O output) |

Table 7-2: PWM pin structures (during internal rom execution)

| PWMEE/PWMOE (OPTION BITS) | PWMEOM/PWMOOM /PWM6OM/PWM7OM | PIO.X (X = 0-7) | PIN OUTPUT | PIN STRUCTURES |
|---|---|---|---|---|
| 1 (Disable) | X | X | External access | Push Pull |
| 0 (Enable) | X | X | External access | Push Pull (strong) |

Table 7-3: PWM pin structures (during external rom execution)

**Note**: PWMEOM/PWMOOM/PWM6OM/PWM7OM are cleared to zero when CPU in reset state. Thus, the port pins that multi-function with PWM will be tristated on default. User is required to set the bits to 1 to enable GPIO/PWM outputs.

**EXTENDED INTERRUPT ENABLE**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ES1 | EX5 | EX4 | EWDI | EX3 | EX2 | - | EI2C |

Mnemonic: EIE Address: E8h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7 | ES1 | Enable Serial Port 1 interrupts. |
| 6 | EX5 | Enable External Interrupt 5. |
| 5 | EX4 | Enable External Interrupt 4. |
| 4 | EWDI | Enable Watchdog timer interrupt. |
| 3 | EX3 | Enable External Interrupt 3. |
| 2 | EX2 | Enable External Interrupt 2. |
| 1 | - | Reserved. |
| 0 | EI2C | Enable I2C interrupt. |

**I2C CONTROL REGISTER**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | ENS | STA | STO | SI | AA | I2CIN | - |

Mnemonic: I2CON Address: E9h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7 | - | Reserved. |
| 6 | ENS | I2C serial function block enable bit. When ENS=1 the I2C serial function enables. The port latches of SDA and SCL must be set to logic high. |
| 5 | STA | I2C START Flag. Setting STA to logic 1 to enter master mode, the I2C hardware sends a START or repeat START condition to bus when the bus is free. |
| 4 | STO | I2C STOP Flag. In master mode, setting STO to transmit a STOP condition to bus then I2C hardware will check the bus condition if a STOP condition is detected this flag will be cleared by hardware automatically. In a slave mode, setting STO resets I2C hardware to the defined “not addressed” slave mode. |
| 3 | SI | I2C Interrupt Flag. When a new SIO state is present in the S1STA register, the SI flag is set by hardware, and if the EA and EI2C bits are both set, the I2C interrupt is requested. SI must be cleared by software. |
| 2 | AA | Assert Acknowledge Flag. When AA=1 an acknowledged (low level to SDA) will be returned during the acknowledge clock pulse on the SCL line. When AA=0 an acknowledged (high level to SDA) will be returned during the acknowledge clock pulse on the SCL line. |
| 1 | I2CIN | By default it is zero and input are allows to come in through SDA pin. As when it is 1 input is disallow and to prevent leakage current. During Power-Down mode input is disallow. |
| 0 | - | Reserved. |

**I2C ADDRESS REGISTER**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | I2ADDR.7 | I2ADDR.6 | I2ADDR.5 | I2ADDR.4 | I2ADDR.3 | I2ADDR.2 | I2ADDR.1 | GC |

Mnemonic: I2ADDR Address: EAh

| BIT | NAME | FUNCTION |
|---|---|---|
| 7-1 | I2ADDR | I2C Slave Address. The contents of the register are irrelevant when I2C is in master mode. In the slave mode, the seven most significant bits must be loaded with the MCU's own slave address. The I2C hardware will react if the contents of I2ADDR are matched with the received slave address. |
| 0 | GC | Enable General Call Function. The GC bit is set the I2C port hardware will respond to General Call address (00H). Clear GC bit to disable general call function. |

**NVM HIGH BYTE ADDRESS**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | NVMADDRH.10 | NVMADDRH.9 | NVMADDRH.8 |

Mnemonic: NVMADDRH Address: EBh

| BIT | NAME | FUNCTION |
|---|---|---|
| 7-3 | - | Reserved. |
| 2-0 | NVMADDRH.10 ~ NVMADDRH.8 | NVM High byte address |

**I2C DATA REGISTER**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | I2DAT.7 | I2DAT.6 | I2DAT.5 | I2DAT.4 | I2DAT.3 | I2DAT.2 | I2DAT.1 | I2DAT.0 |

Mnemonic: I2DAT Address: ECh

I2DAT.7-0 The data register of I2C channel.

**I2C STATUS REGISTER**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | B7 | B6 | B5 | B4 | B3 | 0 | 0 | 0 |

Mnemonic: I2STATUS Address: EDh

| BIT | NAME | FUNCTION |
|---|---|---|
| 7-0 | I2STATUS | The Status Register of I2C. The three least significant bits are always 0. The five most significant bits contain the status code. There are 23 possible status codes. When I2STATUS contains F8H, no serial interrupt is requested. All other I2STATUS values correspond to defined I2C states. When each of these states is entered, the I2C1 interrupt is requested (SI = 1). A valid status code is present in I2STATUS one machine cycle after SI is set by hardware and is still present one machine cycle after SI has been reset by software. In addition, states 00H stands for a Bus Error. A Bus Error occurs when a START or STOP condition is present at an illegal position in the formation frame. Example of illegal position are during the serial transfer of an address byte, a data byte or an acknowledge bit. |

### I2C BAUD RATE CONTROL REGISTER

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | I2CLK.7 | I2CLK.6 | I2CLK.5 | I2CLK.4 | I2CLK.3 | I2CLK.2 | I2CLK.1 | I2CLK.0 |

Mnemonic: I2CLK Address: EEh

| BIT | NAME | FUNCTION |
|---|---|---|
| 7-0 | I2CLK | I2C clock rate control. |

### I2C TIMER COUNTER REGISTER

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | ENTI | DIV4 | TIF |

Mnemonic: I2TIMER Address: EFh

| BIT | NAME | FUNCTION |
|---|---|---|
| 7-3 | - | Reserved. |
| 2 | ENTI | Enable I2C 14-bits Time-out Counter. Setting ENTI to logic high will firstly reset the time-out counter and then start up counting. Clearing ENTI disables the 14-bit time-out counter. ENTI can be set to logic high only when SI=0. |
| 1 | DIV4 | I2C Time-out Counter Clock Frequency Selection.<br>0 = the clock frequency is coherent to the system clock Fosc.<br>1 = the clock frequency is Fosc/4. |
| 0 | TIF | I2C Time-out Flag. When the time-out counter overflows hardware will set this flag and request the I2C interrupt if I2C interrupt is enabled (EI2C=1). This bit must be cleared by software. |

### B REGISTER

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | B.7 | B.6 | B.5 | B.4 | B.3 | B.2 | B.1 | B.0 |

Mnemonic: B Address: F0h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7-0 | B | The B register is the standard 8032 accumulator. |

### SERIAL PERIPHERAL CONTROL REGISTER

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SSOE | SPE | LSBFE | MSTR | CPOL | CPHA | SPR1 | SPR0 |

Mnemonic: SPCR Address: F3h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7 | SSOE | Slave Select Output Enable Bit. The $\overline{SS}$ output feature is enabled only in master mode by asserting the SSOE bit. In slave mode (/SS) input is not affected by SSOE bit. See table below. |
| 6 | SPE | Serial Peripheral System Enable Bit.<br>When the SPE bit is set, SPI block functions is enable. When MODF is set, SPE always reads 0.<br>0 = SPI system disabled.<br>1 = SPI system enabled. |
| 5 | LSBFE | LSB - First Enable. This bit does not affect the position of the MSB and LSB in the data register. Reads and writes of the data register always have the MSB in bit 7. In master mode, a change of this bit will abort a transmission in progress and force the SPI system into idle state.<br>1 = Data is transferred least significant bit first.<br>0 = Data is transferred most significant bit first. |
| 4 | MSTR | Master Mode Select Bit. It is customary to have an external pull-up resistor on lines that are driven by open drain devices.<br>0 = Slave mode.<br>1 = Master mode. |
| 3 | CPOL | Clock Polarity Bit. When the clock polarity bit is cleared and data is not being transferred, the SPCLK pin of the master device has a steady state low value. When CPOL is set, SPCLK idles high. |
| 2 | CPHA | Clock Phase Bit. The clock phase bit, in conjunction with the CPOL bit, controls the clock-data relationship between master and slave. The CPHA bit selects one of two different clocking protocols. |
| 1-0 | SPR | SPI Baud Rate Selection Bits. These bits specify the SPI baud rates. |

| DRSS | SSOE | MASTER MODE | SLAVE MODE |
|---|---|---|---|
| 0 | 0 | /SS input ( With Mode Fault ) | /SS Input ( Not affected by SSOE ) |
| 0 | 1 | Reserved | /SS Input ( Not affected by SSOE ) |
| 1 | 0 | /SS General purpose I/O ( No Mode Fault ) | /SS Input ( Not affected by SSOE ) |
| 1 | 1 | /SS output ( No Mode Fault ) | /SS Input ( Not affected by SSOE ) |

**Note**: In master mode, a change of LSBFE, MSTR, CPOL, CPHA and SPR [1:0] will abort a transmission in progress and force the SPI system into idle state.

**SERIAL PERIPHERAL STATUS REGISTER**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SPIF | WCOL | SPIOVF | MODF | DRSS | - | - | - |

Mnemonic: SPSR Address: F4h

| BIT | NAME | FUNCTION |
|---|---|---|

| | | |
|---|---|---|
| 7 | SPIF | SPI Interrupt Complete Flag. SPIF is set upon completion of data transfer between this device and external device or when new data has been received and copied to the SPDR. If SPIF goes high, and if ESPI is set, a serial peripheral interrupt is generated. When SPIF is set; it must be clear by software and attempts to write SPDR are inhibited if SPIF set. |
| 6 | WCOL | Write Collision Flag. If a writer collision occurs on SPI bus, WCOL is set to high by hardware. WCOL must be clear by software. |
| 5 | SPIOVF | SPI overrun flag. SPIOVF is set if a new character is received before a previously received character is read from SPDR. Once this bit is set it will prevent SPDR register form excepting new data and must be cleared first before any new data can be written. This flag is clear by software.<br>0 = No overrun.<br>1 = Overrun detected. |
| 4 | MODF | SPI Mode Error Interrupt Status Flag. MODF is set when hardware detects mode fault. This bit is cleared by software. |
| 3 | DRSS | Data Register Slave Select. Refer to above table in SPCR register. |
| 2-0 | - | Reserved. |

**Note:** Bits WCOL, MODF and SPIF are cleared by software writing "0" to them.

**SERIAL PERIPHERAL DATA I/O REGISTER**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SPD.7 | SPD.6 | SPD.5 | SPD.4 | SPD.3 | SPD.2 | SPD.1 | SPD.0 |

Mnemonic: SPDR Address: F5h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7-0 | SPD | SPDR is used when transmitting or receiving data on serial bus. Only a write to this register initiates transmission or reception of a byte, and this only occurs in the master device. A read of the SPDR is actually a read of a buffer. To prevent an overrun and the loss of the byte that caused the overrun, the first SPIF must be cleared by the time a second transfer of data from the shift Register to the read buffer is initiated. |

**I2C SLAVE ADDRESS MASK ENABLE**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | I2CSADEN.7 | I2CSADEN.6 | I2CSADEN.5 | I2CSADEN.4 | I2CSADEN.3 | I2CSADEN.2 | I2CSADEN.1 | I2CSADEN.0 |

Mnemonic: I2CSADEN Address: F6h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7-0 | I2CSADEN | This register enables the Automatic Address Recognition feature of the I2C. When a bit in the I2CSADEN is set to 1, the same bit location in I2CSADDR1 will be compared with the incoming serial port data. When I2CSADEN.n is 0, the bit becomes don't care in the comparison. This register enables the Automatic Address Recognition feature of the I2C. When all the bits of I2CSADEN are 0, interrupt will occur for any incoming address. The default value is 0xFE. |

### EXTENDED INTERRUPT HIGH PRIORITY

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PS1H | PX5H | PX4H | PWDIH | PX3H | PX2H | - | PI2CH |

Mnemonic: EIPH Address: F7h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7 | PS1H | Serial Port 1 Interrupt High Priority. PS1H = 1 sets it to highest priority level. |
| 6 | PX5H | External Interrupt 5 High Priority. PX5H = 1 sets it to highest priority level. |
| 5 | PX4H | External Interrupt 4 High Priority. PX4H = 1 sets it to highest priority level. |
| 4 | PWDIH | Watchdog Timer Interrupt High Priority. PWDIH = 1 sets it to highest priority level. |
| 3 | PX3H | External Interrupt 3 High Priority. PX3H = 1 sets it to highest priority level. |
| 2 | PX2H | External Interrupt 2 High Priority. PX2H = 1 sets it to highest priority level. |
| 1 | - | Reserved. |
| 0 | PI2CH | I2C Interrupt High Priority. PI2CH = 1 sets it to highest priority level. |

### EXTENDED INTERRUPT PRIORITY

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PS1 | PX5 | PX4 | PWDI | PX3 | PX2 | - | PI2C |

Mnemonic: EIP Address: F8h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7 | PS1 | Serial Port 1 Interrupt Priority. |
| 6 | PX5 | External Interrupt 5 Priority. |
| 5 | PX4 | External Interrupt 4 Priority. |
| 4 | PWDI | Watchdog Timer Interrupt Priority. |
| 3 | PX3 | External Interrupt 3 Priority. |
| 2 | PX2 | External Interrupt 2 Priority. |
| 1 | - | Reserved. |
| 0 | PI2C | I2C Interrupt Priority. |

### EXTENDED INTERRUPT ENABLE 1

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | - | ENVM | ECPTF | ET3 | EBK | EPWM | ESPI |

Mnemonic: EIE1 Address: F9h

| BIT | NAME | FUNCTION |
|---|---|---|
| 7-6 | - | Reserved. |
| 5 | ENVM | NVM Interrupt Enable Bit.<br>0 = Disable NVM interrupt.<br>1 = Enable NVM interrupt. |
| 4 | ECPTF | Capture Interrupt Enable Bit.<br>0 = Disable External capture/reload interrupt.<br>1 = Enable External capture/reload interrupt. |
| 3 | ET3 | Timer 3 Interrupt Enable Bit.<br>0 = Disable Timer 3 Interrupt.<br>1 = Enable Timer 3 Interrupt. |
| 2 | EBK | Brake Interrupt Enable Bit.<br>0 = Brake interrupt disable.<br>1 = Brake interrupt enable. |
| 1 | EPWM | PWM Period Interrupt Enable Bit.<br>0 = PWM period system interrupts disabled.<br>1 = PWM period system interrupts enabled. |
| 0 | ESPI | Serial Peripheral Interrupt Enable Bit. Set the ESPI bit to 1 to request a hardware interrupt sequence each time the SPIF/MODF status flag is set.<br>0 = SPI system interrupts disabled.<br>1 = SPI system interrupts enabled. |

**EXTENDED INTERRUPT PRIORITY 1**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | - | PNVMI | PCPTF | PT3 | PBKF | PPWMF | PSPI |

Mnemonic: EIP1 Address: FAh

| BIT | NAME | FUNCTION |
|---|---|---|
| 7-6 | - | Reserved. |
| 5 | PNVMI | NVM interrupt Priority |
| 4 | PCPTF | Capture/reload Interrupt Priority. |
| 3 | PT3 | Timer 3 Interrupt Priority. |
| 2 | PBKF | PWM Brake Interrupt Priority. |
| 1 | PPWMF | PWM period Interrupt Priority. |
| 0 | PSPI | SPI Interrupt Priority. |

**INPUT CAPTURE 0/PULSE READ COUNTER LOW REGISTER**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CCL0.7/ PCNTL.7 | CCL0.6/ PCNTL.6 | CCL0.5/ PCNTL.5 | CCL0.4/ PCNTL.4 | CCL0.3/ PCNTL.3 | CCL0.2/ PCNTL.2 | CCL0.1/ PCNTL.1 | CCL0.0/ PCNTL.0 |

Mnemonic: CCL0/PCNTL Address: FBh

PCNTL must be read first before reading at PCNTH as reading PCNTL will latch the PLSCNTH automatically into PCNTH; otherwise inaccurate result is read when reading PCNTH first as it will not latch the PLSCNTL data into PCNTL.

**INPUT CAPTURE 0/PULSE READ COUNTER HIGH REGISTER**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CCH0.7/ PCNTH.7 | CCH0.6/ PCNTH.6 | CCH0.5/ PCNTH.5 | CCH0.4/ PCNTH.4 | CCH0.3/ PCNTH.3 | CCH0.2/ PCNTH.2 | CCH0.1/ PCNTH.1 | CCH0.0/ PCNTH.0 |

Mnemonic: CCH0/PCNTH Address: FCh

PCNTL must be read first before reading at PCNTH as reading PCNTL will latch the PLSCNTH automatically into PCNTH.

**INPUT CAPTURE 1/PULSE COUNTER LOW REGISTER**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CCL1.7/ PLSCNTL.7 | CCL1.6/ PLSCNTL.6 | CCL1.5/ PLSCNTL.5 | CCL1.4/ PLSCNTL.4 | CCL1.3/ PLSCNTL.3 | CCL1.2/ PLSCNTL.2 | CCL1.1/ PLSCNTL.1 | CCL1.0/ PLSCNTL.0 |

Mnemonic: CCL1/PLSCNTL Address: FDh

**INPUT CAPTURE 1/PULSE COUNTER HIGH REGISTER**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CCH1.7/ PLSCNTH.7 | CCH1.6/ PLSCNTH.6 | CCH1.5/ PLSCNTH.5 | CCH1.4/ PLSCNTH.4 | CCH1.3/ PLSCNTH.3 | CCH1.2/ PLSCNTH.2 | CCH1.1/ PLSCNTH.1 | CCH1.0/ PLSCNTH.0 |

Mnemonic: CCH1/PLSCNTH Address: FEh

**INTERRUPT CONTROL**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | - | INT5CT1 | INT5CT0 | INT4CT1 | INT4CT0 | INT3CT1 | INT3CT0 |

Mnemonic: INTCTRL Address: FFh

<table>
<tr><th>BIT</th><th>NAME</th><th>FUNCTION</th></tr>
<tr><td>7-6</td><td>-</td><td>Reserved.</td></tr>
<tr><td>5-4</td><td>INT5CT</td><td>Interrupt 5 edge select:
<table>
<tr><td>INT5CT1</td><td>INT5CT0</td><td>Description</td></tr>
<tr><td>0</td><td>0</td><td>Rising edge trigger.</td></tr>
<tr><td>0</td><td>1</td><td>Falling edge trigger.</td></tr>
<tr><td>1</td><td>0</td><td>Rising and falling edge trigger.</td></tr>
<tr><td>1</td><td>1</td><td>Reserved.</td></tr>
</table></td></tr>
<tr><td>3-2</td><td>INT4CT</td><td>Interrupt 4 edge select:
<table>
<tr><td>INT4CT1</td><td>INT4CT0</td><td>Description</td></tr>
<tr><td>0</td><td>0</td><td>Rising edge trigger.</td></tr>
<tr><td>0</td><td>1</td><td>Falling edge trigger.</td></tr>
<tr><td>1</td><td>0</td><td>Rising and falling edge trigger.</td></tr>
<tr><td>1</td><td>1</td><td>Reserved.</td></tr>
</table></td></tr>
<tr><td>1-0</td><td>INT3CT</td><td>Interrupt 3 edge select:
<table>
<tr><td>INT3CT1</td><td>INT3CT0</td><td>Description</td></tr>
<tr><td>0</td><td>0</td><td>Rising edge trigger.</td></tr>
<tr><td>0</td><td>1</td><td>Falling edge trigger.</td></tr>
<tr><td>1</td><td>0</td><td>Rising and falling edge trigger.</td></tr>
<tr><td>1</td><td>1</td><td>Reserved.</td></tr>
</table></td></tr>
</table>

## 8. INSTRUCTION SET

The W79E22X SERIES executes all the instructions of the standard 8051/52 family. The operations of these instructions, as well as their effects on flag and status bits, are exactly the same. However, the timing of these instructions is different in two ways. Firstly, the W79E22X SERIES machine cycle is four clock periods, while the standard-8051/52 machine cycle is twelve clock periods. Secondly, the W79E22X SERIES can fetch only once per machine cycle (i.e., four clocks per fetch), while the standard 8051/52 can fetch twice per machine cycle (i.e., six clocks per fetch).

The timing differences create an advantage for the W79E22X SERIES. There is only one fetch per machine cycle, so the number of machine cycles is usually equal to the number of operands in the instruction. (Jumps and calls do require an additional cycle to calculate the new address.) As a result, the W79E22X SERIES reduces the number of dummy fetches and wasted cycles, and therefore improves overall efficiency, compared to the standard 8051/52.

| OP-CODE | HEX CODE | BYTES | W79E22X SERIES MACHINE CYCLE | W79E22X SERIES CLOCK CYCLES | 8032 CLOCK CYCLES | W79E22X SERIES VS. 8032 SPEED RATIO |
|---|---|---|---|---|---|---|
| NOP | 00 | 1 | 1 | 4 | 12 | 3 |
| ADD A, R0 | 28 | 1 | 1 | 4 | 12 | 3 |
| ADD A, R1 | 29 | 1 | 1 | 4 | 12 | 3 |
| ADD A, R2 | 2A | 1 | 1 | 4 | 12 | 3 |
| ADD A, R3 | 2B | 1 | 1 | 4 | 12 | 3 |
| ADD A, R4 | 2C | 1 | 1 | 4 | 12 | 3 |
| ADD A, R5 | 2D | 1 | 1 | 4 | 12 | 3 |
| ADD A, R6 | 2E | 1 | 1 | 4 | 12 | 3 |
| ADD A, R7 | 2F | 1 | 1 | 4 | 12 | 3 |
| ADD A, @R0 | 26 | 1 | 1 | 4 | 12 | 3 |
| ADD A, @R1 | 27 | 1 | 1 | 4 | 12 | 3 |
| ADD A, direct | 25 | 2 | 2 | 8 | 12 | 1.5 |
| ADD A, #data | 24 | 2 | 2 | 8 | 12 | 1.5 |
| ADDC A, R0 | 38 | 1 | 1 | 4 | 12 | 3 |
| ADDC A, R1 | 39 | 1 | 1 | 4 | 12 | 3 |
| ADDC A, R2 | 3A | 1 | 1 | 4 | 12 | 3 |
| ADDC A, R3 | 3B | 1 | 1 | 4 | 12 | 3 |
| ADDC A, R4 | 3C | 1 | 1 | 4 | 12 | 3 |
| ADDC A, R5 | 3D | 1 | 1 | 4 | 12 | 3 |
| ADDC A, R6 | 3E | 1 | 1 | 4 | 12 | 3 |
| ADDC A, R7 | 3F | 1 | 1 | 4 | 12 | 3 |
| ADDC A, @R0 | 36 | 1 | 1 | 4 | 12 | 3 |
| ADDC A, @R1 | 37 | 1 | 1 | 4 | 12 | 3 |
| ADDC A, direct | 35 | 2 | 2 | 8 | 12 | 1.5 |

Continued

| OP-CODE | HEX CODE | BYTES | W79E22X SERIES MACHINE CYCLE | W79E22X SERIES CLOCK CYCLES | 8032 CLOCK CYCLES | W79E22X SERIES VS. 8032 SPEED RATIO |
|---|---|---|---|---|---|---|
| ADDC A, #data | 34 | 2 | 2 | 8 | 12 | 1.5 |
| SUBB A, R0 | 98 | 1 | 1 | 4 | 12 | 3 |
| SUBB A, R1 | 99 | 1 | 1 | 4 | 12 | 3 |
| SUBB A, R2 | 9A | 1 | 1 | 4 | 12 | 3 |
| SUBB A, R3 | 9B | 1 | 1 | 4 | 12 | 3 |
| SUBB A, R4 | 9C | 1 | 1 | 4 | 12 | 3 |
| SUBB A, R5 | 9D | 1 | 1 | 4 | 12 | 3 |
| SUBB A, R6 | 9E | 1 | 1 | 4 | 12 | 3 |
| SUBB A, R7 | 9F | 1 | 1 | 4 | 12 | 3 |
| SUBB A, @R0 | 96 | 1 | 1 | 4 | 12 | 3 |
| SUBB A, @R1 | 97 | 1 | 1 | 4 | 12 | 3 |
| SUBB A, direct | 95 | 2 | 2 | 8 | 12 | 1.5 |
| SUBB A, #data | 94 | 2 | 2 | 8 | 12 | 1.5 |
| INC A | 04 | 1 | 1 | 4 | 12 | 3 |
| INC R0 | 08 | 1 | 1 | 4 | 12 | 3 |
| INC R1 | 09 | 1 | 1 | 4 | 12 | 3 |
| INC R2 | 0A | 1 | 1 | 4 | 12 | 3 |
| INC R3 | 0B | 1 | 1 | 4 | 12 | 3 |
| INC R4 | 0C | 1 | 1 | 4 | 12 | 3 |
| INC R5 | 0D | 1 | 1 | 4 | 12 | 3 |
| INC R6 | 0E | 1 | 1 | 4 | 12 | 3 |
| INC R7 | 0F | 1 | 1 | 4 | 12 | 3 |
| INC @R0 | 06 | 1 | 1 | 4 | 12 | 3 |
| INC @R1 | 07 | 1 | 1 | 4 | 12 | 3 |
| INC direct | 05 | 2 | 2 | 8 | 12 | 1.5 |
| INC DPTR | A3 | 1 | 2 | 8 | 24 | 3 |
| DEC A | 14 | 1 | 1 | 4 | 12 | 3 |
| DEC R0 | 18 | 1 | 1 | 4 | 12 | 3 |
| DEC R1 | 19 | 1 | 1 | 4 | 12 | 3 |
| DEC R2 | 1A | 1 | 1 | 4 | 12 | 3 |
| DEC R3 | 1B | 1 | 1 | 4 | 12 | 3 |
| DEC R4 | 1C | 1 | 1 | 4 | 12 | 3 |
| DEC R5 | 1D | 1 | 1 | 4 | 12 | 3 |

Continued

| OP-CODE | HEX CODE | BYTES | W79E22X SERIES MACHINE CYCLE | W79E22X SERIES CLOCK CYCLES | 8032 CLOCK CYCLES | W79E22X SERIES VS. 8032 SPEED RATIO |
|---|---|---|---|---|---|---|
| DEC R6 | 1E | 1 | 1 | 4 | 12 | 3 |
| DEC R7 | 1F | 1 | 1 | 4 | 12 | 3 |
| DEC @R0 | 16 | 1 | 1 | 4 | 12 | 3 |
| DEC @R1 | 17 | 1 | 1 | 4 | 12 | 3 |
| DEC direct | 15 | 2 | 2 | 8 | 12 | 1.5 |
| MUL AB | A4 | 1 | 5 | 20 | 48 | 2.4 |
| DIV AB | 84 | 1 | 5 | 20 | 48 | 2.4 |
| DA A | D4 | 1 | 1 | 4 | 12 | 3 |
| ANL A, R0 | 58 | 1 | 1 | 4 | 12 | 3 |
| ANL A, R1 | 59 | 1 | 1 | 4 | 12 | 3 |
| ANL A, R2 | 5A | 1 | 1 | 4 | 12 | 3 |
| ANL A, R3 | 5B | 1 | 1 | 4 | 12 | 3 |
| ANL A, R4 | 5C | 1 | 1 | 4 | 12 | 3 |
| ANL A, R5 | 5D | 1 | 1 | 4 | 12 | 3 |
| ANL A, R6 | 5E | 1 | 1 | 4 | 12 | 3 |
| ANL A, R7 | 5F | 1 | 1 | 4 | 12 | 3 |
| ANL A, @R0 | 56 | 1 | 1 | 4 | 12 | 3 |
| ANL A, @R1 | 57 | 1 | 1 | 4 | 12 | 3 |
| ANL A, direct | 55 | 2 | 2 | 8 | 12 | 1.5 |
| ANL A, #data | 54 | 2 | 2 | 8 | 12 | 1.5 |
| ANL direct, A | 52 | 2 | 2 | 8 | 12 | 1.5 |
| ANL direct, #data | 53 | 3 | 3 | 12 | 24 | 2 |
| ORL A, R0 | 48 | 1 | 1 | 4 | 12 | 3 |
| ORL A, R1 | 49 | 1 | 1 | 4 | 12 | 3 |
| ORL A, R2 | 4A | 1 | 1 | 4 | 12 | 3 |
| ORL A, R3 | 4B | 1 | 1 | 4 | 12 | 3 |
| ORL A, R4 | 4C | 1 | 1 | 4 | 12 | 3 |
| ORL A, R5 | 4D | 1 | 1 | 4 | 12 | 3 |
| ORL A, R6 | 4E | 1 | 1 | 4 | 12 | 3 |
| ORL A, R7 | 4F | 1 | 1 | 4 | 12 | 3 |
| ORL A, @R0 | 46 | 1 | 1 | 4 | 12 | 3 |
| ORL A, @R1 | 47 | 1 | 1 | 4 | 12 | 3 |
| ORL A, direct | 45 | 2 | 2 | 8 | 12 | 1.5 |
| ORL A, #data | 44 | 2 | 2 | 8 | 12 | 1.5 |

Continued

| OP-CODE | HEX CODE | BYTES | W79E22X SERIES MACHINE CYCLE | W79E22X SERIES CLOCK CYCLES | 8032 CLOCK CYCLES | W79E22X SERIES VS. 8032 SPEED RATIO |
|---|---|---|---|---|---|---|
| ORL direct, A | 42 | 2 | 2 | 8 | 12 | 1.5 |
| ORL direct, #data | 43 | 3 | 3 | 12 | 24 | 2 |
| XRL A, R0 | 68 | 1 | 1 | 4 | 12 | 3 |
| XRL A, R1 | 69 | 1 | 1 | 4 | 12 | 3 |
| XRL A, R2 | 6A | 1 | 1 | 4 | 12 | 3 |
| XRL A, R3 | 6B | 1 | 1 | 4 | 12 | 3 |
| XRL A, R4 | 6C | 1 | 1 | 4 | 12 | 3 |
| XRL A, R5 | 6D | 1 | 1 | 4 | 12 | 3 |
| XRL A, R6 | 6E | 1 | 1 | 4 | 12 | 3 |
| XRL A, R7 | 6F | 1 | 1 | 4 | 12 | 3 |
| XRL A, @R0 | 66 | 1 | 1 | 4 | 12 | 3 |
| XRL A, @R1 | 67 | 1 | 1 | 4 | 12 | 3 |
| XRL A, direct | 65 | 2 | 2 | 8 | 12 | 1.5 |
| XRL A, #data | 64 | 2 | 2 | 8 | 12 | 1.5 |
| XRL direct, A | 62 | 2 | 2 | 8 | 12 | 1.5 |
| XRL direct, #data | 63 | 3 | 3 | 12 | 24 | 2 |
| CLR A | E4 | 1 | 1 | 4 | 12 | 3 |
| CPL A | F4 | 1 | 1 | 4 | 12 | 3 |
| RL A | 23 | 1 | 1 | 4 | 12 | 3 |
| RLC A | 33 | 1 | 1 | 4 | 12 | 3 |
| RR A | 03 | 1 | 1 | 4 | 12 | 3 |
| RRC A | 13 | 1 | 1 | 4 | 12 | 3 |
| SWAP A | C4 | 1 | 1 | 4 | 12 | 3 |
| MOV A, R0 | E8 | 1 | 1 | 4 | 12 | 3 |
| MOV A, R1 | E9 | 1 | 1 | 4 | 12 | 3 |
| MOV A, R2 | EA | 1 | 1 | 4 | 12 | 3 |
| MOV A, R3 | EB | 1 | 1 | 4 | 12 | 3 |
| MOV A, R4 | EC | 1 | 1 | 4 | 12 | 3 |
| MOV A, R5 | ED | 1 | 1 | 4 | 12 | 3 |
| MOV A, R6 | EE | 1 | 1 | 4 | 12 | 3 |
| MOV A, R7 | EF | 1 | 1 | 4 | 12 | 3 |
| MOV A, @R0 | E6 | 1 | 1 | 4 | 12 | 3 |
| MOV A, @R1 | E7 | 1 | 1 | 4 | 12 | 3 |

Continued

| OP-CODE | HEX CODE | BYTES | W79E22X SERIES MACHINE CYCLE | W79E22X SERIES CLOCK CYCLES | 8032 CLOCK CYCLES | W79E22X SERIES VS. 8032 SPEED RATIO |
|---|---|---|---|---|---|---|
| MOV A, direct | E5 | 2 | 2 | 8 | 12 | 1.5 |
| MOV A, #data | 74 | 2 | 2 | 8 | 12 | 1.5 |
| MOV R0, A | F8 | 1 | 1 | 4 | 12 | 3 |
| MOV R1, A | F9 | 1 | 1 | 4 | 12 | 3 |
| MOV R2, A | FA | 1 | 1 | 4 | 12 | 3 |
| MOV R3, A | FB | 1 | 1 | 4 | 12 | 3 |
| MOV R4, A | FC | 1 | 1 | 4 | 12 | 3 |
| MOV R5, A | FD | 1 | 1 | 4 | 12 | 3 |
| MOV R6, A | FE | 1 | 1 | 4 | 12 | 3 |
| MOV R7, A | FF | 1 | 1 | 4 | 12 | 3 |
| MOV R0, direct | A8 | 2 | 2 | 8 | 12 | 1.5 |
| MOV R1, direct | A9 | 2 | 2 | 8 | 12 | 1.5 |
| MOV R2, direct | AA | 2 | 2 | 8 | 12 | 1.5 |
| MOV R3, direct | AB | 2 | 2 | 8 | 12 | 1.5 |
| MOV R4, direct | AC | 2 | 2 | 8 | 12 | 1.5 |
| MOV R5, direct | AD | 2 | 2 | 8 | 12 | 1.5 |
| MOV R6, direct | AE | 2 | 2 | 8 | 12 | 1.5 |
| MOV R7, direct | AF | 2 | 2 | 8 | 12 | 1.5 |
| MOV R0, #data | 78 | 2 | 2 | 8 | 12 | 1.5 |
| MOV R1, #data | 79 | 2 | 2 | 8 | 12 | 1.5 |
| MOV R2, #data | 7A | 2 | 2 | 8 | 12 | 1.5 |
| MOV R3, #data | 7B | 2 | 2 | 8 | 12 | 1.5 |
| MOV R4, #data | 7C | 2 | 2 | 8 | 12 | 1.5 |
| MOV R5, #data | 7D | 2 | 2 | 8 | 12 | 1.5 |
| MOV R6, #data | 7E | 2 | 2 | 8 | 12 | 1.5 |
| MOV R7, #data | 7F | 2 | 2 | 8 | 12 | 1.5 |
| MOV @R0, A | F6 | 1 | 1 | 4 | 12 | 3 |
| MOV @R1, A | F7 | 1 | 1 | 4 | 12 | 3 |
| MOV @R0, direct | A6 | 2 | 2 | 8 | 12 | 1.5 |
| MOV @R1, direct | A7 | 2 | 2 | 8 | 12 | 1.5 |
| MOV @R0, #data | 76 | 2 | 2 | 8 | 12 | 1.5 |
| MOV @R1, #data | 77 | 2 | 2 | 8 | 12 | 1.5 |
| MOV direct, A | F5 | 2 | 2 | 8 | 12 | 1.5 |
| MOV direct, R0 | 88 | 2 | 2 | 8 | 12 | 1.5 |

Continued

| OP-CODE | HEX CODE | BYTES | W79E22X SERIES MACHINE CYCLE | W79E22X SERIES CLOCK CYCLES | 8032 CLOCK CYCLES | W79E22X SERIES VS. 8032 SPEED RATIO |
|---|---|---|---|---|---|---|
| MOV direct, R1 | 89 | 2 | 2 | 8 | 12 | 1.5 |
| MOV direct, R2 | 8A | 2 | 2 | 8 | 12 | 1.5 |
| MOV direct, R3 | 8B | 2 | 2 | 8 | 12 | 1.5 |
| MOV direct, R4 | 8C | 2 | 2 | 8 | 12 | 1.5 |
| MOV direct, R5 | 8D | 2 | 2 | 8 | 12 | 1.5 |
| MOV direct, R6 | 8E | 2 | 2 | 8 | 12 | 1.5 |
| MOV direct, R7 | 8F | 2 | 2 | 8 | 12 | 1.5 |
| MOV direct, @R0 | 86 | 2 | 2 | 8 | 12 | 1.5 |
| MOV direct, @R1 | 87 | 2 | 2 | 8 | 12 | 1.5 |
| MOV direct, direct | 85 | 3 | 3 | 12 | 24 | 2 |
| MOV direct, #data | 75 | 3 | 3 | 12 | 24 | 2 |
| MOV DPTR, #data 16 | 90 | 3 | 3 | 12 | 24 | 2 |
| MOVC A, @A+DPTR | 93 | 1 | 2 | 8 | 24 | 3 |
| MOVC A, @A+PC | 83 | 1 | 2 | 8 | 24 | 3 |
| MOVX A, @R0 | E2 | 1 | 2 - 9 | 8 - 36 | 24 | 3 - 0.66 |
| MOVX A, @R1 | E3 | 1 | 2 - 9 | 8 - 36 | 24 | 3 - 0.66 |
| MOVX A, @DPTR | E0 | 1 | 2 - 9 | 8 - 36 | 24 | 3 - 0.66 |
| MOVX @R0, A | F2 | 1 | 2 - 9 | 8 - 36 | 24 | 3 - 0.66 |
| MOVX @R1, A | F3 | 1 | 2 - 9 | 8 - 36 | 24 | 3 - 0.66 |
| MOVX @DPTR, A | F0 | 1 | 2 - 9 | 8 - 36 | 24 | 3 - 0.66 |
| PUSH direct | C0 | 2 | 2 | 8 | 24 | 3 |
| POP direct | D0 | 2 | 2 | 8 | 24 | 3 |
| XCH A, R0 | C8 | 1 | 1 | 4 | 12 | 3 |
| XCH A, R1 | C9 | 1 | 1 | 4 | 12 | 3 |
| XCH A, R2 | CA | 1 | 1 | 4 | 12 | 3 |
| XCH A, R3 | CB | 1 | 1 | 4 | 12 | 3 |
| XCH A, R4 | CC | 1 | 1 | 4 | 12 | 3 |
| XCH A, R5 | CD | 1 | 1 | 4 | 12 | 3 |
| XCH A, R6 | CE | 1 | 1 | 4 | 12 | 3 |
| XCH A, R7 | CF | 1 | 1 | 4 | 12 | 3 |
| XCH A, @R0 | C6 | 1 | 1 | 4 | 12 | 3 |

Continued

| OP-CODE | HEX CODE | BYTES | W79E22X SERIES MACHINE CYCLE | W79E22X SERIES CLOCK CYCLES | 8032 CLOCK CYCLES | W79E22X SERIES VS. 8032 SPEED RATIO |
|---|---|---|---|---|---|---|
| XCH A, @R1 | C7 | 1 | 1 | 4 | 12 | 3 |
| XCHD A, @R0 | D6 | 1 | 1 | 4 | 12 | 3 |
| XCHD A, @R1 | D7 | 1 | 1 | 4 | 12 | 3 |
| XCH A, direct | C5 | 2 | 2 | 8 | 12 | 1.5 |
| CLR C | C3 | 1 | 1 | 4 | 12 | 3 |
| CLR bit | C2 | 2 | 2 | 8 | 12 | 1.5 |
| SETB C | D3 | 1 | 1 | 4 | 12 | 3 |
| SETB bit | D2 | 2 | 2 | 8 | 12 | 1.5 |
| CPL C | B3 | 1 | 1 | 4 | 12 | 3 |
| CPL bit | B2 | 2 | 2 | 8 | 12 | 1.5 |
| ANL C, bit | 82 | 2 | 2 | 8 | 24 | 3 |
| ANL C, /bit | B0 | 2 | 2 | 6 | 24 | 3 |
| ORL C, bit | 72 | 2 | 2 | 8 | 24 | 3 |
| ORL C, /bit | A0 | 2 | 2 | 6 | 24 | 3 |
| MOV C, bit | A2 | 2 | 2 | 8 | 12 | 1.5 |
| MOV bit, C | 92 | 2 | 2 | 8 | 24 | 3 |
| ACALL addr11 | 71, 91, B1, 11, 31, 51, D1, F1 | 2 | 3 | 12 | 24 | 2 |
| LCALL addr16 | 12 | 3 | 4 | 16 | 24 | 1.5 |
| RET | 22 | 1 | 2 | 8 | 24 | 3 |
| RETI | 32 | 1 | 2 | 8 | 24 | 3 |
| AJMP ADDR11 | 01, 21, 41, 61, 81, A1, C1, E1 | 2 | 3 | 12 | 24 | 2 |
| LJMP addr16 | 02 | 3 | 4 | 16 | 24 | 1.5 |
| JMP @A+DPTR | 73 | 1 | 2 | 6 | 24 | 3 |
| SJMP rel | 80 | 2 | 3 | 12 | 24 | 2 |
| JZ rel | 60 | 2 | 3 | 12 | 24 | 2 |
| JNZ rel | 70 | 2 | 3 | 12 | 24 | 2 |
| JC rel | 40 | 2 | 3 | 12 | 24 | 2 |
| JNC rel | 50 | 2 | 3 | 12 | 24 | 2 |
| JB bit, rel | 20 | 3 | 4 | 16 | 24 | 1.5 |
| JNB bit, rel | 30 | 3 | 4 | 16 | 24 | 1.5 |

Continued

| OP-CODE | HEX CODE | BYTES | W79E22X SERIES MACHINE CYCLE | W79E22X SERIES CLOCK CYCLES | 8032 CLOCK CYCLES | W79E22X SERIES VS. 8032 SPEED RATIO |
|---|---|---|---|---|---|---|
| JBC bit, rel | 10 | 3 | 4 | 16 | 24 | 1.5 |
| CJNE A, direct, rel | B5 | 3 | 4 | 16 | 24 | 1.5 |
| CJNE A, #data, rel | B4 | 3 | 4 | 16 | 24 | 1.5 |
| CJNE @R0, #data, rel | B6 | 3 | 4 | 16 | 24 | 1.5 |
| CJNE @R1, #data, rel | B7 | 3 | 4 | 16 | 24 | 1.5 |
| CJNE R0, #data, rel | B8 | 3 | 4 | 16 | 24 | 1.5 |
| CJNE R1, #data, rel | B9 | 3 | 4 | 16 | 24 | 1.5 |
| CJNE R2, #data, rel | BA | 3 | 4 | 16 | 24 | 1.5 |
| CJNE R3, #data, rel | BB | 3 | 4 | 16 | 24 | 1.5 |
| CJNE R4, #data, rel | BC | 3 | 4 | 16 | 24 | 1.5 |
| CJNE R5, #data, rel | BD | 3 | 4 | 16 | 24 | 1.5 |
| CJNE R6, #data, rel | BE | 3 | 4 | 16 | 24 | 1.5 |
| CJNE R7, #data, rel | BF | 3 | 4 | 16 | 24 | 1.5 |
| DJNZ R0, rel | D8 | 2 | 3 | 12 | 24 | 2 |
| DJNZ R1, rel | D9 | 2 | 3 | 12 | 24 | 2 |
| DJNZ R5, rel | DD | 2 | 3 | 12 | 24 | 2 |
| DJNZ R2, rel | DA | 2 | 3 | 12 | 24 | 2 |
| DJNZ R3, rel | DB | 2 | 3 | 12 | 24 | 2 |
| DJNZ R4, rel | DC | 2 | 3 | 12 | 24 | 2 |
| DJNZ R6, rel | DE | 2 | 3 | 12 | 24 | 2 |
| DJNZ R7, rel | DF | 2 | 3 | 12 | 24 | 2 |
| DJNZ direct, rel | D5 | 3 | 4 | 16 | 24 | 1.5 |

Table 8-1: Instruction Set for W79E22X SERIES

## 8.1 Instruction Timing

This section is important because some applications use software instructions to generate timing delays. It also provides more information about timing differences between the W79E22X SERIES and the standard 8051/52.

In W79E22X SERIES, each machine cycle is four clock periods long. Each clock period is called a state, and each machine cycle consists of four states: C1, C2 C3 and C4, in order. Both clock edges are used for internal timing, so the duty cycle of the clock should be as close to 50% as possible.

The W79E22X SERIES does one op-code fetch per machine cycle, so, in most instructions, the number of machine cycles required is equal to the number of bytes in the instruction. There are 256 available op-codes. 128 of them are single-cycle instructions, so many op-codes are executed in just four clock periods. Some of the other op-codes are two-cycle instructions, and most of these have two-byte op-codes. However, there are some instructions that have one-byte instructions yet take two cycles to execute. One important example is the MOVX instruction.

In the standard 8051/52, the MOVX instruction is always two machine cycles long. However, in the W79E22X SERIES, the duration of this instruction is controlled by the software. It can vary from two to nine machine cycles long, and, $\overline{RD}$ and $\overline{WR}$ strobe lines are elongated proportionally. This is called stretching, and it gives a lot of flexibility when accessing fast and slow peripherals. It also reduces the amount of external circuitry and software overhead.

The rest of the instructions are three-, four- or five-cycle instructions. At the end of this section, there are timing diagrams that provide an example of each type of instruction (single-cycle, two-cycle, …).

In summary, there are five types of instructions in the W79E22X SERIES, based on the number of machine cycles, and each machine cycle is four clock periods long. The standard 8051/52 has only three types of instructions, based on the number of machine cycles, but each machine cycle is twelve clock periods long. As a result, even though the number of categories is higher, each instruction in the W79E22X SERIES runs 1.5 to 3 times faster, based on the number of clock periods, than it does in the standard 8051/52.

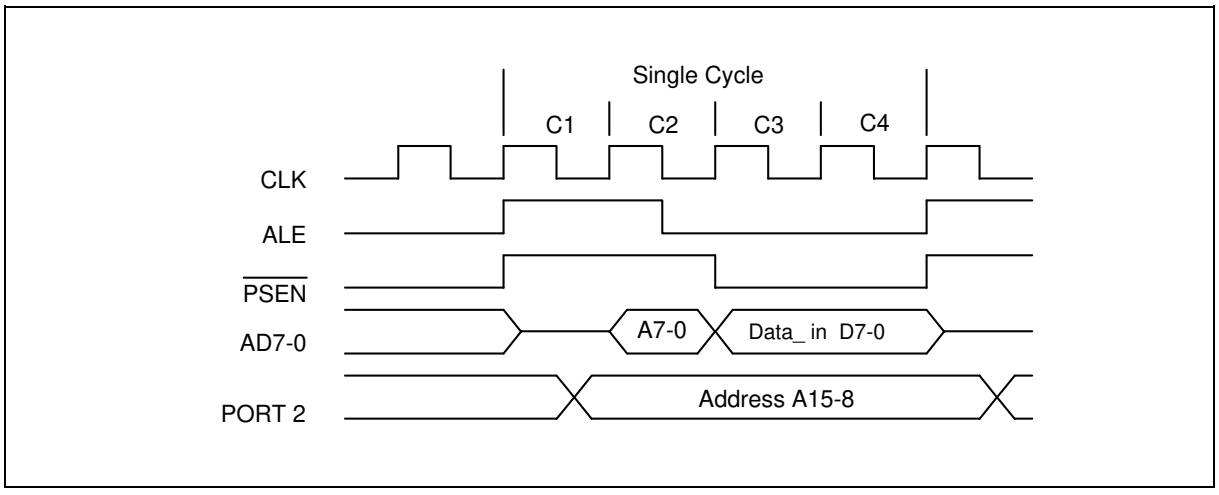



Figure 8-1: Single Cycle Instruction Timing

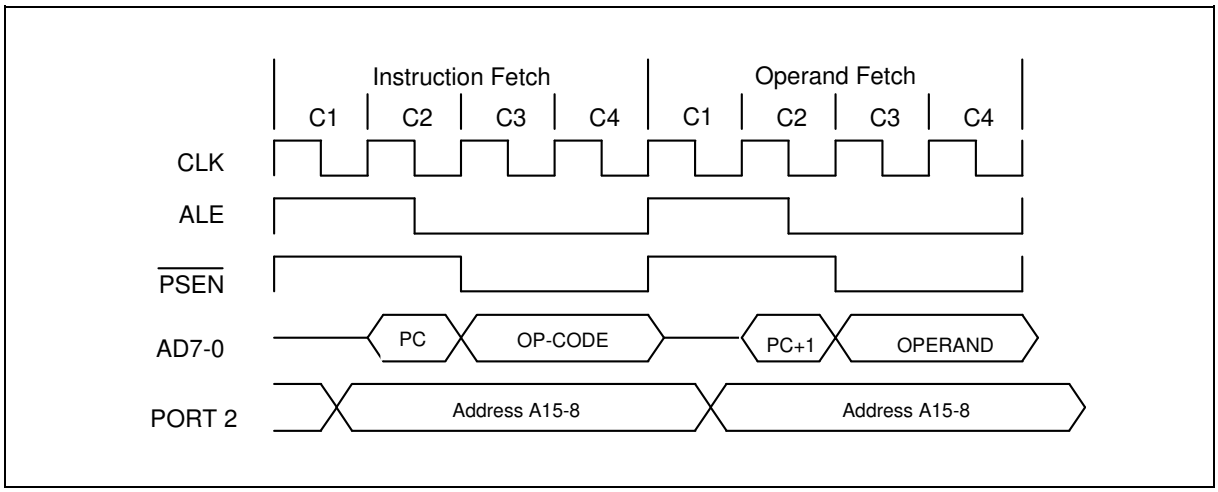



Figure 8-2: Two Cycles Instruction Timing

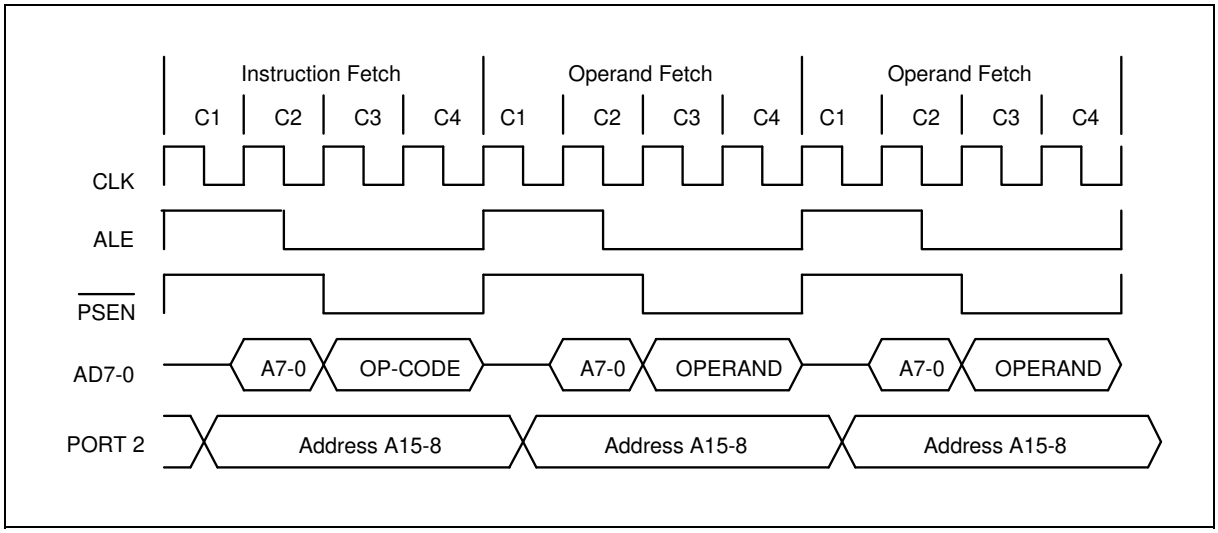



Figure 8-3: Three Cycles Instruction Timing

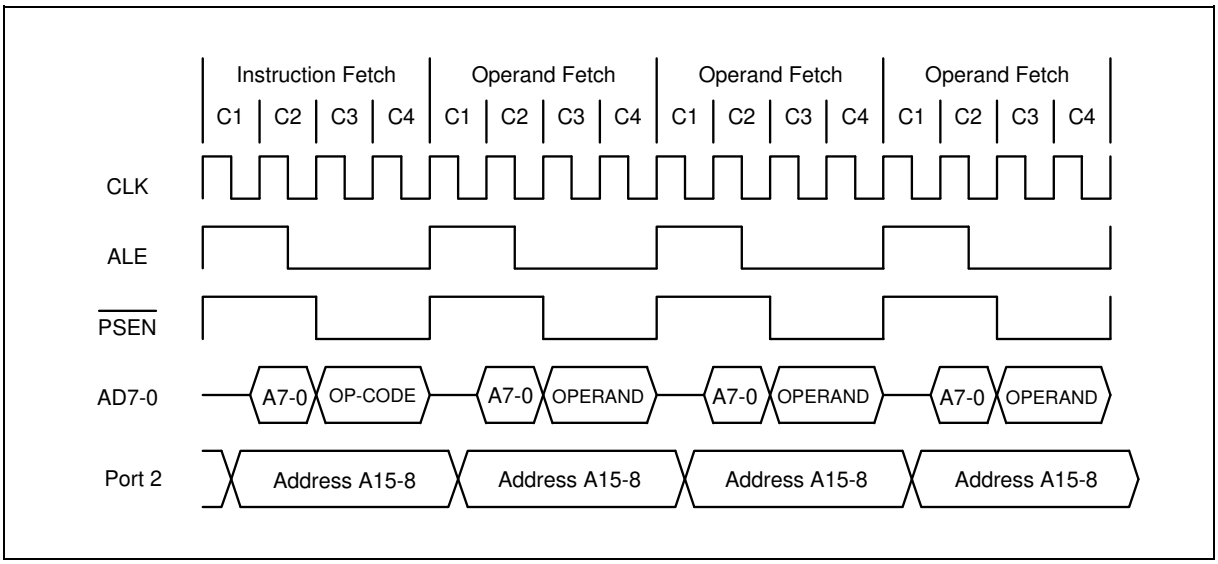



Figure 8-4: Four Cycles Instruction Timing

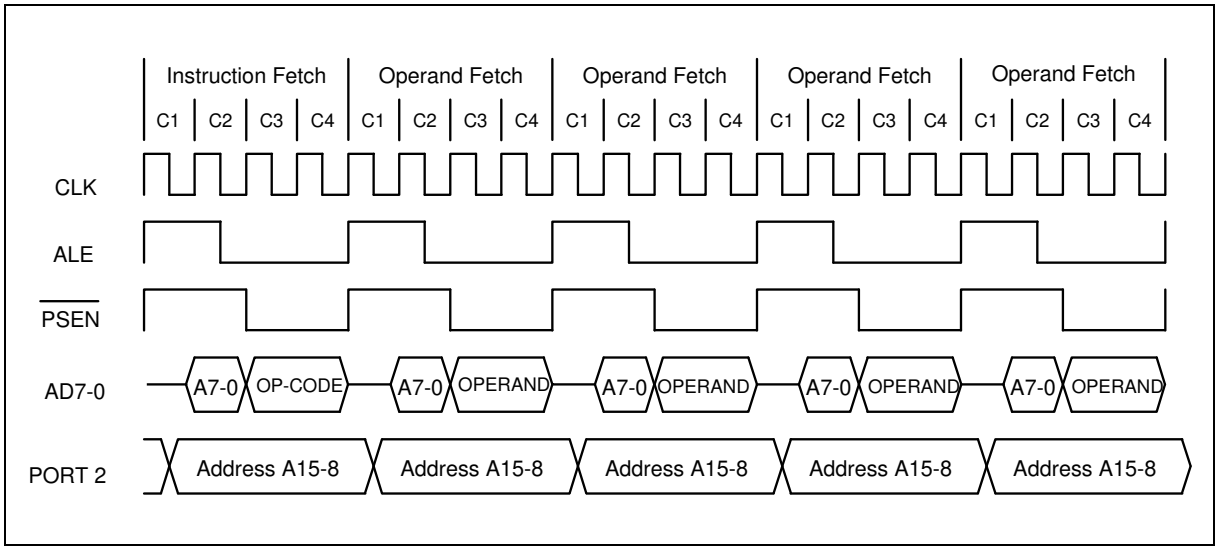



Figure 8-5: Five Cycles Instruction Timing

### 8.1.1 External Data Memory Access Timing

The timing for the MOVX instruction is another feature of the W79E22X SERIES. In the standard 8051/52, the MOVX instruction has a fixed execution time of 2 machine cycles. However, in W79E22X SERIES, the duration of the access can be controlled by the user.

The instruction starts off as a normal op-code fetch that takes four clocks. In the next machine cycle, W79E22X SERIES puts out the external memory address, and the actual access occurs. The user can control the duration of this access by setting the stretch value in CKCON, bits 2 – 0. As shown in the table below, these three bits can range from zero to seven, resulting in MOVX instructions that take two to nine machine cycles. The default value is one, resulting in a MOVX instruction of three machine cycles.

Stretching only affects the MOVX instruction. There is no effect on any other instruction or its timing; it is as if the state of the CPU is held for the desired period. The timing waveforms when the stretch value is zero, one, and two are shown below.

| M2 | M1 | M0 | MACHINE CYCLES | $\overline{RD}$ OR $\overline{WR}$ STROBE WIDTH IN CLOCKS | $\overline{RD}$ OR $\overline{WR}$ STROBE WIDTH @ 25 MHZ | $\overline{RD}$ OR $\overline{WR}$ STROBE WIDTH @ 40 MHZ |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 2 | 2 | 80 nS | 50 nS |
| 0 | 0 | 1 | 3 (default) | 4 | 160 nS | 100 nS |
| 0 | 1 | 0 | 4 | 8 | 320 nS | 200 nS |
| 0 | 1 | 1 | 5 | 12 | 480 nS | 300 nS |
| 1 | 0 | 0 | 6 | 16 | 640 nS | 400 nS |
| 1 | 0 | 1 | 7 | 20 | 800 nS | 500 nS |
| 1 | 1 | 0 | 8 | 24 | 960 nS | 600 nS |
| 1 | 1 | 1 | 9 | 28 | 1120 nS | 700 nS |

Table 8-2: Data Memory Cycle Stretch Values

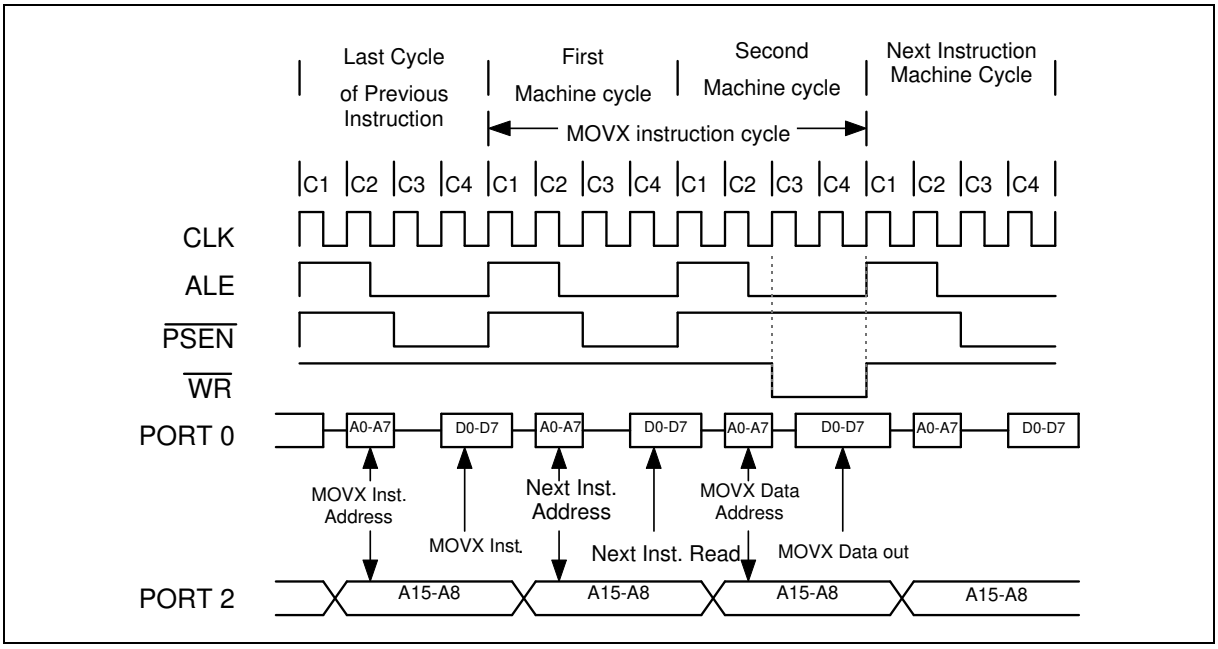



Figure 8-6: Data Memory Write with Stretch Value = 0

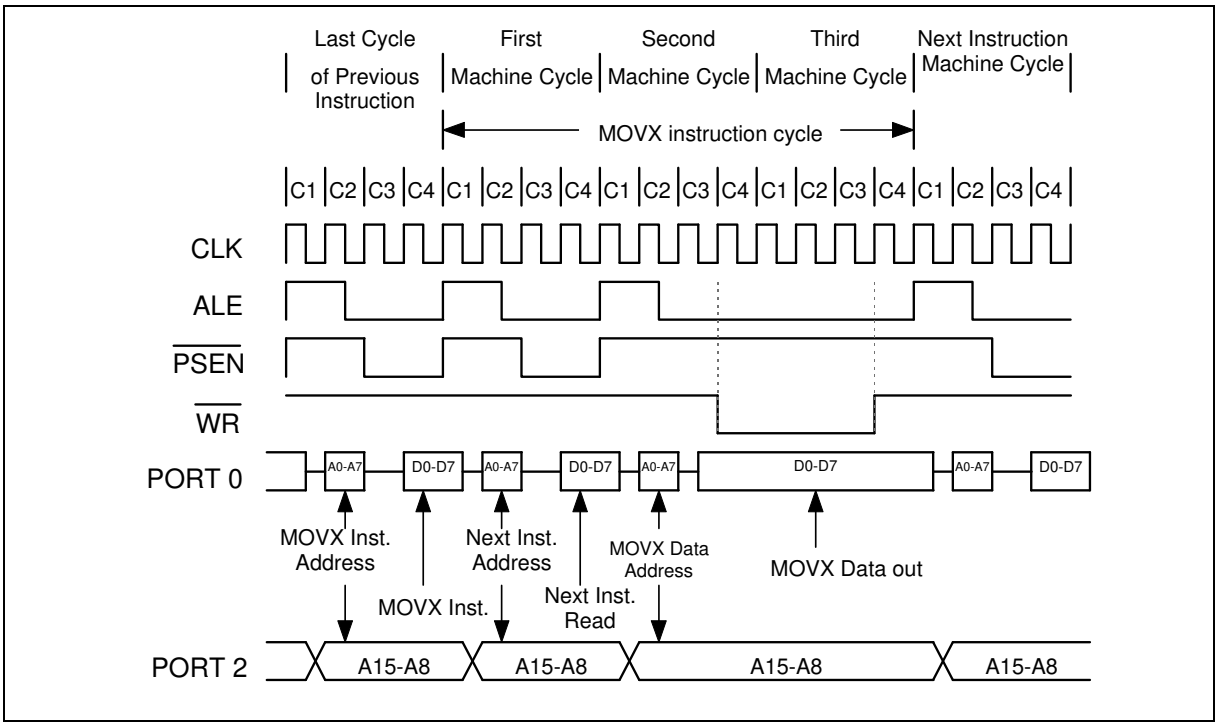



Figure 8-7: Data Memory Write with Stretch Value = 1

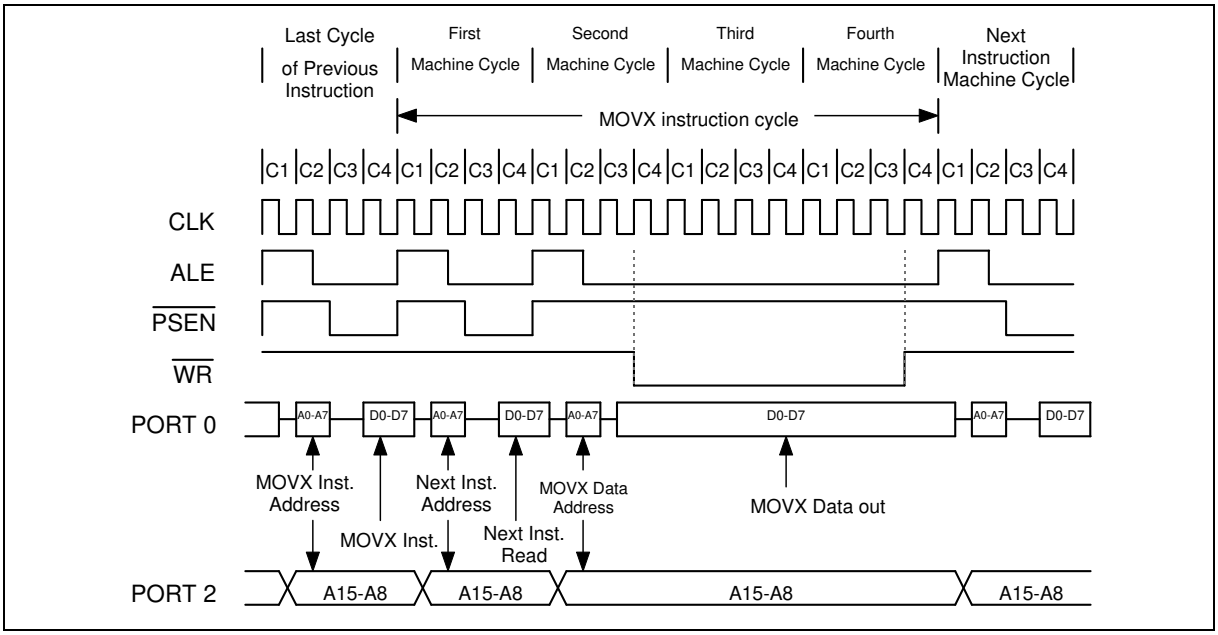



Figure 8-8: Data Memory Write with Stretch Value = 2

# 9. POWER MANAGEMENT

The W79E22X SERIES provides idle mode and power-down mode to control power consumption. These modes are discussed in the next two sections.

## 9.1 Idle Mode

The user can put the device into idle mode by writing 1 to the bit PCON.0. The instruction that sets the idle bit is the last instruction that will be executed before the device goes into Idle Mode. In the Idle mode, the clock to the CPU is halted, but not to the Interrupt, Timer, Watchdog timer, PWM, ADC and Serial ports blocks. This forces the CPU state to be frozen; the Program counter, the Stack Pointer, the Program Status Word, the Accumulator and the other registers hold their contents. The ALE and $\overline{PSEN}$ pins are held high during the Idle state. The port pins hold the logical states they had at the time Idle was activated. The Idle mode can be terminated in two ways. Since the interrupt controller is still active, the activation of any enabled interrupt can wake up the processor. This will automatically clear the Idle bit, terminate the Idle mode, and the Interrupt Service Routine (ISR) will be executed. After the ISR, execution of the program will continue from the instruction which put the device into Idle mode.

The Idle mode can also be exited by activating the reset. The device can be put into reset either by applying a high on the external RST pin, a Power on reset condition or a Watchdog timer reset. The external reset pin has to be held high for at least two machine cycles i.e. 8 clock periods to be recognized as a valid reset. In the reset condition the program counter is reset to 0000h and all the SFRs are set to the reset condition. Since the clock is already running there is no delay and execution starts immediately. In the Idle mode, the Watchdog timer continues to run, and if enabled, a time-out will cause a watchdog timer interrupt which will wake up the device. The software must reset the Watchdog timer in order to preempt the reset which will occur after 512 clock periods of the time-out. When the device is exiting from an Idle mode with a reset, the instruction following the one which put the device into Idle mode is not executed. So there is no danger of unexpected writes.

## 9.2 Power Down Mode

The device can be put into Power Down mode by writing 1 to bit PCON.1. The instruction that does this will be the last instruction to be executed before the device goes into Power Down mode. In the Power Down mode, all the clocks are stopped and the device comes to a halt. All activity is completely stopped and the power consumption is reduced to the lowest possible value. In this state the ALE and $\overline{PSEN}$ pins are pulled low (if PWDNH=0). The port pins output the values held by their respective SFRs.

The device will exit the Power Down mode with a reset or by an external interrupt pin enabled (external interrupts 0 and 1). An external reset can be used to exit the Power down state. The high on RST pin terminates the Power Down mode, and restarts the clock. The program execution will restart from 0000h. In the Power down mode, the clock is stopped, so the Watchdog timer cannot be used to provide the reset to exit Power down mode.

The device can be waken up from the Power Down mode by forcing an external interrupt pin activation, provided the corresponding interrupt is enabled, while the global enable (EA) bit is set. If these conditions are met, then either a low-level or a falling-edge at external interrupt pin will re-start the oscillator. The device will then execute the interrupt service routine for the corresponding external interrupt. After the interrupt service routine is completed, the program execution returns to the instruction after one which put the device into Power Down mode and continues from there.

| MODE | PROGRAM MEMORY | ALE | $\overline{PSEN}$ | PORT0 | PORT1 | PORT2 | PORT3 | PORT4 | PORT5 |
|---|---|---|---|---|---|---|---|---|---|
| Idle | Internal | 1 | 1 | Data | Data | Data | Data | Data | Data |
| Idle | External | 1 | 1 | Float | Data | Address | Data | Data | Data |
| Power Down | Internal | 0 [1]<br>1 [2] | 0 [1]<br>1 [2] | Data | Data | Data | Data | Data | Data |
| Power Down | External | 0 [1]<br>1 [2] | 0 [1]<br>1 [2] | Float | Data | Data | Data | Data | Data |

Table 9-1: Status of external pins during Idle and Power Down

**Note:**

1. When PWDNH=0.
2. When PWDNH=1.

# 10. RESET CONDITIONS

The user has several hardware related options for placing the W79E22X SERIES into reset condition. In general, most register bits go to their reset value irrespective of the reset condition, but there are a few flags whose state depends on the source of reset. The user can use these flags to determine the cause of reset using software. There are three ways of putting the device into reset state. They are External reset, Power-On Reset and Watchdog reset. In general, most registers return to their default values regardless of the source of the reset, but a couple flags depend on the source. As a result, the user can use these flags to determine the cause of the reset.

The rest of this section discusses the three causes of reset and then elaborates on the reset state.

## 10.1 Sources of reset

### 10.1.1 External Reset

The device samples the RST pin every machine cycle during state C4. The RST pin must be held high for at least two machine cycles before the reset circuitry applies an internal reset signal. Thus, this reset is a synchronous operation and requires the clock to be running.

The device remains in the reset state as long as RST is one and remains there up to two machine cycles after RST is deactivated. Then, the device begins program execution at 0000h. There are no flags associated with the external reset, but, since the other two reset sources do have flags, the external reset is the cause if those flags are clear.

### 10.1.2 Power-On Reset (POR)

If the power supply falls below $V_{rst}$, the device goes into the reset state. When the power supply returns to proper levels, the device performs a power-on reset and sets the POR flag. The software should clear the POR flag, or it will be difficult to determine the source of future resets.

### 10.1.3 Watchdog Timer Reset

The Watchdog Timer is a free-running timer with programmable time-out intervals. The program must clear the Watchdog Timer before the time-out interval is reached to restart the count. If the time-out interval is reached, an interrupt flag is set. 512 clocks later, if the Watchdog Reset is enabled and the Watchdog Timer has not been cleared, the Watchdog Timer generates a reset. The reset condition is maintained by the hardware for two machine cycles, and the WTRF bit in WDCON is set. Afterwards, the device begins program execution at 0000h.

## 10.2 Reset State

When the device is reset, most registers return to their initial state. The Watchdog Timer is disabled if the reset source was a power-on reset. The port registers are set to FFh, which puts most of the port pins in a high state and makes Port 0 float (as it does not have on-chip pull-up resistors). The Program Counter is set to 0000h, and the stack pointer is reset to 07h. After this, the device remains in the reset state as long as the reset conditions are satisfied.

Reset does not affect the on-chip RAM, however, so RAM is preserved as long as VDD remains above approximately 2 V, the minimum operating voltage for the RAM. If VDD falls below 2 V, the RAM contents are also lost. In either case, the stack pointer is always reset, so the stack contents are lost.

The WDCON SFR bits are set/cleared in reset condition depends on the source of the reset. The WDCON SFR is set to a 0x0x0xx0b on an external reset. WTRF is set to a 1 on a Watchdog timer reset, but to a 0 on power on/down resets. WTRF is not altered by an external reset. POR is set to 1 by a power-on reset. EWT is cleared to 0 on a Power-on reset and unaffected by other resets. All the bits in this SFR have unrestricted read access. POR, WDIF, EWT and RWT bits require Timed Access (TA) procedure to write. The remaining bits have unrestricted write accesses. Please refer TA register description. Table below lists the different reset values for WDCON due to different sources of reset.

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| WDCON | Watch-Dog Control | D8H | (DF)<br>- | (DE)<br>POR | (DD)<br>- | (DC)<br>- | (DB)<br>WDIF | (DA)<br>WTRF | (D9)<br>EWT | (D8)<br>RWT | x0xx 0000B External reset<br>x0xx 0100B Watchdog reset<br>x1xx 0000B Power on reset |

# 11. INTERRUPTS

The device has a four priority level interrupt structure with 20 interrupt sources. Each of the interrupt sources has an individual priority bit, flag, interrupt vector and enable bit. In addition, all the interrupts can be globally enabled or disabled.

## 11.1 Interrupt Sources

The External Interrupts $\overline{INT0}$ and $\overline{INT1}$ can be either edge triggered or level triggered, depending on bits IT0 and IT1. The bits IE0 and IE1 in the TCON register are the flags which are checked to generate the interrupt. In the edge triggered mode, the INTx inputs are sampled in every machine cycle. If the sample is high in one cycle and low in the next, then a high to low transition is detected and the interrupts request flag IEx in TCON is set. The flag bit requests the interrupt. Since the external interrupts are sampled every machine cycle, they have to be held high or low for at least one complete machine cycle. The IEx flag is automatically cleared when the service routine is called. If the level triggered mode is selected, then the requesting source has to hold the pin low till the interrupt is serviced. The IEx flag will not be cleared by the hardware on entering the service routine. If the interrupt continues to be held low even after the service routine is completed, then the processor may acknowledge another interrupt request from the same source. Note that the external interrupts INT2 are edge trigger only. By default, the individual interrupt flag corresponding to external interrupt 2 to 5 must be cleared manually by software. It can be configured with hardware cleared by setting the corresponding bit HCx in T2MOD register. For instance, if HC2 is set hardware will clear IE2 flag after program enters the interrupt 2 service routine. While for INT3 to INT5 can detect the rising, falling or both edges which function are selectable by software located in INTCTRL [5:0] register. The Timer 0 and 1 Interrupts are generated by the TF0 and TF1 flags. These flags are set by the overflow in the Timer 0 and Timer 1. The TF0 and TF1 flags are automatically cleared by the hardware when the timer interrupt is serviced. The Timer 2 interrupt is generated by a logical OR of the TF2 and the EXF2 flags. These flags are set by overflow or capture/reload events in the timer 2 operation. The hardware does not clear these flags when a timer 2 interrupt is executed. Software has to resolve the cause of the interrupt between TF2 and EXF2 and clear the appropriate flag.

When ADC conversion is completed hardware will set flag ADCI to logic high to request ADC interrupt if bit EADC (IE.6) is in high state. ADCI is cleared by software only.

The I2C function can generate interrupt, if EI2C and EA bits are enabled, when SI Flag is set due to a new I2C status code is generated, SI flag is generated by hardware and must be cleared by software.

The Watchdog timer can be used as a system monitor or a simple timer. In either case, when the time-out count is reached, the Watchdog timer interrupt flag WDIF (WDCON.3) is set. If the interrupt is enabled by the enable bit EIE.4, then an interrupt will occur.

All the bits that generate interrupts can be set or reset by hardware, and thereby software initiated interrupts can be generated. Each of the individual interrupts can be enabled or disabled by setting or clearing a bit in the IE SFR. IE also has a global enable/disable bit EA, which can be cleared to disable all interrupts.

## 11.2 Priority Level Structure

There are four priority levels for the interrupts; highest, high, low and lowest. The other interrupt source can be individually set to either high or low levels. Naturally, a higher priority interrupt cannot be interrupted by a lower priority interrupt. However there exists a predefined hierarchy amongst the interrupts themselves. This hierarchy comes into play when the interrupt controller has to resolve simultaneous requests having the same priority level. This hierarchy is defined as shown below; the interrupts are numbered starting from the highest priority to the lowest.

| SOURCE | FLAG | VECTOR ADDRESS | FLAG CLEARED BY | PRIORITY LEVEL |
|---|---|---|---|---|
| External Interrupt 0 | IE0 | 0003H | Hardware, Follow the inverse of pin | 1(highest) |
| Timer 0 Overflow | TF0 | 000BH | Hardware, software | 2 |
| External Interrupt 1 | IE1 | 0013H | Hardware, Follow the inverse of pin | 3 |
| Timer 1 Overflow | TF1 | 001BH | Hardware, software | 4 |
| Serial Port | RI + TI | 0023H | Software | 5 |
| Timer 2 Overflow | TF2 + EXF2 | 002BH | Software | 6 |
| A/D Converter | ADCI | 0033H | Software | 7 |
| I2C Channel | I2C1 SI | 003BH | Software | 8 |
| Serial Port 1 | RI_1 + TI_1 | 007BH | Software | 9 |
| SPI interrupt | SPIF + MODF + SPIOVF | 0083H | Software | 10 |
| External Interrupt 2 | IE2 | 0043H | Hardware, software | 11 |
| External Interrupt 3 | IE3 | 004BH | Hardware, software | 12 |
| External Interrupt 4 | IE4 | 0053H | Hardware, software | 13 |
| External Interrupt 5 | IE5 | 005BH | Hardware, software | 14 |
| PWM Period | PWMF | 0073H | Software | 15 |
| PWM Brake | BKF | 006BH | Software | 16 |
| Timer 3 Overflow | TF3 | 008BH | Software | 17 |
| Capture Input/Direction Interrupt/QEI | CPTF0/QEIF+ CPTF1/DIRF+ CPTF2 | 0093H | Software | 18 |
| NVM Interrupt | NVMF | 009BH | Software | 19 |
| Watchdog Timer | WDIF | 0063H | Software | 20 |

Table 11- 1: Priority structure of interrupts

The interrupt flags are sampled every machine cycle. In the same machine cycle, the sampled interrupts are polled and their priority is resolved. If certain conditions are met, the hardware will execute an internally generated LCALL instruction which will vector the process to the appropriate interrupt vector address. The conditions for generating the LCALL are;

1. An interrupt of equal or higher priority is not currently being serviced.

2. The current polling cycle is the last machine cycle of the instruction currently being executed.

3. The current instruction does not involve a write to IE, EIE, EIE1, IP, EIP, EIP1, IPH, EIPH or EIP1H registers and is not a RETI.

If any of these conditions are not met, then the LCALL will not be generated. The polling cycle is repeated every machine cycle, with the interrupts sampled in the same machine cycle. If an interrupt

flag is active in one cycle but not responded to, and is not active when the above conditions are met, the denied interrupt will not be serviced. This means that active interrupts are not remembered; every polling cycle is new.

The processor responds to a valid interrupt by executing an LCALL instruction to the appropriate service routine. This may or may not clear the flag which caused the interrupt. In case of Timer interrupts, the TF0 or TF1 flags are cleared by hardware whenever the processor vectors to the appropriate timer service routine. In case of external interrupt, INT0 and INT1, the flags are cleared only if they are edge triggered. In case of Serial interrupts, the flags are not cleared by hardware. In the case of Timer 2 interrupt, the flags are not cleared by hardware. The Watchdog timer interrupt flag WDIF has to be cleared by software. The hardware LCALL behaves exactly like the software LCALL instruction. This instruction saves the Program Counter contents onto the Stack, but does not save the Program Status Word PSW. The PC is reloaded with the vector address of that interrupt which caused the LCALL. These address of vector for the different sources are shown in Table 11- 1: Priority structure of interrupts.

| PRIORITY BITS | | INTERRUPT PRIORITY LEVEL |
|---|---|---|
| IPH/EIPH/EIP1H | IP/EIP/EIP1 | |
| 0 | 0 | Level 0 (lowest priority) |
| 0 | 1 | Level 1 |
| 1 | 0 | Level 2 |
| 1 | 1 | Level 3 (highest priority) |

Table 11- 2: Four-level interrupt priority

Each interrupt source can be individually programmed to one of four priority levels by setting or clearing bits in the IP, IPH, EIP, EIPH, EIP1 and EIP1H registers. An interrupt service routine in progress can be interrupted by a higher priority interrupt, but not by another interrupt of the same or lower priority. The highest priority interrupt service cannot be interrupted by any other interrupt source. So, if two requests of different priority levels are received simultaneously, the request of higher priority level is serviced.

If requests of the same priority level are received simultaneously, an internal polling sequence determines which request is serviced. This is called the arbitration ranking. Note that the arbitration ranking is only used to resolve simultaneous requests of the same priority level.

As below Table summarizes the interrupt sources, flag bits, vector addresses, enable bits, priority bits, arbitration ranking, and whether each interrupt may wake up the CPU from Power Down mode.

| SOURCE | FLAG | VECTOR ADDRESS | INTERRUPT ENABLE BITS | INTERRUPT PRIORITY CONTROL BITS | ARBITRATION RANKING | POWER DOWN WAKEUP |
|---|---|---|---|---|---|---|
| External Interrupt 0 | IE0 | 0003H | EX0 (IE.0) | IPH.0,IP.0 | 1(highest) | Yes |
| Timer 0 Overflow | TF0 | 000BH | ET0 (IE.1) | IPH.1,IP.1 | 2 | No |
| External Interrupt 1 | IE1 | 0013H | EX1 (IE.2) | IPH.2,IP.2 | 3 | Yes |
| Timer 1 Overflow | TF1 | 001BH | ET1 (IE.3) | IPH.3,IP.3 | 4 | No |
| Serial Port | RI + TI | 0023H | ES (IE.4) | IPH.4,IP.4 | 5 | No |
| Timer 2 Overflow | TF2 + EXF2 | 002BH | ET2 (IE.5) | IPH.5,IP.5 | 6 | No |
| A/D Converter | ADCI | 0033H | EADC (IE.6) | IPH.6,IP.6 | 7 | No |
| I2C Channel | I2C SI | 003BH | EI2C (EIE.0) | EIPH.0, EIP.0 | 8 | No |
| Serial Port 1 | RI_1 + TI_1 | 007BH | ES1 (EIE.7) | EIPH.7, EIP.7 | 9 | No |
| SPI interrupt | SPIF/MODF/ SPIOVF | 0083H | ESPI (EIE1.0) | EIP1H.0, EIP1.0 | 10 | No |
| External Interrupt 2 | IE2 | 0043H | EX2 (EIE.2) | EIPH.2, EIP.2 | 11 | No |
| External Interrupt 3 | IE3 | 004BH | EX3 (EIE.3) | EIPH.3, EIP.3 | 12 | No |
| External Interrupt 4 | IE4 | 0053H | EX4 (EIE.5) | EIPH.5, EIP.5 | 13 | No |
| External Interrupt 5 | IE5 | 005BH | EX5 (EIE.6) | EIPH.6, EIP.6 | 14 | No |
| PWM Period | PWMF | 0073H | EPWM (EIE1.1) | EIP1H.1, EIP1.1 | 15 | No |
| PWM Brake | BKF | 006BH | EBK (EIE1.2) | EIP1H.2, EIP1.2 | 16 | No |
| Timer 3 Overflow | TF3 | 008BH | ET3 (EIE1.3) | EIP1H.3, EIP1.3 | 17 | No |
| Capture Input/Direction Interrupt/QEI | CPTF0/QEIF + CPTF1/DIRF + CPTF2 | 0093H | ECPTF (EIE1.4) | EIP1H.4, EIP1.4 | 18 | No |
| NVM Interrupt | NVMF | 009BH | ENVMI (EIE1.5) | EIP1H.5, EIP1.5 | 19 | No |
| Watchdog Timer | WDIF | 0063H | EWDI (EIE.4) | EIPH.4, EIP.4 | 20 | No |

Table 11- 3: Vector location for Interrupt sources and power down wakeup

### 11.2.1 Response Time

The response time for each interrupt source depends on several factors like nature of the interrupt and the instruction under progress. In the case of external interrupt INT0 to INT5, they are sampled at C3 of every machine cycle and then their corresponding interrupt flags IE0 and IE1 will be set or reset. Similarly, the Serial port flags RI/RI_1 and TI/TI_1 are set in C4 of last machine cycle. The Timer 0 and 1 overflow flags are set at C3 of the machine cycle in which overflow has occurred. These flag values are polled only in the next machine cycle. If a request is active and all three conditions are met, then the hardware generated LCALL is executed. This call itself takes four machine cycles to be completed. Thus there is a minimum time of five machine cycles between the interrupt flag being set and the interrupt service routine being executed.

A longer response time should be anticipated if any of the three conditions are not met. If a higher or equal priority is being serviced, then the interrupt latency time obviously depends on the nature of the service routine currently being executed. If the polling cycle is not the last machine cycle of the instruction being executed, then an additional delay is introduced. The maximum response time (if no other interrupt is in service) occurs if the device is performing a write to IE, IP, IPH, EIE, EIP, EIPH, EIE1, EIP1 or EIP1H and then executes a MUL or DIV instruction. From the time an interrupt source is activated, the longest reaction time is 12 machine cycles. This includes 1 machine cycle to detect the interrupt, 2 machine cycles to complete the IE, IP, IPH, EIE, EIP, EIPH, EIE1, EIP1 or EIP1H access, 5 machine cycles to complete the MUL or DIV instruction and 4 machine cycles to complete the hardware LCALL to the interrupt vector location.

Thus in a single-interrupt system the interrupt response time will always be more than 5 machine cycles and not more than 12 machine cycles. The maximum latency of 12 machine cycle is 48 clock cycles. Note that in the standard 8051 the maximum latency is 8 machine cycles which equals 96 machine cycles. This is a 50% reduction in terms of clock periods.

# 12. PROGRAMMABLE TIMERS/COUNTERS

The W79E22X SERIES has three 16-bit programmable timer/counters.

## 12.1 Timer/Counters 0 & 1

TM0 and TM1 are 16-bit Timer/Counters, and there are nearly identical. Each of these Timer/Counters has two 8 bit registers which form the 16 bits counting register. For Timer/Counter 0, it consists of TH0, the upper 8 bits register, and TL0, the lower 8 bit register. Similarly Timer/Counter 1 has two 8 bits registers; TH1 and TL1. The two timers can be configured to operate either as timers, counting machine cycles or as counters counting external inputs.

When configured as a "Timer", the timer counts clock cycles. In "Counter" mode, the register is incremented on the falling edge of the corresponding external input pins, T0 for Timer 0 and T1 for Timer 1. The T0 and T1 inputs are sampled in every machine cycle at C4. If the sampled value is high in one machine cycle and low in the next, then a valid high to low transition on the pin is recognized and the count register is incremented. Since it takes two machine cycles to recognize a negative transition on the pin, therefore the maximum counting rate is 1/8 of the master clock frequency. In both "Timer" and "Counter" mode, the count register is updated at C3. Therefore, in the "Timer" mode, the recognized negative transition on pin T0 and T1 can cause the count register value to be updated only in the machine cycle following the one in which the negative edge was detected.

The "Timer" or "Counter" function is selected by the "$C/\overline{T}$" bit in the TMOD Special Function Register. Each Timer/Counter has one selection bit for its own; bit 2 of TMOD selects the function for Timer/Counter 0 and bit 6 of TMOD selects the function for Timer/Counter 1. In addition each Timer/Counter can be set to operate in any one of four possible modes. The mode selection is done by bits M0 and M1 in the TMOD SFR.

### 12.1.1 Time-Base Selection

The W79E22X SERIES can operate like the standard 8051/52 family, counting at the rate of 1/12 of the clock speed, or in turbo mode, counting at the rate of 1/4 clock speed. The speed is controlled by the T0M and T1M bits in CKCON, and the default value is zero, which uses the standard 8051/52 speed.

### 12.1.2 Mode 0

In Mode 0, the timer/counter is a 13-bit counter. The 13-bit counter consists of THx (8 MSB) and the five lower bits of TLx (5 LSB). The upper three bits of TLx are ignored. The timer/counter is enabled when TRx is set and either GATE is 0 or $\overline{INTx}$ is 1. When $C/\overline{T}$ is 0, the timer/counter counts clock cycles; when $C/\overline{T}$ is 1, it counts falling edges on T0 (P3.4 for Timer 0) or T1 (P3.5 for Timer 1). For clock cycles, the time base may be 1/12 or 1/4 clock speed, and the falling edge of the clock increments the counter. When the 13-bit value moves from 1FFFh to 0000h, the timer overflow flag TFx is set, and an interrupt occurs if enabled. This is illustrated in next figure below.

### 12.1.3 Mode 1

Mode 1 is the same as Mode 0, except that the timer/counter is 16 bits, instead of 13 bits.

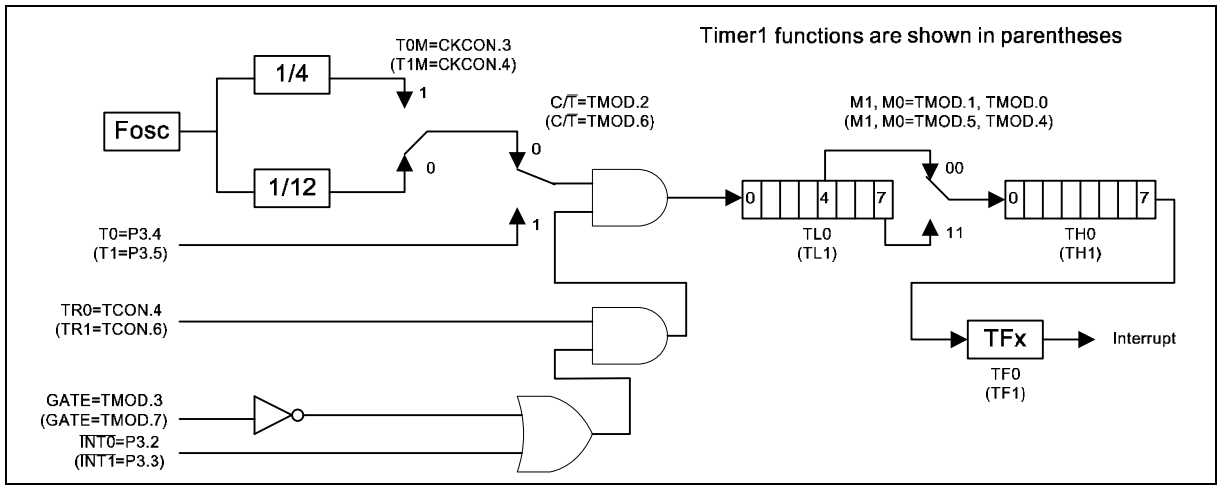



Figure 12-1: Timer/Counters 0 & 1 in Mode 0 and Mode 1

### 12.1.4 Mode 2

In Mode 2, the timer/counter is in the Auto Reload Mode. In this mode, TLx acts as an 8 bits count register, while THx holds the reload value. When the TLx register overflows from FFh to 00h, the TFx bit in TCON is set and TLx is reloaded with the contents of THx, and the counting process continues. The reload operation leaves the contents of the THx register unchanged. Counting is enabled by the TRx bit and proper setting of GATE and $\overline{INTx}$ pins. As in the other two modes 0 and 1, mode 2 allows counting of either clock cycles (clock/12 or clock/4) or pulses on pin Tn.

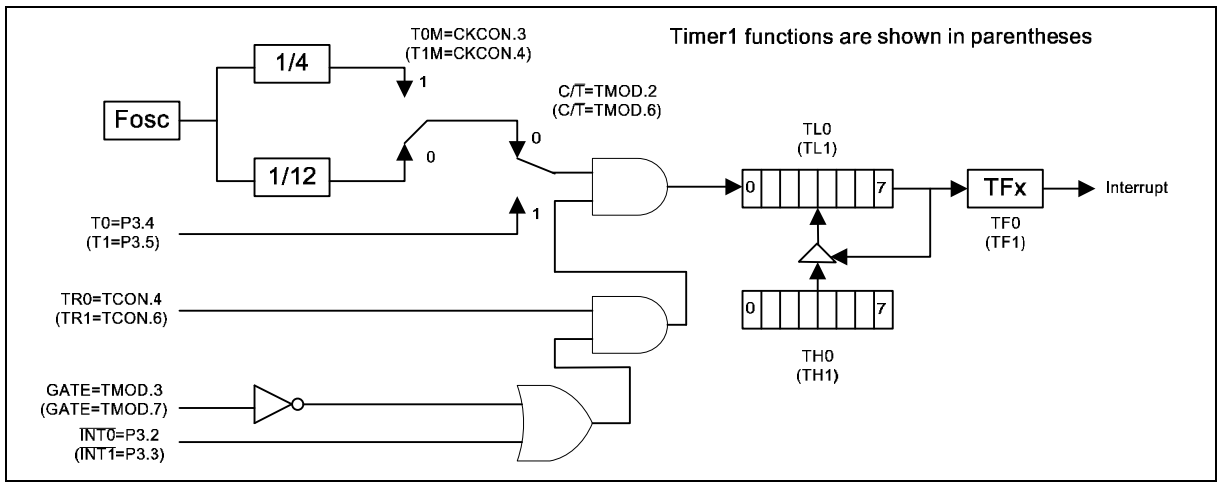



Figure 12-2: Timer/Counter 0 & 1 in Mode 2

### 12.1.5 Mode 3

Mode 3 is used when an extra 8-bit timer is needed. It has different effect on Timer 0 and Timer 1. TL0 and TH0 become two separate 8 bits counters. TL0 uses the Timer 0 control bits C/$\overline{T}$, GATE, TR0, $\overline{INT0}$ and TF0, and it can be used to count clock cycles (clock/12 or clock/4) or falling edges on pin T0, as determined by C/$\overline{T}$ (TMOD.2). TH0 becomes a clock-cycle counter (clock/12 or clock/4) and takes over the Timer 1 enable bit TR1 and overflow flag TF1. In contrast, mode 3 simply freezes Timer 1. If Timer 0 is in mode 3, Timer 1 can still be used in modes 0, 1 and 2, but it no longer has control over TR1 and TF1. Therefore when Timer 0 is in Mode 3, Timer 1 can only count oscillator cycles, and it does not have an interrupt or flag. With these limitations, baud rate generation is its most practical application, but other time-base functions may be achieved by reading the registers.

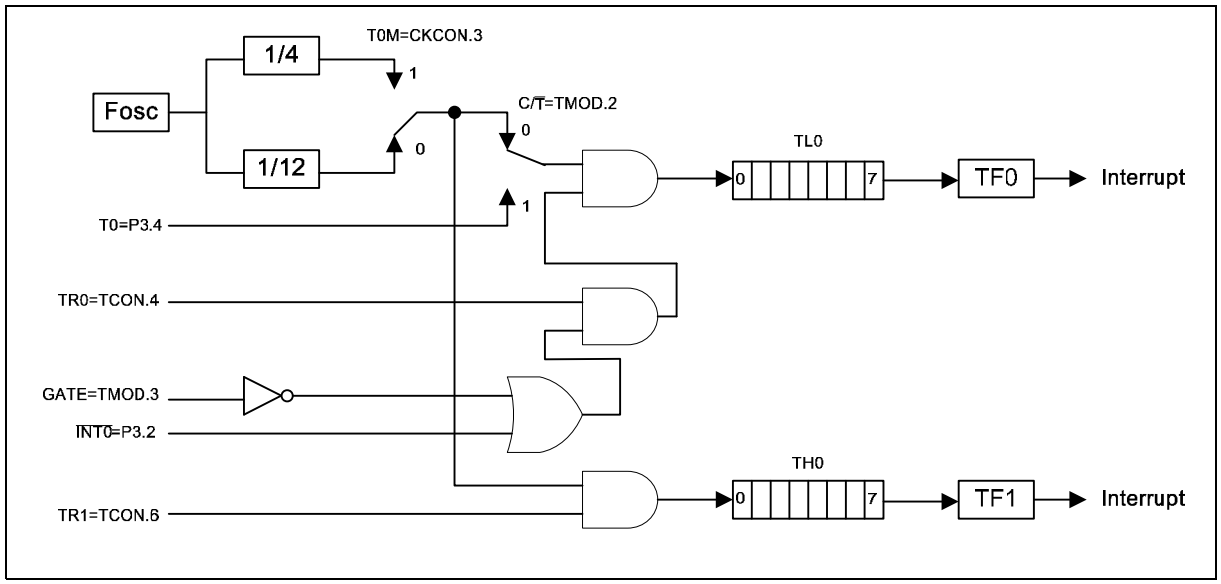



Figure 12-3: Timer/Counter Mode 3

## 12.2 Timer/Counter 2

Timer/Counter 2 is a 16-bit up/down-counter equipped with a capture/reload capability. The clock source for Timer/Counter 2 may be the external T2 pin (C/$\overline{T2}$ = 1) or the crystal oscillator (C/$\overline{T2}$ = 0), divided by 12 or 4. The clock is enabled and disabled by TR2. Timer/Counter 2 runs in one of four operating modes, each of which is discussed below.

### 12.2.1 Capture Mode

Capture mode is enabled by setting CP/$\overline{RL2}$ in T2CON to 1. In capture mode, Timer/Counter 2 is a 16-bit up-counter. When the counter rolls over from FFFFh to 0000h, the timer overflow flag TF2 is set, and an interrupt is generated, if enabled.

If the EXEN2 bit is set, a negative transition on the T2EX pin captures the current value of TL2 and TH2 in the RCAP2L and RCAP2H registers. It also sets the EXF2 bit in T2CON, which generates an interrupt if enabled. In addition, if the T2CR bit in T2MOD is set, the W79E22X SERIES resets Timer 2 automatically after each capture. This is illustrated below.

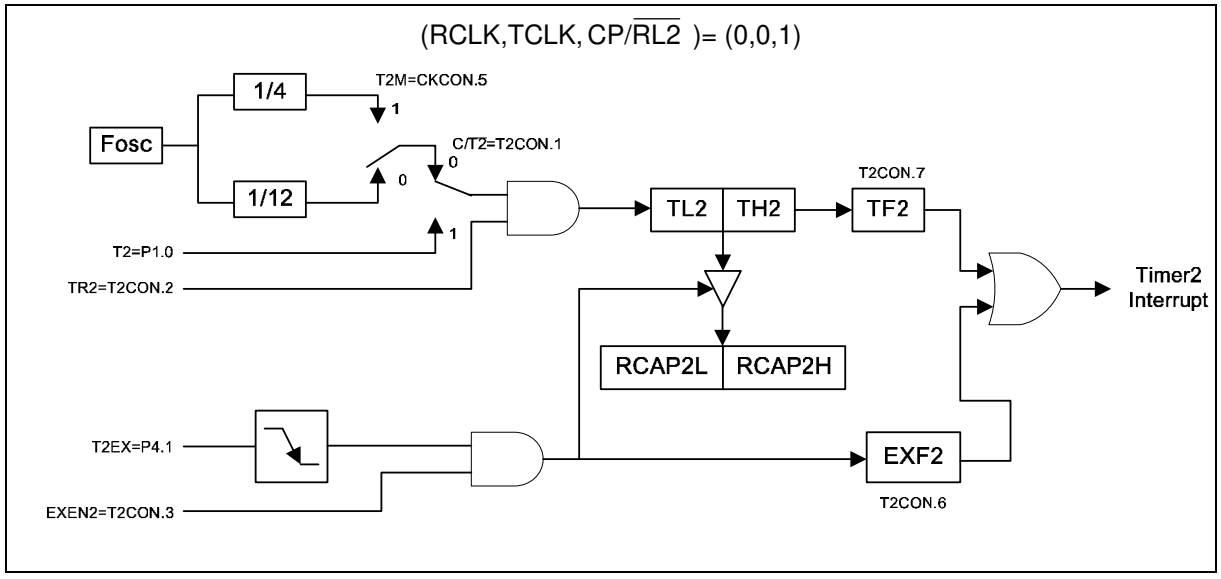



Figure 12-4: Timer2 16-Bit Capture Mode

### 12.2.2 Auto-reload Mode, Counting up

This mode is enabled by clearing CP/$\overline{RL2}$ in T2CON register and DCEN in T2MOD. In this mode, Timer/Counter 2 is a 16-bit up-counter. When the counter rolls over from FFFFh to 0000h, the timer overflow flag TF2 is set, and TL2 and TH2 capture the contents of RCAP2L and RCAP2H, respectively. Alternatively, if EXEN2 is set, a negative transition on the T2EX pin causes a reload, which also sets the EXF2 bit in T2CON.

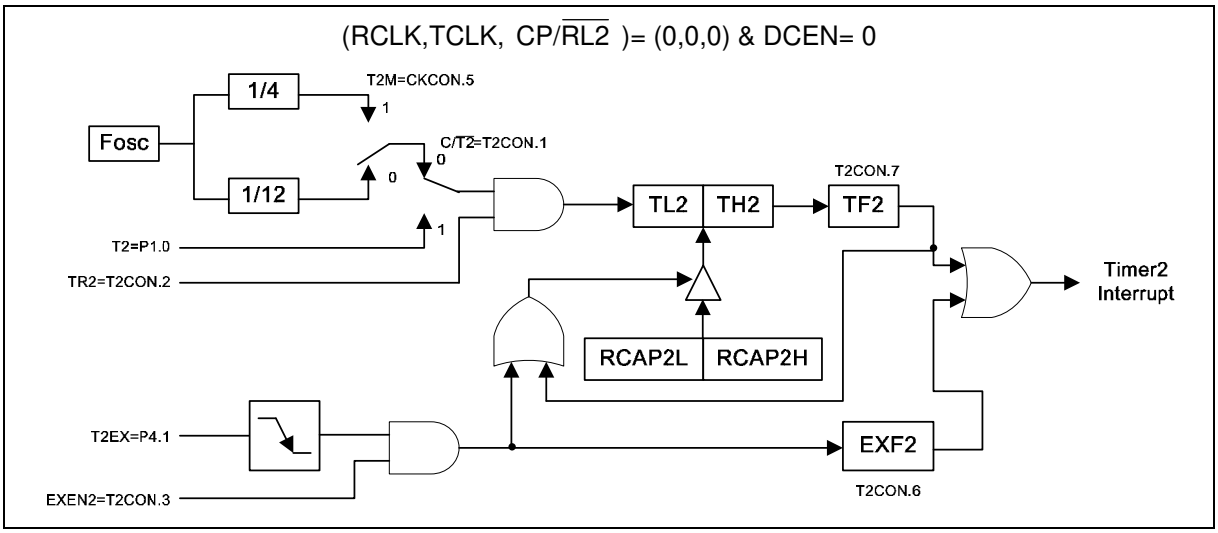



Figure 12-5: 16-Bit Auto-reload Mode, Counting Up

### 12.2.3 Auto-reload Mode, Counting Up/Down

This mode is enabled by clearing CP / $\overline{RL2}$ in T2CON and setting DCEN in T2MOD. In this mode, Timer/Counter 2 is a 16-bit up/down-counter, whose direction is controlled by the T2EX pin (1 = up, 0 = down). If Timer/Counter 2 is counting up, an overflow reloads TL2 and TH2 with the contents of the capture registers RCAP2L and RCAP2H. If Timer/Counter 2 is counting down, TL2 and TH2 are loaded with FFFFh when the contents of Timer/Counter 2 equal the capture registers RCAP2L and RCAP2H. Regardless of direction, reloading sets the TF2 bit. It also toggles the EXF2 bit, but the EXF2 bit can not generate an interrupt in this mode. This is illustrated below.

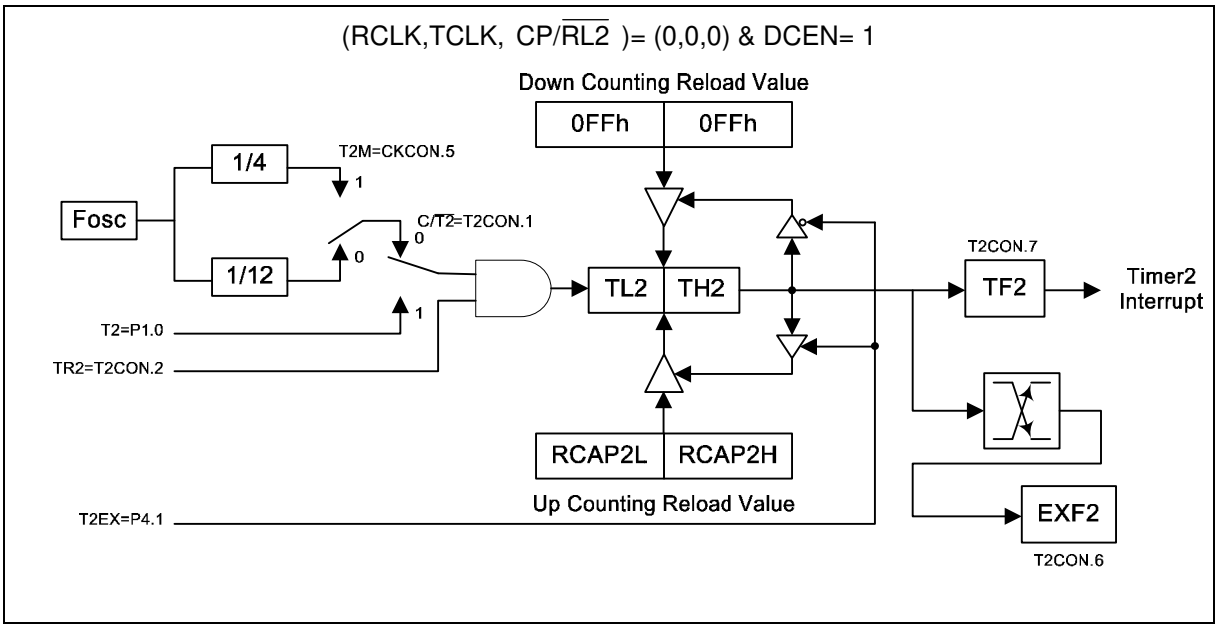



Figure 12-6: 16-Bit Auto-reload Up/Down Counter

### 12.2.4 Baud Rate Generator Mode

Baud rate generator mode is enabled by setting either RCLK or TCLK in T2CON. In baud rate generator mode, Timer/Counter 2 is a 16-bit counter with auto-reload when the count rolls over from FFFFh. However, rolling-over does not set TF2. If EXEN2 is set, then a negative transition on the T2EX pin sets EXF2 bit in the T2CON register and causes an interrupt request.

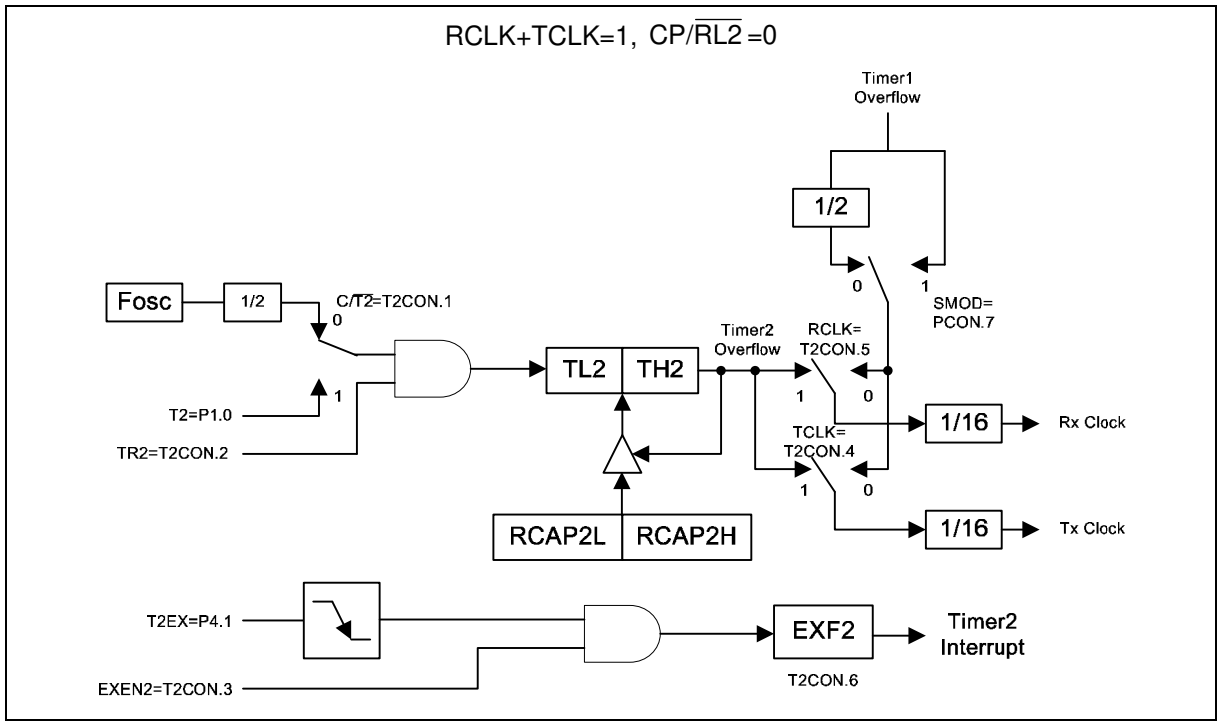



Figure 12-7: Baud Rate Generator Mode

## 13. WATCHDOG TIMER

The Watchdog Timer is a free-running timer that can be programmed to serve as a system monitor, a time-base generator or an event timer. It is basically a set of dividers that divide the system clock to determine the time-out interval. When the time-out occurs, a flag is set, which can generate an interrupt or a system reset, if enabled. The interrupt will occur if its interrupt and global interrupt enables are set. The interrupt and reset functions are independent of each other and may be used separately or together.

The main use of the Watchdog Timer is as a system monitor. In case of power glitches or electromagnetic interference, the processor may begin to execute errant code. The Watchdog Timer helps W79E22X SERIES recovers from these states. During development, the code is first written without the watchdog interrupt or reset. Then, the watchdog interrupt is enabled to identify code locations where the interrupt occurs, and instructions are inserted to reset the Watchdog Timer in these locations. In the final version, the watchdog interrupt is disabled, and the watchdog reset is enabled. If errant code is executed, the Watchdog Timer is not reset at the required times, so a Watchdog Timer reset occurs.

When used as a simple timer, the reset and interrupt functions are disabled. The Watchdog Timer can be started by RWT and sets the WDIF flag after the selected time interval. Meanwhile, the program polls the WDIF flag to find out when the selected time interval has passed. Alternatively, the Watchdog Timer can also be used as a very long timer. In this case, the interrupt feature is enabled.

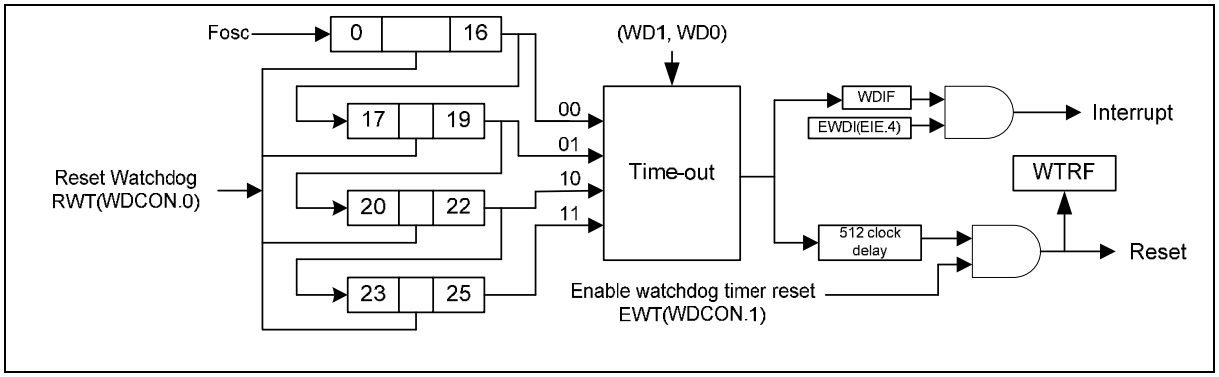



Figure 13-1: Watchdog Timer

The Watchdog Timer should be started by RWT because this ensures that the timer starts from a known state. The RWT bit is self-clearing; i.e., after writing a 1 to this bit, the bit is automatically cleared. After setting RWT, the Watchdog Timer begins counting clock cycles. The time-out interval is selected by WD1 and WD0 (CKCON.7 and CKCON.6).

| WD1 | WD0 | INTERRUPT TIME-OUT | RESET TIME-OUT | NUMBER OF CLOCKS | TIME @ 10 MHZ | TIME @ 11.0592 MHZ | TIME @ 25 MHZ |
|---|---|---|---|---|---|---|---|
| 0 | 0 | $2^{17}$ | $2^{17}$ + 512 | 131072 | 13.11 mS | 11.85 mS | 5.24 mS |
| 0 | 1 | $2^{20}$ | $2^{20}$ + 512 | 1048576 | 104.86 mS | 94.81 mS | 41.94 mS |
| 1 | 0 | $2^{23}$ | $2^{23}$ + 512 | 8388608 | 838.86 mS | 758.52 mS | 335.54 mS |
| 1 | 1 | $2^{26}$ | $2^{26}$ + 512 | 67108864 | 6710.89 mS | 6068.15 mS | 2684.35 mS |

Table 13-1: Time-out values for the Watchdog Timer

When the selected time-out occurs, the watchdog interrupt flag WDIF (WDCON.3) is set. After watchdog time-out, and if Watchdog Timer Reset EWT (WDCON.1) is enabled, the Watchdog Timer will cause a reset 512 clocks later. This reset lasts two machine cycles, and the Watchdog Timer reset flag WTRF (WDCON.2) is set, which indicates that the Watchdog Timer caused the reset. RWT can be used to clear Watchdog timer before a time-out occurs.

The Watchdog Timer is disabled by a power-on/fail reset. The external reset and Watchdog Timer reset can not disable Watchdog Timer, instead it only restart the Timer.

The control bits that support the Watchdog Timer are described as below:

**Watchdog Timer Control (WDCON)**

| BIT | NAME | FUNCTION |
|---|---|---|
| 7 | - | Reserved. |
| 6 | POR | Power-on reset flag. The hardware sets this flag during power–up, and it can only be cleared by software. This flag can also be written by software. |
| 5-4 | - | Reserved. |
| 3 | WDIF | Watchdog Timer Interrupt Flag. If the watchdog interrupt is enabled, the hardware sets this bit to indicate that the watchdog interrupt has occurred. If the interrupt is not enabled, this bit indicates that the time-out period has elapsed. This bit must be cleared by software. |
| 2 | WTRF | Watchdog Timer Reset Flag. If EWT is 0, the Watchdog Timer has no affect on this bit. Otherwise, the hardware sets this bit when the Watchdog Timer causes a reset. It can be cleared by software or a power-fail reset. It can be also read by software, which helps determine the cause of a reset. |
| 1 | EWT | Enable Watchdog-Timer Reset. Set this bit to enable the Watchdog Timer Reset function. |
| 0 | RWT | Reset Watchdog Timer. Set this bit to reset the Watchdog Timer before a time-out occurs. This bit is automatically cleared by the hardware. |

The POR, EWT, WDIF and RWT bits are protected by the Timed Access procedure. This procedure prevents software, especially errant code, from accidentally enabling or disabling the Watchdog Timer. An example is provided below.

```
        org     63h
        mov     TA,#AAH
        mov     TA,#55H
        clr     WDIF
        jnb     execute_reset_flag,bypass_reset     ; Test if CPU need to reset.
        jmp     $                                   ; Wait to reset
bypass_reset:
        mov     TA,#AAH
        mov     TA,#55H
        setb    RWT
        reti

        org     300h
start:
        mov     ckcon,#01h                          ; Select 2 ^ 17 timer
;       mov     ckcon,#61h                          ; Select 2 ^ 20 timer
;       mov     ckcon,#81h                          ; Select 2 ^ 23 timer
;       mov     ckcon,#c1h                          ; Select 2 ^ 26 timer
        mov     TA,#aah
        mov     TA,#55h
        mov     WDCON,#00000011B
        setb    EWDI
        setb    ea
        jmp     $                                   ; Wait time out
```

**Clock Control**

WD1, WD0: CKCON.7, CKCON.6 - Watchdog Timer Mode select bits. These two bits select the time-out interval for the Watchdog Timer. The reset interval is 512 clocks longer than the selected interval. The default time-out is $2^{17}$ clocks, the shortest time-out period.

# 14. PULSE-WIDTH-MODULATED (PWM) OUTPUTS

## 14.1 PWM Features

The PWM block supports the following features;

- Four 12-bit PWM channels or complementary pairs:
  - 4 independent PWM outputs: PWM0, PWM2, PWM4 & PWM6.
  - 4 complementary PWM pairs with insertion of programmable dead-time: (PWM0,PWM1), (PWM2,PWM3), (PWM4,PWM5), (PWM6,PWM7)
- Three operation mode:
  Edge aligned mode, Center aligned mode and Single shot mode.
- Programmable dead-time insertion between paired PWMs.
- Output override control for Electrically Commutated Motor operation.
- Hardware/software brake protection.
- Support 2 independent interrupts:
  - Interrupt request when up/down counter comparison matched or underflow.
  - Interrupt request when external brake asserted.
- Flexible operation in debug mode.
- High Source/Sink current.

The outputs for PWM0 to PWM7 are on P2[5:0] (PWM[5:0]) and P5[1:0] (PWM [7:6]) respectively. After CPU reset, the internal output of each PWM channel depends on the output controls and polarity settings. The interval between successive outputs is controlled by a 12–bit up/down counter which uses the oscillator frequency with configurable internal clock prescaler as its input. The PWM counter clock, has the frequency as the clock source $F_{PWM} = F_{OSC}$/Prescaler. The following Figure 14-1: PWM Block Diagram below is the block diagram for PWM.

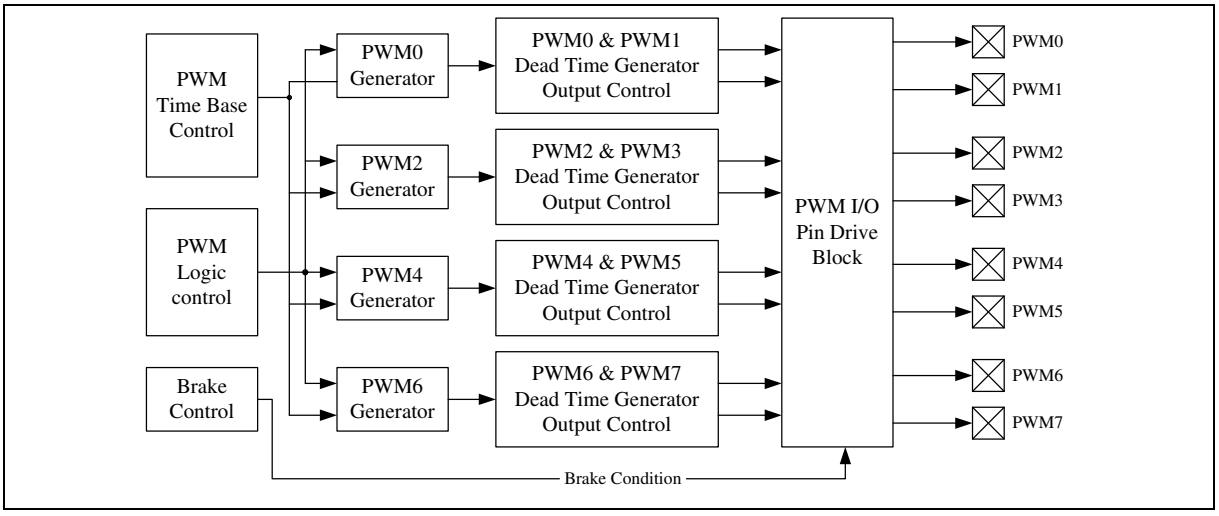



Figure 14-1: PWM Block Diagram

## 14.2 PWM Control Registers

The overall functioning of the PWM module is controlled by the contents of the PWMCON1 register. The operation of most of the control bits is straightforward. For example PWM0I is an invert bit for each output which causes results in the output to have the opposite value compared to its non-inverted output. The transfer of the data from the Counter and Compare registers to the control registers is controlled by the PWMCON1.6 (load) while PWMCON1.7 (PWMRUN) allows the PWM to be either in the run or idle state.

If the Brake pin is not used to control the brake function, the “Brake when PWM is not running” function can be used to cause the outputs to have a given state when the PWM is halted. This approach should be used only in time critical situations when there is not sufficient time to use the approach outlined above, since going from the Brake state to run without causing an undefined state on the outputs is not straightforward. A discussion on this topic is included in the section on PWMCON2.

The Brake function, which is controlled by the contents of the PWMCON2 register, is somewhat unique. In general, when Brake is asserted, the eight PWM outputs are forced to a user selected state, namely the state selected by PWMCON3. As shown in the description of the operation of the PWMCON2 register, if PWMCON2.4, BKEN, is a “1” brake is asserted under the control PWMCON2.7, BKCH, and PWMCON2.5, BPEN. As shown, if both are a “0”, brake is asserted. If PWMCON2.7 is a “1”, brake is asserted when the PWMRUN bit, PWMCON1.7, is a “0”. If PWMCON2.6, BKPS, is a “1”, brake is asserted when the Brake Pin, P1.1, has the same polarity as PWMCON2.6. When brake is asserted in response to this pin, the PWMRUN bit in PWMCON1.7 is automatically cleared, and BKF (PWMCON4.0) flag will be set. When both BKCH and BPEN are “1”, BKF will be set when Brake pin is asserted, but PWM generator continues to run. With this special condition, the PWM output does not follow PWMnB, instead it output continuously as per normal without affected by the brake.

Since the Brake Pin being asserted will automatically clear the PWMRUN (PWMCON1.7) and BKF (PWMCON4.0) flag will be set, the user program can poll this bit or enable PWM’s brake interrupt to determine when the Brake Pin causes a brake to occur. The other method for detecting a brake caused by the Brake Pin would be to tie the Brake Pin to one of the external interrupt pins. This latter approach is needed if the Brake signal is of insufficient length to ensure that it can be captured by a polling routine. When, after being asserted, the condition causing the brake is removed, the PWM outputs go to whatever state that had immediately prior to the brake. This means that in order to go from brake being asserted to having the PWM run without going through an indeterminate state, care must be taken. If the Brake Pin causes brake to be asserted, the following prototype code will allow the PWM to go from brake to run smoothly by software polling BKF flag or enable PWM’s interrupt.

• Rewrite PWMCON2 to change from Brake Pin enabled to S/W Brake.

• Write PWM (0, 2, 4, 6) Compare register to always “1”, FFFh, or always “0”, 000h, to initialize PWM output to a High or Low, respectively.

• Clear BKF flag.

• Set PWMCON1 to enable PWMRUN and Load.

• Poll Brake Pin until it is no longer active.

• Poll PWMCON1 to find that Load Bit PWMCON1.6 is “0”. When “0”:

• Write PWMP (0, 2, 4, and 6) Counter register for desired pulse widths and counter reload values.

• Set PWMCON1 to Run and Transfer.

Note that if a narrow pulse on the Brake Pin causes brake to be asserted, it may not be possible to go through the above code before the end of the pulse. In this case, in addition to the code shown, an external latch on the Brake Pin may be required to ensure that there is a smooth transition in going from brake to run. A compare value greater than the counter reloaded value resulted in the PWM output being high. In addition there are two special cases. A compare value of all zeroes, 000H, causes the output to remain permanently Low. A compare value of all ones, FFFH, results in the PWM output remaining permanently High.

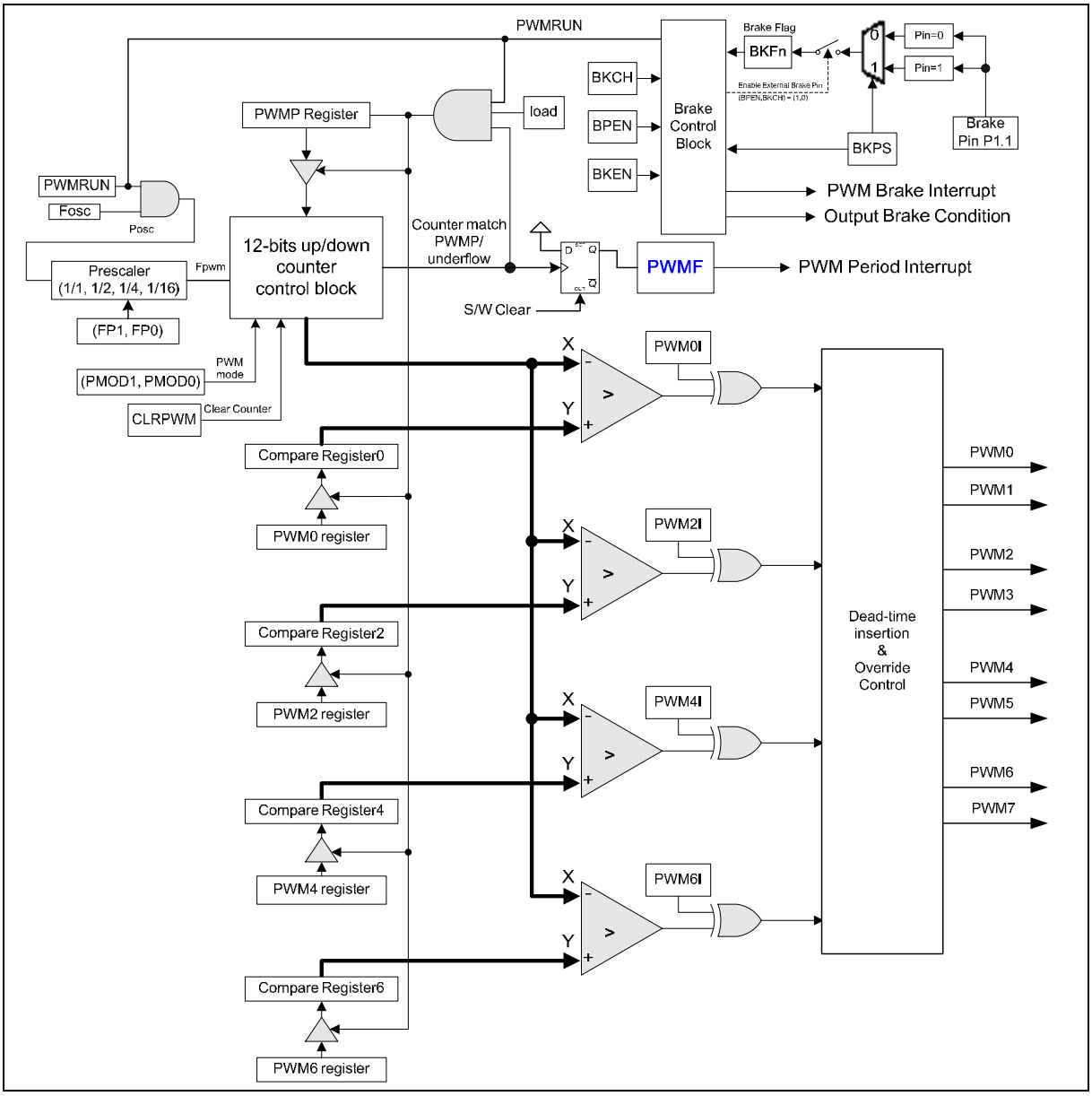



Figure 14-2: PWM Time-base Generator and Brake Function

The PWMP register fact that writes are not into the Counter register that controls the counter; rather they are into a holding register. As described below the transfer of data from this holding register, into

the register which contains the actual reload value, is controlled by the user's program. The width of each PWM output pulse is determined by the value in the appropriate compare register. Each PWM register pair of (PWMPH,PWMPL), (PWM0H,PWM0L), (PWM2H,PWM2L), (PWM4H,PWM4L) and (PWM6H,PWM6L),in the format of 12-bit width by combining 4 LSB of high byte and 8 MSB bits of low byte, decides the PWM period and each channel's duty cycle. The following equations show the formula for period and duty for each pwm operation mode:

**Edge aligned**:

Period = (pwmp +1) * ioclock period * 1/prescaler

Duty = duty * ioclock period

**Single shot**:

Period = (pwmp) * ioclock period /prescaler (no meaning since it is not periodic)

Duty = (duty) * ioclock period/prescaler (for prescaler 1, 1/2/2, 1/4)

(duty-1) * ioclock period /prescaler < Duty < (duty) * ioclock period/prescaler (for prescaler 1/16)

**Centre aligned**:

Period = (pwmp* 2) * ioclock period /prescaler

Duty = (duty*2 - 1) * ioclock period /prescaler

**Note**: "duty" refers to PWM0~3 register value.

## 14.3 PWM Pin Structures

As show in the following diagrams, PWM pin structures are controllable through PWM options bits (PWMEE/PWMOE) and SFR PWMCON4 bits (PWMEOM/PWMOOM/PWM6OM/PWM7OM).

| PWMEE/PWMOE (OPTION BITS) | PWMEOM/PWMOOM /PWM6OM/PWM7OM | PIO.X (X = 0-7) | PIN STRUCTURES |
|---|---|---|---|
| X | 0 | X | Tri-state |
| 1 (Disable) | 1 | X | Quasi (I/O output) |
| 0 (Enable) | 1 | 0 | Push Pull (PWM output) |
| 0 (Enable) | 1 | 1 | Push Pull (I/O output) |

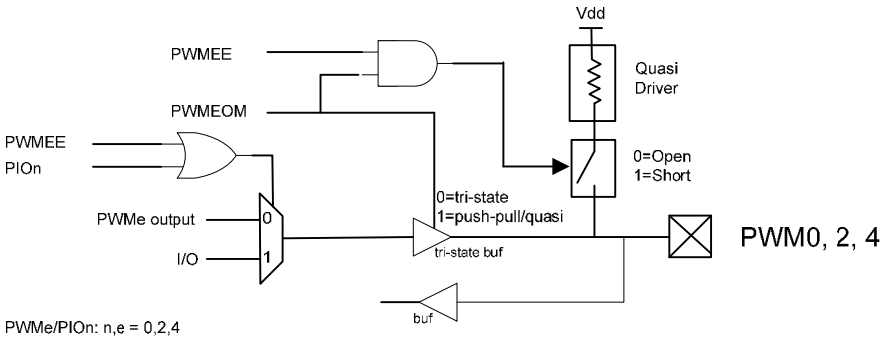
Table 14-1: PWM pin structures (during internal rom execution)

| PWMEE/PWMOE (OPTION BITS) | PWMEOM/PWMOOM /PWM6OM/PWM7OM | PIO.X (X = 0-7) | PIN OUTPUT | PIN STRUCTURES |
|---|---|---|---|---|
| 1 (Disable) | X | X | External access | Push Pull |
| 0 (Enable) | X | X | External access | Push Pull (strong) |

Table 14-2: PWM pin structures (during external rom execution)

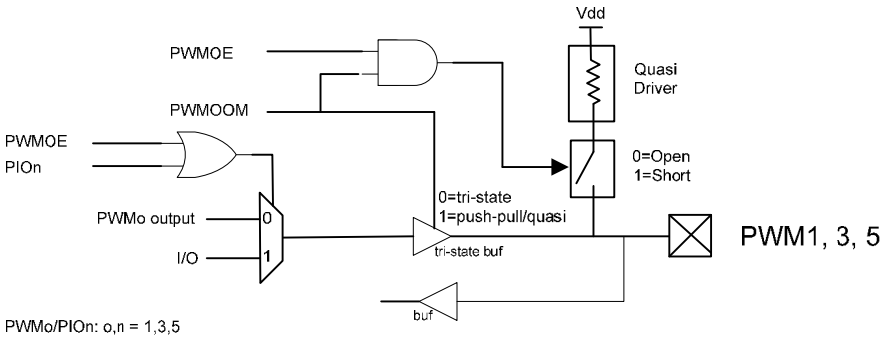
**Note: PWMEOM/PWMOOM/PWM6OM/PWM7OM are cleared to zero when CPU in reset state.**

**Thus, the port pins that multi-function with PWM will be tri-stated on default. User is required to set the bits to enable GPIO/PWM outputs.**



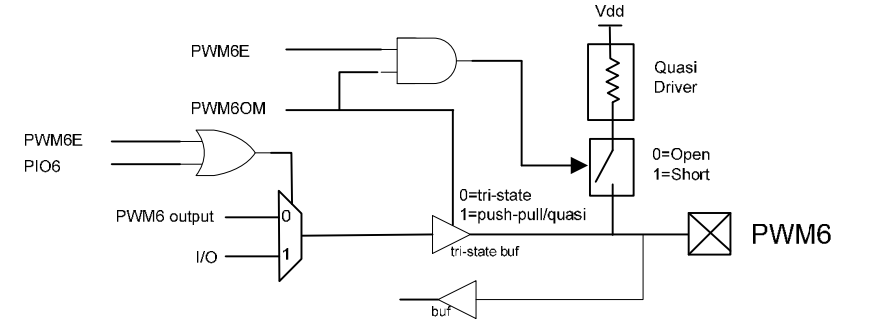

PWMe/PIOn: n,e = 0,2,4
PWMEE: PWM Channel 0, 2 & 4 Enable (Option Bit)
PWMEOM: PWM Channel 0, 2 & 4 Output Mode (SFR PWMCON4.7 Bit)

Figure 14-3: PWM0, 2 & 4 I/O pins




PWMo/PIOn: o,n = 1,3,5
PWMOE: PWM Channel 1, 3 & 5 Enable (Option Bit)
PWMOOM: PWM Channel 1, 3 & 5 Output Mode (SFR PWMCON4.6 Bit)

Figure 14-4: PWM1, 3 & 5 I/O pins




PWM6E: PWM Channel 6 Enable (Option Bit)
PWM6OM: PWM Channel 6 Output Mode (SFR PWMCON4.5 Bit)

Figure 14-5: PWM6 I/O pin

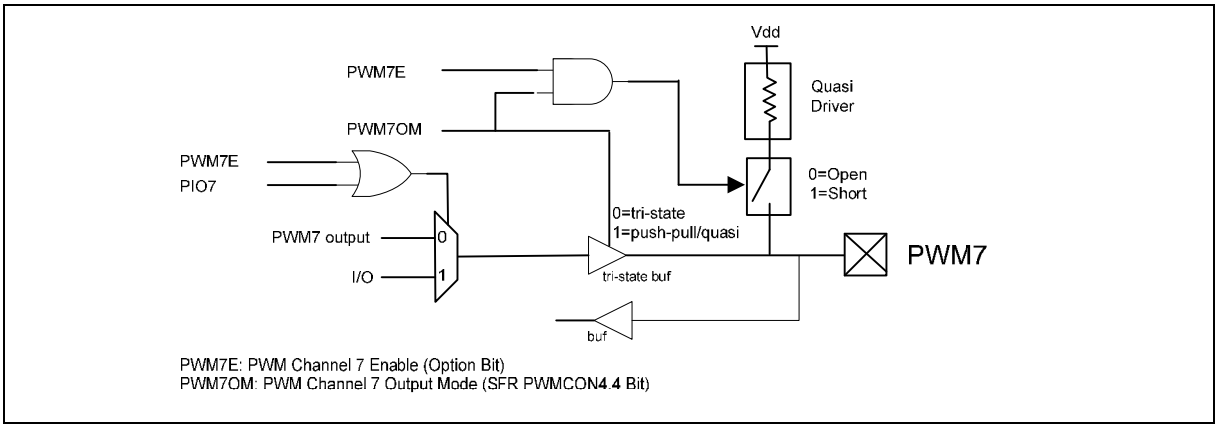



Figure 14-6: PWM7 I/O pin

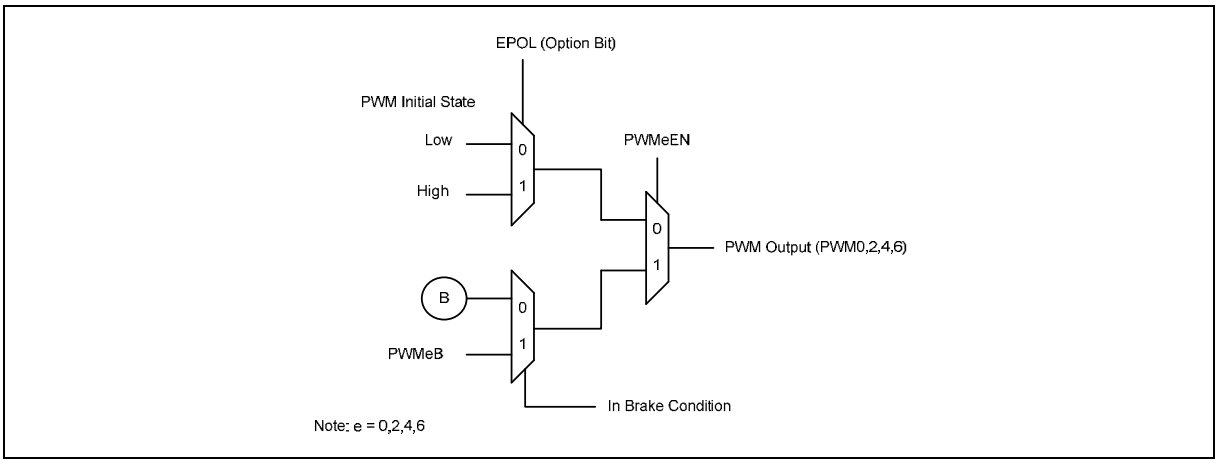



Figure 14-7: Even PWM Output

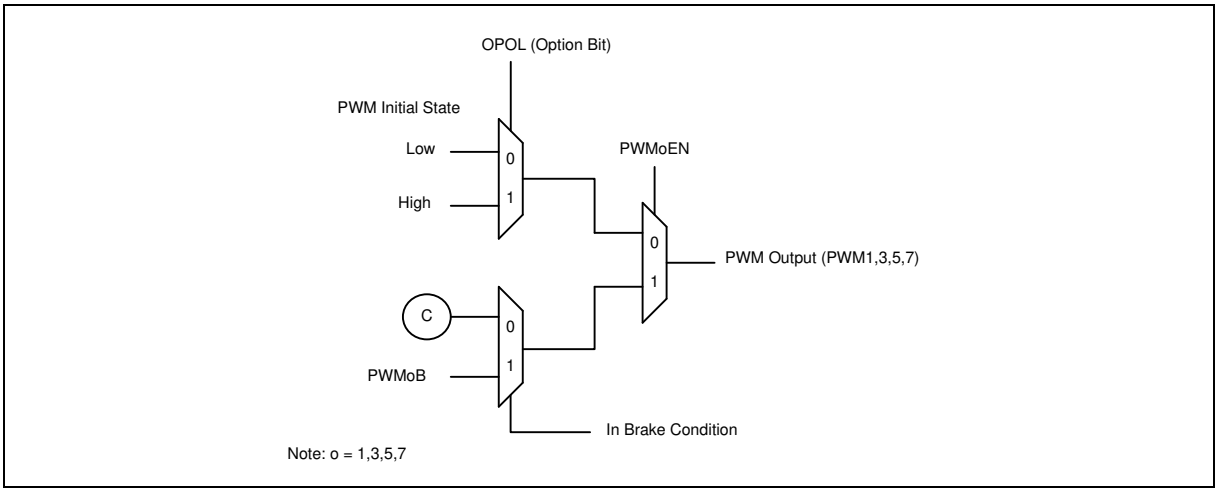



Figure 14-8: Odd PWM Output

## 14.4 Complementary PWM with Dead-time and Override functions

In this module there are four duty-cycle generators, from 0 through 3. The total of eight PWM output pins in this module, from 0 through 7. The eight PWM outputs are grouped into output pairs of even and odd numbered outputs. In complimentary modes, the odd PWM pins must always be the complement of the corresponding even PWM pin. For example, PWM1 will be the complement of PWM0. PWM3 will be the complement of PWM2, PWM5 will be the complement of PWM4 and PWM7 will be the complement of PWM6. Complementary mode is enabled only when both PWMeEN and the corresponding PWMoEN are set to high. The time base for the PWM module is provided by its own 12-bit timer, which also incorporates selectable prescaler options.

**Note:** PWM pairs of (PWM2, 3), (PWM4, 5) and (PWM6, 7) are in the same structure as pair of (PWM0, 1). (Refer to Figure 14-9).

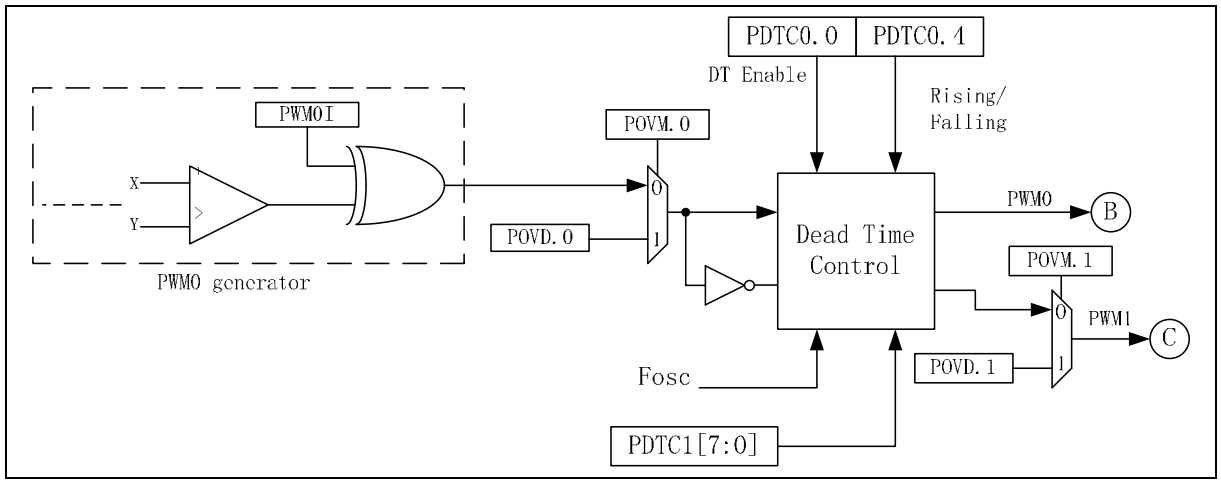



Figure 14-9: Complementary PWM with Dead-time and Override functions

## 14.5 Dead-Time Insertion

The dead time generator inserts an “off” period called “dead time” between the turnings off of one pin to the turning on of the complementary pin of the paired pins. This is to prevent damage to the power switching devices that will be connected to the PWM output pins. Each complementary output pair for the PWM module has 6-bits counter used to produce the dead time insertion. Each dead time unit has a rising and falling edge detector connected to the duty cycle comparison output. The dead time is loaded into the timer on the detected PWM edge event. Depending on whether the edge is rising or falling, one of the transitions on the complementary outputs is delayed until the timer counts down to zero. A timing diagram indicating the dead time insertion for one pair of PWM outputs is shown in Figure 14-10 and Figure 14-11.

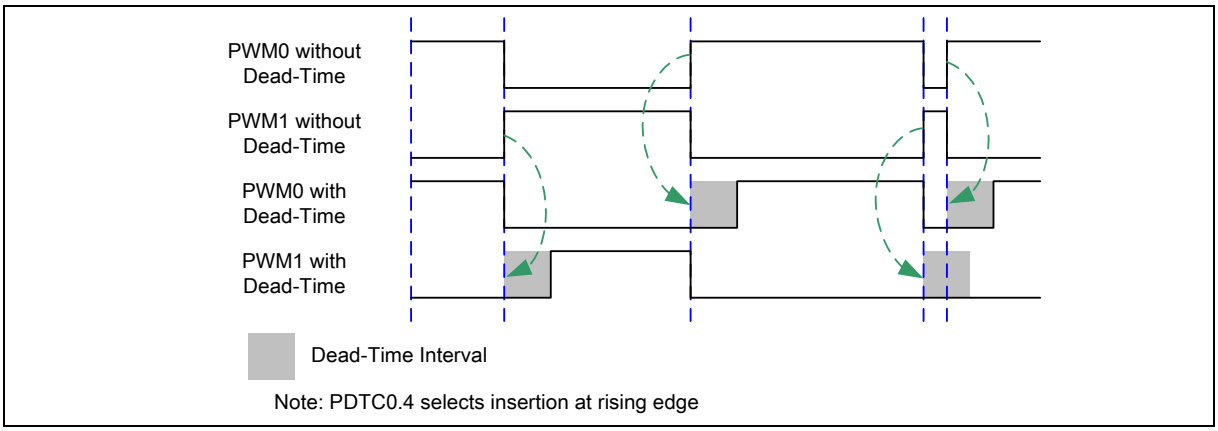



Figure 14-10: Effect of Dead-Time for complementary pairs (rising edge)

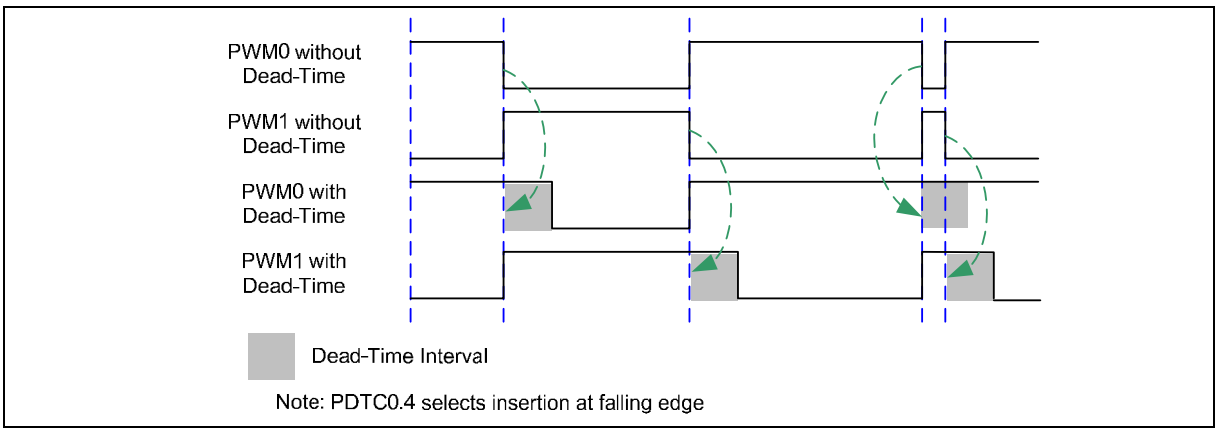



Figure 14-11: Effect of Dead-Time for complementary pairs (falling edge)

**Note: User need to take care that power switches should not be use when PWM pair is asserted (high) at the same time.**

**PDTCO and PDTC1 have time access protection in writing access.** In Power inverter application, a dead time insertion avoids the upper and lower switches of the half bridge from being active at the same time. Hence the dead time control is crucial to proper operation of a system. Some amount of time must be provided between turning off of one PWM output in a complementary pair and turning on the other transistor as the power output devices cannot switch instantaneously.

## 14.6 PWM Output Override

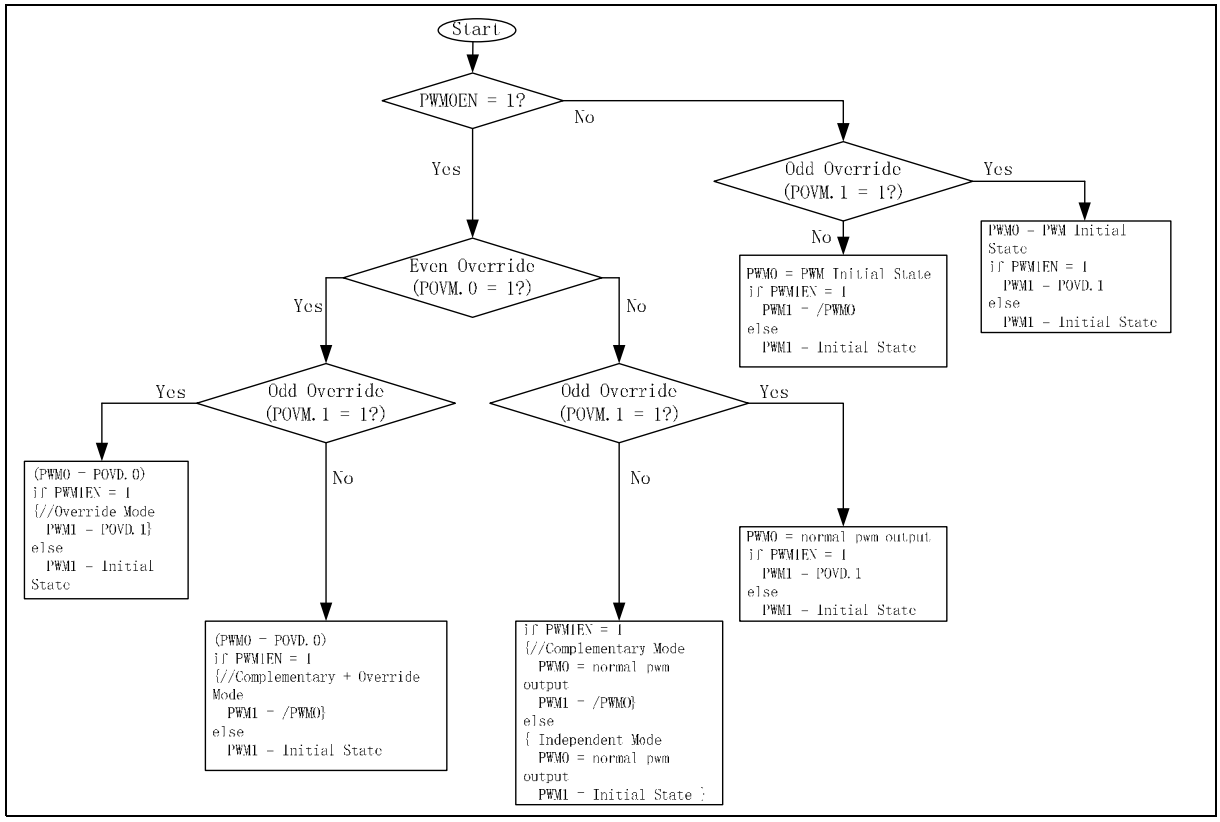



Figure 14-12: Override Flow Diagram

Each of the PWM output channels can be manually overridden by using the appropriate bits in the POVD and POVM registers. This function allow user to drive the PWM I/O pins to specified logic states independent of the duty cycle comparison units. The PWM override bits are useful when controlling various types of Electrically Commutated Motor (ECM) like a BLDC motor. The POVD register contains eight bits, POVD[7:0]. It determines which PWM I/O pins will be overridden. On reset, POVD is 00H.

The POVM register contains eight bits, POVM[7:0]. It determines the state of the PWM I/O pins when a particular output is overridden via the POVD bits. On reset, POVM is 00H. The POVM[7:0] bits are active-high. When the POVM[7:0] bits are set, the corresponding POVD[7:0] bit will have effect on the PWM output. When one of the POVM bits is set, the output on the corresponding PWM I/O pin will be determined by the state of corresponding POVD bit. When a POVM bit is clear, the PWM pin will be driven to its active state. The odd channel is always the complement of the even channel with dead time inserted.. Figure 14-13 demonstrates the override function in complementary mode.

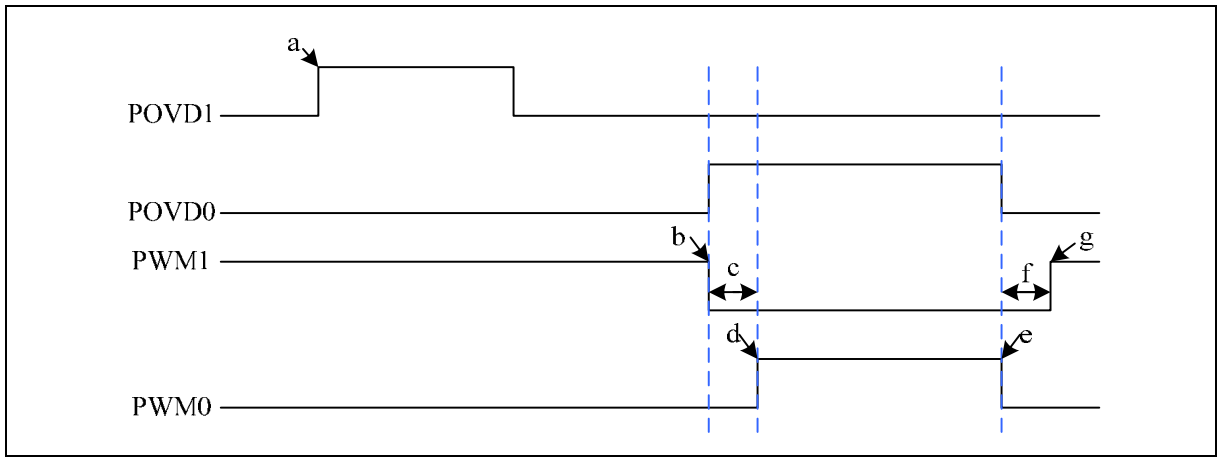



Figure 14-13: Override bit in complementary mode

Assume rising edge dead time insertion; refer to Figure 14-12: Override Flow Diagram.

Example: POVM0 = 1 and POVM1 = 0; PWM0EN and PWM1EN = 1;

a. Odd override bits have no effect in complementary mode.
b. Even override bit is activated, which causes the Odd PWM to deactivate.
c. Dead-Time insertion.
d. Even PWM activated after the dead-time.
e. Even override bit is deactivated, which causes the Even PWM to deactivate.
f. Dead-Time insertion.
g. Odd PWM is activated after the dead time.

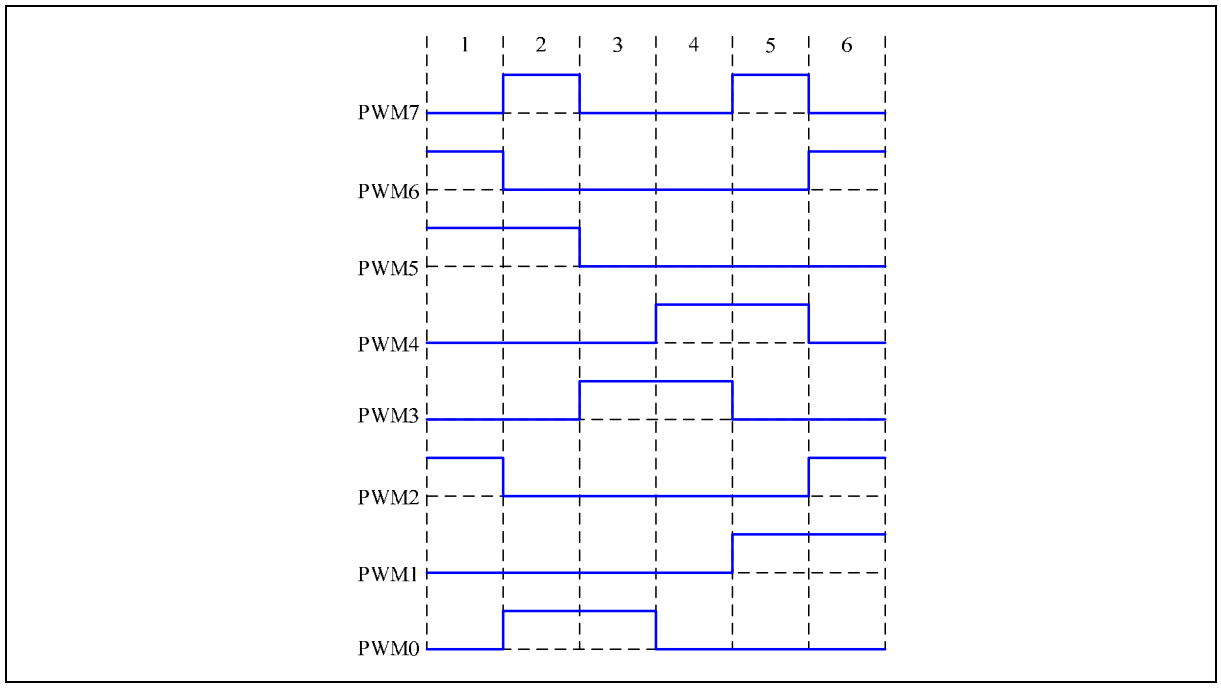



Figure 14-14: Example 1 of Output Even & Odd Override

| STATE | POVM<br>PWMEEN = 1<br>PWMOEN = 1 | POVD |
|---|---|---|
| 1 | 1111 1111b | 0110 0100b |
| 2 | 1111 1111b | 1010 0001b |
| 3 | 1111 1111b | 0000 1001b |
| 4 | 1111 1111b | 0001 1000b |
| 5 | 1111 1111b | 1001 0010b |
| 6 | 1111 1111b | 0100 0110b |

Table 14-3: Example 1 of Output Even & Odd Override

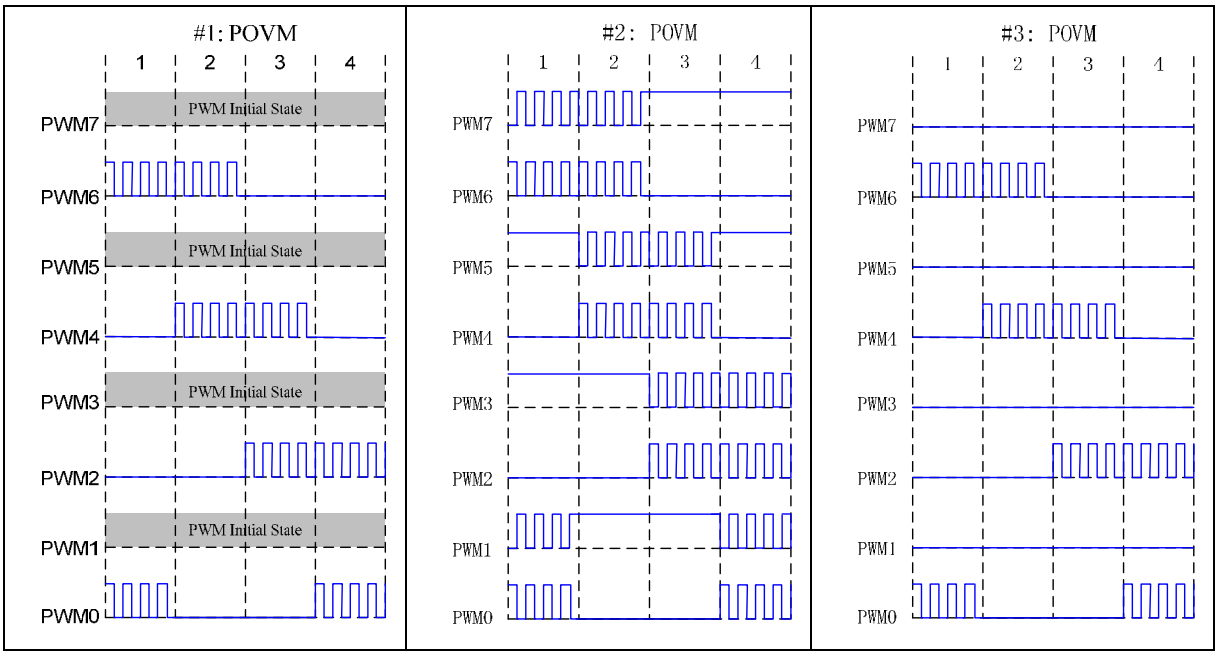



Figure 14-15: Example 2 of Output Override

| State | #1: POVM<br>(Odd not overridenot in complementary)<br>(PWMeEN = 1, PWMoEN = 0) | #2: POVM<br>(Odd not Override in complementary)<br>(PWMeEN = 1, PWMoEN = 1) | #3: POVM<br>(Odd with Override not in complementary)<br>(PWMeEN = 1, PWMoEN = 1) | POVD |
|---|---|---|---|---|
| 1 | 0001 0100b | 0001 0100b | 1011 1110 | 0000 0000b |
| 2 | 0000 0101b | 0000 0101b | 1010 1111 | 0000 0000b |
| 3 | 0100 0001b | 0100 0001b | 1110 1011 | 0000 0000b |
| 4 | 0101 0000b | 0101 0000b | 1111 1010 | 0000 0000b |

Table 14-4: Example 2 of Output Override

## 14.7 Edge Aligned PWM (up-counter)

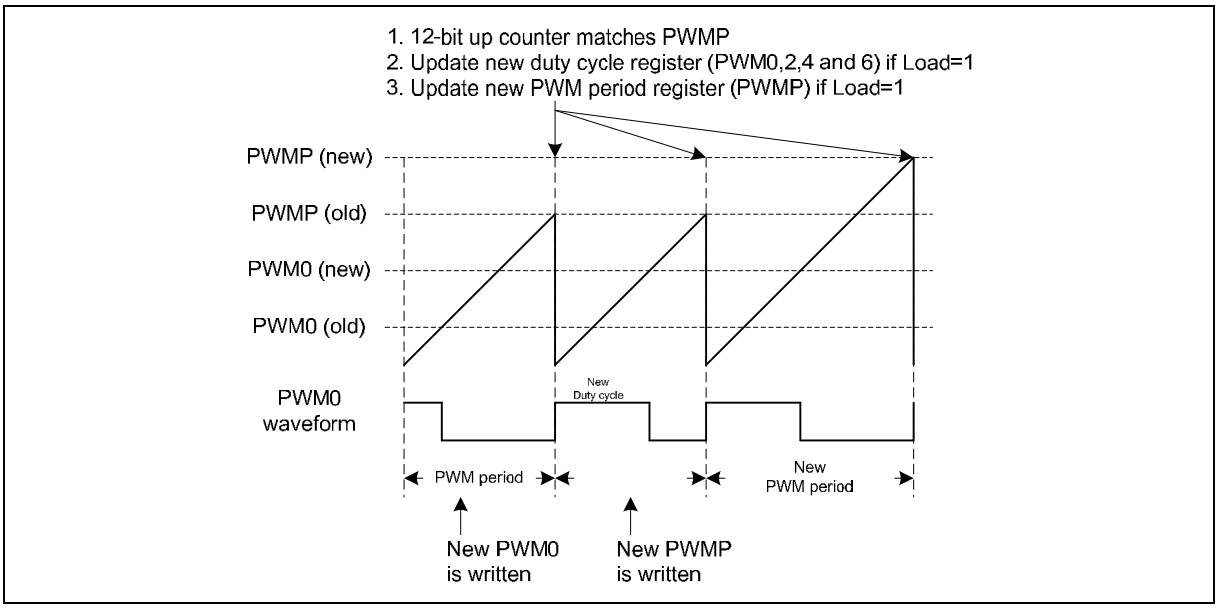



Figure 14-16: Edge-Aligned PWM

In edge-aligned PWM Output mode, the 12 bits counter will starts counting from 0 to match with the value of the duty cycle PWM0 (old). When the match occurs, it will toggle the PWM0 output waveform to low. After CPU resets, the value of PWM0 waveform at starts of counter depend on the polarity setting located in the Option bits. At this point a new PWM0 (new) is written. The counter will continue counting to match with the value of the period register, PWMP (old) and toggle the PWM0 waveform to high. Please take note that PWM0 and PWMP is a double-buffered register used to set the duty cycle and counting period for the PWM time base respectively. For the 1$^{st}$ buffer it is accessible by user while the 2$^{nd}$ buffer holds the actual compare value used in the present period. Load bit must be set to 1 to enable the value to be loaded in to the 2$^{nd}$ buffer register when counter underflow/match.

When the counter matches the PWMP (old) it will automatically update the new duty cycle register and the counter will again starts counting upwards to match the value PWM0 (new). At this point it will toggle the PWM0 waveform to low. New PWMP is written at this point. When the counter continues counting to match the PWMP (old), again the PWM0 waveform will be toggle to high. The counter starts counting from 0 again; at this point the value is PWM0 (new) and PWMP (new) to be match by the counter and once the counter matches these values it will be toggle at the PWM output.

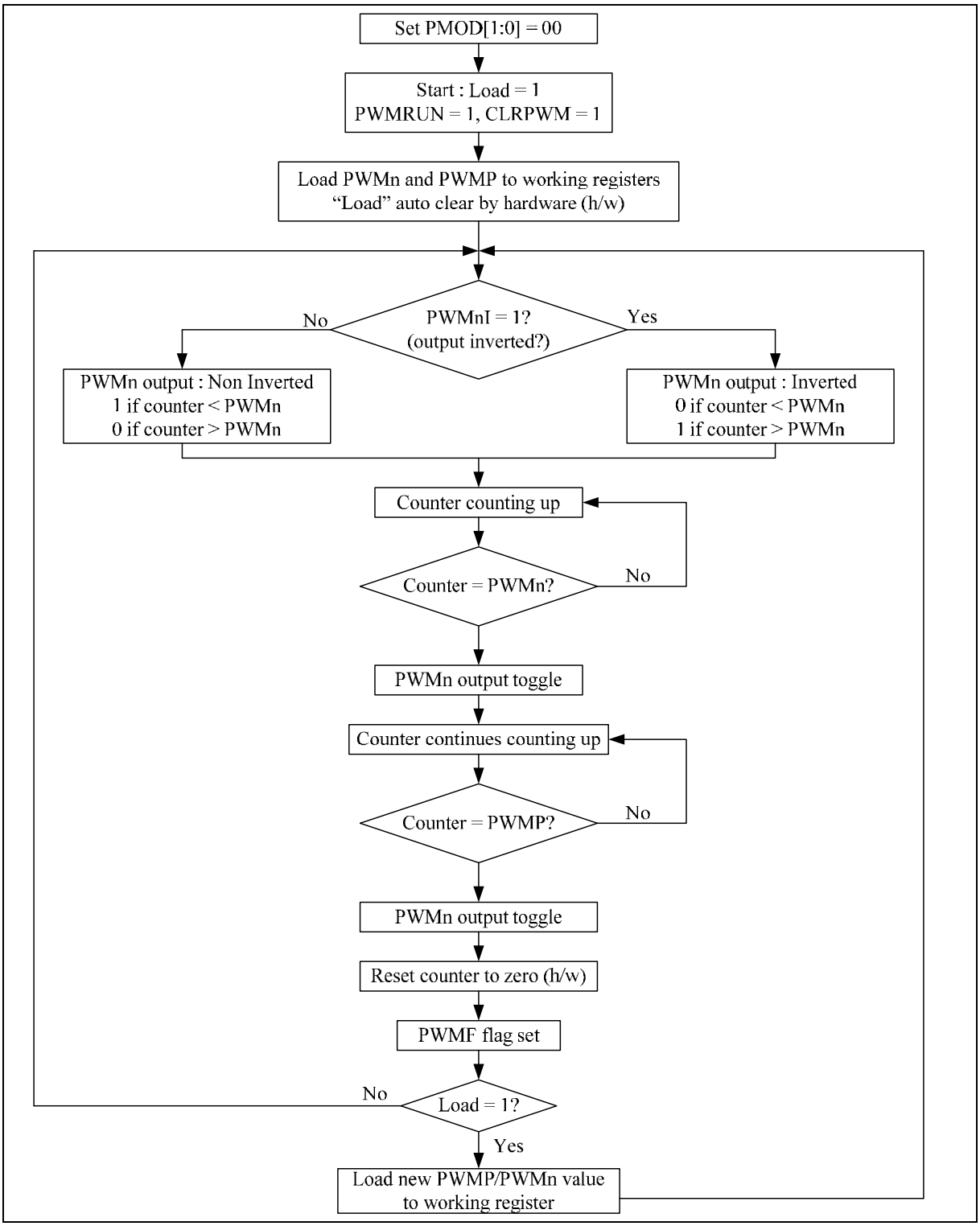



Figure 14-17: Edge-Aligned Flow Diagram

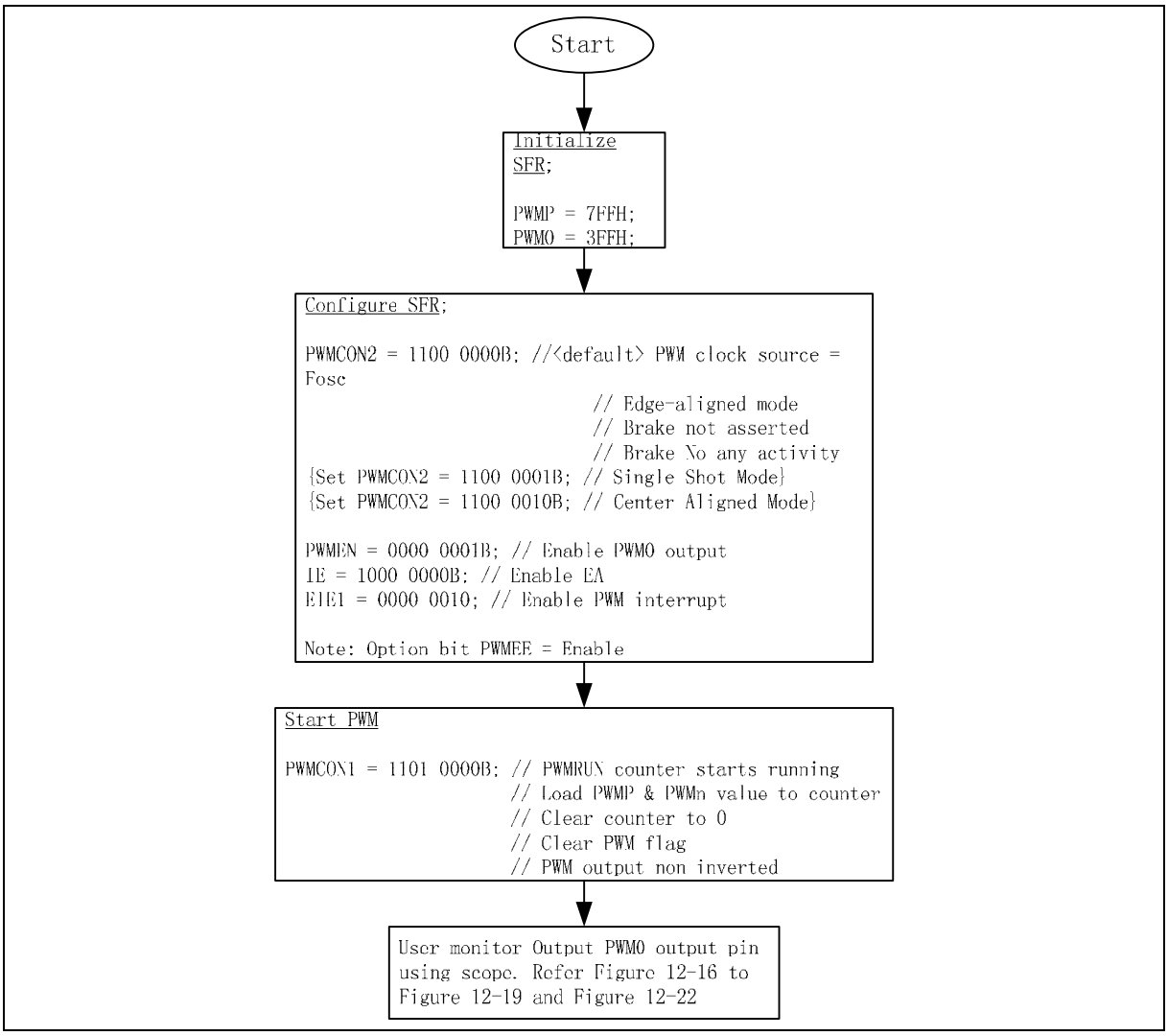



Figure 14-18: Program Flow for Edge-Aligned mode

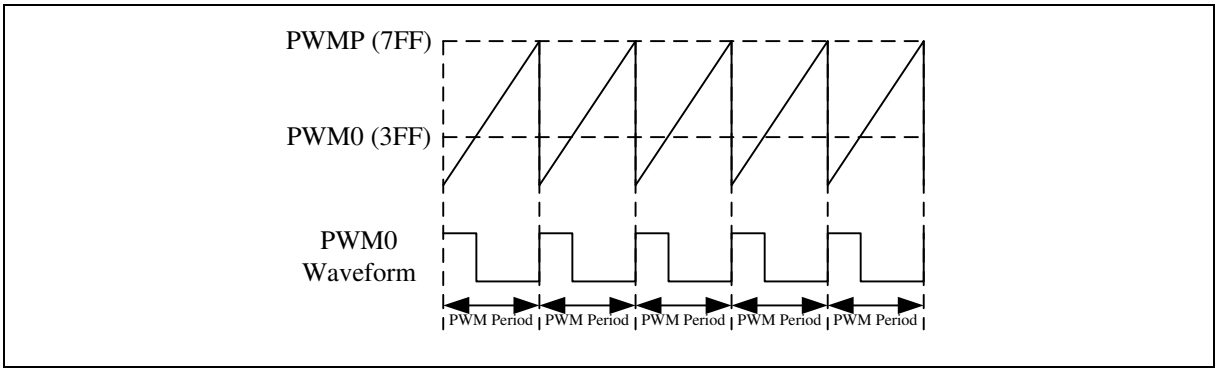



Figure 14-19: PWM0 Edge Aligned Waveform Output

## 14.8 Center Aligned PWM (up/down counter)

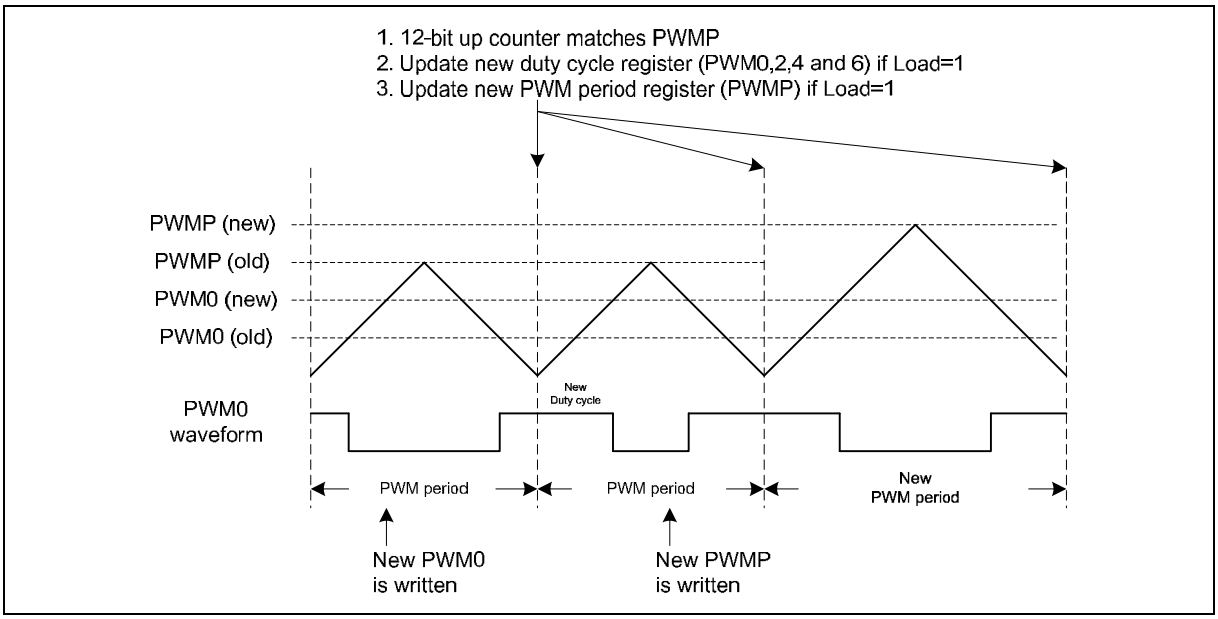



Figure 14-20: Center-Aligned Mode

Center-aligned PWM signals are produced by the module when the PWM time base is configured in an Up/Down Counting mode (see Figure 14-20). The counter will start counting-up from 0 to match the value of PWM0 (old); this will cause the toggling of the PWM0 output to low. The CPU reset states determine the starts value of PWM0 waveform at starts of counter lies on the polarity setting located in the Option bits. At this time the new PWM0 is written to the register. Counter continue to count and match with the PWMP (old). Upon reaching this states counter is configured automatically to down counting and toggle the PWM0 output when counter matches the PWM0 (old) value. Interrupt request when up/down counter underflow. Once the counter reaches 0 it will update the duty cycle register with Load = 1. Up-counting is continues with the matching at PWM0 (new) follow by a low toggle at the PWM0 output. By this time the PWMP buffer is still consist of the PWMP (old) value. A new PWMP is written. So the counter will still matches with this value and continues with down counting and matched the PWM0 (new) and toggle the PWM0 output.

Again updates on the PWM period register is reflected on the 3$^{rd}$ cycle of the diagram by starts counting from 0 to match the PWM0 (new) and toggle at the PWM0 output to low. Counter is continuing up-counting, upon reaching the PWMP (new) it is matched. Then counter is down counting automatically to match at the PWM0 (new) to get a toggle high at PWM0 output.

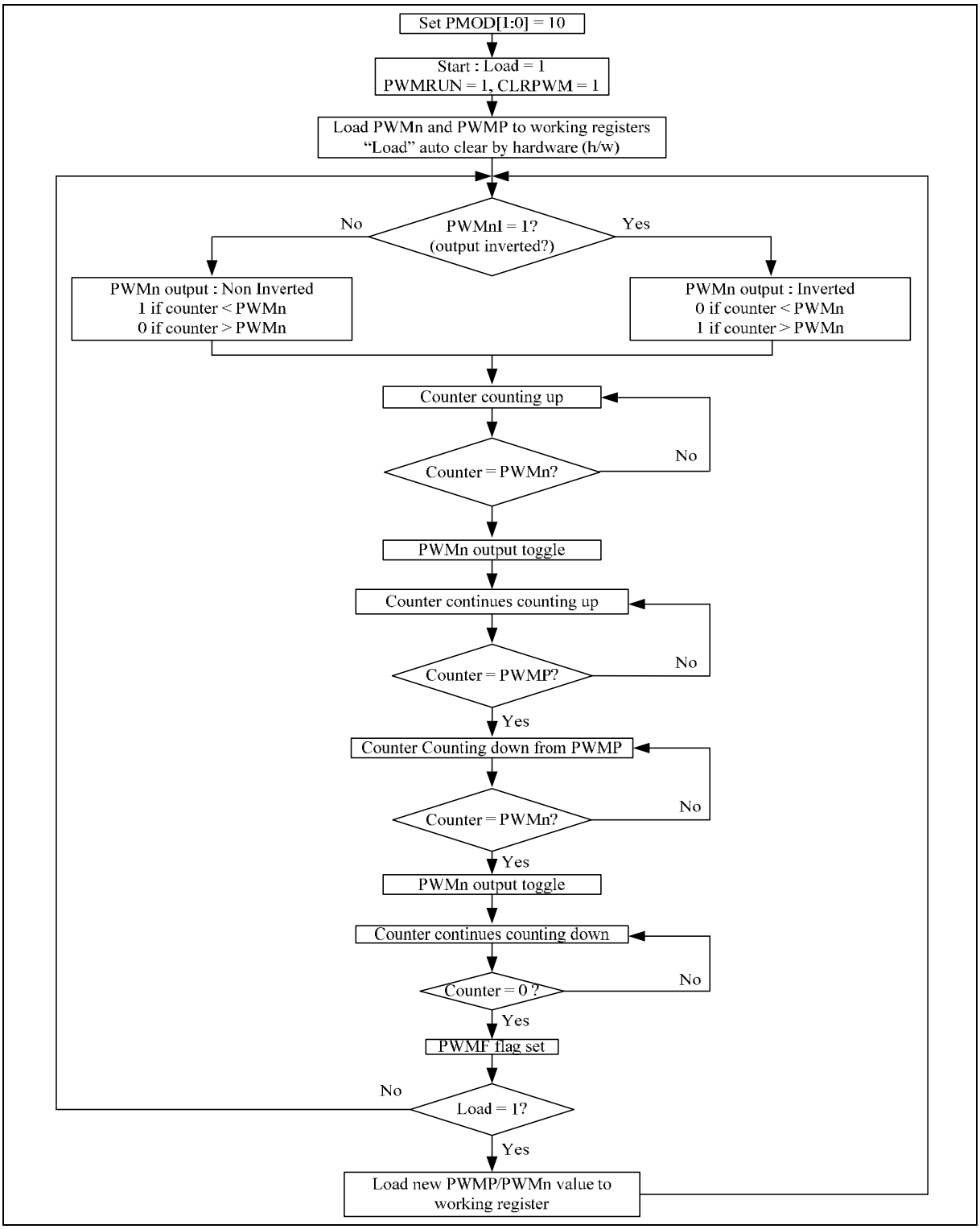



Figure 14-21: Center-aligned Flow Diagram

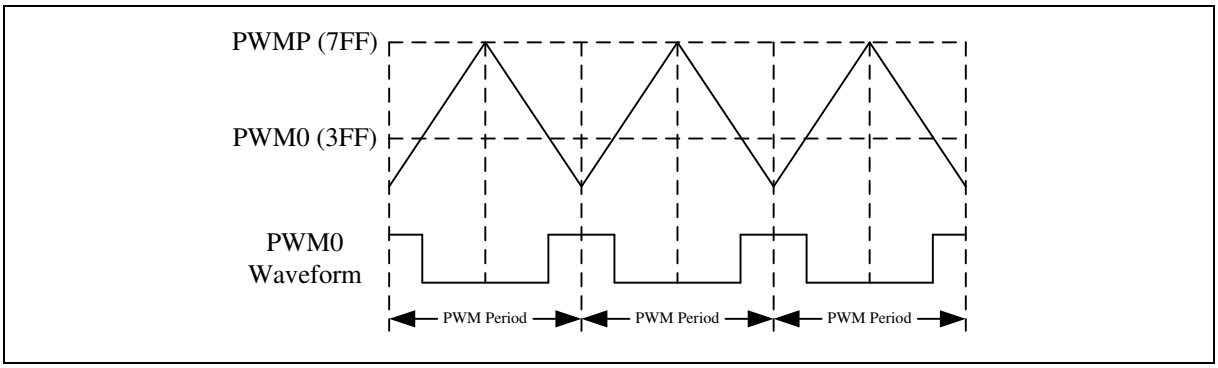



Figure 14-22: PWM0 Center Aligned Waveform Output

## 14.9 Single Shot (Up-Counter)

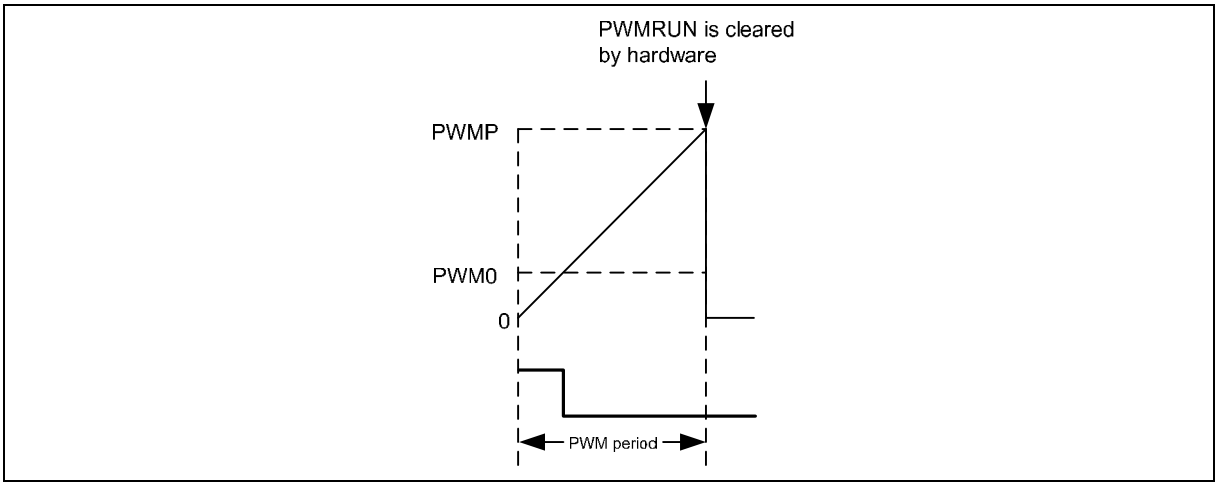



Figure 14-23: Single Shot Mode

The single shot mode PWM module will produce single pulse output. Single-pulse operation is configured when the PMOD1:PMOD0 bits are set to '01' in PWMCON3 register. This mode of operation is useful for driving certain types of ECMs. In this mode, the PWM counter will start counting upwards when the PWMRUN is set to 1. When the counter value matches with the PWMP register, PWM interrupt will be generated if it is enable and PWMF is set and counter will reset to zero on the following input clock edge and PWMRUN will be cleared by hardware. Duty cycle of PWM channels are determined by the respective PWMx registers, where x = 0,2,4,6

Example Steps of setting up Single Shot:-

1. Set initial state = 0 (controlled by EPOL option bit)
2. PWM0EN=0, POVM.0=0, PWM0I=0, PWM0=0000H(for keep comparator output in low state), PWMP=0001H(let the period as short as possible)
3. PWMRUN=1(Do a dummy PWMRUN for loading PWM0 to compare register0, which make comparator output LOW always.
4. PWM0EN=1, now the PWM0 pin should be still in 0 state.
5. PWMP=xxxxH(controls a period), PWM0=yyyyH(controls duty or pulse width)
6. PWMRUN=1(this time a real PWM single shot signal user wanted. The wave form should be the upper one.

**Note: In single shot mode, it's important that user sets CLRPWM together with PWMRUN and LOAD in order to have PWMn and PWMP loaded into working registers immediately.**

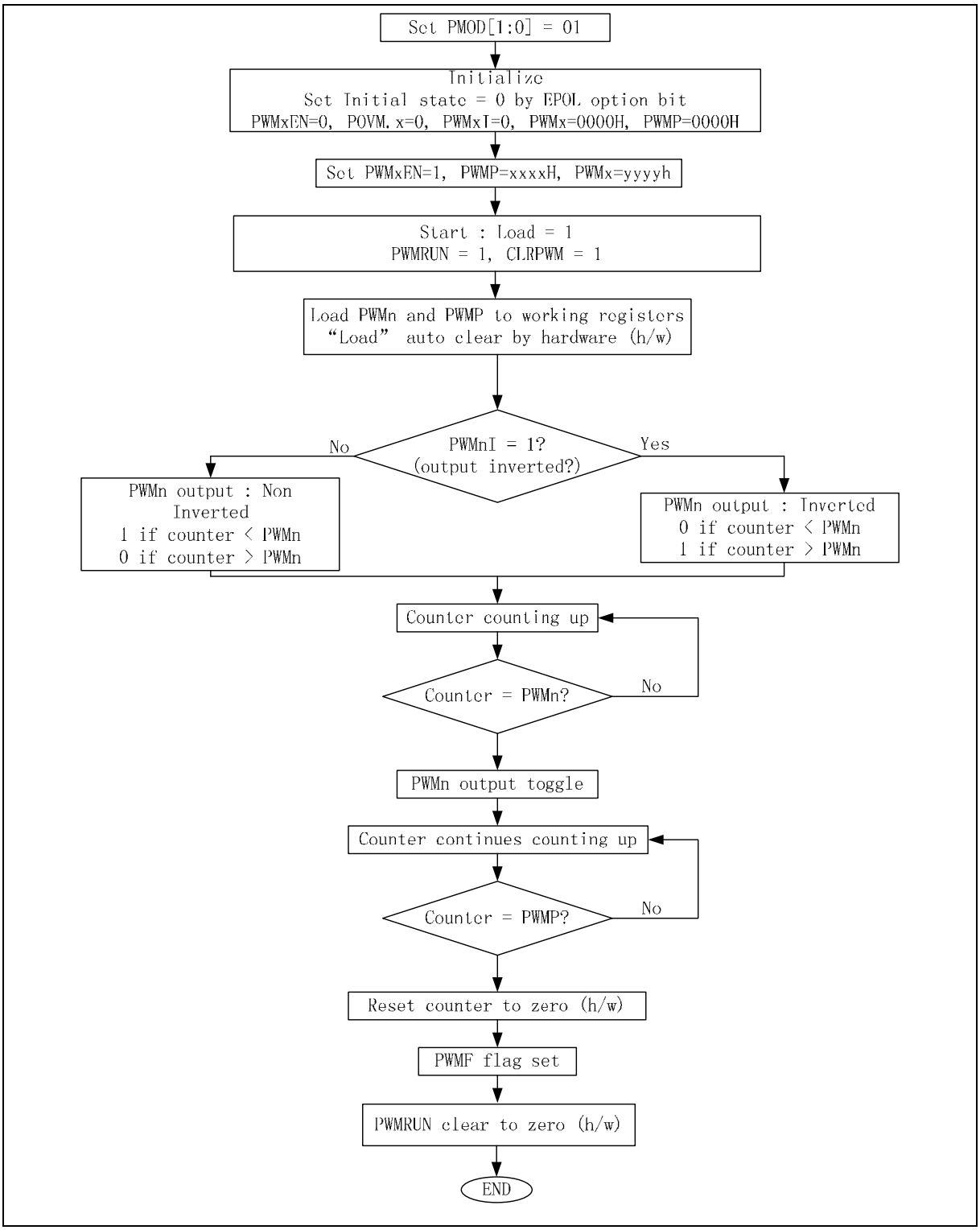



Figure 14-24: Single-Shot Flow Diagram

## 14.10 Smart Fault Detector

This is a brake detection logic that is new to support external brake conditions that already exist. A dedicated SFR FSPLT is added for this function. The SFR consists of smart fault detector control and status bits. It basically consists of a clock divider, 8 bits counter, comparator and 4 selectable compare values. The following diagram show the general block diagram.

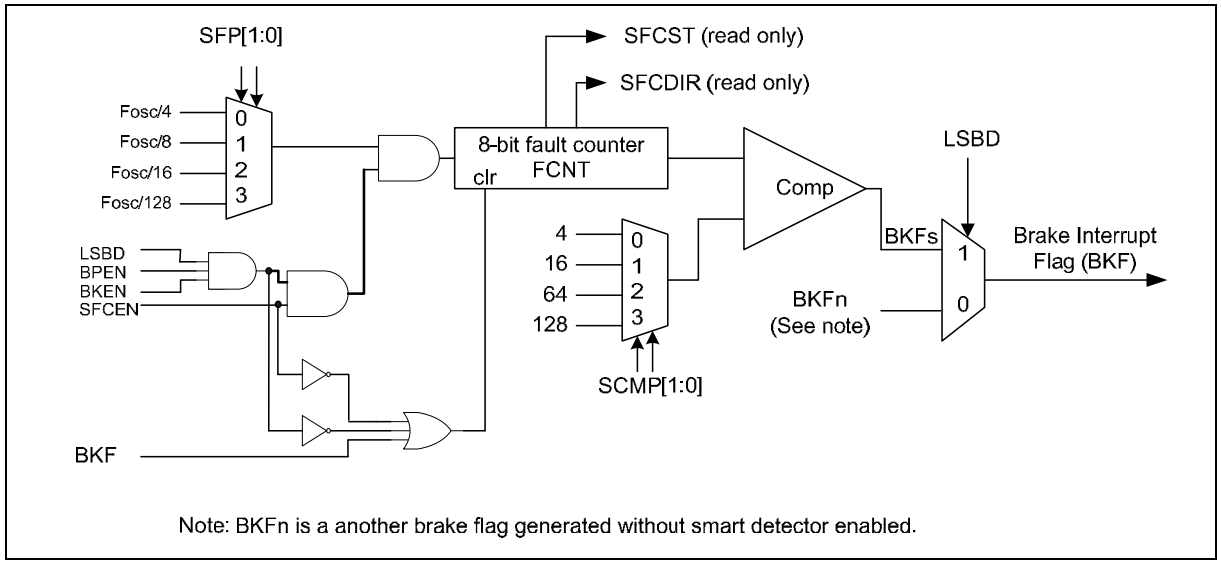



Figure 14-25: Smart Fault Detector

The smart fault detector is enabled when bit LSBD = 1 (FSPLT.0). This logic detects low level brake pin. The 8 bits counter is enabled by SFCEN bit located in SFR FSPLT.3. The counter is clock by Fosc divider selectable by SFP1-0 control bits (FSPLT.5-4). The comparator compares the 8 bits counter value with the compare value selectable with SCMP1-0 (FSPLT1-0).

Upon initial detection of low level at brake pin, the 8 bits counter will be active. This will cause the counter to increment. While the counter is active and there is high level detected at brake pin, the counter will decrement. See next figure for timing diagram. When the counter value reaches compare value, BKF will be asserted.

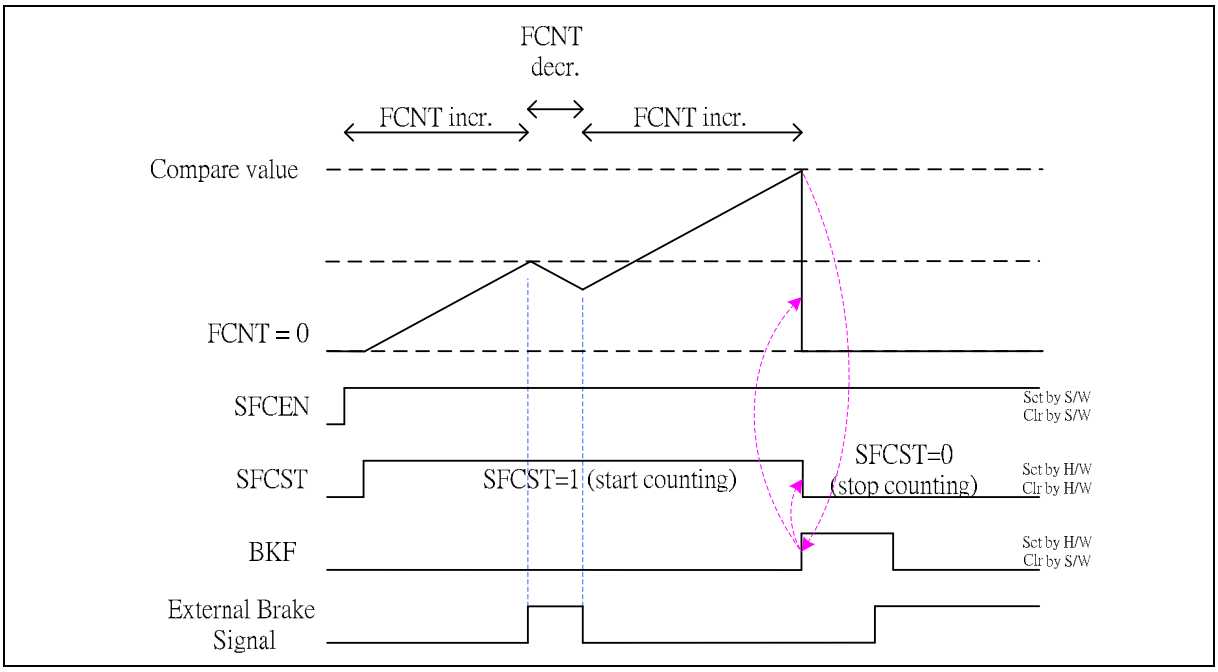



Figure 14-26: Smart Fault Detector timing diagram

The smart fault detector consists of 2 status bits; SFCST and SFCDIR. A SFCST show status of 8 bits counter is active or in-active, while SFCDIR shows the counter's counting direction. When SFCST = 0, SFCDIR keeps its' state.

The s/w can manually disable and clear the 8 bits counter, by clearing SFCEN to 0.

The following tables show the tabulate accumulated low level Brake time with various Fosc/x dividers and compares value, at 40MHz and 20MHz.

| FOSC/X | 1/4 | 1/8 | 1/16 | 1/128 |
|---|---|---|---|---|
| SCMP[1:0] | 10,000,000 | 5,000,000 | 2,500,000 | 312,500 |
| 4 | 0.40us | 0.80us | 1.60us | 12.80us |
| 16 | 1.60us | 3.20us | 6.40us | 51.20us |
| 64 | 6.40us | 12.80us | 25.60us | 204.80us |
| 128 | 12.80us | 25.60us | 51.20us | 409.60us |

Table 14-5: Example the accumulated low level time at 40 MHz

| FOSC/X | 1/4 | 1/8 | 1/16 | 1/128 |
|---|---|---|---|---|
| SCMP[1:0] | 5,000,000 | 2,500,000 | 1,250,000 | 156,250 |
| 4 | 0.80us | 1.60us | 3.20us | 25.60us |
| 16 | 3.20us | 6.40us | 12.80us | 102.40us |
| 64 | 12.80us | 25.60us | 51.20us | 409.60us |
| 128 | 25.60us | 51.20us | 102.40us | 819.20us |

Table 14-6: Example the accumulated low level time at 20 MHz

## 14.11 PWM Power-down/Wakeup Procedures

The following flow diagrams describe the possible pwm procedures users require to take care prior to the product power-down/wake-up. The power-down procedure below will result in PWM output a low state after power-down. To output a high state, users may set PWMn to FFFh and initial state set to high through option bit (EPOL/OPOL).

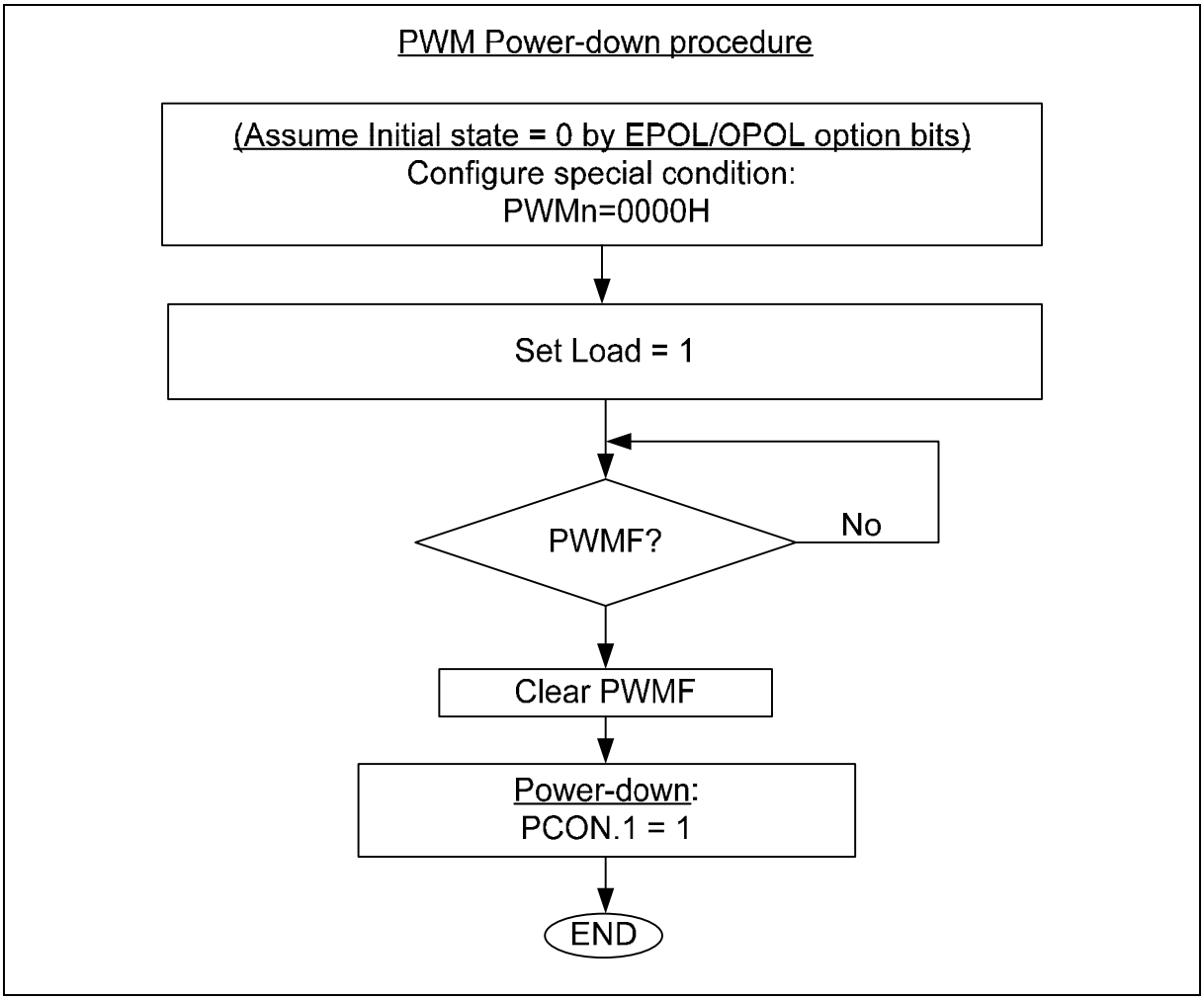



Figure 14-27: Example of PWM power-down procedure (pwm output low state)

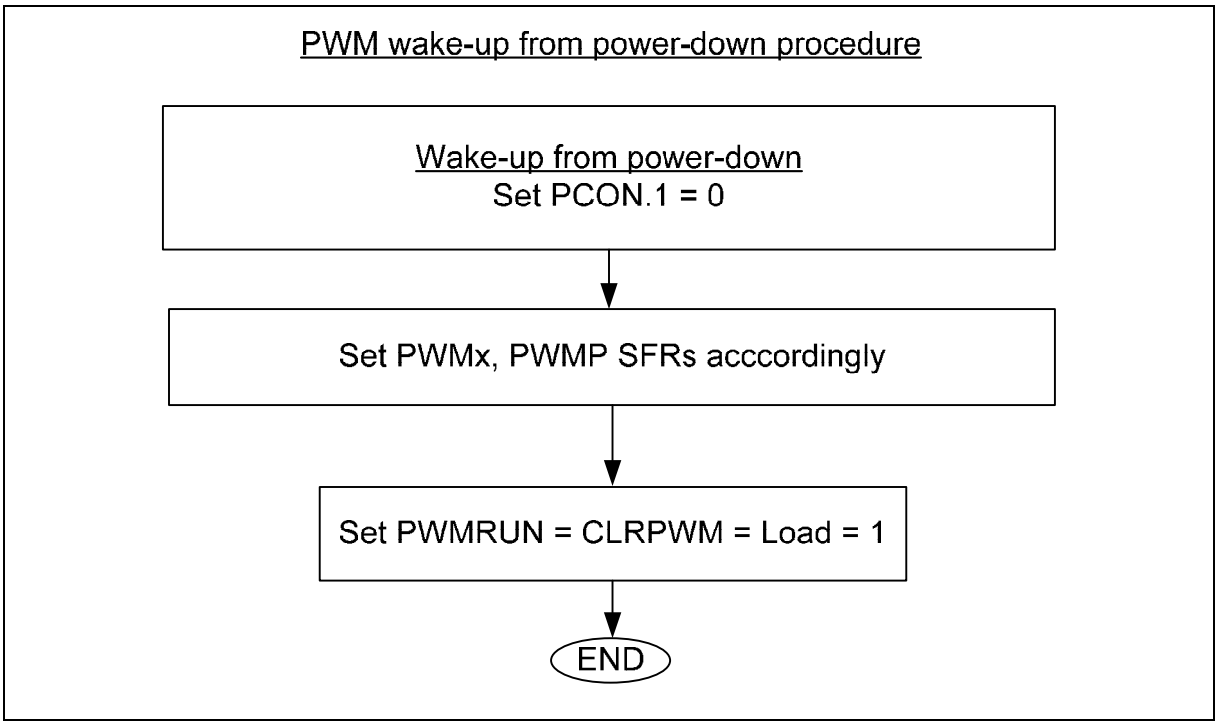



Figure 14-28: Example of PWM wake-up from power-down procedure

# 15. MOTION FEEDBACK MODULE

Motion feedback module is a peripheral module designed for motion feedback applications. This module includes two sub-modules:

- Input Capture Module (IC).
- Quadrature Encoder Interface (QEI).

There are three 16-bit registers cascaded by two 8-bit SFR in motion feedback module, but with different definitions in each sub-module. Together with Timer 3, these modules provide a number of options for motion and control applications. Most of the features for the QEI and IC sub-modules are fully programmable thus making a flexible peripheral structure that can accommodate a wide range of uses. A simplified block diagram of the entire Motion Feedback module is shown in Figure 15-2.

**Note**: The input pins are common to the IC and QEI sub-modules, only one of these two sub-modules may be used at any given time. IC sub-module is the default value upon reset.

## 15.1 Input Capture Module (IC)

The capture modules are function to detect and measure pulse width and period of a square wave.

It supports 3 capture inputs and digital noise rejection filter. The modules are configured by CAPCON0 and CAPCON1 SFR registers. Input Capture 0, 1 & 2 have their own edge detector but share with one timer i.e. Timer 3. The Input Capture pins structure are Schmitt trigger. For this operation it basically consists of;

- 3 capture module function blocks.
- Timer 3 block.

Each capture module block consists of 2 bytes of capture registers, noise filter and programmable edge triggers. Noise Filter is used to filter the unwanted glitch or pulse on the trigger input pin.

The noise filter can be enabled through bit ENFx (CAPCON1). If enabled, the capture logic required to sample 4 consecutive same capture input value in order to recognize an edge as a capture event. A possible implementation of digital noise filter is as follow; the interval between pulses requirement for input capture is 1 machine cycle width, which is the same as the pulse width required to guarantee a trigger for all trigger edge mode. For less than 3 system clocks, anything less than 3 clocks will not have any trigger and pulse width of 3 or more but less than 4 clocks will trigger but will not guarantee 100% because input sampling is at stage C3 of the machine cycle.

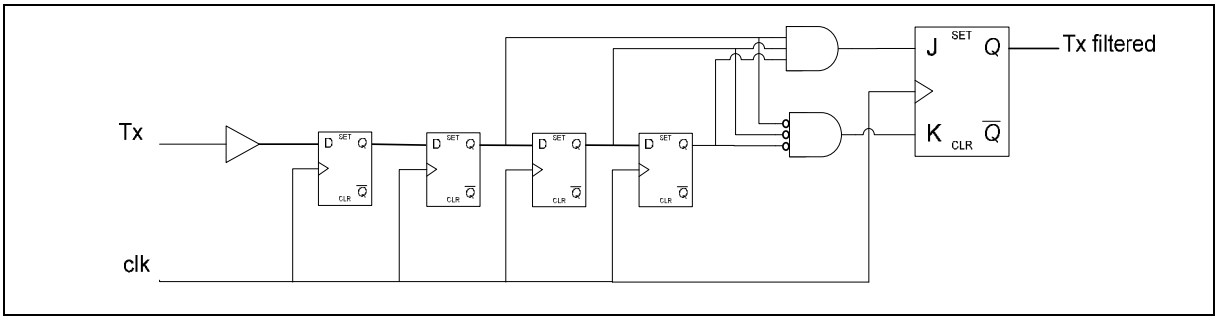
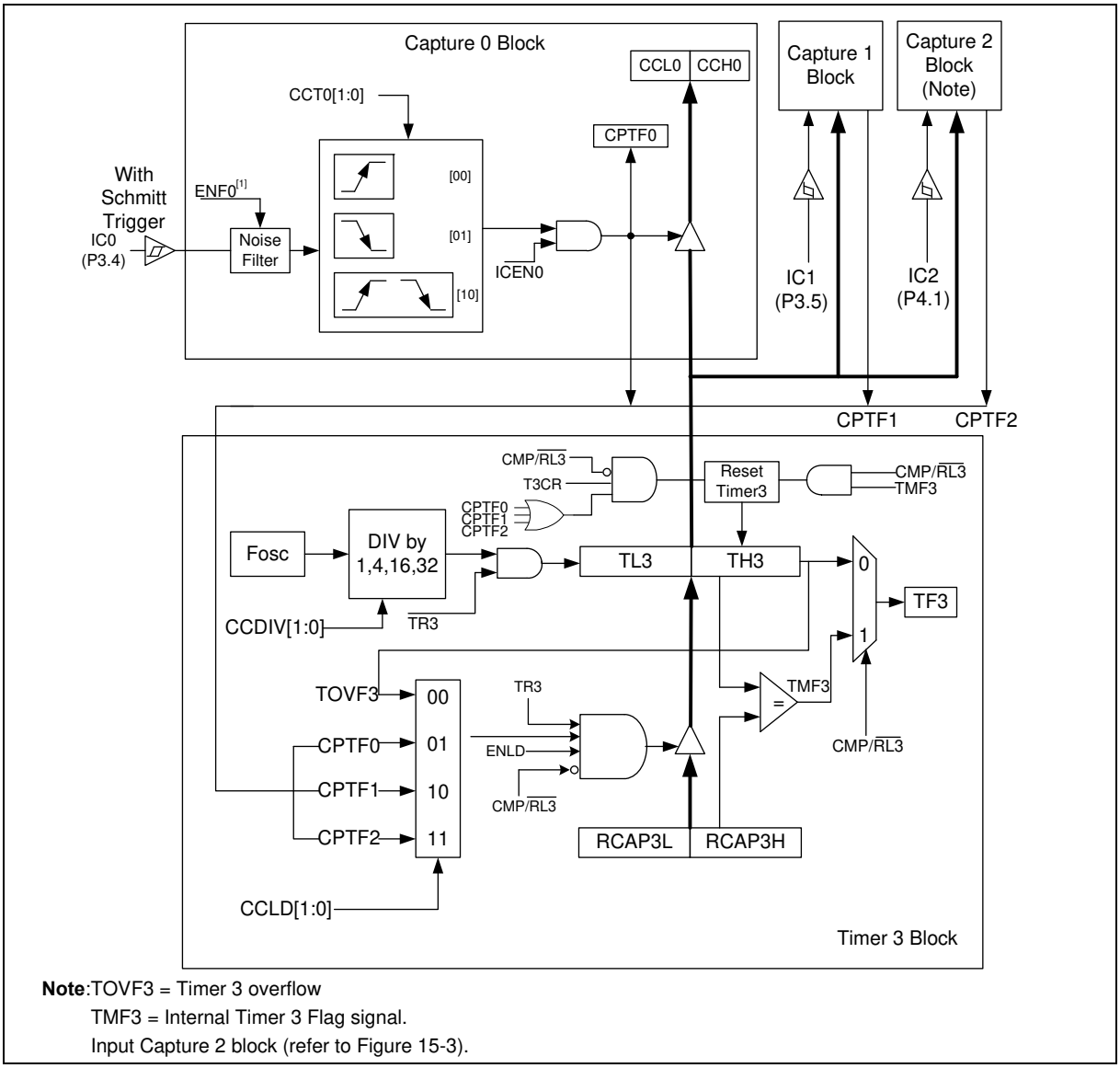



Figure 15-1: Noise Filter

The trigger option is programmable through CCTx [1:0] (CAPCON0). It supports positive edge, negative edge and both edge triggers. Each capture module consists of an enable, ICEN0~2. [Note: x=0, 1, 2 for capture 0, 1, 2 block].

Capture blocks are triggered by external pins IC0, IC1 and IC2, respectively. If ICENx is enabled, each time the external pin triggers, the content of the free running 16 bits counter, TL3 & TH3 (from Timer 3 block) will be captured/transferred into the corresponding capture registers, CCLx and CCHx. This action also causes the corresponding CPTFx flag bit in CAPCON1 to be set, and generate an interrupt (if enabled by ECPTF bit in SFR, EIE1.4). The CPTF0-2 flags are logical "OR" to the interrupt module. Input Capture 0~2 share one interrupt named Capture Interrupt. Flag is set by hardware and cleared by software.

Setting the T3CR bit (T3MOD.3), will allow hardware to reset timer 3 automatically after the value of TL3 and TH3 have been captured. Priority is given to T3CR to reset counter after capture the timer value into the capture register. When CMP/$\overline{RL3}$ = 0 (reload mode, with T3CR=0 and ENLD=1), RCAP3 will be loaded into Timer 3 counter upon overflow. While the rest of the condition of combination of setting for T3CR and ENLD will reset the counter to 0000H.




**Note**:TOVF3 = Timer 3 overflow
TMF3 = Internal Timer 3 Flag signal.
Input Capture 2 block (refer to Figure 15-3).

Figure 15-2: Timer3/Capture/Compare/Reload modules

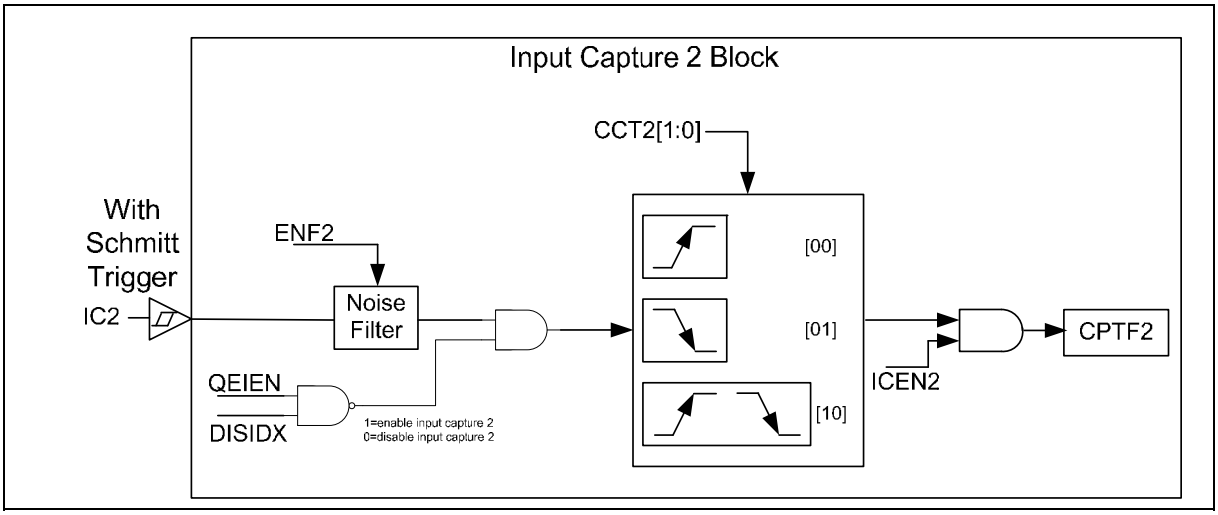



Figure 15-3: Input Capture 2 block diagram

**Note: When QEI enabled (QEIEN=1), input capture 2 (IC2) still can detect edge changes.**

.

The following table shows the bits setting for enabling input capture 2 edge detection.

| QEIEN | DISIDX | ICEN2 | INPUT CAPTURE 2 EDGE DETECTION |
|---|---|---|---|
| 0 | X(don't care) | 0 | Disabled. |
| | | 1 | Enabled. |
| 1 | 0 | 0 | Disabled. |
| | | 1 | Enabled. |
| 1 | 1 | X | Disabled, |

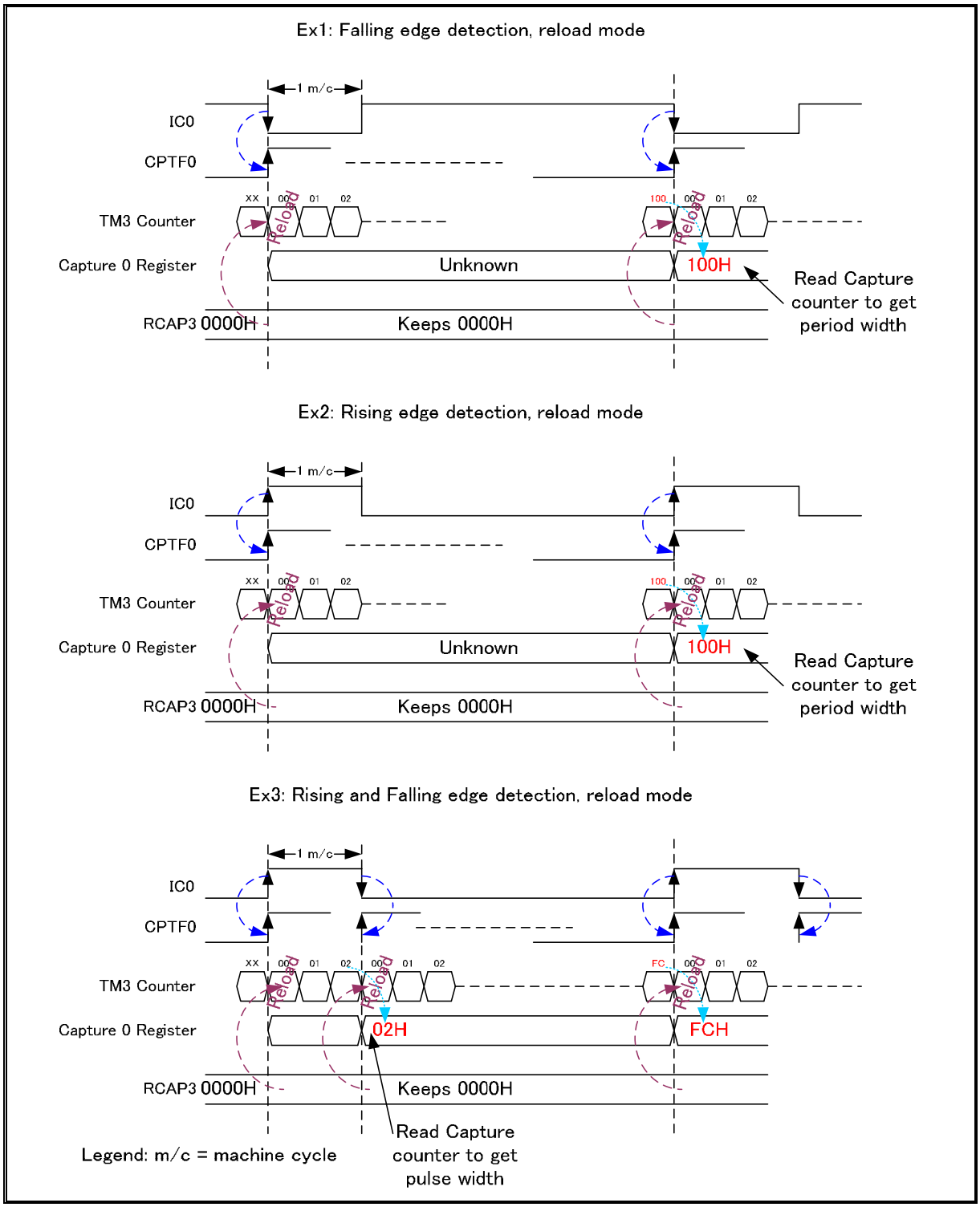



Figure 15-4: Timing diagram for Input Capture

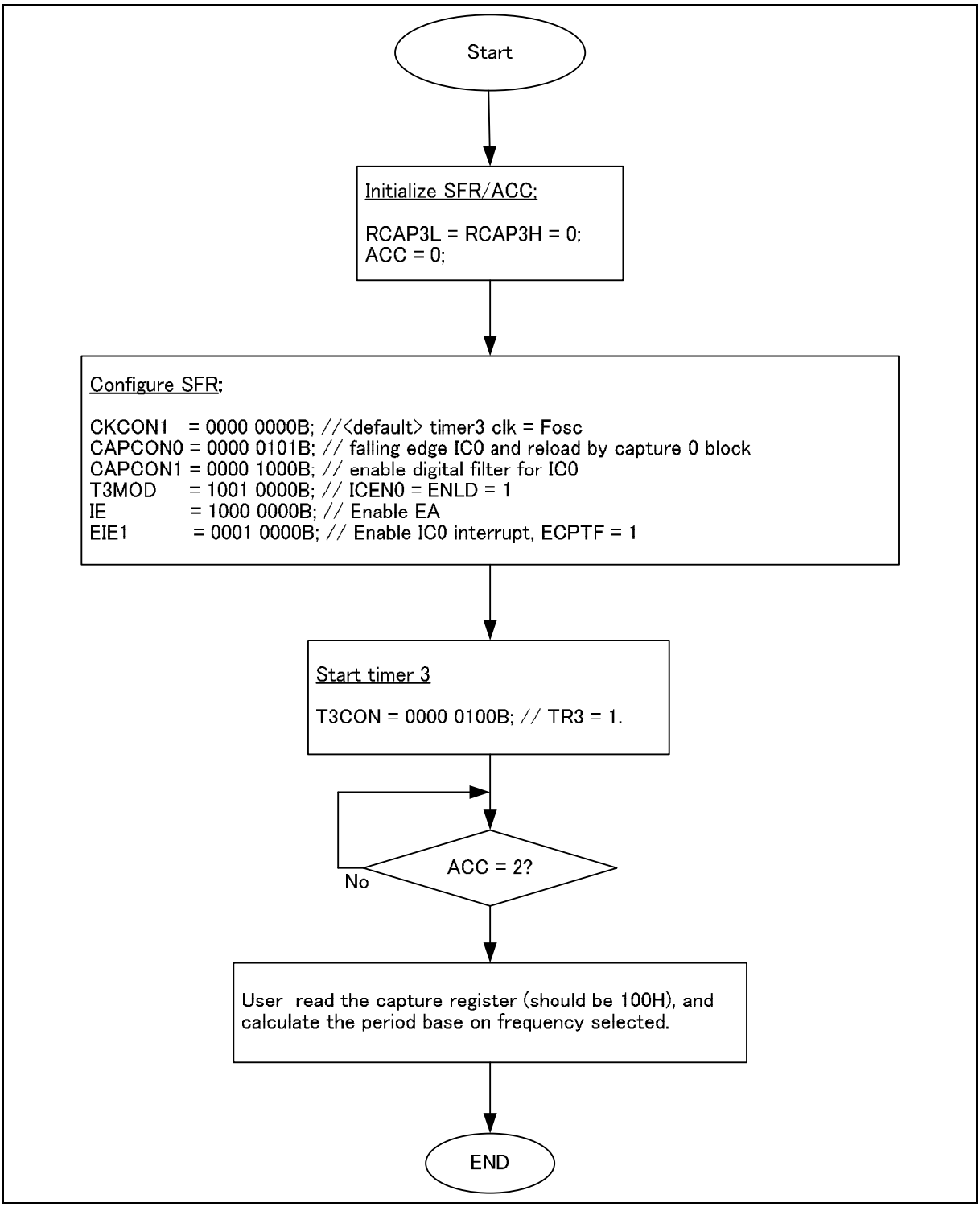



Figure 15-5: Program flow for measurement with IC0 between pulses with falling edge detection (ACC is incremented in interrupt service routine).

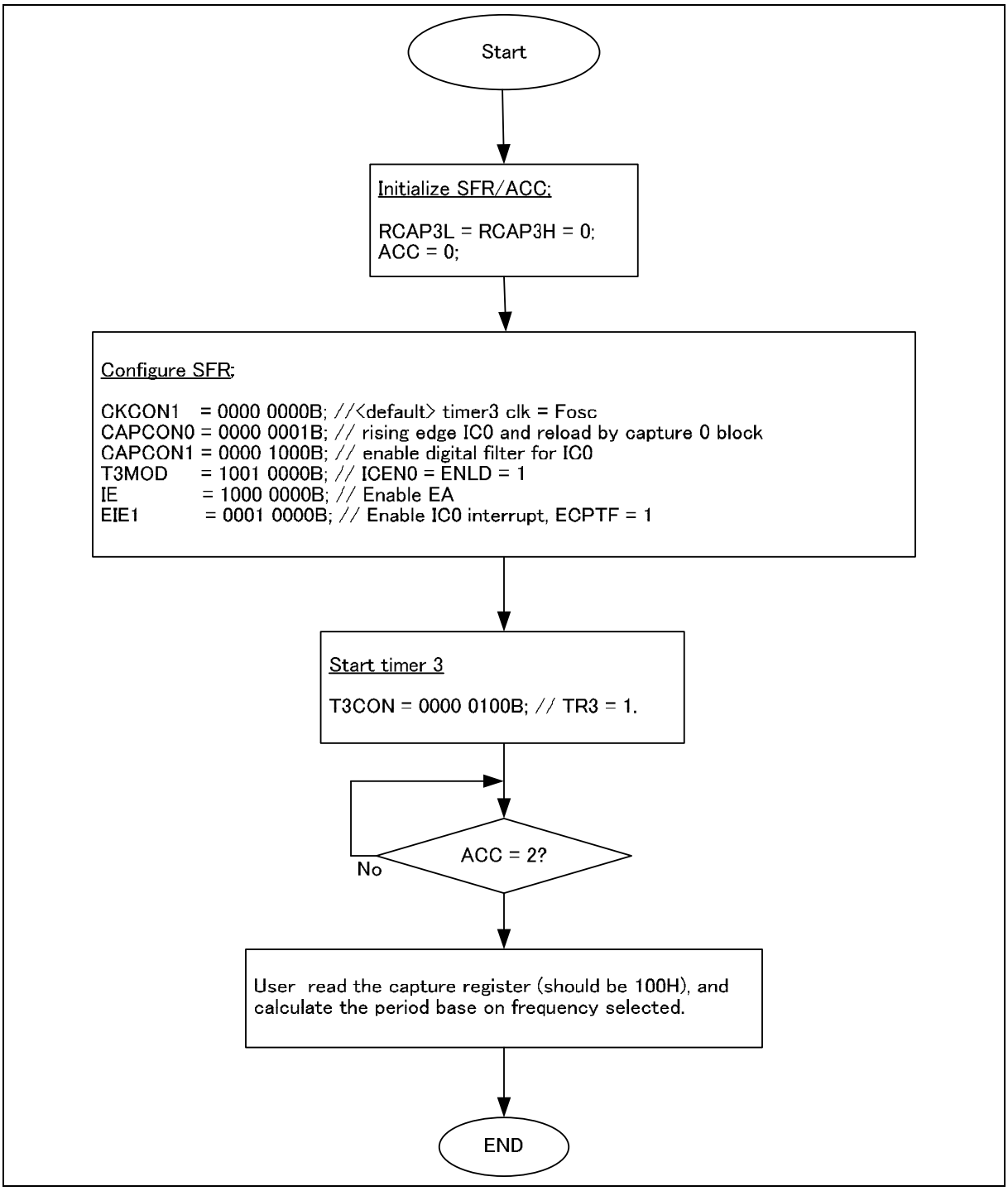



Figure 15-6: Program flow for measurement with IC0 between pulses with rising edge detection (ACC is incremented in interrupt service routine).

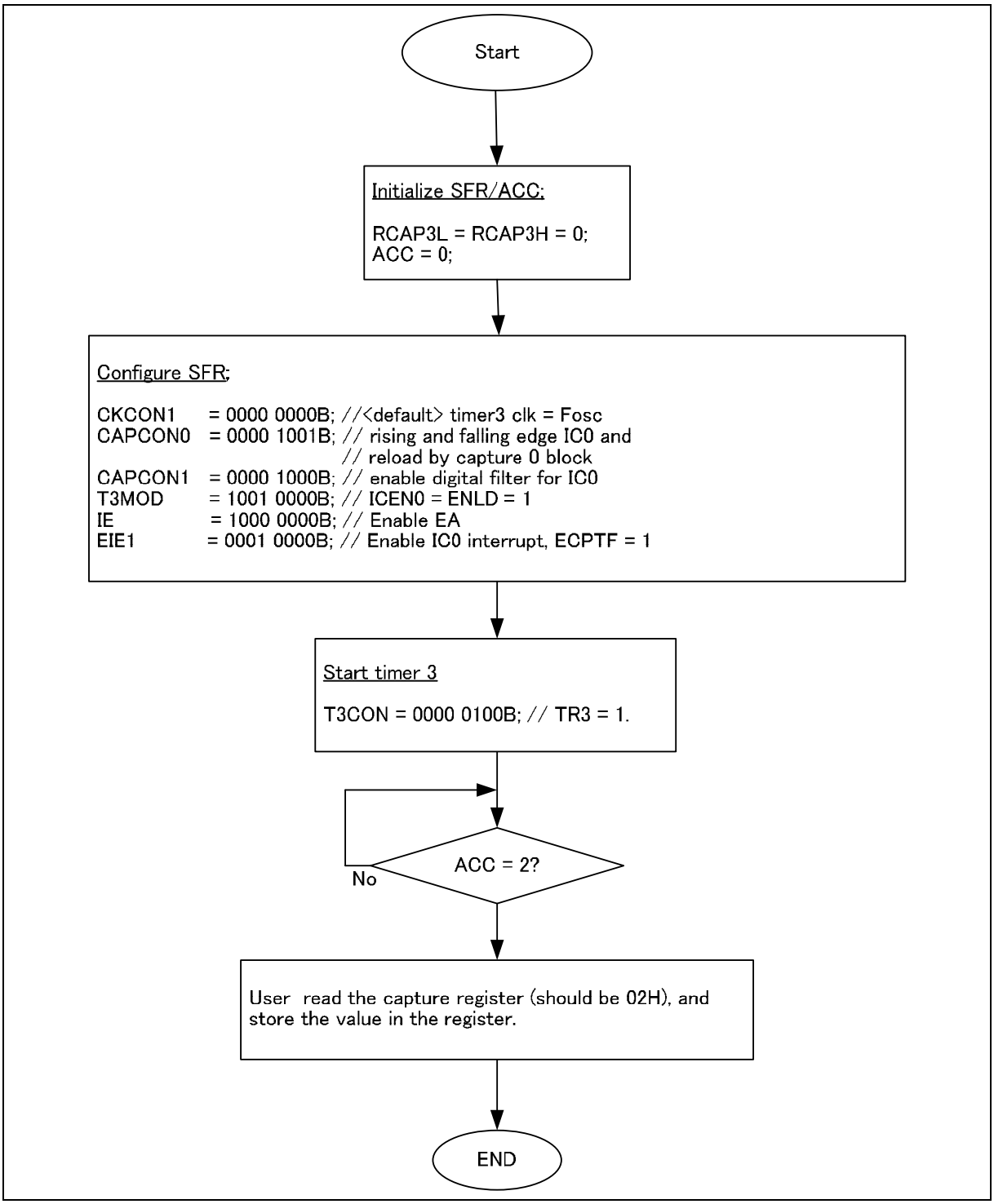



Figure 15-7: Program flow for measurement with IC0 pulse width with rising and falling edge detection (ACC is incremented in interrupt service routine).

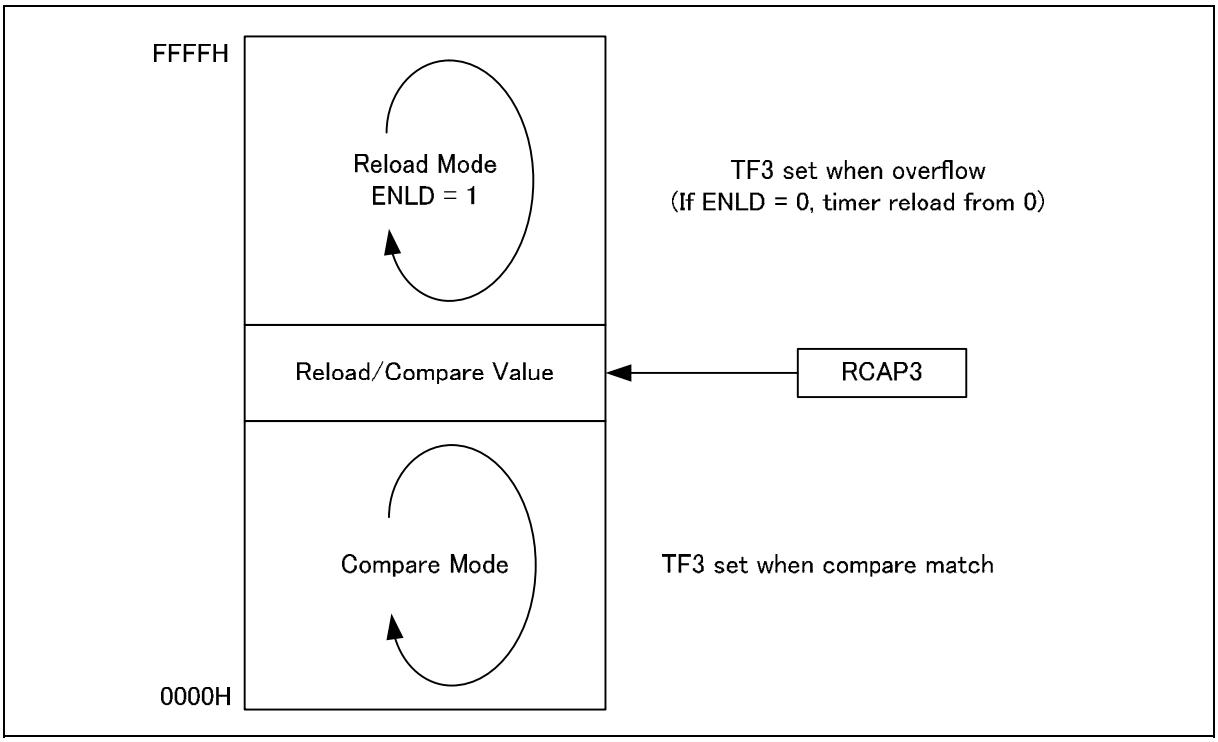



Figure 15-8: Compare/Reload Function

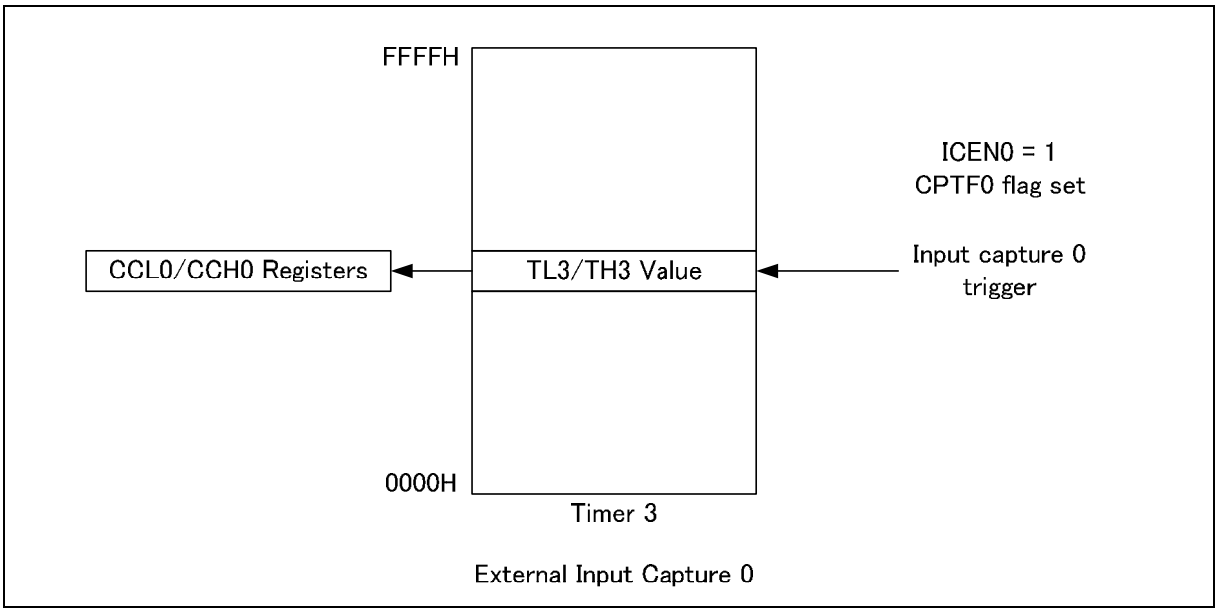



Figure 15-9: Input Capture 0 Triggers

### 15.1.1 Compare Mode

Timer 3 can be configured for compare mode. The compare mode is enabled by setting the CMP/$\overline{RL3}$ bit to 1 in the T3CON register. RCAP3 will serves as a compare register. As Timer 3 counting up,

upon matching with RCAP3 value, TF3 will be set (which will generate an interrupt request if enable Timer 3 interrupt ET3 is enabled) and the timer reload from 0 and starts counting again.

### 15.1.2 Reload Mode

Timer 3 can be also be configured for reload mode. The reload mode is enabled by clearing the CMP/$\overline{RL3}$ bit to 0 in the T3CON register. In this mode, RCAP serves as a reload register. When timer 3 overflows, a reload is generated that causes the contents of the RCAP3L and RCAP3H registers to be reloaded into the TL3 and TH3 registers, if ENLD is set. TF3 flag is set, and interrupt request is generated if enable Timer 3 interrupt ET3 is enabled. However, if ENLD = 0, timer 3 will be reload with 0, and count up again.

Alternatively, other reload source is also possible by the input capture pins by configuring the CCLD [1:0] bit. If the ICENx bit is set, then a trigger of external IC0, IC1 or IC2 pin (respectively) will also cause a reload. This action also sets the CPTF0, CPTF1 or CPTF2 flag bit in SFR CAPCON1, respectively.

## 15.2 Quadrature Encoder Interface (QEI)

The Quadrature Encoder Interface (QEI) decodes speed of rotation and motion sensor information. It can be used in any application that uses quadrature encoder for feedback. The QEI block supports the features as below:

- Two QEI phase inputs: QEA and QEB.
- 16-bit Up/Down Pulse Counter (PLSCNT) with 16-bit read access latched buffer (PCNT).
- Four pulse counter update modes:
  - Mode0: x4 free-counting mode.
  - Mode1: x2 free-counting mode.
  - Mode2: x4 compare-counting mode.
  - Mode3: x2 compare-counting mode.
- Three interrupt sources:
  - Pulse counter interrupt (CPTF0/QEIF).
  - Direction index of motion detection with direction interrupt (CPTF1/DIRF).
  - Input Capture 2 interrupt (CPTF2).
- The three 16-bit SFRs in QEI share the same addresses with the capture counter registers.

| INPUT CAPTURE MODE | QEI MODE |
|---|---|
| Capture0 Counter Register (CCH0, CCL0) | Pulse Read Counter Register (PCNTH, PCNTL) |
| Capture1 Counter Register (CCH1, CCL1) | Pulse Counter Register (PLSCNTH, PLSCNTL) |
| Capture2 Counter Register (CCH2, CCL2) | Maximum Counter Register (MAXCNTH, MAXCNTL) |

In QEI mode, IC1 and IC0 work as QEB and QEA inputs respectively. QEA and QEB accept the outputs from a quadrature encoded source, such as incremental optical shaft encoder. Two channels, A and B, nominally 90 degrees out of phase, are required.

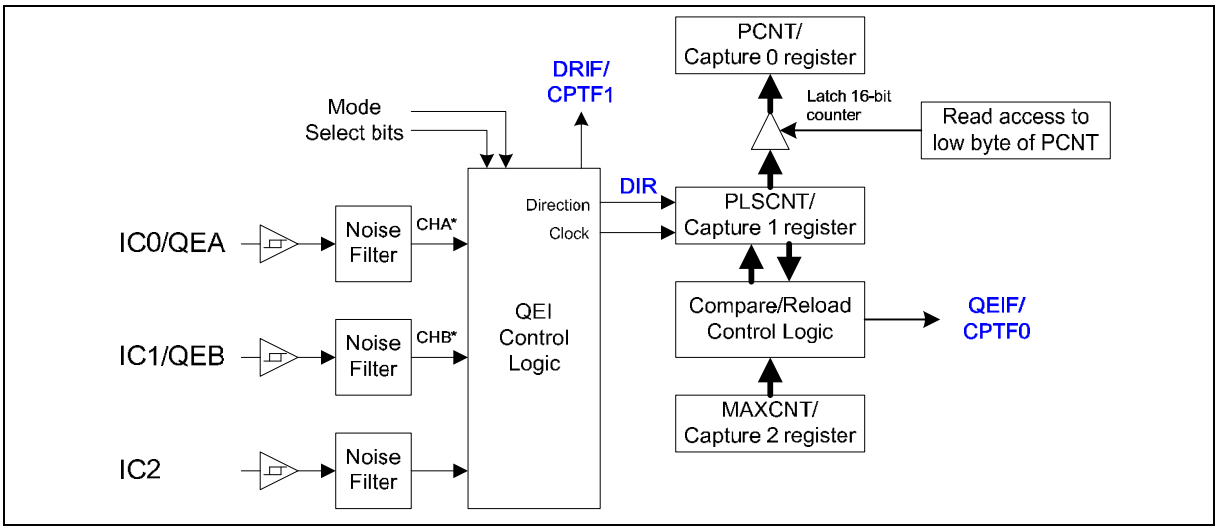



Figure 15-10: QEI Block Diagram

The QEI control logic detects the relation of phase lead/lag between QEA and QEB to produce direction index (DIR) and clock to control pulse counter. The comparator/reload logic compares the pulse counter and maximum count and control the function of reloading pulse counter in compare-counting mode. In Free-counting mode, the pulse counter will counts until the 65535 value. In Compare-counting mode, the pulse counter will count to MAXCNT value. The value of the pulse counter is not affected during QEI mode changes, nor when the QEI is disabled altogether.

In QEI mode, when IC2 edge (rising/falling edge is programmable through CAPCON0) has been detected, CPTF2 will be set (if QEIEN=ICEN2=1 and DISIDX=0), and the only way to clear it is by software.

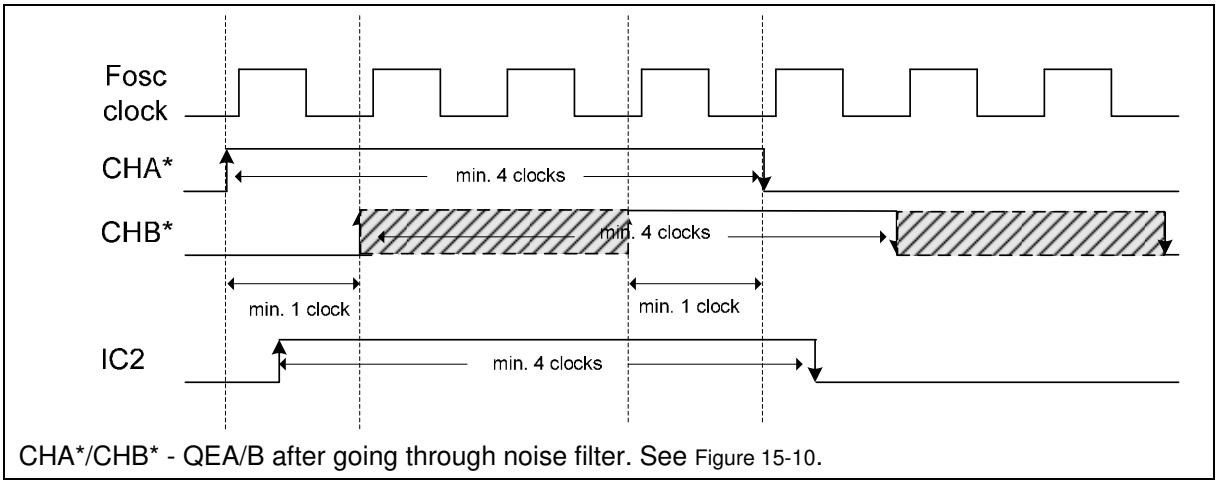



Figure 15-11: QEA/QEB/IC2 timing requirement.

### 15.2.1 Free-counting mode

Pulse counter up or down counts according to direction index (DIR). When overflow or underflow occurs, it sets flag QEIF.

### 15.2.2 Compare-counting mode

Pulse counter up or down counts according to direction index (DIR). On up counting, QEIF will be asserted when PLSCNT overflows from MAXCNT to zero on the next QEA edge for x2 counting mode, and on QEA/QEB edge for x4 counting mode. On down counting, QEIF will be asserted when PLSCNT underflows from zero to MAXCNT on the next QEA edge for x2 counting mode, and on QEA/QEB edge for x4 counting mode. This mode provides the position of a rotor to user. If a quadrature encoder output 1024 pulses to QEA per round, user can write MAXCNT with 4095 in x4 mode or 2047 in x2 mode and reset PLSCNT at initial before rotor runs. When the PLSCNT reaches MAXCNT, it means rotor runs one round on next QEA edge.

### 15.2.3 X2/X4 Counting modes

In X2 counting mode, the pulse counter increases or decreases one on every QEA edge based on the phase relationship of QEA and QEB signals, however:-

In X4 counting mode, the pulse counter increases or decreases one on every QEA and QEB edge based on the phase relationship of QEA and QEB signals.

### 15.2.4 Direction of Count

If QEA lead QEB, the pulse counter is increased by 1. If QEA lags QEB, the pulse counter is decreased by 1. The QEI control logic generates a signal that sets the DIR bit (QEICON.3); this in turn determines the direction of the count. When QEA leads QEB, DIR is set (= 1), and the position counter increments on every active edge. When QEA lags QEB, DIR is cleared, and the position counter decrements on every active edge. Refer to below table.

| CURRENT SIGNAL DETECTED | PREVIOUS SIGNAL DETECTED | | | | COUNTING CONTROL (DIR) |
|---|---|---|---|---|---|
| | RISING | | FALLING | | |
| | QEA | QEB | QEA | QEB | |
| QEA rising | | | | ✓ | INC (1) |
| | | ✓ | | | DEC (0) |
| QEA falling | | | | ✓ | DEC (0) |
| | | ✓ | | | INC (1) |
| QEB rising | ✓ | | | | INC (1) |
| | | | ✓ | | DEC (0) |
| QEB falling | | | ✓ | | INC (1) |
| | ✓ | | | | DEC (0) |

Table 15-1: Direction of count

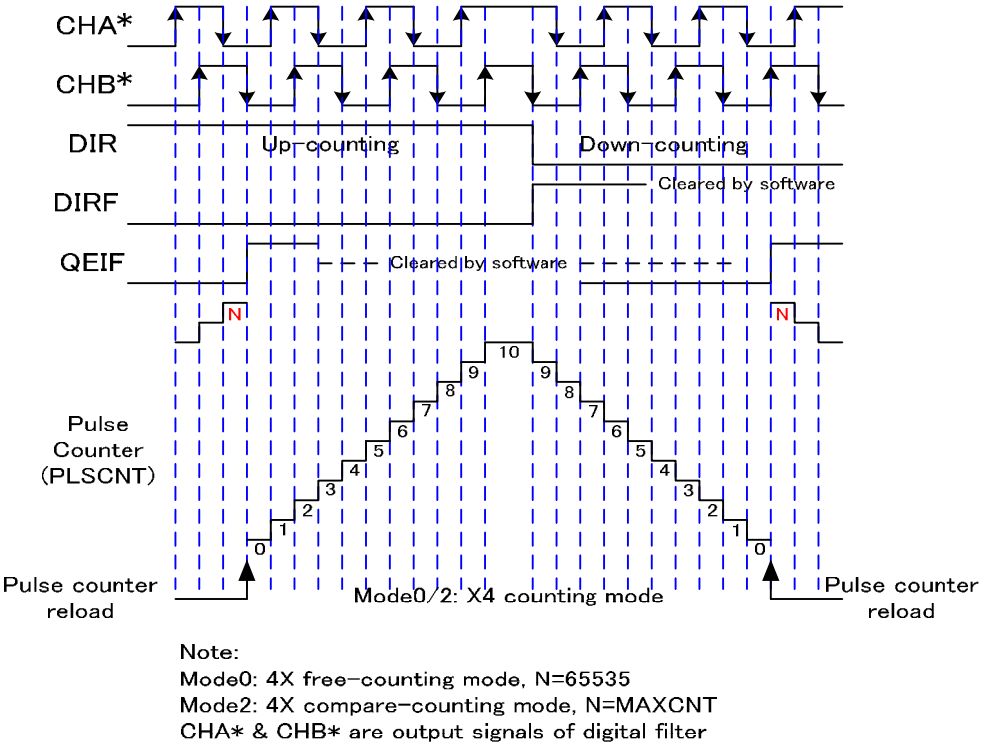



Figure 15-12: X4 Counting Mode

QEI x4 Counting mode provides for a finer resolution of the rotor position, since the counter increments or decrements more frequently for each QEA/QEB input pulse pair than in QEI x2 mode. This mode is selected by setting the QEI Mode Select bits to '00b' or '10b'. In this mode, the QEI logic detects every edge on every QEA and QEB input edges.

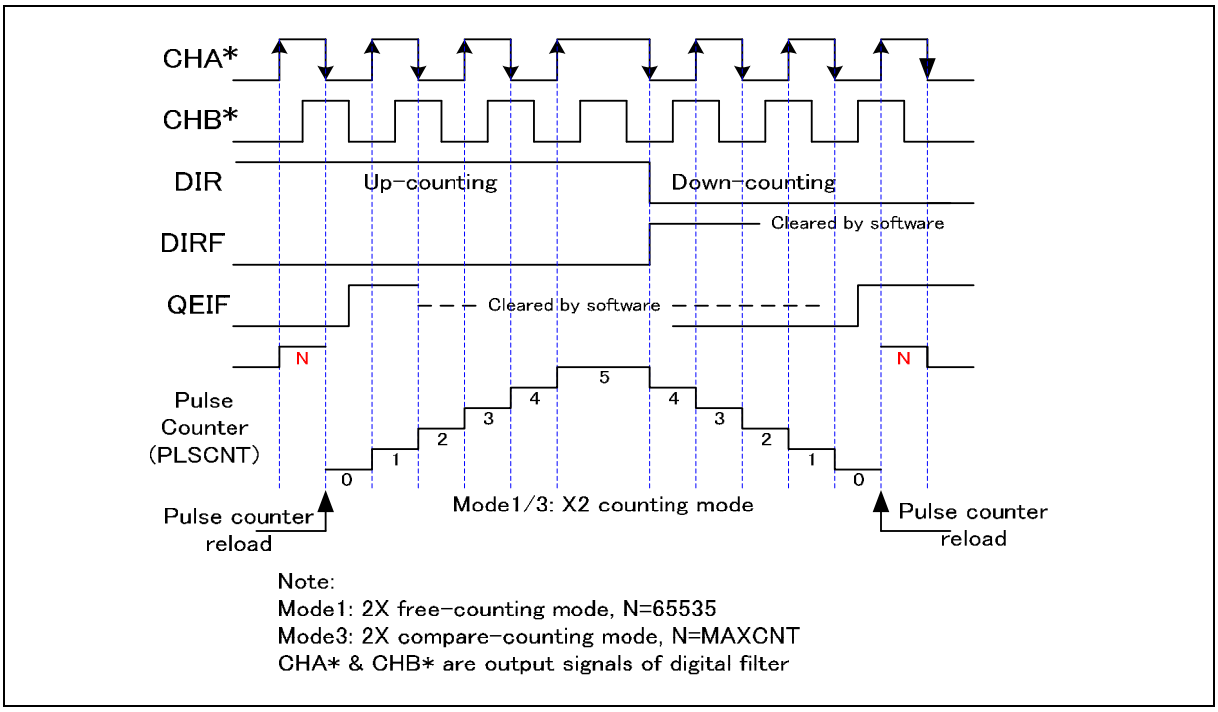



Figure 15-13: X2 Counting Mode

QEI x2 Counting mode is selected by setting the QEI Mode Select bits (QEIM1:QEIM0) to '01b' or '11b'. In this mode, the QEI logic detects every edge on the QEA input only. Every rising and falling edge on the QEA signal clocks the pulse counter.

### 15.2.5 Up-Counting

Under the forward direction the DIR bit is 1 when up-counting. Software needs to clear the QEIF flag. For the free-counting mode counter will counts until it matches 65535 and next edges on the forward direction will set the QEIF high and reset the PLSCNT to zero. For compare-counting mode counter counts until the MAXCNT value and reload the counter to zero and starts counting up. Changes of direction trigger a down-count and PLSCNT decreasing in counter value. For X2 mode, only CHA edge will set QEIF while for X4 mode both CHA and CHB edges will set QEIF.

### 15.2.6 Down-Counting

A change of direction will causes the counter to down-count for x2/x4 counting mode. It is indicated with the DIR bit as 0 and DIRF flag is set to 1. At this stage the PLSCNT will starts to down-count from the MAXCNT value for compare-counting mode and while in free-counting mode it will starts to down-count from 65535. The pulse counter will reload with MAXCNT when it down counts to zero in compare-counting mode and sets QEIF to high in the next edge. In free-counting mode the counter will count to 16 bits value before it reload the pulse counter with the value 65535 and set the QEIF high in the next edge. For X2 mode, only CHA edge will set QEIF while for X4 mode both CHA and CHB edges will set QEIF.

## 16. SERIAL PORT

The W79E22X SERIES has two enhanced serial ports that are functionally similar to the serial port of the original 8052 family. Both the serial ports are full-duplex ports, and the W79E22X SERIES provides additional features, such as Frame Error Detection and Automatic Address Recognition. The serial port 0 can use Timer 1 or 2 as baud rate generator, but the serial port 1 only uses Timer 1 as baud rate generator. However, note that if both serial ports are enabled the baud rate setting control of UART1 is also from the setting of UART0. The serial ports are capable of synchronous and asynchronous communication. In synchronous mode, the W79E22X SERIES generates the clock and operates in half-duplex mode. In asynchronous mode, the serial ports can simultaneously transmit and receive data. The transmit registers and the receive buffers are both addressed as SBUF (SBUF1 for the second serial port), but any write to SBUF/SBUF1 writes to the transmit register while any read from SBUF/SBUF1 reads from the receive buffer. Both serial ports can operate in four modes, as described below. The descriptions are for serial port 0, however, it also apply to the second serial port.

### 16.1 Mode 0

This mode provides half-duplex, synchronous communication with external devices. In this mode, serial data is transmitted and received on the RXD line, and the W79E22X SERIES provides the shift clock on TxD, whether the device is transmitting or receiving. Eight bits are transmitted or received per frame, LSB first. The baud rate is 1/12 or 1/4 of the oscillator frequency, as determined by the SM2 bit (SCON.5; 0 = 1/12; 1 = 1/4). This programmable baud rate is the only difference between the standard 8051/52 and the W79E22X SERIES in mode 0.

Any write to SBUF starts transmission. The shift clock is activated, and data is shifted out on RxD until all eight bits are transmitted. If SM2 is 1, the data appears on RxD one clock period before the falling edge of the shift clock on TxD. Then, the clock remains low for two clock periods before going high again. If SM2 is 0, the data appears on RxD three clock periods before the falling edge of the shift clock on TxD, and the clock on TxD remains low for six clock periods before going high again. This ensures that, at the receiving end, the data on the RxD line can be clocked on the rising edge of the shift clock or latched when the clock is low. The TI flag is set high in C1 following the end of transmission. The functional block diagram is shown below.

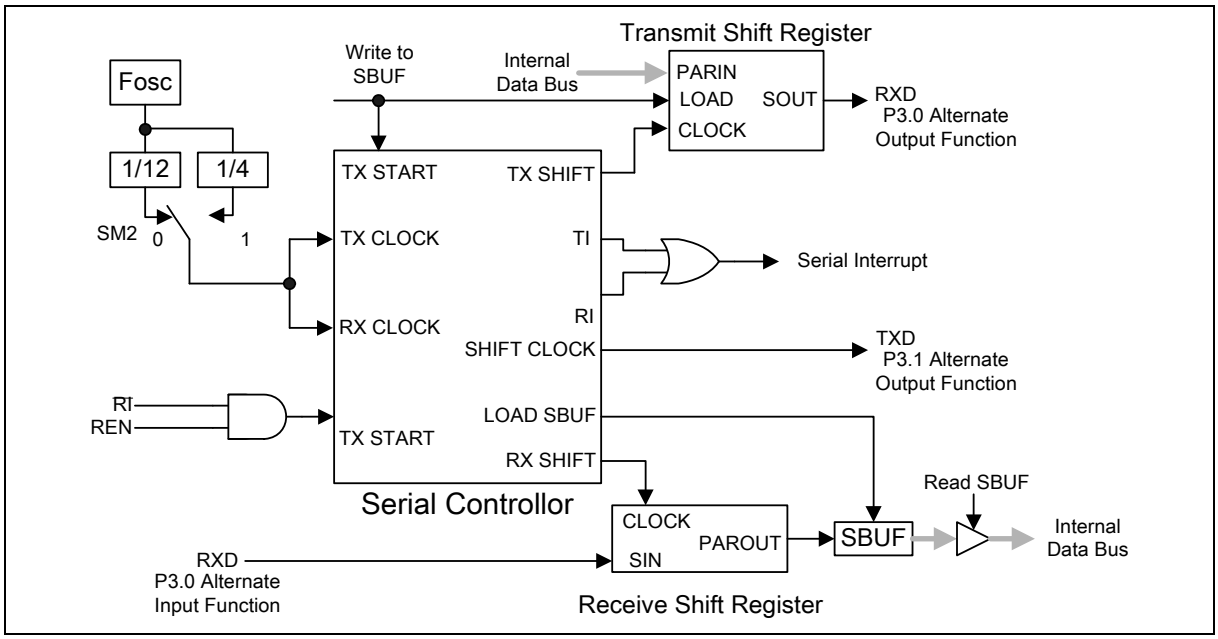


Figure 16-1 Serial Port Mode 0

The serial port receives data when REN is 1 and RI is zero. The shift clock (TxD) is activated, and the serial port latches data on the rising edge of the shift clock. The external device should, therefore, present data on the falling edge of the shift clock. This process continues until all eight bits have been received. The RI flag is set in C1 following the last rising edge of the shift clock, which stops reception until RI is cleared by the software.

## 16.2 Mode 1

In Mode 1, full-duplex asynchronous communication is used. Frames consist of ten bits transmitted on TXD and received on RXD. The ten bits consist of a start bit (0), eight data bits (LSB first), and a stop bit (1). When receiving, the stop bit goes into RB8 in SCON. The baud rate in this mode is 1/16 or 1/32 of the Timer 1 overflow, and since Timer 1 can be set to a wide range of values, a wide variation of baud rates is possible.

Transmission begins with a write to SBUF but is synchronized with the divide-by-16 counter, not the write to SBUF. The start bit is put on TxD at C1 following the first roll-over of the divide-by-16 counter, and the next bit is placed at C1 following the next rollover. After all eight bits are transmitted, the stop bit is transmitted. The TI flag is set in the next C1 state, or the tenth rollover of the divide-by-16 counter after the write to SBUF.

Reception is enabled when REN is high, and the serial port starts receiving data when it detects a falling edge on RxD. The falling-edge detector monitors the RxD line at 16 times the selected baud rate. When a falling edge is detected, the divide-by-16 counter is reset to align the bit boundaries with the rollovers of the counter. The 16 states of the counter divide the bit time into 16 slices. Bit detection is done on a best-of-three basis using samples at the 8th, 9th and 10th counter states. If the first bit after the falling edge is not 0, the start bit is invalid, reception is aborted immediately, and the serial port resumes looking for a falling edge on RxD. If a valid start bit is detected, the rest of the bits are shifted into SBUF. After shifting in eight data bits, the stop bit is received. Then, if;

1. RI is 0, and
2. SM2 is 0 or the received stop bit is 1,

the stop bit goes into RB8, the eight data bits go into SBUF, and RI is set. Otherwise, the received frame is lost. In the middle of the stop bit, the receiver resumes looking for a falling edge on RxD.

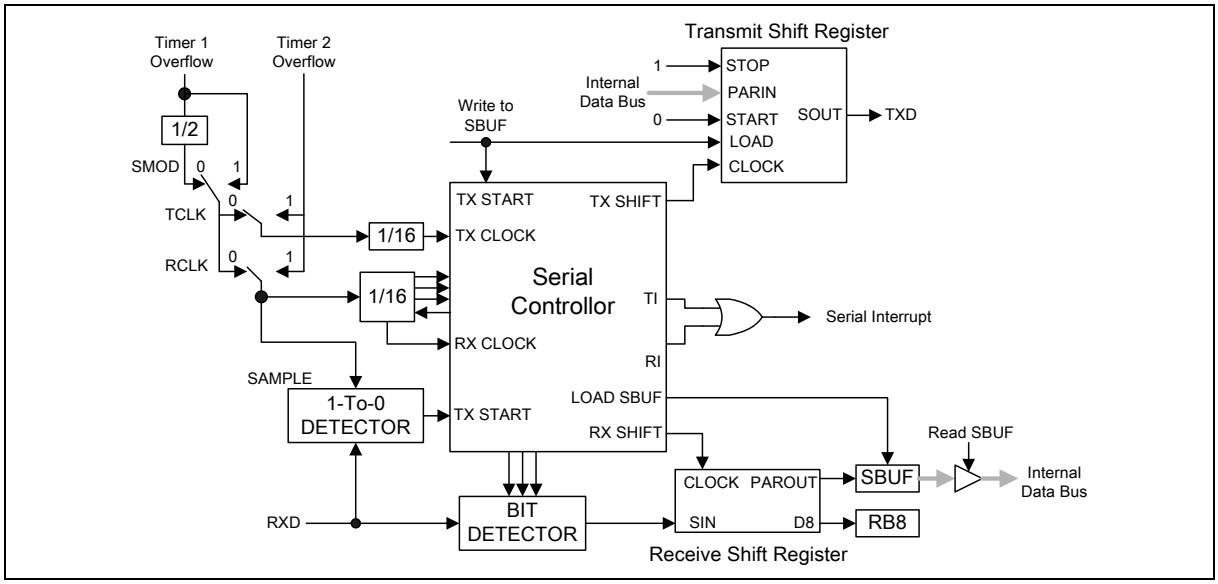



Figure 16-2 Serial Port Mode 1

## 16.3 Mode 2

In Mode 2, full-duplex asynchronous communication is used. Frames consist of eleven bits: one start bit (0), eight data bits (LSB first), a programmable ninth bit (TB8) and a stop bit (0). When receiving, the ninth bit is put into RB8. The baud rate is 1/16 or 1/32 of the oscillator frequency, as determined by SMOD in PCON.

Transmission begins with a write to SBUF but is synchronized with the divide-by-16 counter, not the write to SBUF. The start bit is put on TxD pin at C1 following the first roll-over of the divide-by-16 counter, and the next bit is placed on TxD at C1 following the next rollover. After all nine bits of data are transmitted, the stop bit is transmitted. The TI flag is set in the next C1 state, or the 11th rollover of the divide-by-16 counter after the write to SBUF.

Reception is enabled when REN is high, and the serial port starts receiving data when it detects a falling edge on RxD. The falling-edge detector monitors the RxD line at 16 times the selected baud rate. When a falling edge is detected, the divide-by-16 counter is reset to align the bit boundaries with the rollovers of the counter. The 16 states of the counter divide the bit time into 16 slices. Bit detection is done on a best-of-three basis using samples at the 8th, 9th and 10th counter states. If the first bit after the falling edge is not 0, the start bit is invalid, reception is aborted, and the serial port resumes looking for a falling edge on RxD. If a valid start bit is detected, the rest of the bits are shifted into SBUF. After shifting in nine data bits, the stop bit is received. Then, if;

1. RI is 0, and
2. SM2 is 0 or the received stop bit is 1,

the stop bit goes into RB8, the eight data bits go into SBUF, and RI is set. Otherwise, the received frame may be lost. In the middle of the stop bit, the receiver resumes looking for a falling edge on RxD. The functional description is shown in the figure below.

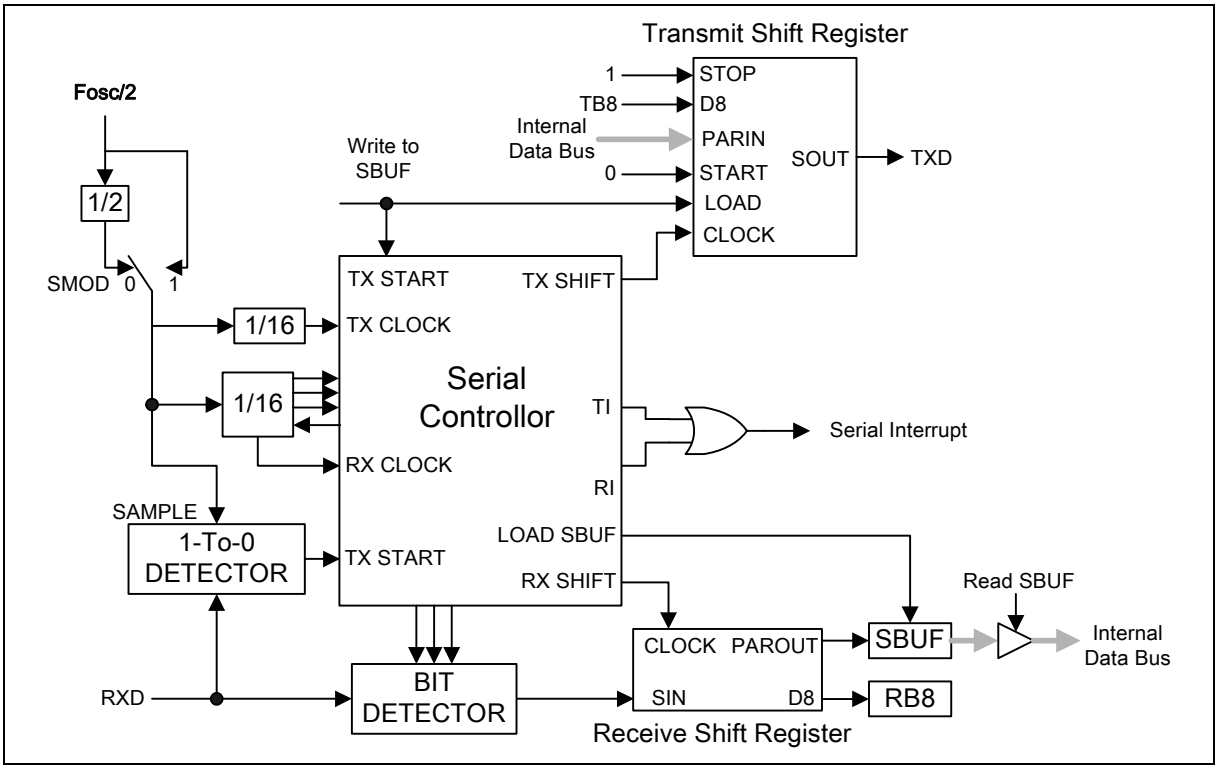



Figure 16-3 Serial Port Mode 2

## 16.4 Mode 3

This mode is the same as Mode 2, except that the baud rate is programmable. The program must select the mode and baud rate in SCON before any communication can take place. Timer 1 should be initialized if Mode 1 or Mode 3 will be used.

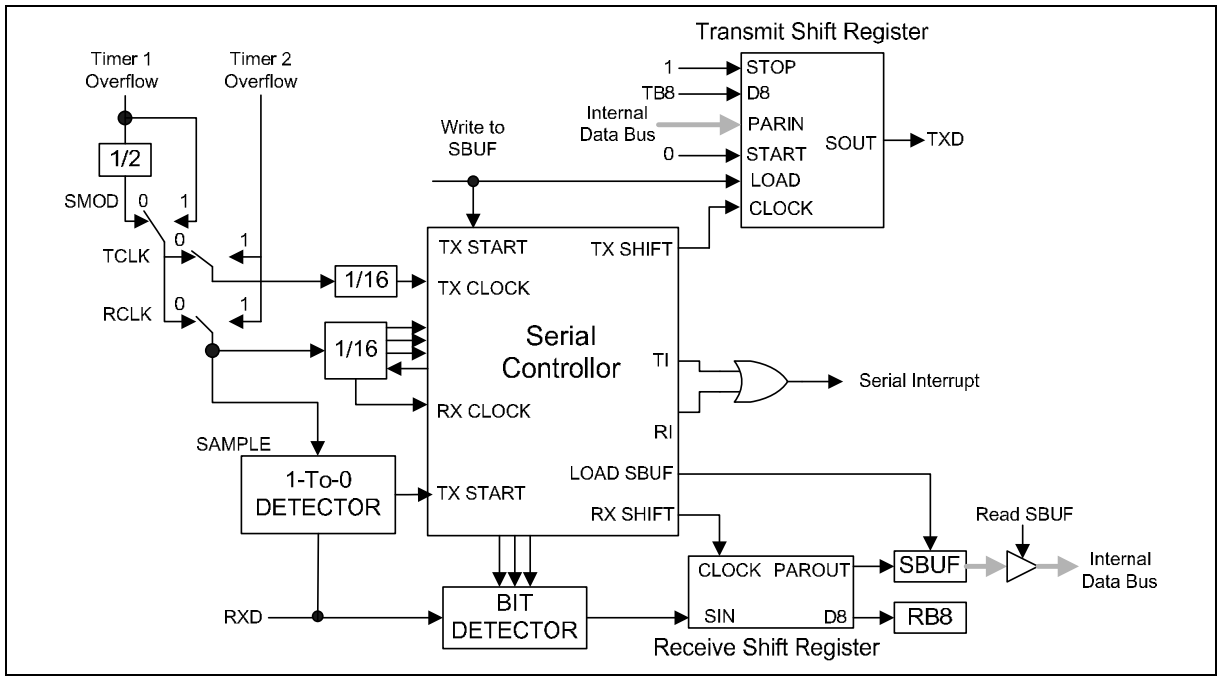



Figure 16-4 Serial Port Mode 3

| SM0 | SM1 | MODE | TYPE | BAUD CLOCK | FRAME SIZE | START BIT | STOP BIT | 9TH BIT FUNCTION |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | Synch. | 4 or 12 OSC | 8 bits | No | No | None |
| 0 | 1 | 1 | Asynch. | Timer 1 or 2 | 10 bits | 1 | 1 | None |
| 1 | 0 | 2 | Asynch. | 32 or 64 OSC | 11 bits | 1 | 1 | 0, 1 |
| 1 | 1 | 3 | Asynch. | Timer 1 or 2 | 11 bits | 1 | 1 | 0, 1 |

Table 16-1: Serial Ports Modes

## 16.5 Framing Error Detection

A frame error occurs when a valid stop bit is not detected. This could indicate incorrect serial data communication. Typically, a frame error is due to noise or contention on the serial communication line. The W79E22X SERIES has the ability to detect framing errors and set a flag that can be checked by software.

The frame error FE (FE_1) bit is located in SCON.7. This bit is SM0 in the standard 8051/52 family, but, in the W79E22X SERIES, it serves a dual function and is called SM0/FE. There are actually two separate flags, SM0 and FE. The flag that is actually accessed as SCON.7 is determined by SMOD0 (PCON.6). When SMOD0 is set to 1, the FE flag is accessed. When SMOD0 is set to 0, the SM0 flag is accessed.

The FE bit is set to 1 by the hardware, but it must be cleared by the software. Once FE is set, any frames received afterwards, even those without errors, do not clear the FE flag. The flag has to be cleared by the software. Note that SMOD0 must be set to 1 while reading or writing FE.

## 16.6 Multiprocessor Communications

Multiprocessor communication is available in modes 1, 2 and 3 and makes use of the 9th data bit and the automatic address recognition feature. This approach eliminates the software overhead required to check every received address and greatly simplifies the program.

In modes 2 and 3, address bytes are distinguished from data bytes by 9th bit set, which is set high in address bytes. When the master processor wants to transmit a block of data to one of the slaves, it first sends the address of the target slave(s). The slave processors have already set their SM2 bits high so that they are only interrupted by an address byte. The automatic address recognition feature then ensures that only the addressed slave is actually interrupted. This feature compares the received byte to the slave's Given or Broadcast address and only sets the RI flag if the bytes match. This slave then clears the SM2 bit, clearing the way to receive the data bytes. The unaddressed slaves are not affected, as they are still waiting for their address.

In mode 1, the 9th bit is the stop bit, which is 1 in valid frames. Therefore, if SM2 is 1, RI is only set if a valid frame is received and if the received byte matches the Given or Broadcast address.

The master processor can selectively communicate with groups of slaves using the Given Address or all the slaves can be addressed together using the Broadcast Address. The addresses for each slave are defined by the SADDR and SADEN registers. The slave address is the 8-bit value specified in SADDR. SADEN is a mask for the value in SADDR. If a bit position in SADEN is 0, then the corresponding bit position in SADDR is a don't-care condition in the address comparison. Only those bit positions in SADDR whose corresponding bits in SADEN are 1 are used to obtain the Given Address. This provides flexibility to address multiple slaves without changing addresses in SADDR.

The following example shows how to setup the Given Addresses to address different slaves.

Slave 1:

SADDR 1010 0100
SADEN 1111 1010
Given 1010 0x0x

Slave 2:

SADDR 1010 0111
SADEN 1111 1001
Given 1010 0xx1

The Given Address for slaves 1 and 2 differ in the LSB. In slave 1, it is a don't-care, while, in slave 2, it is 1. Thus, to communicate with only slave 1, the master must send an address with LSB = 0 (1010 0000). Similarly, bit 1 is 0 for slave 1 and don't-care for slave 2. Hence, to communicate only with slave 2, the master has to transmit an address with bit 1 = 1 (1010 0011). If the master wishes to communicate with both slaves simultaneously, then the address must have bit 0 = 1 and bit 1 = 0. Since bit 3 is don't-care for both slaves, two different addresses can address both slaves (1010 0001 and 1010 0101).

The master can communicate with all the slaves simultaneously using the Broadcast Address. The Broadcast Address is formed from the logical OR of the SADDR and SADEN registers. The zeros in the result are don't–care values. In most cases, the Broadcast Address is FFh. In the previous case, the Broadcast Address is (1111111X) for slave 1 and (11111111) for slave 2.

The SADDR and SADEN registers are located at addresses A9h and B9h, respectively. These two registers default to 00h, so the Given Address and Broadcast Address default to XXXX XXXX (i.e., all bits don't-care), which effectively removes the multiprocessor communications feature

## 17. I2C SERIAL PORTS

The I2C bus uses two wires (SCL and SDA) to transfer information between devices connected to the bus. The main features of the I2C bus are:

– Bi-directional data transfer between masters and slaves.

– Multi-master bus (no central master).

– Arbitration between simultaneously transmitting masters without corruption of serial data on the bus.

– Serial clock synchronization allows devices with different bit rates to communicate via one serial bus.

– Serial clock synchronization can be used as a handshake mechanism to suspend and resume serial transfer.

– The I2C bus may be used for test and diagnostic purposes.

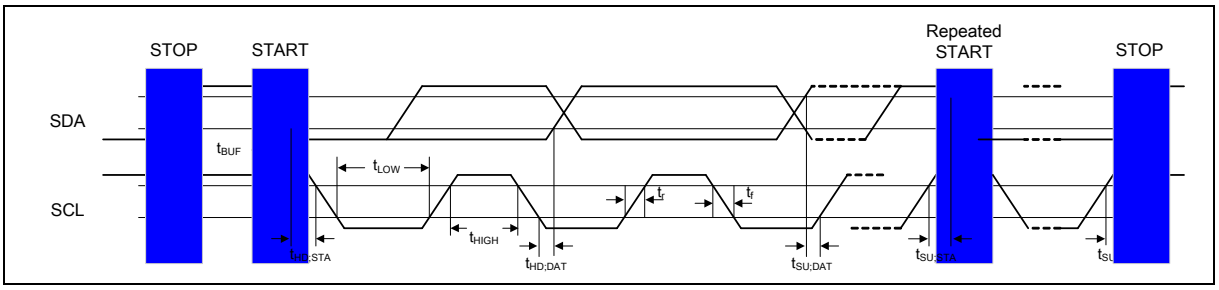



Figure 17-1: I2C Bus Timing

The device's on-chip I2C logic provides the serial interface that meets the I2C bus standard mode specification. The I2C logic handles bytes transfer autonomously. It also keeps track of serial transfers, and a status register (I2STATUS) reflects the status of the I2C bus.

The I2C port, SCL and SDA are at P2.6 and P2.7. When the I/O pins are used as I2C port, user must set the pins to logic high in advance. When I2C port is enabled by setting ENS to high, the internal states will be controlled by I2CON and I2C logic hardware. Once a new status code is generated and stored in I2STATUS, the I2C interrupt flag (SI) will be set automatically. If both EA and EI2C are also in logic high, the I2C interrupt is requested. The 5 most significant bits of I2STATUS stores the internal state code, the lowest 3 bits are always zero and the content keeps stable until SI is cleared by software.

### 17.1 SIO Port

The SIO port is a serial I/O port, which supports all transfer modes from and to the I2C bus. The SIO port handles byte transfers autonomously. To enable this port, the bit ENS1 in I2CON should be set to '1'. The CPU interfaces to the SIO port through the seven special function registers. The detail description of these registers can be found in the I2C Control registers section. The SIO H/W interfaces to the I2C bus via two pins: SDA (P2.7, serial data line) and SCL (P2.6, serial clock line). Pull up resistor is needed for Pin P2.6 and P2.7 for I2C operation as these are 2 open drain pins.

### 17.2 The I2C Control Registers

The I2C has 1 control register (I2CON) to control the transmit/receive flow, 1 data register (I2DAT) to buffer the Tx/Rx data, 1 status register (I2STATUS) to catch the state of Tx/Rx, recognizable slave address register for slave mode use and 1 clock rate control block for master mode to generate the variable baud rate.

| SYMBOL | DEFINITION | ADDRESS | MSB | | BIT_ADDRESS, SYMBOL | | | | LSB | | RESET |
|---|---|---|---|---|---|---|---|---|---|---|---|
| I2TIMER | I2C Timer Counter Register | EFH | - | - | - | - | - | ENTI | DIV4 | TIF | xxxx x000B |
| I2CLK | I2C Clock Rate | EEH | I2CLK.7 | I2CLK.6 | I2CLK.5 | I2CLK.4 | I2CLK.3 | I2CLK.2 | I2CLK.1 | I2CLK.0 | 0000 0000B |
| I2STATUS | I2C Status Register | EDH | I2STATUS.7 | I2STATUS.6 | I2STATUS.5 | I2STATUS.4 | I2STATUS.3 | - | - | - | 1111 1000B |
| I2DAT | I2C Data | ECH | I2DAT.7 | I2DAT.6 | I2DAT.5 | I2DAT.4 | I2DAT.3 | I2DAT.2 | I2DAT.1 | I2DAT.0 | 0000 0000B |
| I2ADDR | I2C Slave Address | EAH | ADDR.7 | ADDR.6 | ADDR.5 | ADDR.4 | ADDR.3 | ADDR.2 | ADDR.1 | GC | 0000 0000B |
| I2CON | I2C Control Register | E9H | - | ENS | STA | STO | SI | AA | I2CIN | - | x000 000xB |
| I2CSADEN | I2C Maskable Slave Address | F6H | I2CSADEN.7 | I2CSADEN.6 | I2CSADEN.5 | I2CSADEN.4 | I2CSADEN.3 | I2CSADEN.2 | I2CSADEN.1 | I2CSADEN.0 | 1111 1110B |

Table 17-1: Control Registers of I2C Ports

### 17.2.1 Slave Address Registers, I2ADDR

I2C port is equipped with one slave address register. The contents of the register are irrelevant when I2C is in master mode. In the slave mode, the seven most significant bits must be loaded with the MCU's own slave address. The I2C hardware will react if the contents of I2ADDR are matched with the received slave address.

The I2C ports support the "General Call" function. If the GC bit is set the I2C port1 hardware will respond to General Call address (00H). Clear GC bit to disable general call function.

When GC bit is set, the device is in slave mode which can receive the General Call address(00H) sent by Master on the I2C bus. This special slave mode is referred to as GC mode.

### 17.2.2 Data Register, I2DAT

This register contains a byte of serial data to be transmitted or a byte which has just been received. The CPU can read from or write to this 8-bit directly addressable SFR while it is not in the process of shifting a byte. Data in I2DAT remains stable as long as SI is set. The MSB is shifted out first.While data is being shifted out, data on the bus is simultaneously being shifted in; I2DAT always contains the last data byte present on the bus. Thus, in the event of arbitration lost, the transition from master transmitter to slave receiver is made with the correct data in I2DAT.

I2DAT and the acknowledge bit form a 9-bit shift register which shifts in or out an 8-bit byte, followed by an acknowledge bit. The acknowledge bit is controlled by the hardware and cannot be accessed by the CPU. Serial data is shifted into I2DAT on the rising edges of serial clock pulses on the SCL line. When a byte has been shifted into I2DAT, the serial data is available in I2DAT, and the acknowledge bit (ACK or NACK) is returned by the control logic during the ninth clock pulse. Serial data is shifted out from I2DAT on the falling edges of SCL clock pulses, and is shifted into I2DAT on the rising edges of SCL clock pulses.

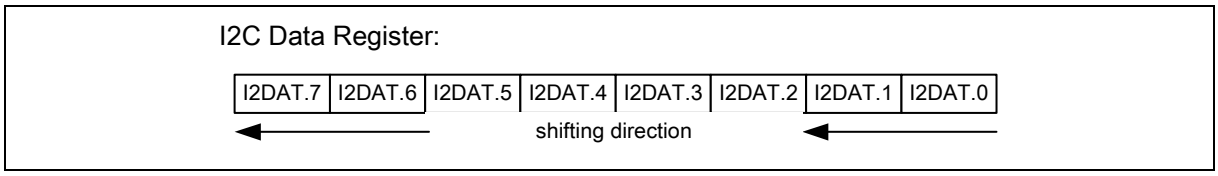



Figure 17-2: I2C Data Shift

### 17.2.3 Control Register, I2CON

The CPU can read from and write to this 8-bit, directly addressable SFR. Two bits are affected by hardware: the SI bit is set when the I2C hardware requests a serial interrupt, and the STO bit is cleared when a STOP condition is present on the bus. The STO bit is also cleared when ENS = "0".

ENS I2C serial function block enable bit. When ENS=1 the I2C serial function enables. The port latches of SDA1 and SCL1 must be set to logic high.

| | |
|---|---|
| STA | I2C START Flag. Setting STA to logic 1 to enter master mode, the I2C hardware sends a START or repeat START condition to bus when the bus is free. |
| STO | I2C STOP Flag. In master mode, setting STO to transmit a STOP condition to bus then I2C hardware will check the bus condition if a STOP condition is detected this flag will be cleared by hardware automatically. In a slave mode, setting STO resets I2C hardware to the "not addressed slave mode". |
| SI | I2C Port 1 Interrupt Flag. When a new SIO state is present in the S1STA register, the SI flag is set by hardware, and if the EA and EI2C1 bits are both set, the I2C1 interrupt is requested. SI must be cleared by software. |
| AA | Assert Acknowledge control bit. When AA=1 prior to address or data received, an acknowledged (low level to SDA) will be returned during the acknowledge clock pulse on the SCL line when; 1.) A slave is acknowledging the address sent from master, 2.) The receiver devices are acknowledging the data sent by transmitter. When AA=0 prior to address or data received, a not acknowledged (high level to SDA) will be returned during the acknowledge clock pulse on the SCL line. |
| I2CIN | By default it is zero and input are allows to come in through SDA pin. As when it is 1 input is disallow and to prevent leakage current. During Power-Down mode input is disallow. |

### 17.2.4 Status Register, I2STATUS

I2STATUS is an 8-bit read-only register. The three least significant bits are always 0. The five most significant bits contain the status code. There are 23 possible status codes. When I2STATUS contains F8H, no serial interrupt is requested. All other I2STATUS values correspond to defined I2C ports states. When each of these states is entered, a status interrupt is requested (SI = 1). A valid status code is present in I2STATUS one machine cycle after SI is set by hardware and is still present one machine cycle after SI has been reset by software.

In addition, state 00H stands for a Bus Error. A Bus Error occurs when a START or STOP condition is present at an illegal position in the format frame. Examples of illegal positions are during the serial transfer of an address byte, a data byte or an acknowledge bit.

### 17.2.5 I2C Clock Baud Rate Control, I2CLK

The data baud rate of I2C is determines by I2CLK register when I2C port is in a master mode. It is not important when I2C port is in a slave mode. In the slave modes, SIO will automatically synchronize with any clock frequency up to 400 KHz from master I2C device.

The data baud rate of I2C setting conforms to the following equation.

Data Baud Rate of I2C = $F_{CPU}$ / (I2CLK + 1), where $F_{CPU} = F_{OSC}/4$.

For example, if $F_{OSC}$=16MHz, the I2CLK=40(28H), the data baud rate of I2C = (16MHz/4)/(40+1) = 97.56K bits/sec.

### 17.2.6 I2C Time-out Counter, I2Timer

In W79E22X SERIES, the I2C logic block provides a 14-bit timer-out counter that helps user to deal with bus pending problem. When SI is cleared user can set ENTI=1 to start the time-out counter. If I2C bus is pended too long to get any valid signal from devices on bus, the time-out counter overflows cause TIF=1 to request an I2C interrupt. The I2C interrupt is requested in the condition of either SI=1 or TIF=1. Flags SI and TIF must be cleared by software.

### 17.2.7 I2C Maskable Slave Address

This register enables the Automatic Address Recognition feature of the I2C. When a bit in the I2CSADEN is set to 1, the same bit location in I2CSADDR1 will be compared with the incoming serial

port data. When I2CSADEN.n is 0, then the bit becomes a don't-care in the comparison. This register enables the Automatic Address Recognition feature of the I2C. When all the bits of I2CSADEN are 0, interrupt will occur for any incoming address.

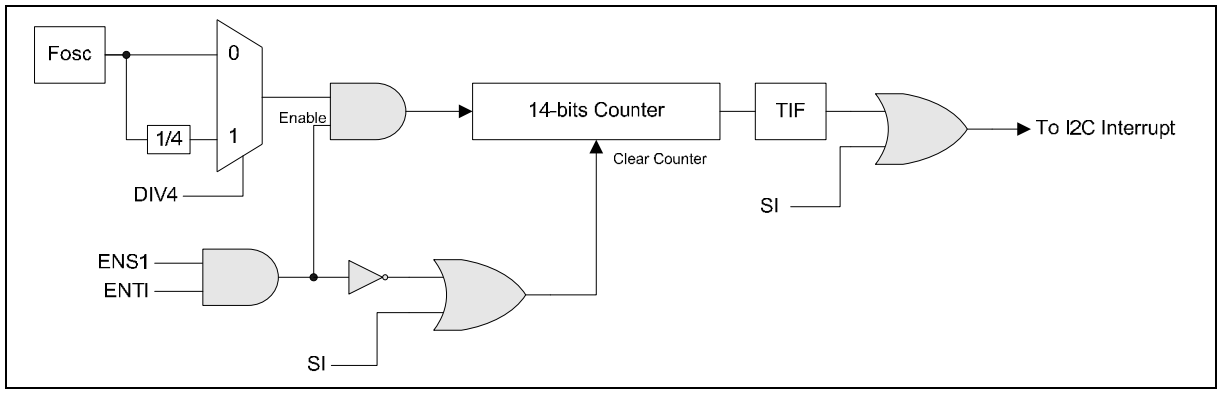



Figure 17-3: I2C Time-out Block Diagram

## 17.3 Modes of Operation

The on-chip I2C ports support five operation modes, Master transmitter, Master receiver, Slave transmitter, Slave receiver, and GC call.

In a given application, I2C port may operate as a master or as a slave. In the slave mode, the I2C port hardware looks for its own slave address and the general call address. If one of these addresses is detected, and if the slave is willing to receive or transmit data from/to master(by setting the AA bit), acknowledge pulse will be transmitted out on the 9$^{th}$ clock, hence an interrupt is requested on both master and slave devices if interrupt is enabled. When the microcontroller wishes to become the bus master, the hardware waits until the bus is free before the master mode is entered so that a possible slave action is not interrupted. If bus arbitration is lost in the master mode, I2C port switches to the slave mode immediately and can detect its own slave address in the same serial transfer.

### 17.3.1 Master Transmitter Mode

Serial data output through SDA while SCL outputs the serial clock. The first byte transmitted contains the slave address of the receiving device (7 bits) and the data direction bit. In this case the data direction bit (R/W) will be logic 0, and we say that a “W” is transmitted. Thus the first byte transmitted is SLA+W. Serial data is transmitted 8 bits at a time. After each byte is transmitted, an acknowledge bit is received. START and STOP conditions are output to indicate the beginning and the end of a serial transfer.

### 17.3.2 Master Receiver Mode

In this case the data direction bit (R/W) will be logic 1, and we say that an “R” is transmitted. Thus the first byte transmitted is SLA+R. Serial data is received via SDA while SCL outputs the serial clock. Serial data is received 8 bits at a time. After each byte is received, an acknowledge bit is transmitted. START and STOP conditions are output to indicate the beginning and end of a serial transfer.

### 17.3.3 Slave Receiver Mode

Serial data and the serial clock are received through SDA and SCL. After each byte is received, an acknowledge bit is transmitted. START and STOP conditions are recognized as the beginning and

end of a serial transfer. Address recognition is performed by hardware after reception of the slave address and direction bit.

### 17.3.4 Slave Transmitter Mode

The first byte is received and handled as in the slave receiver mode. However, in this mode, the direction bit will indicate that the transfer direction is reversed. Serial data is transmitted via SDA while the serial clock is input through SCL. START and STOP conditions are recognized as the beginning and end of a serial transfer.

## 17.4 Data Transfer Flow in Five Operating Modes

The five operating modes are: Master/Transmitter, Master/Receiver, Slave/Transmitter, Slave/Receiver and GC Call. Bits STA, STO and AA in I2CON register will determine the next state of the SIO hardware after SI flag is cleared. Upon complexion of the new action, a new status code will be updated and the SI flag will be set. If the I2C interrupt control bits (EA and EI2) are enabled, appropriate action or software branch of the new status code can be performed in the Interrupt service routine.

Data transfers in each mode are shown in the following figures.

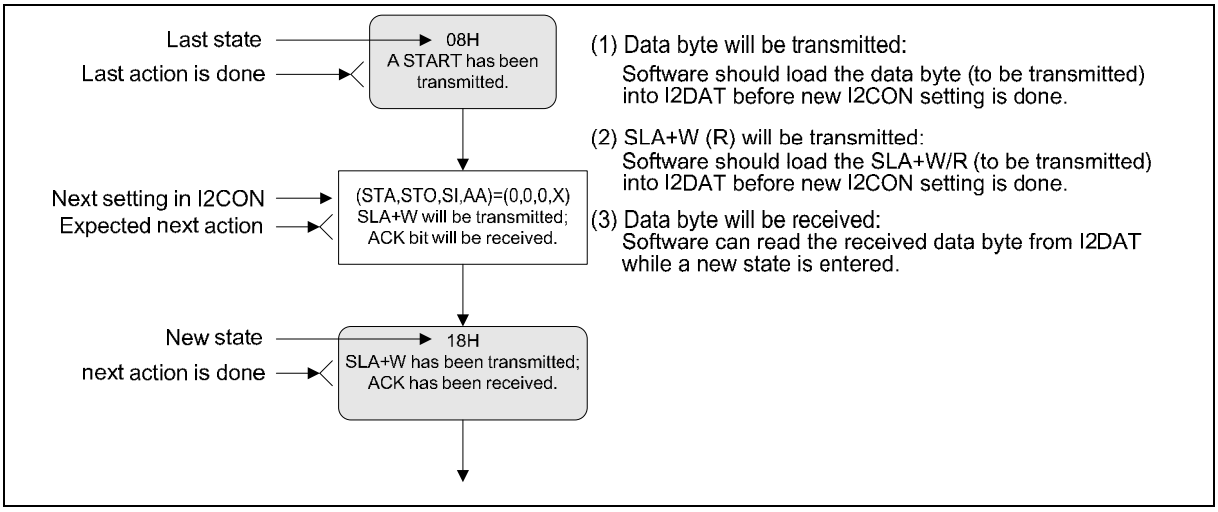
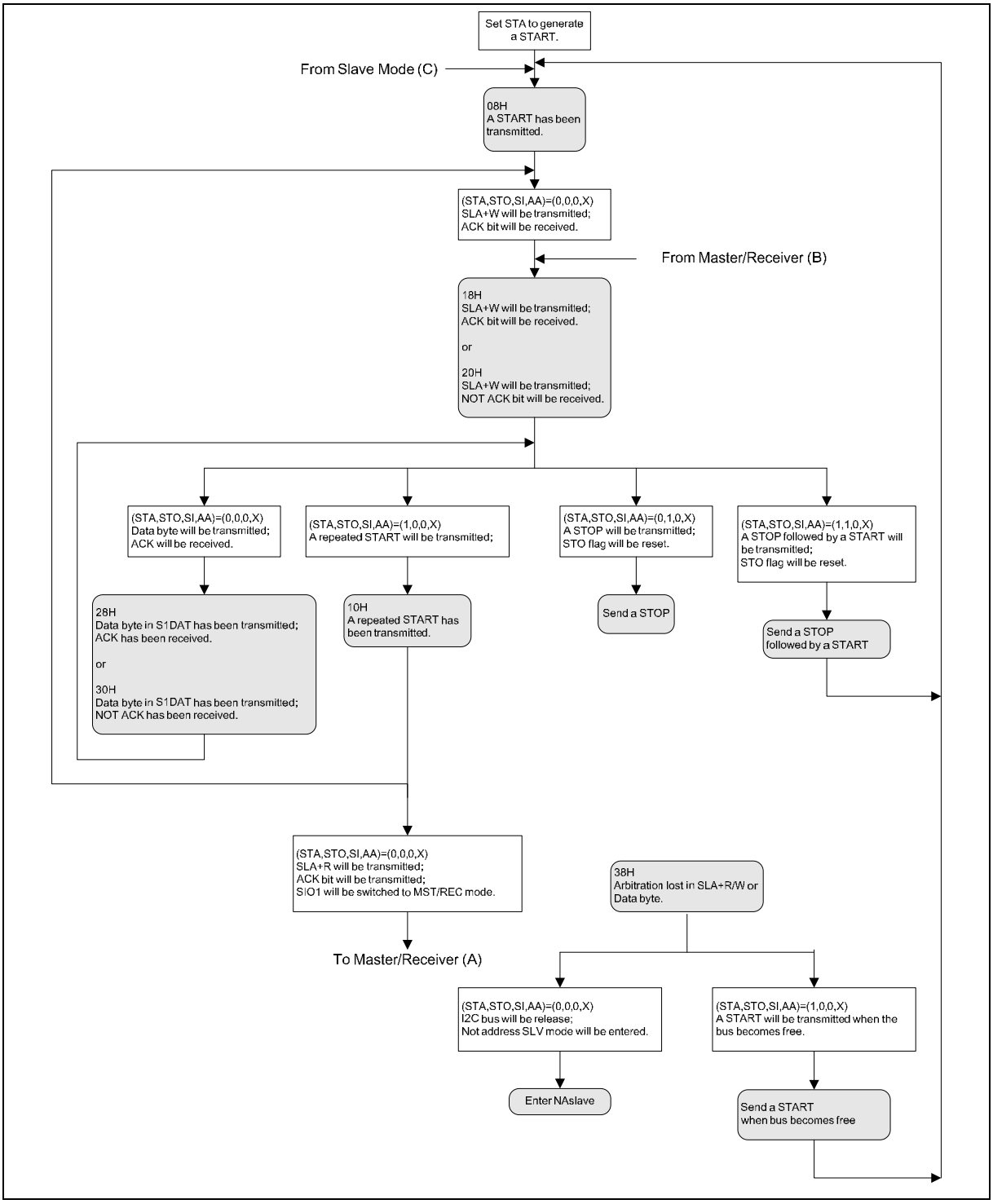



Figure 17-4: Legend for I2C flow charts

### 17.4.1 Master/Transmitter Mode



### 17.4.2 Figure 17-5: Master Transmitter ModeMaster/Receiver Mode

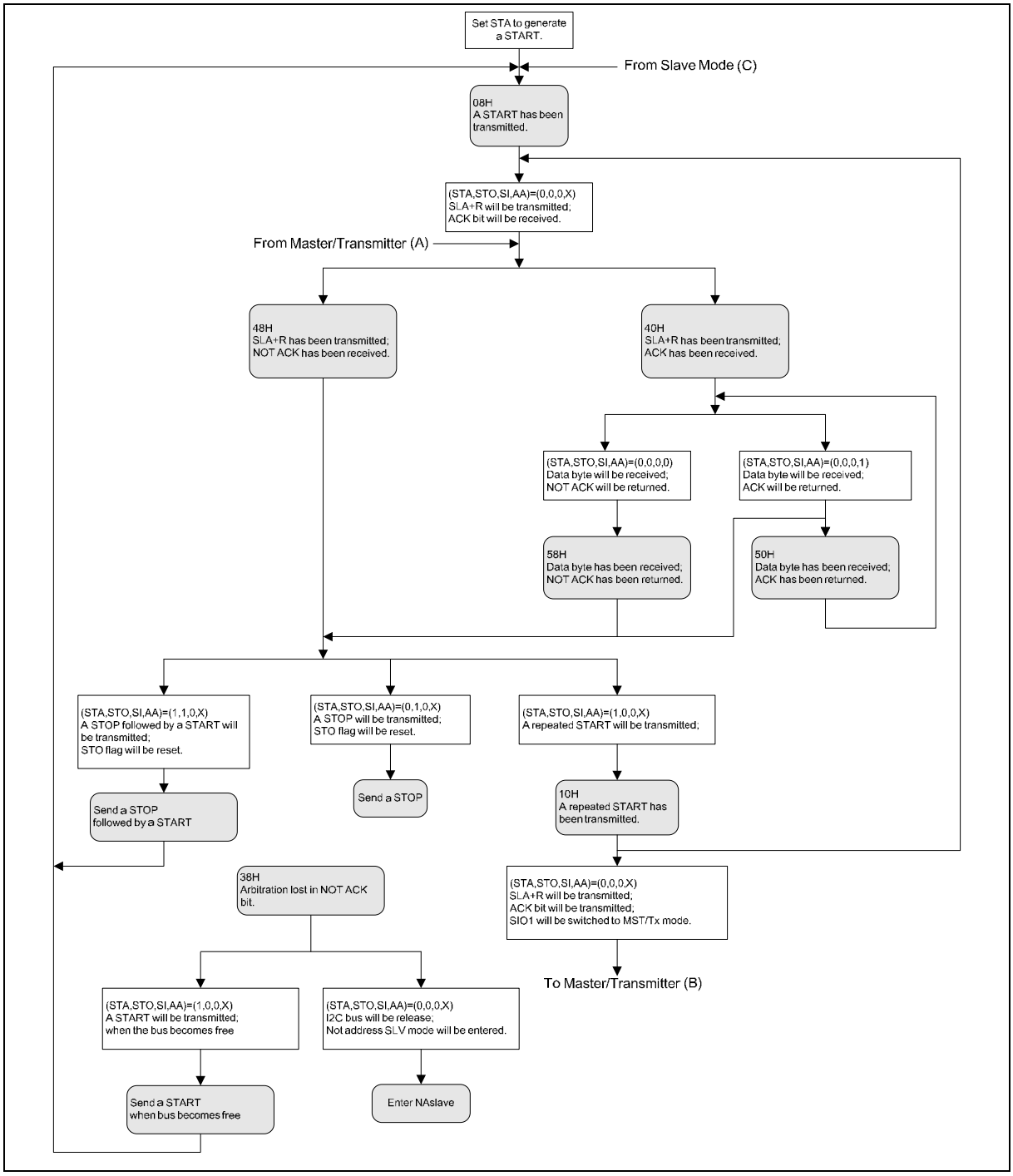



Figure 17-6: Master Receiver Mode

### 17.4.3 Slave/Transmitter Mode

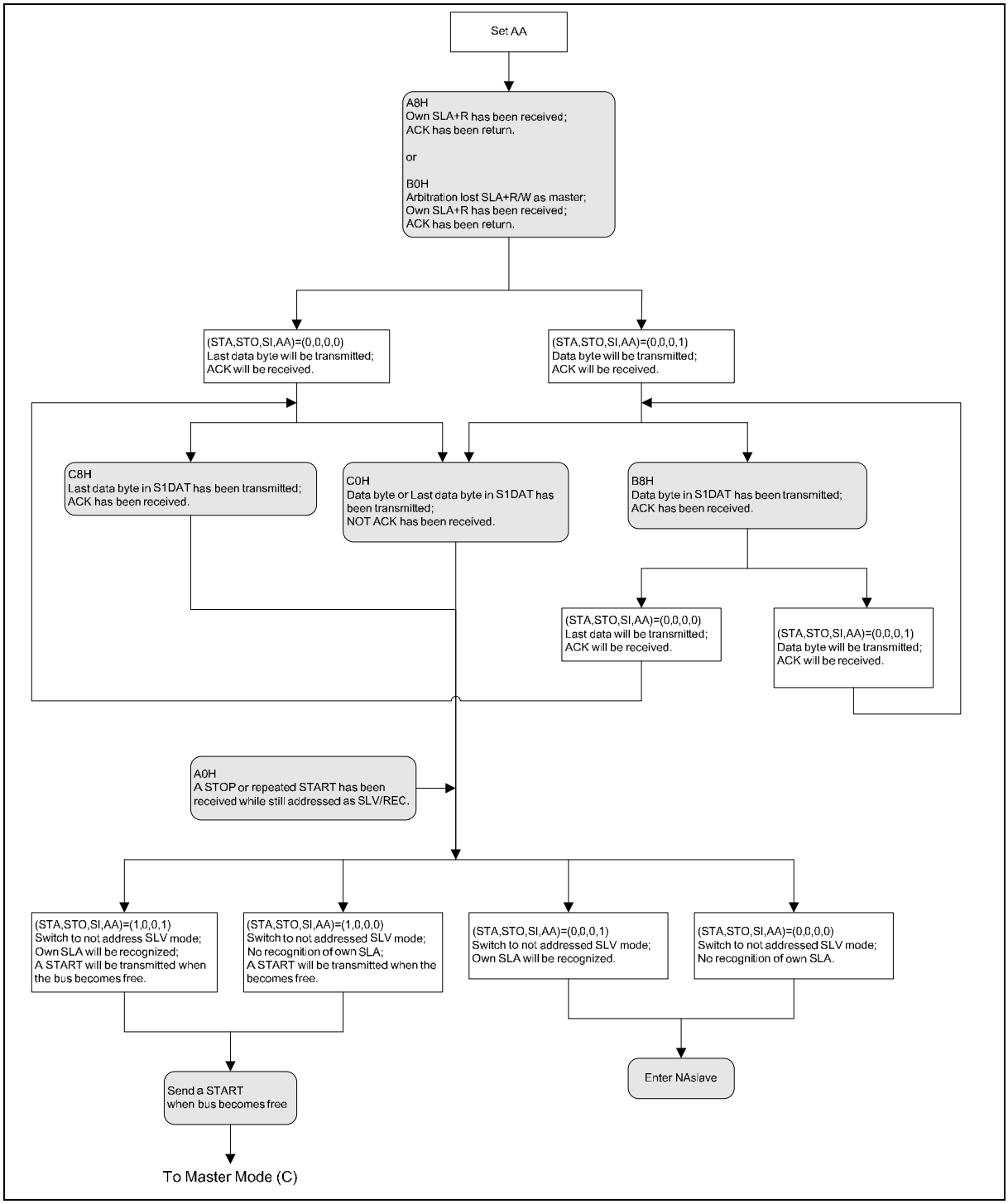



Figure 17-7: Slave Transmitter Mode

### 17.4.4 Slave/Receiver Mode

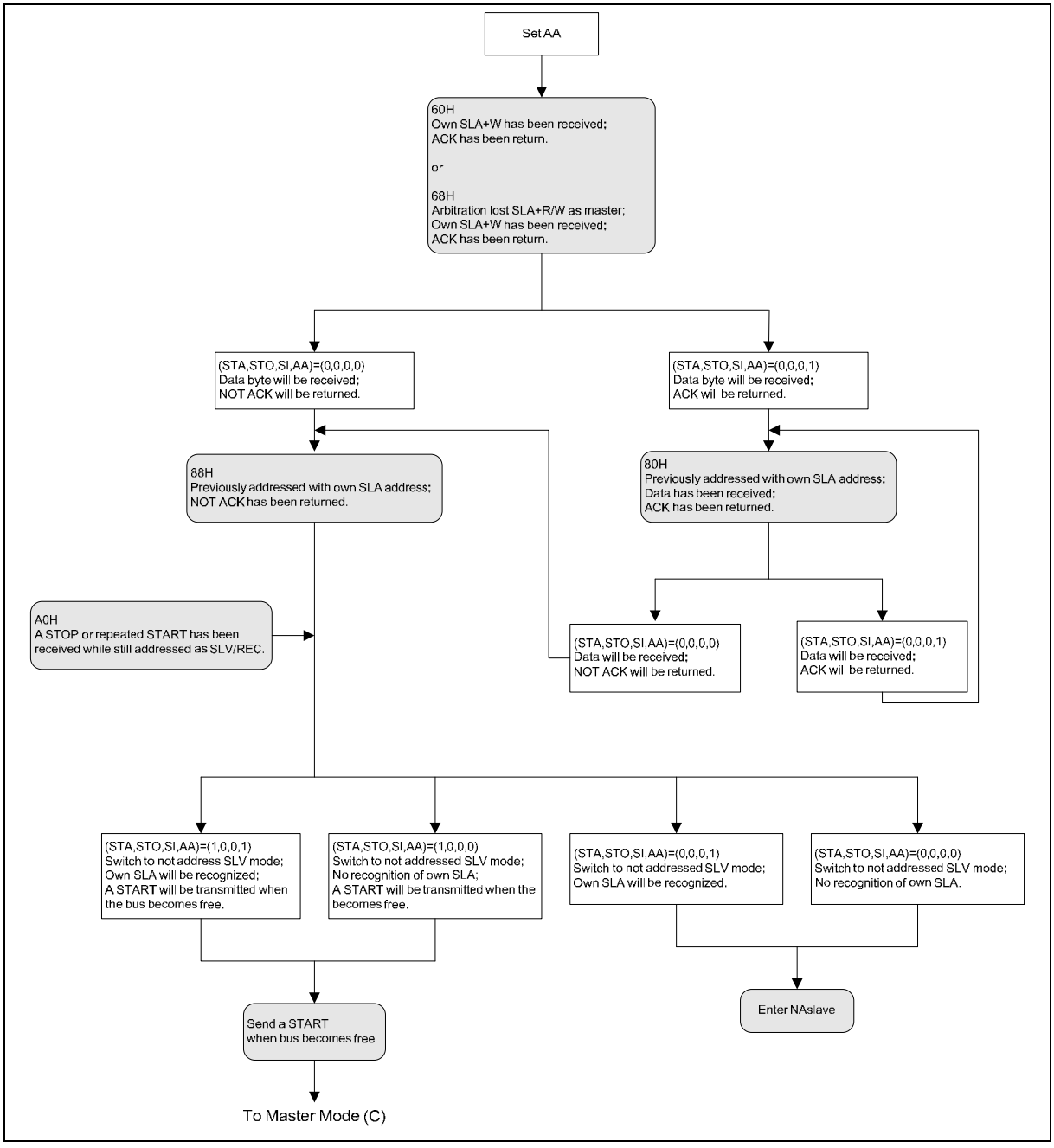



Figure 17-8: Slave Receiver Mode

### 17.4.5 GC Mode

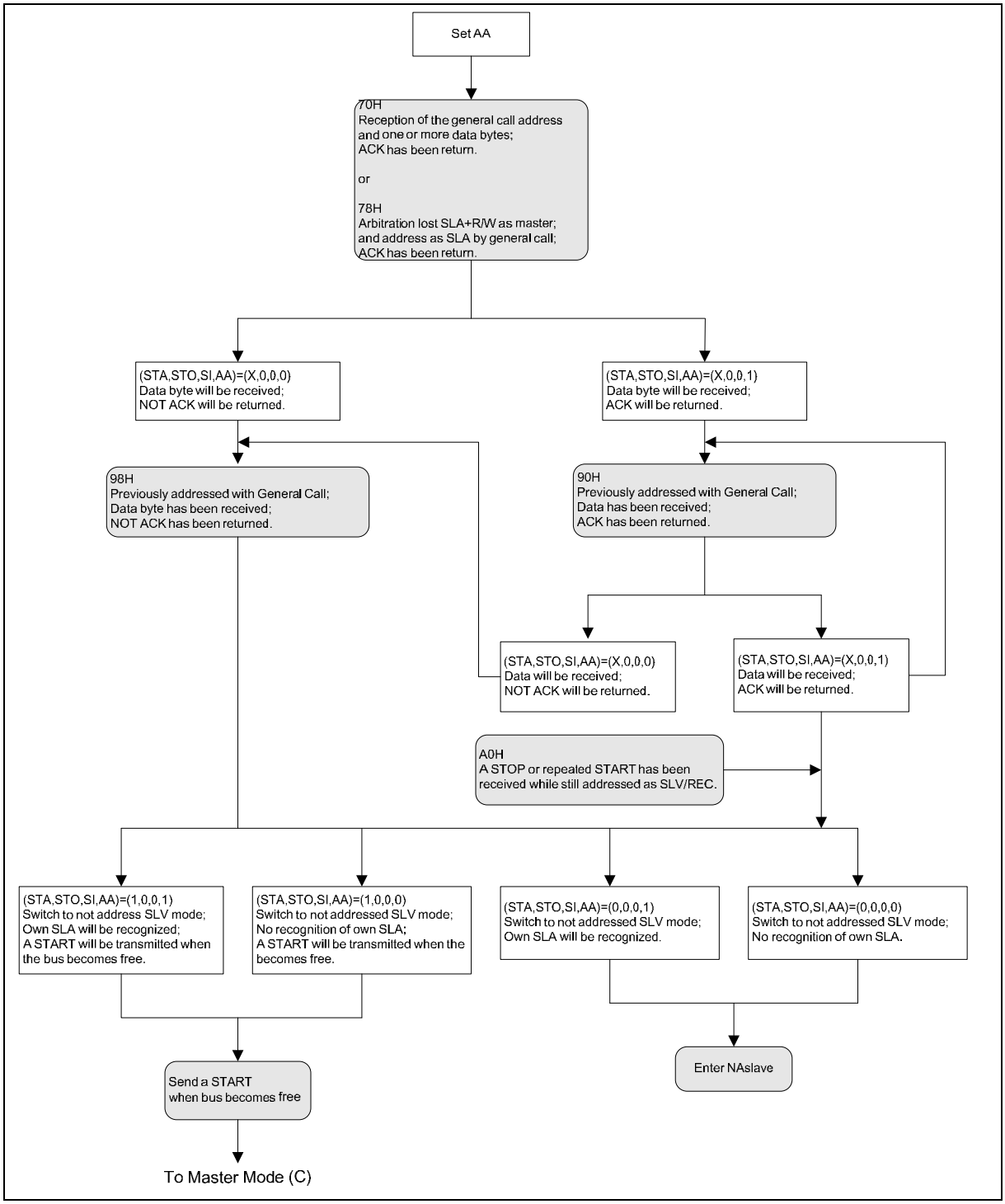



Figure 17-9: General Call Address

# 18. SERIAL PERIPHERAL INTERFACE (SPI)

## 18.1 General descriptions

This device consists of SPI block to support high speed serial communication. It's capable of supporting data transfer rates 5Mbit/s. This device's SPI support the following features;

- Master and slave mode.
- Slave select output.
- Programmable serial clock's polarity and phase.
- Receive double buffered data register.
- LSB first enable.
- Write collision detection.
- Transfer complete interrupt.

## 18.2 Block descriptions

The Figure 18-1 shows SPI block diagram. It provides an overview of SPI architecture in this device. The main blocks of SPI are the register blocks, control logics, baud rate control and pin control logics;

a. Shift register and read data buffer. It is single buffered in the transmit direction and double buffered in the receive direction. Transmit data cannot be written to the shifter until the previous transfer is complete. Receive logics consist of parallel read data buffer so the shifter is free to accept a second data, as the first received data will be transferred to the read data buffer.
b. SPI Control block. This provide control functions to configure the device for SPI enable, master or slave, clock phase and polarity, LSB access first selection, and Slave Select output enable.
c. Baud rate control. These control logics divide CPU clock to 4 different selectable clocks 1/8, 1/32, 1/128 and 1/2/256. Its' selection is controllable through SPR [1:0] bits.

| SPR1 | SPR0 | DIVIDER | BAUD RATE |
|---|---|---|---|
| 0 | 0 | 8 | 5MHz |
| 0 | 1 | 32 | 1.25MHz |
| 1 | 0 | 128 | 312.50kHz |
| 1 | 1 | 256 | 156.25kHz |

Table 18-1 SPI Baud Rate Selection ($F_{OSC}$ @ 40MHz)

d. SPI registers. There are three SPI registers to support its operations, they are;
   - SPI control registers (SPCR)
   - SPI status registers (SPSR)
   - SPI data register (SPDR)

   These registers provide control, status, data storage functions and baud rate selection control. Detail bits descriptions are found at SFR section. When using SPI pull-up must be apply at bit PUP0 = 1.
e. Pin control logic. Controls behavior of SPI interface pins.

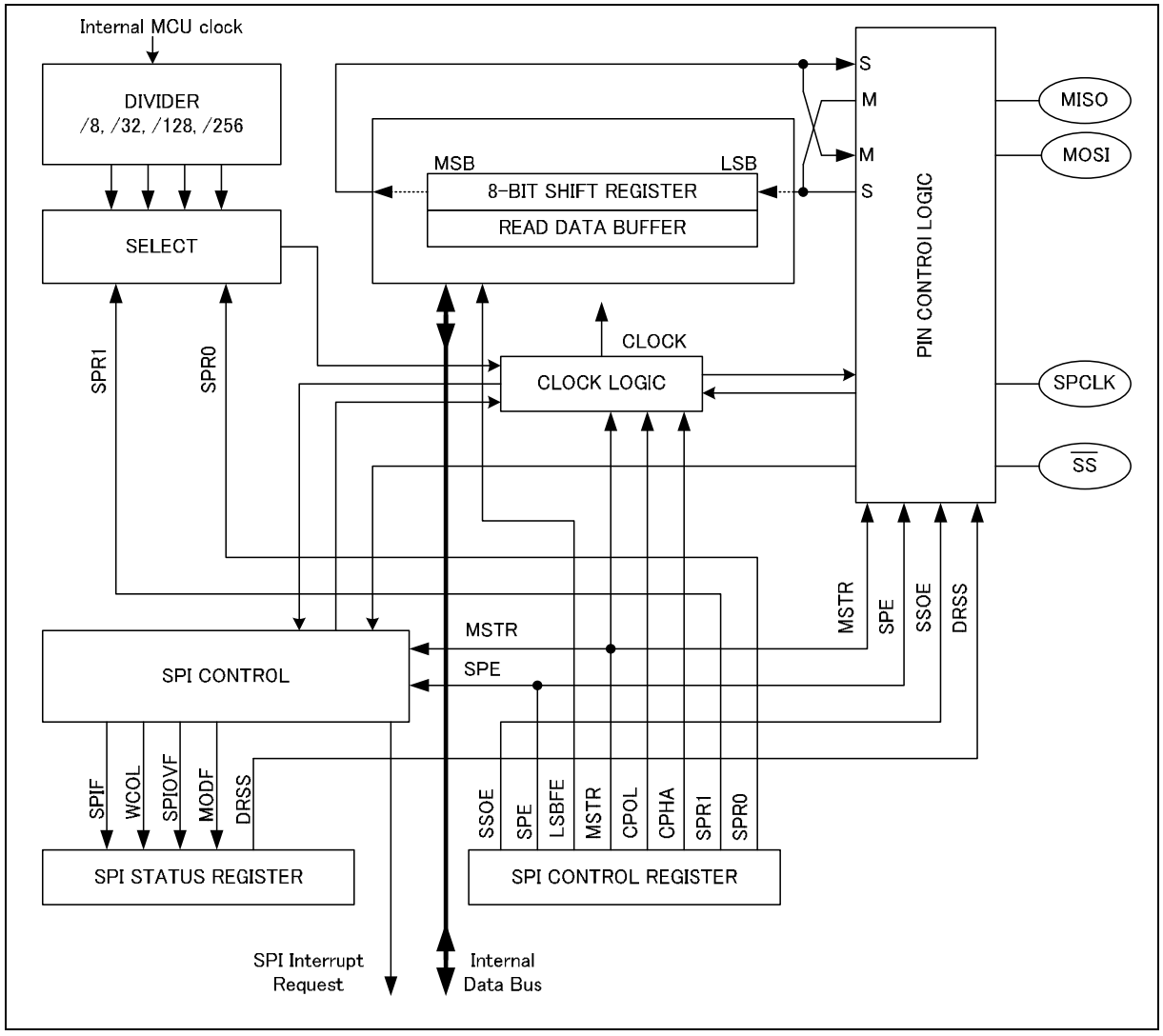



Figure 18-1: SPI block diagram

## 18.3 Functional descriptions

### 18.3.1 Master mode

The device can configure the SPI to operate as a master or as a slave, through MSTR bit. When the MSTR bit is set, master mode is selected, when MSTR bit is cleared, slave mode is selected. During master mode, only master SPI device can initiate transmission. A transmission begins by writing to the master SPDR register. The bytes begin shifting out on MOSI pin under the control of SPCLK. The master places data on MOSI line a half-cycle before SPCLK edge that the slave device uses to latch the data bit. The $\overline{SS}$ must stay low before data transactions and stay low for the duration of the transactions.

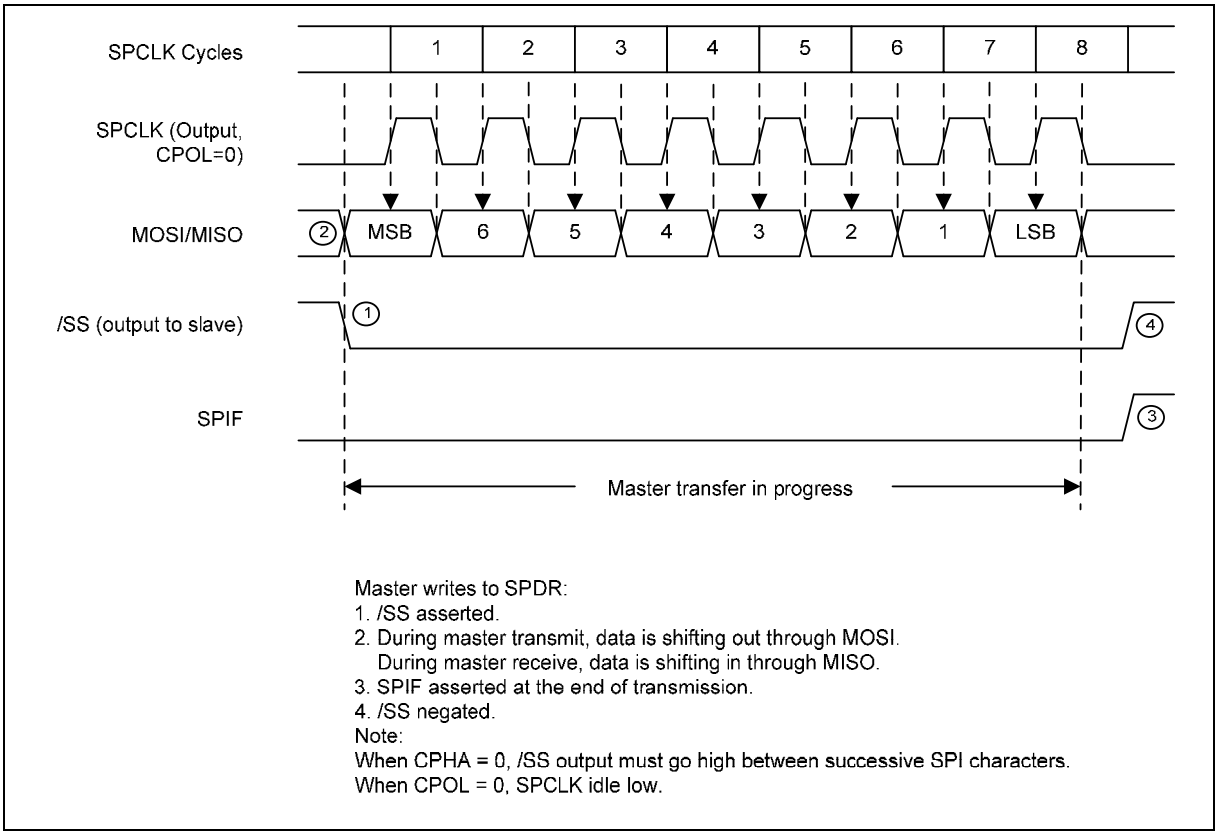
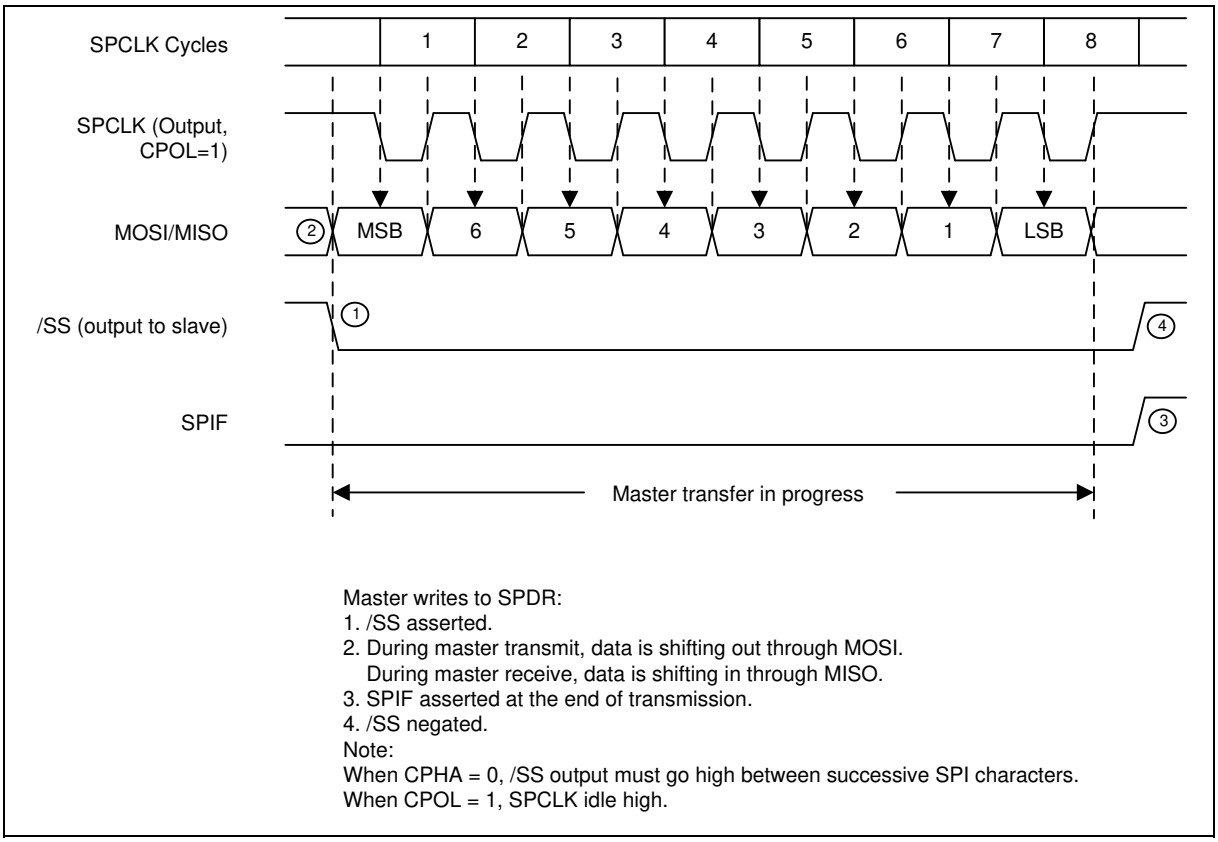



Figure 18-2: Master Mode Transmission (CPOL = 0, CPHA = 0)




Master writes to SPDR:
1. /SS asserted.
2. During master transmit, data is shifting out through MOSI.
   During master receive, data is shifting in through MISO.
3. SPIF asserted at the end of transmission.
4. /SS negated.

Note:
When CPHA = 0, /SS output must go high between successive SPI characters.
When CPOL = 1, SPCLK idle high.

Figure 18-3: Master Mode Transmission (CPOL = 1, CPHA = 0)

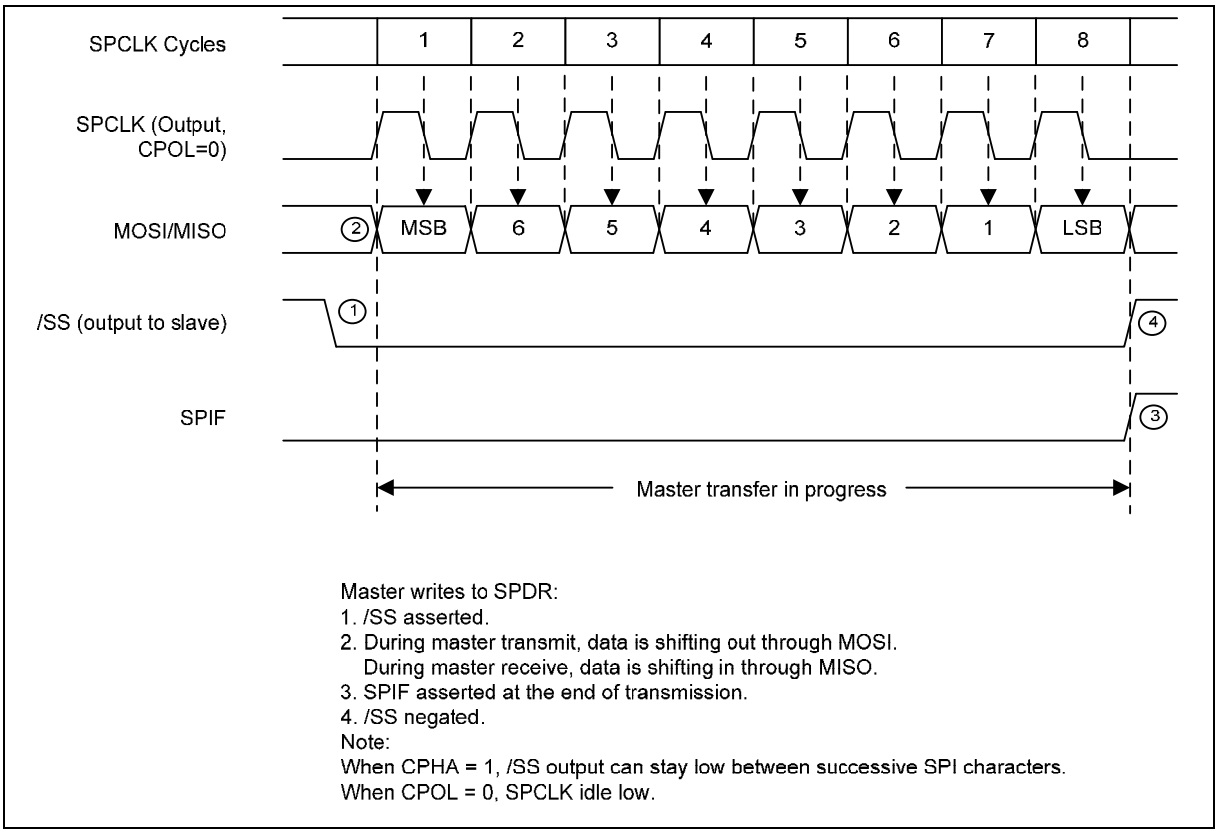



Figure 18-4: Master Mode Transmission (CPOL = 0, CPHA = 1)

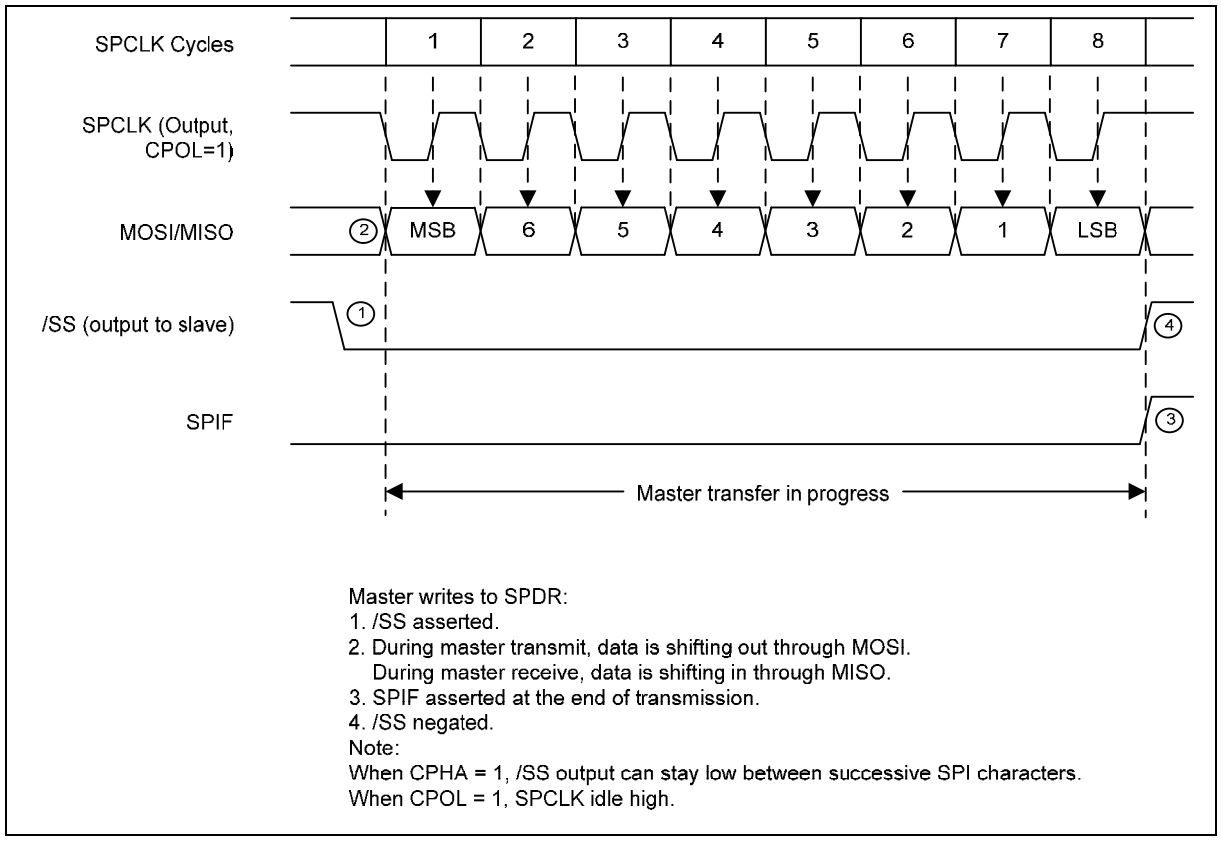



Figure 18-5: Master Mode Transmission (CPOL = 1, CPHA = 1)

### 18.3.2 Slave Mode

When in slave mode, the SPCLK pin becomes input and it will be clock by another master SPI device. The $\overline{SS}$ pin also becomes input. Similarly, before data transmissions occurs, and remain low until the transmission completed. If $\overline{SS}$ goes high, the SPI is forced into idle state. If the $\overline{SS}$ is forced to high at the middle of transmission, the transmission will be aborted and the receiving shifter buffer will be high and goes into idle states.

Data flows from master to slave on MOSI pin and flows from slave to master on MISO pin. The SPDR is used when transmitting or receiving data on the serial bus. Only a write to this register initiates transmission or reception of a byte, and this only occurs in the master device. At the completion of transferring a byte of data, the SPIF status bit is set in both the master and slave devices.

A read of the SPDR is actually a read of a buffer. To prevent an overrun and the loss of the byte that caused the overrun, the first SPIF must be cleared by the time a second transfer of data from the shift register to the read buffer is initiated.

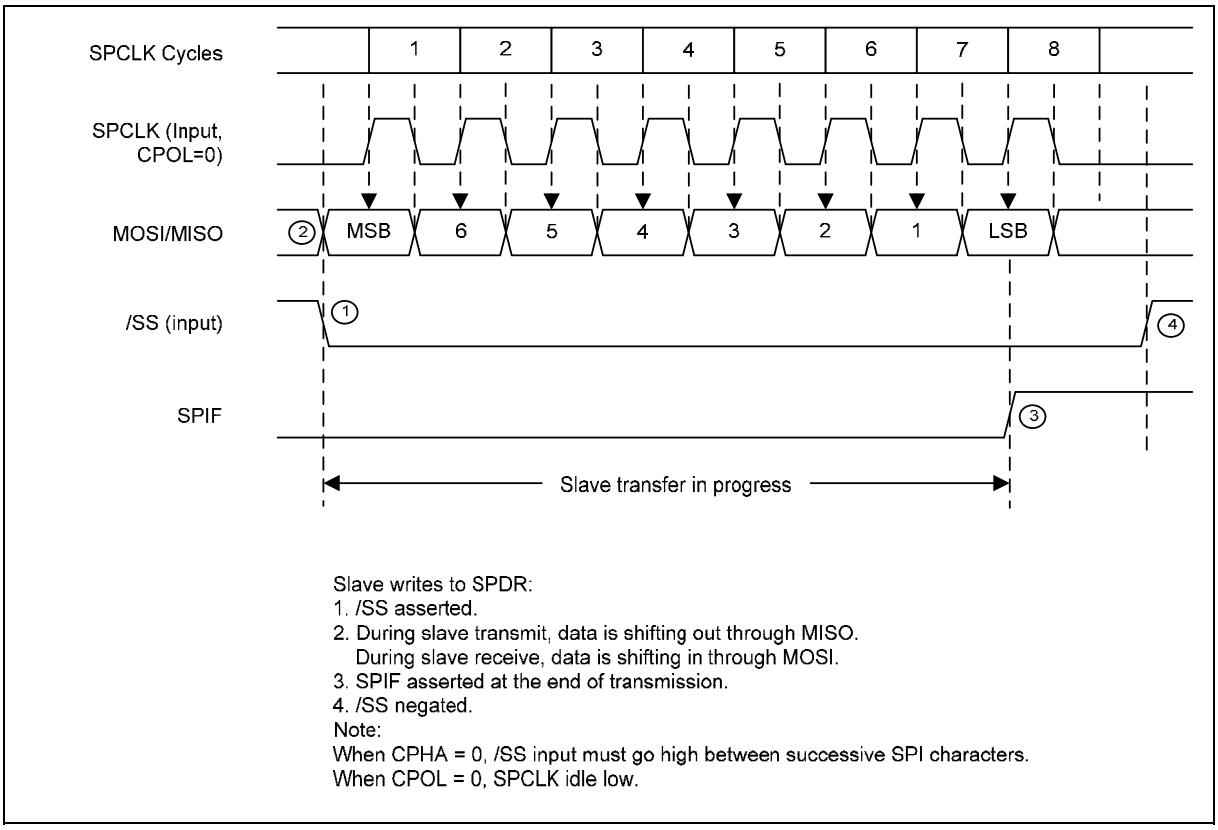



Figure 18-6: Slave Mode Transmission (CPOL = 0, CPHA = 0)

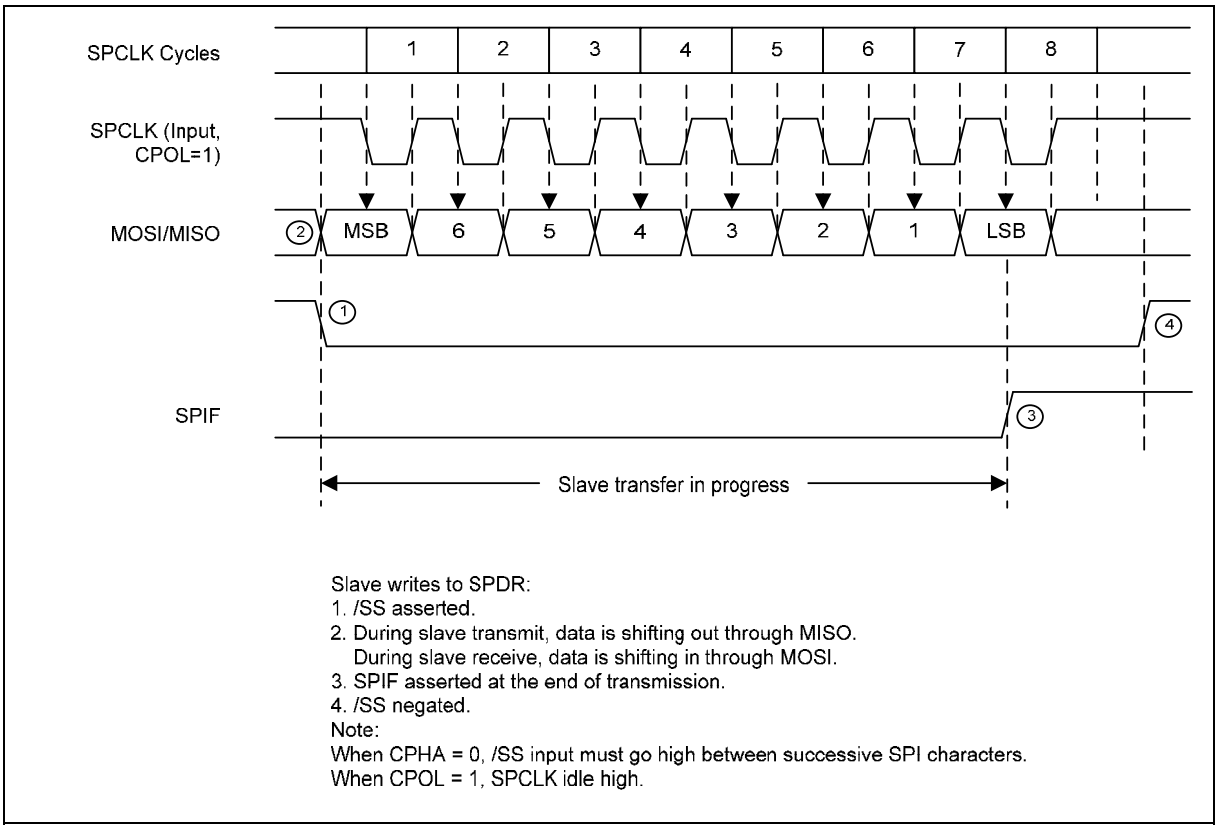



Figure 18-7: Slave Mode Transmission (CPOL = 1, CPHA = 0)

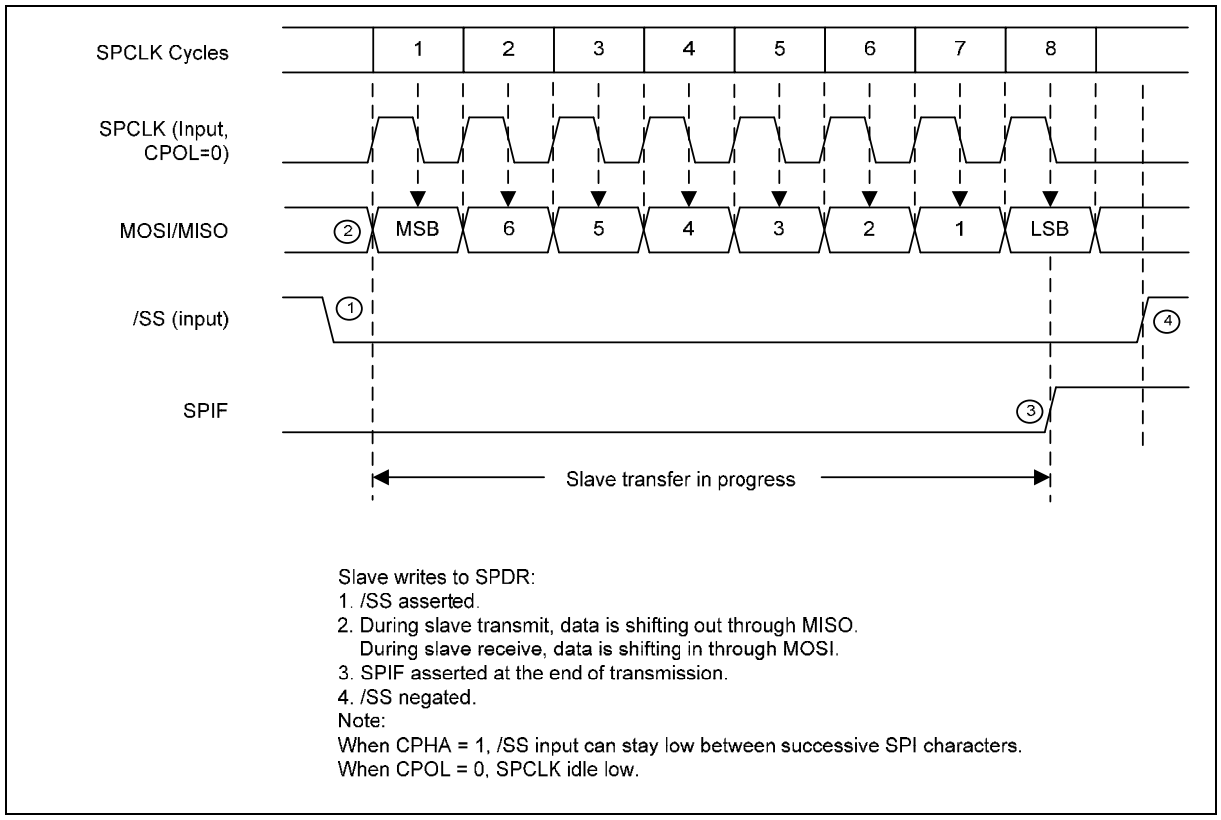



Figure 18-8: Slave Mode Transmission (CPOL = 0, CPHA = 1)

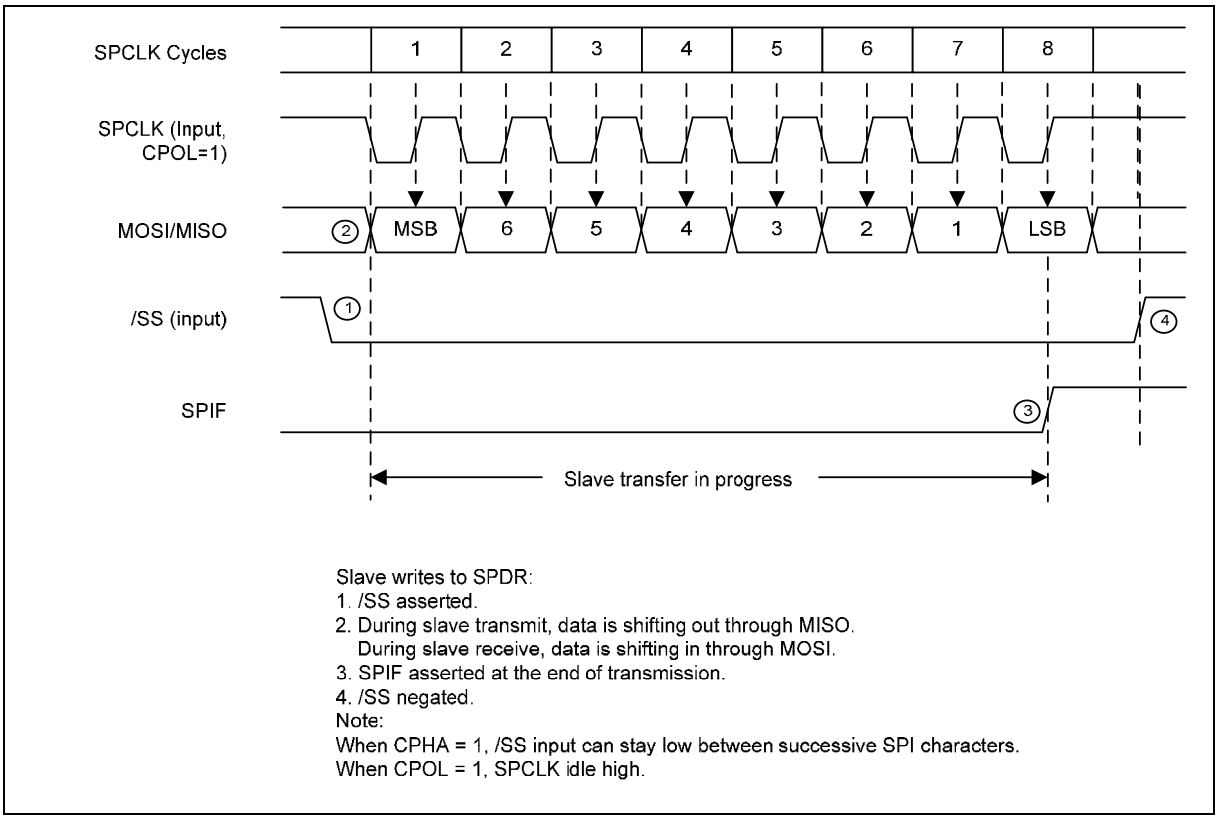



Figure 18-9: Slave Mode Transmission (CPOL = 1, CPHA = 1)

### 18.3.3 Slave select

The slave select ($\overline{SS}$) input of a slave device must be externally asserted before a master device can exchange data with the slave device. $\overline{SS}$ must be low before data transactions and must stay low for the duration of the transaction. The $\overline{SS}$ line of the master must be held high. The other three lines are dedicated to the SPI whenever the serial peripheral interface is on.

The state of the master and slave CPHA bits affects the operation of $\overline{SS}$. CPHA settings should be identical for master and slave. When CPHA = 0, the shift clock is the OR of $\overline{SS}$ with SPCLK. In this clock phase mode, $\overline{SS}$ must go high between successive characters in an SPI message. When CPHA = 1, $\overline{SS}$ can be left low between successive SPI characters. In cases where there is only one SPI slave MCU, its $\overline{SS}$ line can be tied to VSS as long as only CPHA = 1 clock mode is used.

### 18.3.4 /SS output

Available in master mode only, $\overline{SS}$ output is enabled with the SSOE bit in the SPCR register and DRSS bit in the SPSR register. The $\overline{SS}$ output pin is connected to the $\overline{SS}$ input pin of the slave device. The $\overline{SS}$ output automatically goes low for each transmission when selecting external device and it goes high during each idling state to deselect external devices.

| DRSS | SSOE | MASTER MODE | SLAVE MODE |
|---|---|---|---|
| 0 | 0 | $\overline{SS}$ input ( With Mode Fault ) | $\overline{SS}$ Input ( Not affected by SSOE ) |
| 0 | 1 | Reserved | $\overline{SS}$ Input ( Not affected by SSOE ) |
| 1 | 0 | $\overline{SS}$ General purpose I/O ( No Mode Fault ) | $\overline{SS}$ Input ( Not affected by SSOE ) |
| 1 | 1 | $\overline{SS}$ output ( No Mode Fault ) | $\overline{SS}$ Input ( Not affected by SSOE ) |

During master mode (with SSOE=DRSS= 0), mode fault will be set if $\overline{SS}$ pin is detected low. When mode fault is detected hardware will clear MSTR bit and SPE bit in the meantime it will also generated interrupt request, if ESPI is enabled.

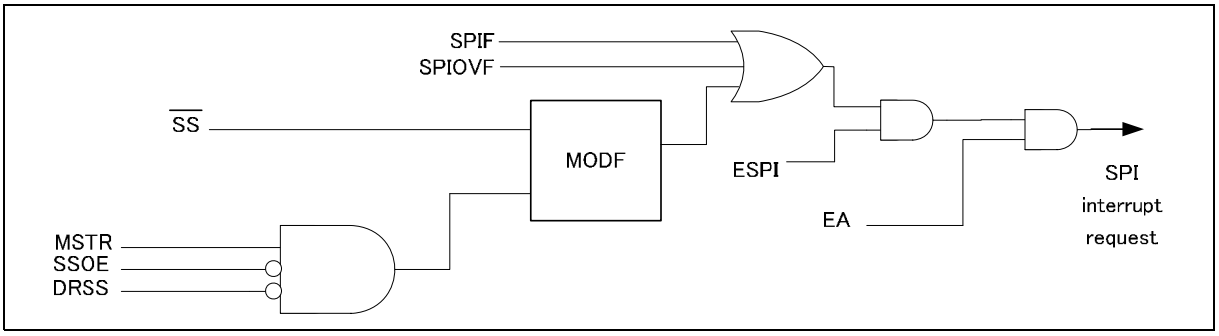



Figure 18-10: SPI interrupt request

### 18.3.5 SPI I/O pins mode

When SPI is disabled (SPE = 0) the corresponding I/O is following the original setting and act as a normal I/O. In the case of SPI is enabled (SPE = 1) the SPI pins I/O mode follow the below table. For $\overline{SS}$ pin it is always at Quasi-bidirectional mode whether it is configured as master or slave.

| | MISO | MOSI | CLK | /SS |
|---|---|---|---|---|
| **Master** | Input | Output | Output | Output*: DRSS=0,SSOE=0<br>Input: DRSS=1, SSOE=1 |
| **Slave** | Output** during /SS = Low<br>Else Input mode | Input | Input | Input |

Input = Quasi-bidirectional mode

Output = Push-pull mode

**Output*** = this output mode in /SS is Quasi-bidirectional output mode. Master needs to detect mode fault during master outputs /SS low.

**Output**** = In SLAVE mode, MISO is in output mode only during the time of $\overline{SS}$ =Low, otherwise it must keep in input mode (Quasi-bidirectional).

### 18.3.6 Programmable serial clock's phase and polarity

The clock polarity CPOL control bit selects active high or active low SPCLK clock, and has no significant effect on the transfer format. The clock phase CPHA control bit selects one of two different transfer protocols by sampling data on odd numbered SPCLK edges or on even numbered SPCLK edges. Thus, both these bits enable selection of four possible clock formats to be used by SPI system.

The clock phase and polarity should be identical for the master SPI device and the communicating slave device.

When CPHA equals 0, the $\overline{SS}$ line must be negated and reasserted between each successive serial byte. Also, if the slave writes data to the SPI data register (SPDR) while $\overline{SS}$ is low, a write collision error results. When CPHA equals 1, the $\overline{SS}$ line can remain low between successive transfers. The figures from

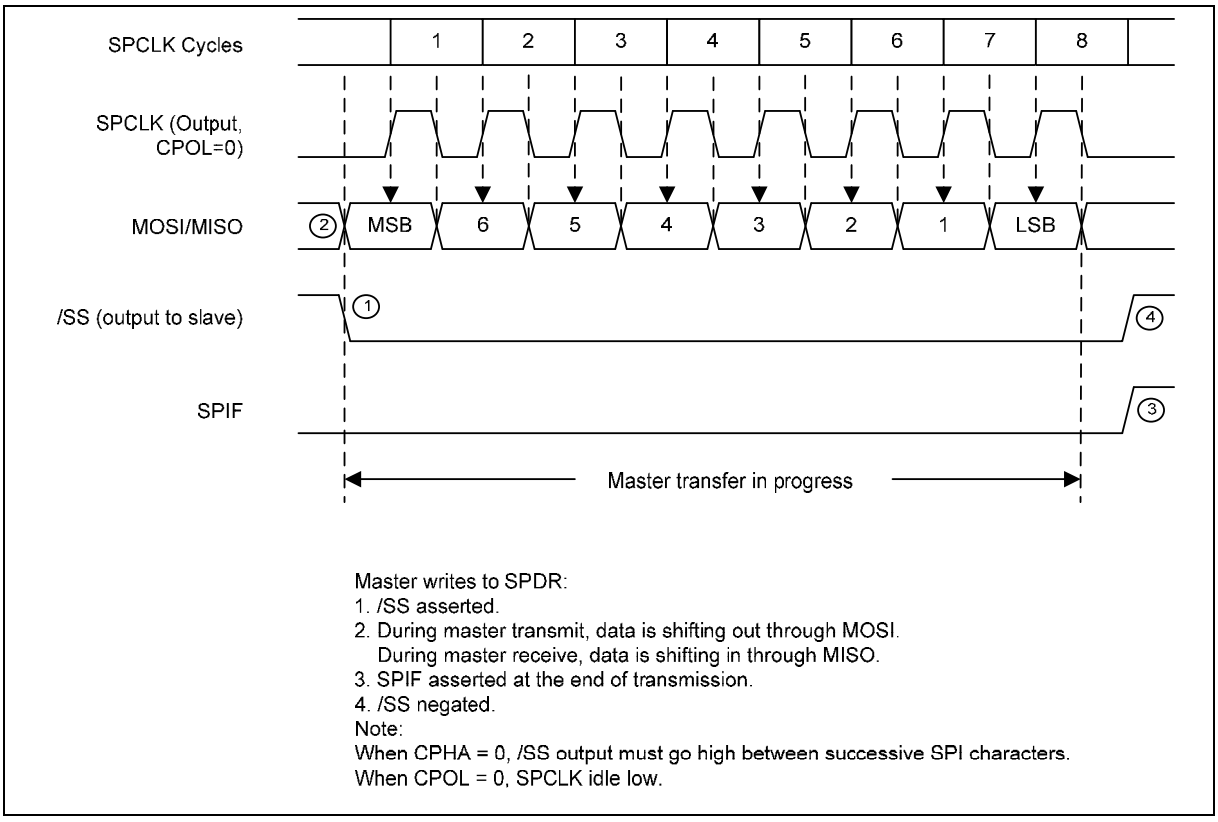



Figure 18-2 to

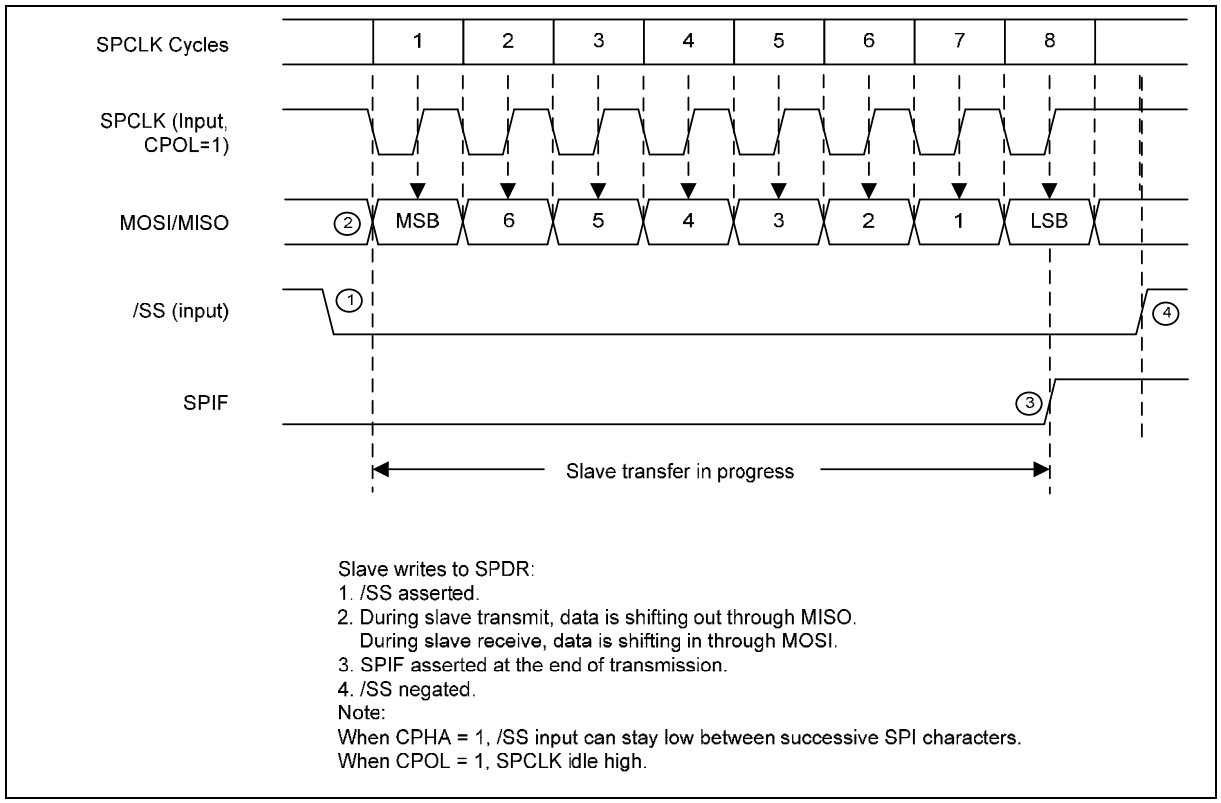



Figure 18-9 show the SPI transfer format, with different CPOL and CPHA. When CPHA = 0, data is sample on the first edge of SPCLK and when CPHA = 1 data is sample on the second edge of the SPCLK. Prior to changing CPOL setting, SPE must be disabled first.

### 18.3.7 Receive double buffered data register

This device is single buffered in the transmit direction and double buffered in the receive direction. This means that new data for transmission cannot be written to the shifter until the previous transfer is complete; however, received data is transferred into a parallel read data buffer so the shifter is free to accept a second serial byte.

As long as the first byte is read out of the read data buffer before the next byte is ready to be transferred, no overrun condition occurs. If overrun occur, SPIOVF is set. Second byte serial data cannot be transferred successfully into the data register during overrun condition and the data register will remains the value of the previous byte. The figure below shows the receive data timing waveform when overrun occur.

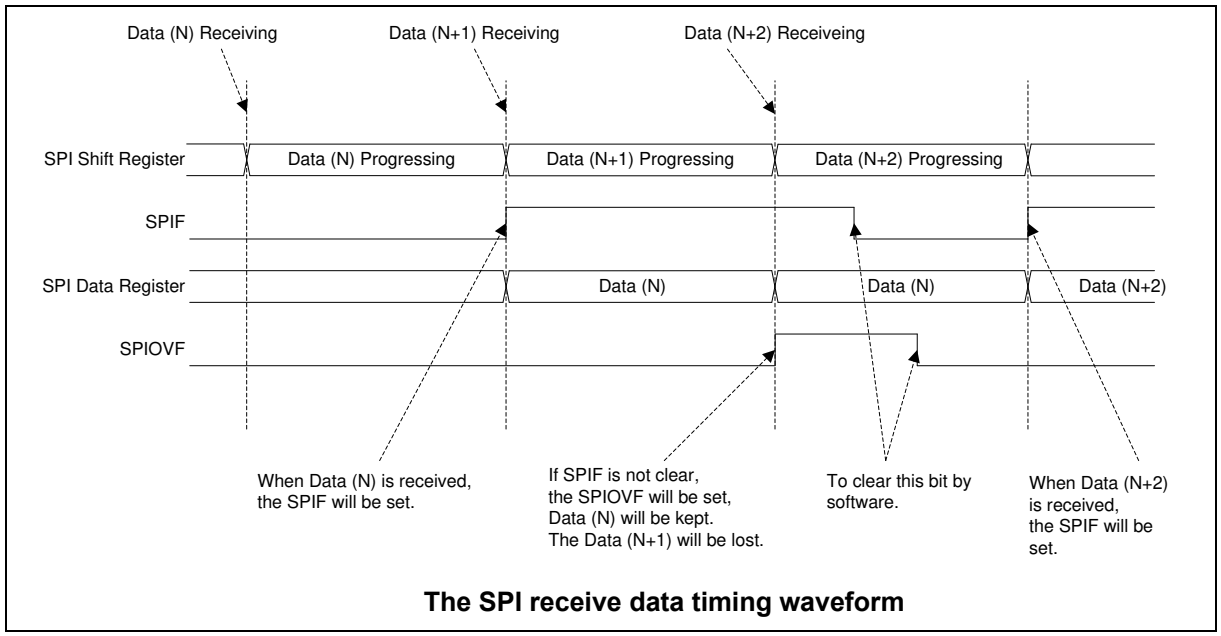



Figure 18-11: SPI Overrun Timing Waveform

### 18.3.8 LSB first enable

By default, this device transfer the SPI data most significant bit first. This device provides a control bit SPCR.LSBFE to allow support of transfer of SPI data in least significant bit first.

### 18.3.9 Write Collision detection

Write collision indicates that an attempt was made to write data to the SPDR while a transfer was in progress. SPDR is not double buffered in the transmit direction, any writes to SPDR cause data to be written directly into the SPI shift register. This write corrupts any transfer in progress, a write collision error is generated (WCOL will be set). The transfer continues undisturbed, and the write data that caused the error is not written to the shifter. A write collision is normally a slave error because a slave has no control over when a master initiates a transfer. A master knows when a transfer is in progress, so there is no reason for a master to generate a write-collision error, although the SPI logic can detect write collisions in both master and slave devices. WCOL flag is clear by software.

### 18.3.10 Transfer complete interrupt

This device consists of an interrupt flag at SPIF. This flag will be set upon completion of data transfer with external device, or when a new data have been received and copied to SPDR. If interrupt is enable (through ESPI), the SPI interrupt request will be generated, if global enable bit EA is also enabled. SPIF is software clear.

### 18.3.11 Mode Fault

Error arises in a multiple-master system when more than one SPI device simultaneously tries to be a master. This error is called a mode fault.

When the SPI system is configured as a master and the /SS input line goes to active low, a mode fault error has occurred — usually because two devices have attempted to act as master at the same time. In cases where more than one device is concurrently configured as a master, there is a chance of contention between two pin drivers. For push-pull CMOS drivers, this contention can cause permanent damage. The mode fault mechanism attempts to protect the device by disabling the drivers. The MSTR and SPE control bits in the SPCR associated with the SPI are cleared by hardware and an interrupt is generated subject to masking by the ESPI control bit.

Other precautions may need to be taken to prevent driver damage. If two devices are made masters at the same time, mode fault does not help protect either one unless one of them selects the other as slave. The amount of damage possible depends on the length of time both devices attempt to act as master.

MODF bit is set automatically by SPI hardware, if the MSTR control bit is set and the slave select input pin becomes 0. This condition is not permitted in normal operation. In the case where /SS is set, it is an output pin rather than being dedicated as the /SS input for the SPI system. In this special case, the mode fault function is inhibited and MODF remains cleared. This flag is cleared by software.

The following shows the sample hardware connection and s/w flow for multi-master/slave environment. It shows how s/w handles mode fault.

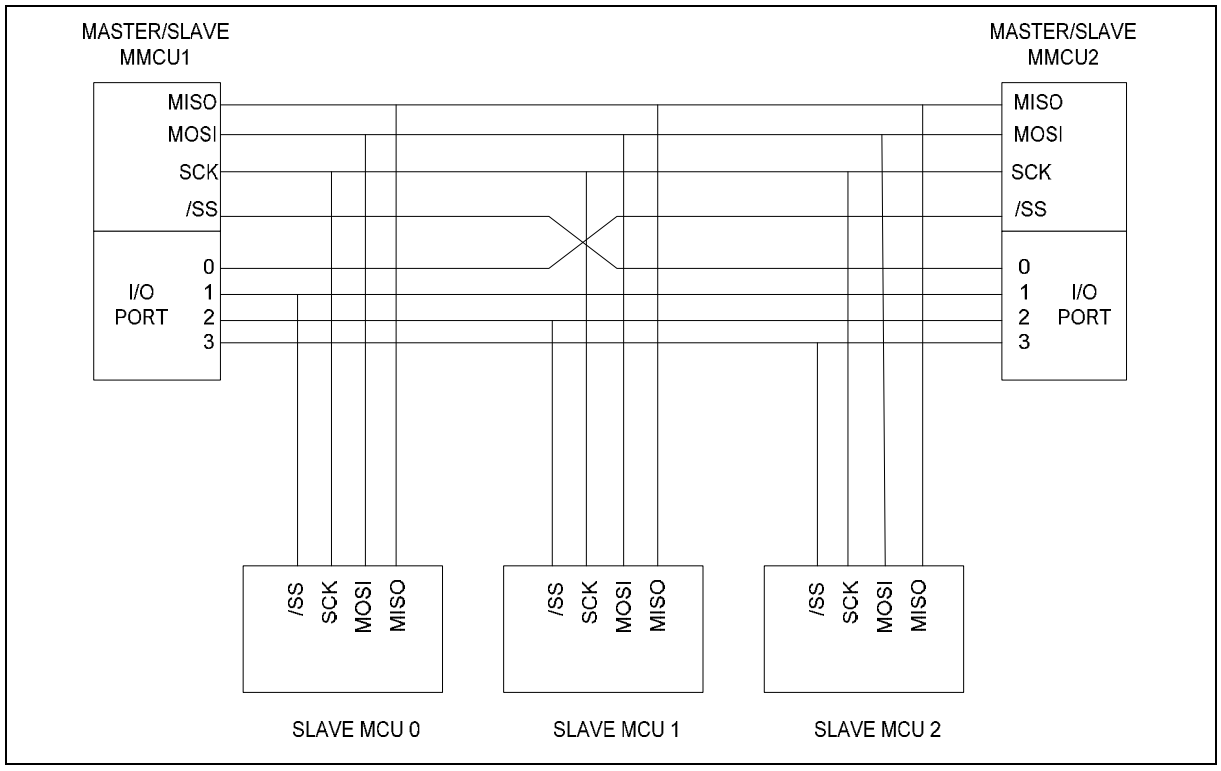



Figure 18-12: SPI multi-master slave environment

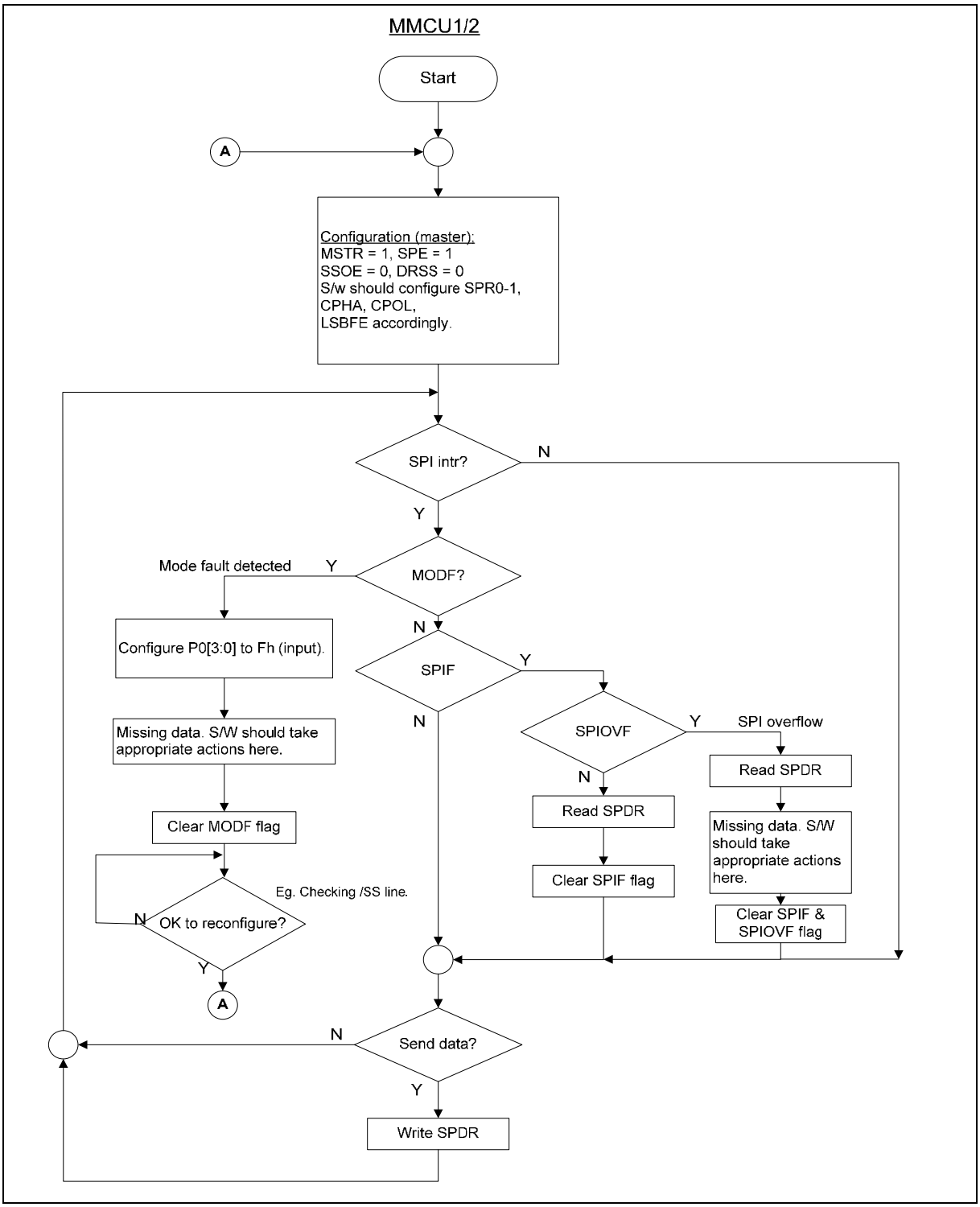



Figure 18-13: SPI multi-master slave s/w flow diagram

## 19. ANALOG-TO-DIGITAL CONVERTER

The ADC contains a digital-to-analog converter (DAC) that converts the contents of a successive approximation register to a voltage ($V_{DAC}$), which is compared to the analog input voltage (Vin). The output of the comparator is then fed back to the successive approximation control logic that controls the successive approximation register. This is illustrated in the figure below.

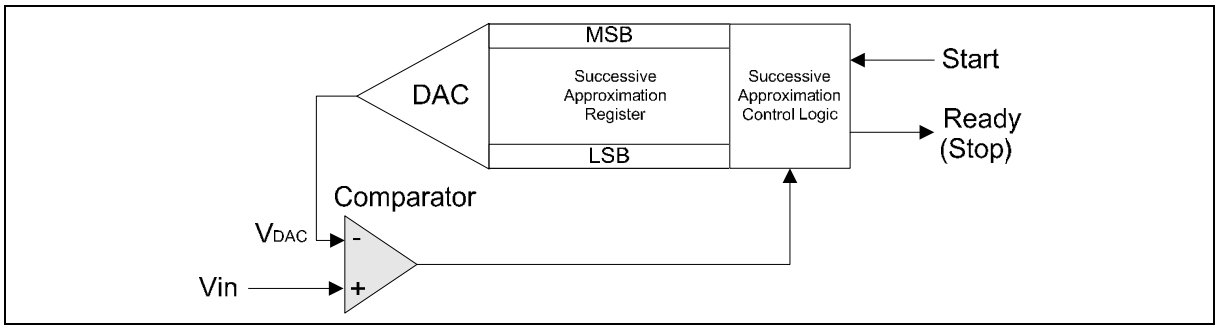



Figure 19-1: Successive Approximation ADC

### 19.1 Operation of ADC

A conversion can be initiated by software only or by either hardware or software. The software only start mode is selected when control bit ADCCON.5 (ADCEX) =0. A conversion is then started by setting control bit ADCCON.3 (ADCS) to 1. The hardware or software start mode is selected when ADCCON.5 =1, and a conversion may be started by setting ADCCON.3 = 1 as above or by applying a rising edge to external pin STADC (P4.0). When a conversion is started by applying a rising edge, a low level must be applied to STADC for at least one machine cycle followed by a high level for at least one machine cycle.

User sets ADCS to start converting then ADCS remains high while ADC is converting signal and will be automatically cleared by hardware when ADC conversion is completed. The end of the 10-bit conversion is flagged by control bit ADCCON.4 (ADCI). The upper 8 bits of the result are held in special function register ADCH, and the two remaining bits are held in ADCL.1 (ADC.1) and ADCL.0 (ADC.0). The user may ignore the two least significant bits in ADCL and use the ADC as an 8-bit converter (8 upper bits in ADCH). In any event, the total actual conversion time is 50 ADC clock input cycles.

Control bits from ADCCON.0 to ADCCON.2 are used to control an analog multiplexer which selects one of eight analog channels. An ADC conversion in progress is unaffected by an external or software ADC start. The result of a completed conversion remains unaffected provided ADCI = logic 1. The result of a completed conversion (ADCI = logic 1) remains unaffected when entering the idle mode.

The device supports maximum 8 analog input ports. 8 analog input ports share the I/O pins from P1.0 to P1.7. These I/O pins are switched to analog input ports by setting the bits of ADC Input Pin Select Register (DDIO) to logic 1.

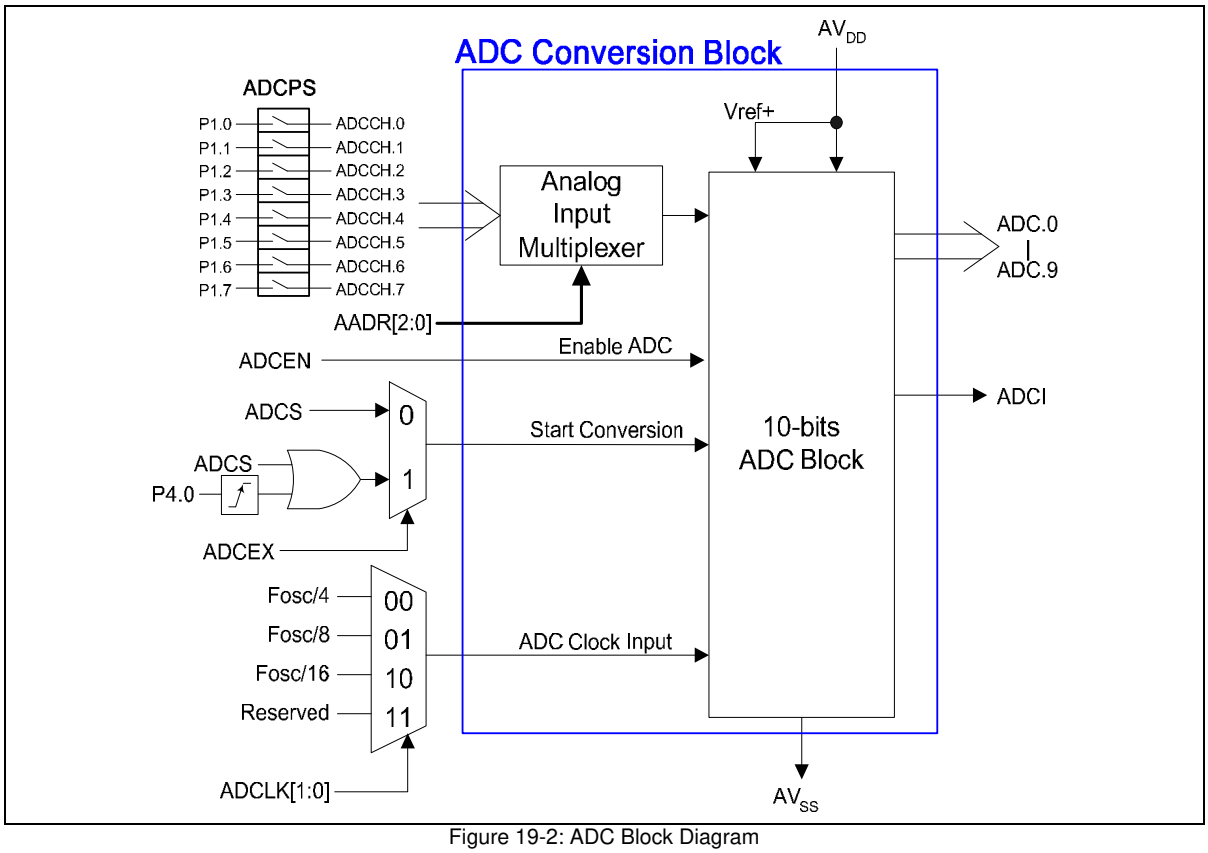



Figure 19-2: ADC Block Diagram

## 19.2 ADC Resolution and Analog Supply

The ADC circuit has its own supply pins ($AV_{DD}$ and $AV_{SS}$) and one pins (Vref+) connected to each end of the DAC's resistance-ladder that the $AV_{DD}$ and Vref+ are connected to $V_{DD}$ and $AV_{SS}$ is connected to $V_{SS}$. The ladder has 1023 equally spaced taps, separated by a resistance of "R". The first tap is located 0.5xR above $AV_{SS}$, and the last tap is located 0.5xR below Vref+. This gives a total ladder resistance of 1024xR. This structure ensures that the DAC is monotonic and results in a symmetrical quantization error.

For input voltages between $AV_{SS}$ and [(Vref+) + ½ LSB], the 10-bit result of an A/D conversion will be 0000000000B = 000H. For input voltages between [(Vref+) – 3/2 LSB] and Vref+, the result of a conversion will be 1111111111B = 3FFH. AVref+ and $AV_{SS}$ may be between $AV_{DD}$ + 0.2V and $AV_{SS}$ – 0.2 V. Avref+ should be positive with respect to $AV_{SS}$, and the input voltage (Vin) should be between AVref+ and $AV_{SS}$.

The result can always be calculated from the following formula:

$$\text{Result} = 1024 \times \frac{\text{Vin}}{\text{AVref}+} \quad \text{or} \quad \text{Result} = 1024 \times \frac{V_{DD}}{V_{SS}}$$

## 20. TIMED ACCESS PROTECTION

The W79E22X SERIES has features like the Watchdog Timer, wait-state control signal and power-on/fail reset flag that are crucial to the proper operation of the system. If these features are unprotected, errant code may write critical control bits, resulting in incorrect operation and loss of control. To prevent this, the W79E22X SERIES provides has a timed-access protection scheme that controls write access to critical bits.

In this scheme, protected bits have a timed write-enable window. A write is successful only if this window is active; otherwise, the write is discarded. The write-enable window is opened in two steps. First, the software writes AAh to the Timed Access (TA) register. This starts a counter, which expires in three machine cycles. Then, if the software writes 55h to the TA register before the counter expires, the write-enable window is opened for three machine cycles. After three machine cycles, the window automatically closes, and the procedure must be repeated again to access protected bits.

The suggested code for opening the write-enable window is;

```
TA      REG     0C7h                ; Define new register TA, located at 0C7h
        MOV     TA, #0AAh
        MOV     TA, #055h
```

Five examples, some correct and some incorrect, of using timed-access protection are shown below.

Example 1: Valid access

```
        MOV     TA, #0AAh           ; 3 M/C ; Note: M/C = Machine Cycles
        MOV     TA, #055h           ; 3 M/C
        MOV     WDCON, #00h         ; 3 M/C
```

Example 2: Valid access

```
        MOV     TA, #0AAh           ; 3 M/C
        MOV     TA, #055h           ; 3 M/C
        NOP                         ; 1 M/C
        SETB    EWT                 ; 2 M/C
```

Example 3: Valid access

```
        MOV     TA, #0Aah           ; 3 M/C
        MOV     TA, #055h           ; 3 M/C
        ORL     WDCON, #00000010B   ; 3M/C
```

Example 4: Invalid access

```
        MOV     TA, #0AAh           ; 3 M/C
        MOV     TA, #055h           ; 3 M/C
        NOP                         ; 1 M/C
        NOP                         ; 1 M/C
        CLR     POR                 ; 2 M/C
```

Example 5: Invalid Access

```
MOV   TA, #0AAh      3 M/C
NOP                  1 M/C
MOV   TA, #055h      3 M/C
SETB  EWT            2 M/C
```

In the first three examples, the protected bits are written before the window closes. In Example 4, however, the write occurs after the window has closed, so there is no change in the protected bit. In Example 5, the second write to TA occurs four machine cycles after the first write, so the timed access window in not opened at all, and the write to the protected bit fails.

## 21. PORT 4 STRUCTURE

Port 4 is a multi-function port that performs general purpose I/O port and chip-select strobe signals including read strobe, write strobe and read/write strobe signals. The 4 alternate modes are selected by P4xM1 and P4xM0. The function of chip-select strobe output provides that user can activate external devices by access to some specific address region.

**Port 4 Control Register A**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | P41M1 | P41M0 | P41C1 | P41C0 | P40M1 | P40M0 | P40C1 | P40C0 |

Mnemonic: P4CONA Address: 92h

**Port 4 Control Register B**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | P43M1 | P43M0 | P43C1 | P43C0 | P42M1 | P42M0 | P42C1 | P42C0 |

Mnemonic: P4CONB Address: 93h

| BIT NAME | FUNCTION |
|---|---|
| P4xM1, P4xM0 | Port 4 alternate modes.<br>=00: Mode 0. P4.x is a general purpose I/O port which is the same as Port 1.<br>=01: Mode 1. P4.x is a Read Strobe signal for chip select purpose. The address range depends on the SFR P4xAH, P4xAL and bits P4xC1, P4xC0.<br>=10: Mode 2. P4.x is a Write Strobe signal for chip select purpose. The address range depends on the SFR P4xAH, P4xAL and bits P4xC1, P4xC0.<br>=11: Mode 3. P4.x is a Read/Write Strobe signal for chip select purpose. The address range depends on the SFR P4xAH, P4xAL and bits P4xC1, P4xC0 |
| P4xC1, P4xC0 | Port 4 Chip-select Mode address comparison:<br>=00: Compare the full address (16 bits length) with the base address registers P4xAH and P4xAL.<br>=01: Compare the 15 high bits (A15-A1) of address bus with the base address registers P4xAH and P4xAL.<br>=10: Compare the 14 high bits (A15-A2) of address bus with the base address registers P4xAH and P4xAL.<br>=11: Compare the 8 high bits (A15-A8) of address bus with the base address registers P4xAH and P4xAL. |

P40AH, P40AL:

The Base address registers for comparator of P4.0. P40AH contains the high-order byte of address; P40AL contains the low-order byte of address.

P41AH, P41AL:

The Base address registers for comparator of P4.1. P41AH contains the high-order byte of address; P41AL contains the low-order byte of address.

P42AH, P42AL:

The Base address registers for comparator of P4.2. P42AH contains the high-order byte of address; P42AL contains the low-order byte of address.

P43AH, P43AL:

The Base address registers for comparator of P4.3. P43AH contains the high-order byte of address; P43AL contains the low-order byte of address.

**PORT 4**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | P4.3 | P4.2 | P4.1 | P4.0 |

Mnemonic: P4 Address: A5h

P4.3-0 Port 4 is a bi-directional I/O port with internal pull-ups.

**PORT 4 CHIP-SELECT POLARITY**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | P43INV | P42INV | P42INV | P40INV | - | PWDNH | RMWFP | PUP0 |

Mnemonic: P4CSIN Address: A2h

P4xINV The active polarity of P4.x when it is set as a chip-select strobe output. High = Active High. Low = Active Low.

PWDNH Set PWDNH to logic 1 then ALE and PSEN will keep high state, clear this bit to logic 0 then ALE and PSEN will output low during power down mode.

RMWFP Control Read Path of Instruction “Read-Modify-Write”. When this bit is set, the read path of executing “read-modify-write” instruction is from port pin otherwise from SFR.

PUP0 Enable Port 0 weak pull up.

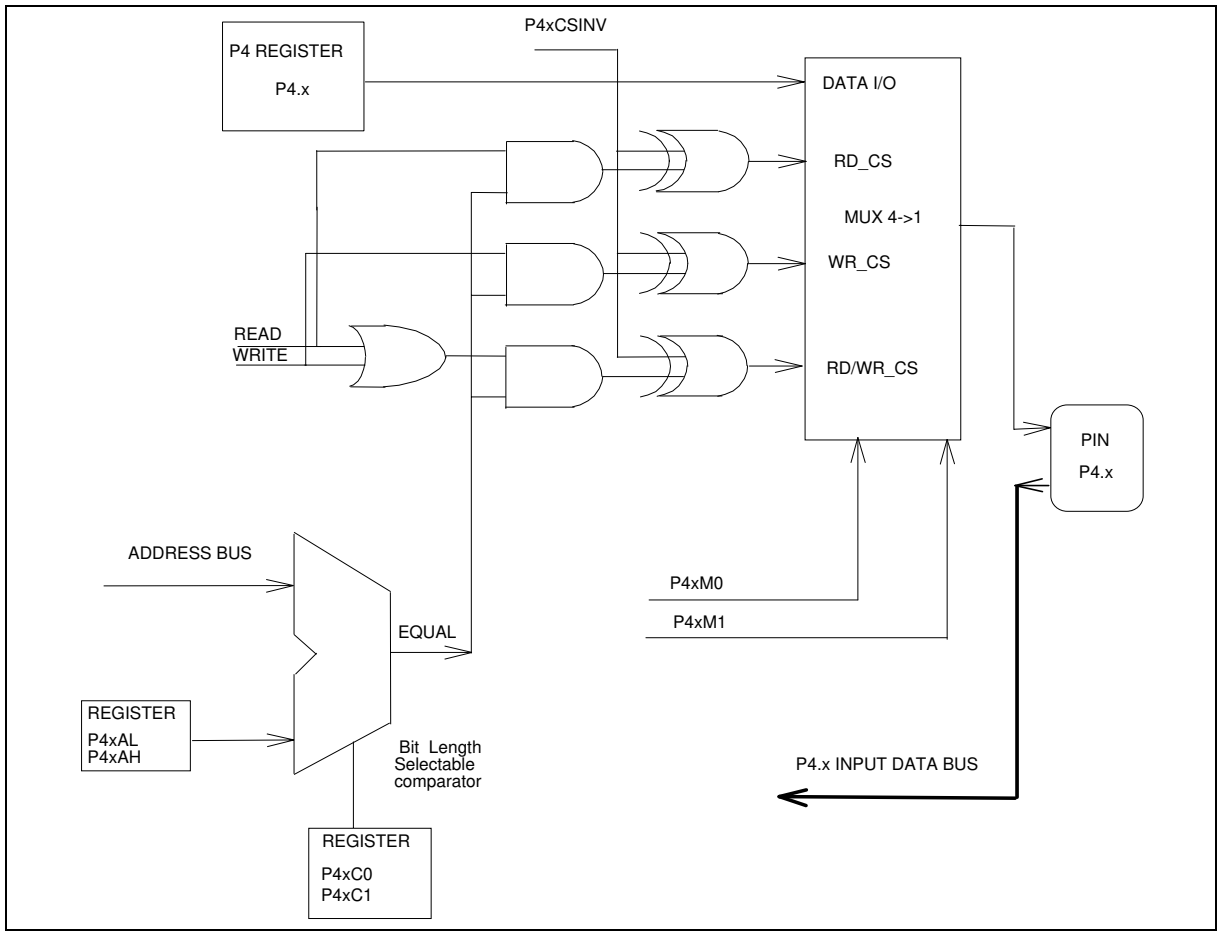



Figure 21-1 Port 4 Structure Diagram

Here is an example to program the P4.0 as a write strobe signal at the I/O port address 1234H ~1237H and positive polarity, and P4.1 ~ P4.3 are used as general I/O ports.

```
MOV     P40AH, #12H
MOV     P40AL, #34H             ;Define the base I/O address 1234H for P4.0 as an special function
MOV     P4CONA, #00001010B ;Define the P4.0 as a write strobe signal pin and the compared address is [A15:A2]
MOV     P4CONB, #00H            ;P4.1~P4.3 as general I/O port which are the same as PORT1
MOV     P4CSIN, #10H            ;Write the P40CSINV =1 to inverse the P4.0 write strobe polarity
```

Then any instruction writes data to address from 1234H to 1237H, for example MOVX @DPTR,A (with DPTR=1234H~1237H), will generate the positive polarity write strobe signal at pin P4.0. And the instruction of “MOV P4, #XX” will output the bit3 to bit1 of data #XX to pin P4.3~ P4.1.

**Note: P4.2 and P4.3 pins are available in 48L LQFP package only.**

# 22. IN-SYSTEM PROGRAMMING

## 22.1 The Loader Program Locates at LDFlash Memory

CPU is Free Run at APFlash memory. CHPCON register had been set #03H value before CPU has entered idle state. CPU will switch to LDFlash memory and execute a reset action. H/W reboot mode will switch to LDFlash memory, too. Set SFRCN register where it locates at user's loader program to update APFlash bank 0 memory. Set a SWRESET (CHPCON=#83H) to switch back APFlash after CPU has updated APFlash program. CPU will restart to run program from reset state.

## 22.2 The Loader Program Locates at APFlash Memory

CPU is Free Run at APFlash memory. CHPCON register had been set #01H value before CPU has entered idle state. Set SFRCN register to update LDFlash or another bank of APFlash program. CPU will continue to run user's APFlash program after CPU has updated program. Please refer demonstrative code to understand other detail description.

## 23. OPTION BITS

This device has two CONFIG bits (CONFIG0, CONFIG1) that must be define at power up and can not be set after the program start of execution. Those features are configured through the use of two flash EPROM bytes, and the flash EPROM can be programmed and verified repeatedly. Until the code inside the Flash EPROM is confirmed OK, the code can be protected. The protection of flash EPROM and those operations of the configuration bits are described below.

### 23.1 Config0

| BIT | DESCRIPTION |
|---|---|
| B0 | =0: Lock data out |
| B1 | =0: MOVC Inhibited |
| B2 | =0; 1/2/2K Data Flash EPROM lock bit |
| B3 | Reserved |
| B4 | =1: Disable H/W reboot by P3.6 and P3.7<br>=0: Enable H/W reboot by P3.6 and P3.7 |
| B5 | =1: Disable H/W reboot by P4.3<br>=0: Enable H/W reboot by P4.3<br>**Note: Support in 48L LQFP package only.** |
| B6 | Reserved |
| B7 | =1: Crystal > 24MHz<br>=0: Crystal < 24MHz |

Table 23-1 Config0 Option Bits

**B0: Lock bit**

This bit is used to protect the customer's program code in the W79E22X SERIES. After the programmer finishes the programming and verifies sequence B0 can be cleared to logic 0 to protect code from reading by any access path. Once this bit is set to logic 0, both the Flash EPROM data and Special Setting Registers can not be accessed again.

**B1: MOVC Inhibit**

This bit is used to restrict the accessible region of the MOVC instruction. It can prevent the MOVC instruction in external program memory from reading the internal program code. When this bit is set to logic 0, a MOVC instruction in external program memory space will be able to access code only in the external memory, not in the internal memory. A MOVC instruction in internal program memory space will always be able to access the ROM data in both internal and external memory. If this bit is logic 1, there are no restrictions on the MOVC instruction.

**B4: H/W Reboot with P3.6 and P3.7**

If this bit is set to logic 0, enable to reboot 4k LD Flash mode while RST =H, P3.6 = L and P3.7 = L state. CPU will start from LD Flash to update the user's program.

**B5: H/W Reboot with P4.3**

If this bit is set to logic 0, enable to reboot 4k LD Flash mode while RST =H and P4.3 = L state. CPU will start from LD Flash to update the user's program

**B7: Select clock frequency.**

If clock frequency is over 24MHz, then set this bit is H. If clock frequency is less than 24MHz, then clear this bit.

## 23.2 Config1

| BITS | NAME | FUNCTION |
|---|---|---|
| **Bit 0** | **PWMOE** | PWM Odd Channel 1, 3 and 5 Enable.<br>1: (Default). Disable odd PWM outputs to corresponding pins and the ports keep in Quasi bi-direction type.<br>0: Enable odd PWM outputs to corresponding pins and the ports are switched to push-pull type |
| **Bit 1** | **PWMEE** | PWM Even Channel 0, 2 and 4 Enable.<br>1: (Default). Disable even PWM outputs to corresponding pins and the ports keep in Quasi bi-direction type.<br>0: Enable even PWM outputs to corresponding pins and the ports are switched to push-pull type |
| **Bit 2** | **OPOL** | Define the polarity of PWM output after CPU reset, OPOL controls odd PWM outputs.<br>1: Initial output high<br>0: Initial output low |
| **Bit 3** | **EPOL** | Define the polarity of PWM output after CPU reset, EPOL control even PWM outputs.<br>1: Initial output high<br>0: Initial output low |
| **Bit 4-5** | **-** | Reserved. |
| **Bit 6** | **PWM6E** | PWM Channel 6 Output Enable.<br>1: (Default). Disable PWM6 outputs to corresponding pins and the ports keep in Quasi bi-direction type.<br>0: Enable PWM6 outputs to corresponding pins and the ports are switched to push-pull type |
| **Bit 7** | **PWM7E** | PWM Channel 7 Output Enable.<br>1: (Default). Disable PWM7 outputs to corresponding pins and the ports keep in Quasi bi-direction type.<br>0: Enable PWM7 outputs to corresponding pins and the ports are switched to push-pull type |

Table 23-2: Config 1 Option Bits

# 24. ELECTRICAL CHARACTERISTICS

## 24.1 Absolute Maximum Ratings

| SYMBOL | PARAMETER | CONDITION | RATING | UNIT |
|---|---|---|---|---|
| DC Power Supply | $V_{DD} - V_{SS}$ | -0.3 | +7.0 | V |
| Input Voltage | $V_{IN}$ | $V_{SS}$ -0.3 | $V_{DD}$ +0.3 | V |
| Operating Temperature | $T_A$ | -40 | +85 | °C |
| Storage Temperature | $T_{st}$ | -55 | +150 | °C |

**Note:** Exposure to conditions beyond those listed under Absolute Maximum Ratings may adversely affect the life and reliability of the device.

## 24.2 DC Characteristics

(VDD – VSS = 5V ±10%, TA = 25°C, Fosc = 20 MHz, unless otherwise specified.)

| PARAMETER | SYMBOL | SPECIFICATION | | | UNIT | TEST CONDITIONS |
|---|---|---|---|---|---|---|
| | | MIN | TYP | MAX | | |
| Operating Voltage | $V_{DD1}$ | 4.5 | | 5.5 | V | $V_{DD}$ =4.5V ~ 5.5V @ 40MHz |
| | $V_{DD2}$ | 2.7 | | 5.5 | | $V_{DD}$ =2.7V ~ 5.5V @ 20MHz |
| | $V_{DD3}$ | 4.5 | | 5.5 | | $V_{DD}$ =4.5V ~ 5.5V @ 24MHz (external access) |
| | $V_{DD4}$ | 3.0 | | | | NVM program/erase operation. |
| Operating Current | $I_{DD1}$ | - | 58 | 65 | mA | Run NOP |
| | | | | | | $V_{DD}$=5.5V at 40MHz |
| | $I_{DD2}$ | - | 37 | 45 | mA | Run NOP |
| | | | | | | $V_{DD}$=5.5V at 20MHz |
| | $I_{DD3}$ | - | 15 | 20 | mA | Run NOP |
| | | | | | | $V_{DD}$=3.0V at 20MHz |
| | $I_{DD4}$ | - | 12 | 16 | mA | Run NOP |
| | | | | | | $V_{DD}$=2.7V at 20MHz |
| | $I_{DD5}$ | - | 50 | 60 | mA | RST = VDD |
| | | | | | | $V_{DD}$=5.5V at 40MHz |
| | $I_{DD6}$ | - | 33 | 38 | mA | RST = VDD |
| | | | | | | $V_{DD}$=5.5V at 20MHz |
| | $I_{DD7}$ | - | 12 | 17 | mA | RST = VDD |
| | | | | | | $V_{DD}$=3.0V at 20MHz |
| | $I_{DD8}$ | - | 10 | 15 | mA | RST = VDD |
| | | | | | | $V_{DD}$=2.7V at 20MHz |

DC Characteristics, continued

| PARAMETER | SYMBOL | SPECIFICATION | | | UNIT | TEST CONDITIONS |
|---|---|---|---|---|---|---|
| | | MIN | TYP | MAX | | |
| Idle Current | $I_{IDLE1}$ | | 35 | 42 | mA | $V_{DD}$=5.5V at 40MHz (I/O High) |
| | | | | | | $V_{DD}$=5.5V at 40MHz (I/O Low) |
| | $I_{IDLE2}$ | - | 20 | 25 | mA | $V_{DD}$=5.5V at 20MHz (I/O High) |
| | | | | | mA | $V_{DD}$=5.5V at 20MHz (I/O Low) |
| | $I_{IDLE3}$ | - | 9 | 14 | mA | $V_{DD}$=3.0V at 20MHz (I/O High) |
| | | | | | | $V_{DD}$=3.0V at 20MHz (I/O Low) |
| | $I_{IDLE4}$ | - | 8 | 11 | mA | $V_{DD}$=2.7V at 20MHz |
| Power Down Current | $I_{PWDN}$ | - | - | 10 | uA | $V_{DD}$=2.7V~5.5V |
| Input Current P1, P2, P3, P4, P5 | $I_{IN1}$ | -95 | -55 | 10 | uA | $V_{DD}$=5.5V |
| | | | | | | $V_{IN}$=0V or $V_{DD}$ |
| Input Current RST | $I_{IN2}$ | -10 | 50 | 300 | uA | $V_{DD}$=5.5V |
| | | | | | | 0<$V_{IN}$<$V_{DD}$ |
| Input Leakage Current P0, /EA | $I_{LK}$ | -10 | 0 | 10 | uA | $V_{DD}$=5.5V |
| | | | | | | 0V<$V_{IN}$<$V_{DD}$ |
| Logic 1 to 0 Transition Current P1, P2, P3, P4, P5 [*4] | $I_{TL}$ [*4] | -500 | - | -200 | uA | $V_{DD}$ = 5.5V |
| | | | | | | $V_{IN}$ » 2.85V |
| Input Low Voltage P0, P1, P2, P3, P4, P5, /EA (Schmitt input) | $V_{IL1}$ | 0 | - | 0.3 $V_{DD}$ | V | $V_{DD}$ = 4.5V |
| Input High Voltage P0, P1, P2, P3, P4, P5, /EA (Schmitt input) | $V_{IH1}$ | 0.7 $V_{DD}$ | - | $V_{DD}$+0.2 | V | $V_{DD}$ = 5.5V |
| Hysteresis Voltage | $V_{HY}$ | - | 0.2 $V_{DD}$ | - | | |
| Input Low Voltage RST [*1] | $V_{IL21}$ | 0 | - | 1.6 | V | $V_{DD}$=4.5V |
| | $V_{IL22}$ | 0 | - | 0.8 | | $V_{DD}$=2.7V |
| Input High Voltage RST [*1] | $V_{IH21}$ | 3.5 | - | VDD+0.2 | V | $V_{DD}$=5.5V |
| | $V_{IH22}$ | 2 | - | VDD+0.2 | | $V_{DD}$=2.7V |
| Input Low Voltage XTAL1[*3] | $V_{IL31}$ | 0 | - | 0.8 | V | $V_{DD}$=4.5V |
| | $V_{IL32}$ | 0 | - | 0.4 | | $V_{DD}$=2.7V |
| Input High Voltage XTAL1[*3] | $V_{IH31}$ | 4.0 | - | $V_{DD}$+0.2 | V | $V_{DD}$=5.5V |
| | $V_{IH32}$ | 2.5 | - | $V_{DD}$+0.2 | | $V_{DD}$=2.7V |

DC Characteristics, continued

| PARAMETER | SYMBOL | SPECIFICATION | | | UNIT | TEST CONDITIONS |
|---|---|---|---|---|---|---|
| | | MIN | TYP | MAX | | |
| Source Current P2.0~P2.5, P5.0~P5.1 (PUSH-PULL Mode) PWM pins | $I_{SR11}$ | -22 | -31 | -42 | mA | $V_{DD}$ = 4.5V, $V_S$ = 2.4V |
| | $I_{SR12}$ | -6 | -9 | -13 | | $V_{DD}$ = 2.7V, $V_S$ = 2.0V |
| Source Current P1, P2, P3, P4, P5 (Quasi-bidirectional mode) | $I_{SR21}$ | -200 | -300 | -430 | uA | $V_{DD}$ = 4.5V, $V_S$ = 2.4V |
| | $I_{SR22}$ | -50 | -82 | -115 | | $V_{DD}$ =2.7V, $V_S$ = 2.0V |
| Sink Current P2.0~P2.5, P5.0~P5.1 (PUSH-PULL Mode) | $I_{SK11}$ | 18 | 22 | 32 | mA | $V_{DD}$ = 4.5V, $V_S$ = 0.45V |
| | $I_{SK12}$ | 10 | 15 | 25 | | $V_{DD}$ = 2.7V, $V_S$ = 0.45V |
| Sink Current P0,P1,P2, P3,P4,P5 (Quasi-bidirectional mode) | $I_{SK21}$ | 4 | 5 | 6 | mA | $V_{DD}$ = 4.5V, $V_S$ = 0.45V |
| | $I_{SK22}$ | 3 | 3.5 | 5 | | $V_{DD}$ = 2.7V, $V_S$ = 0.45V |
| Output Low Voltage P2.0~P2.5, P5.0~P5.1 (PUSH-PULL Mode) | $V_{OL11}$ | - | 0.35 | - | V | $V_{DD}$ = 4.5V, $I_{OL}$ = 20 mA |
| | $V_{OL12}$ | - | 0.07 | - | | $V_{DD}$ = 2.7V, $I_{OL}$ = 3.2 mA |
| Output Low Voltage P0, P1, P2, P3, P4, P5 (Quasi-bidirectional Mode) | $V_{OL21}$ | - | 0.35 | - | V | $V_{DD}$ = 4.5V, $I_{OL}$ = 4.0 mA |
| | $V_{OL22}$ | - | 0.35 | - | | $V_{DD}$ = 2.7V, $I_{OL}$ = 3.0 mA |
| Output High Voltage P2.0~P2.5, P5.0~P5.1 (PUSH-PULL Mode) | $V_{OH11}$ | - | 3.3 | - | V | $V_{DD}$ = 4.5V, $I_{OH}$ = -20mA |
| | $V_{OH12}$ | - | 2.5 | - | | $V_{DD}$ = 2.7V, $I_{OH}$ =-3.2mA |
| Output High Voltage P1, P2, P3, P4, P5, P6, P7 (Quasi-bidirectional Mode) | $V_{OH21}$ | - | 4.1 | - | V | $V_{DD}$ = 4.5V, $I_{OH}$ =-100uA |
| | $V_{OH22}$ | - | 2.52 | - | | $V_{DD}$ = 2.7V, $I_{OH}$ = -30uA |
| Sink current [*2] P0, P2, ALE, /PSEN | Isk31 | 3 | 5 | 8 | mA | $V_{DD}$=4.5V, Vs = 0.45V |
| | Isk32 | 2.5 | 3.5 | 6 | mA | $V_{DD}$=2.7V, Vs = 0.45V |
| Source current [*2] P0, P2, ALE, /PSEN | Isr31 | -6 | -7.5 | -9 | mA | $V_{DD}$=4.5V, Vs = 2.4V |
| | Isr32 | -1 | -2 | -3 | mA | $V_{DD}$=2.7V, Vs = 2.0V |

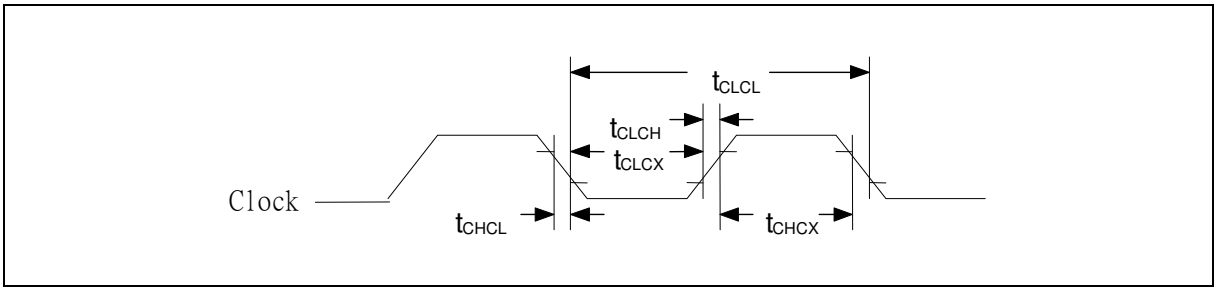
**Notes:** *1. RST pin is a Schmitt trigger input. RST has internal pull-low resistors about 60kΩ.

*2. P0, P2, ALE and /PSEN are tested in the external access mode.

*3. XTAL1 is a CMOS input.

*4. Pins of P1, P2, P3, P4, P5 can source a transition current when they are being externally driven from 1 to 0. The transition current reaches its maximum value when VIN approximates to 2V.

## 24.3 AC Characteristics




**Note:** Duty cycle is 50%.

### 24.3.1 External Clock Characteristics

| PARAMETER | SYMBOL | MIN. | TYP. | MAX. | UNITS | NOTES |
|---|---|---|---|---|---|---|
| Clock High Time | $t_{CHCX}$ | 12 | - | - | nS | |
| Clock Low Time | $t_{CLCX}$ | 12 | - | - | nS | |
| Clock Rise Time | $t_{CLCH}$ | - | - | 10 | nS | |
| Clock Fall Time | $t_{CHCL}$ | - | - | 10 | nS | |

### 24.3.2 AC Specification

($V_{DD} - V_{SS}$ = 5V ±10%, TA = 25°C, Fosc = 20 MHz, unless otherwise specified.)

| PARAMETER | SYMBOL | VARIABLE CLOCK MIN. | VARIABLE CLOCK MAX. | UNITS |
|---|---|---|---|---|
| Oscillator Frequency | $1/t_{CLCL}$ | 0 | 40[1] | MHz |
| Oscillator Frequency | $1/t_{CLCL}$ | 0 | 24[2] | MHz |
| ALE Pulse Width | $t_{LHLL}$ | $1.5t_{CLCL}$ - 5 | | nS |
| Address Valid to ALE Low | $t_{AVLL}$ | $0.5t_{CLCL}$ - 5 | | nS |
| Address Hold After ALE Low | $t_{LLAX1}$ | $0.5t_{CLCL}$ - 5 | | nS |
| Address Hold After ALE Low for MOVX Write | $t_{LLAX2}$ | $0.5t_{CLCL}$ - 5 | | nS |
| ALE Low to Valid Instruction In | $t_{LLIV}$ | | $2.5t_{CLCL}$ - 20 | nS |
| ALE Low to PSEN Low | $t_{LLPL}$ | $0.5t_{CLCL}$ - 5 | | nS |
| $\overline{PSEN}$ Pulse Width | $t_{PLPH}$ | $2.0t_{CLCL}$ - 5 | | nS |
| $\overline{PSEN}$ Low to Valid Instruction In | $t_{PLIV}$ | | $2.0t_{CLCL}$ - 20 | nS |
| Input Instruction Hold After PSEN | $t_{PXIX}$ | 0 | | nS |
| Input Instruction Float After PSEN | $t_{PXIZ}$ | | $t_{CLCL}$ - 5 | nS |
| Port 0 Address to Valid Instr. In | $t_{AVIV1}$ | | $3.0t_{CLCL}$ - 20 | nS |

AC Specification, continued

| PARAMETER | SYMBOL | VARIABLE CLOCK MIN. | VARIABLE CLOCK MAX. | UNITS |
|---|---|---|---|---|
| Port 2 Address to Valid Instr. In | $t_{AVIV2}$ | | $3.5t_{CLCL}$ - 20 | nS |
| $\overline{PSEN}$ Low to Address Float | $t_{PLAZ}$ | 0 | | nS |
| Data Hold After Read | $t_{RHDX}$ | 0 | | nS |
| Data Float After Read | $t_{RHDZ}$ | | $t_{CLCL}$ - 5 | nS |
| $\overline{RD}$ Low to Address Float | $t_{RLAZ}$ | | $0.5t_{CLCL}$ - 5 | nS |

**Note:** 1. CPU executes the program stored in the internal APFlash at $V_{DD}$=5.0V
2. CPU executes the program stored in the external memory at $V_{DD}$=5.0V

### 24.3.3 MOVX Characteristics Using Stretch Memory Cycle

| PARAMETER | SYMBOL | VARIABLE CLOCK MIN. | VARIABLE CLOCK MAX. | UNITS | STRECH |
|---|---|---|---|---|---|
| Data Access ALE Pulse Width | $t_{LLHL2}$ | $1.5t_{CLCL}$ - 5<br>$2.0t_{CLCL}$ - 5 | | nS | $t_{MCS}$ = 0<br>$t_{MCS}$>0 |
| Address Hold After ALE Low for MOVX write | $t_{LLAX2}$ | $0.5t_{CLCL}$ - 5 | | nS | |
| $\overline{RD}$ Pulse Width | $t_{RLRH}$ | $2.0t_{CLCL}$ - 5<br>$t_{MCS}$ - 10 | | nS | $t_{MCS}$ = 0<br>$t_{MCS}$>0 |
| $\overline{WR}$ Pulse Width | $t_{WLWH}$ | $2.0t_{CLCL}$ - 5<br>$t_{MCS}$ - 10 | | nS | $t_{MCS}$ = 0<br>$t_{MCS}$>0 |
| $\overline{RD}$ Low to Valid Data In | $t_{RLDV}$ | | $2.0t_{CLCL}$ - 20<br>$t_{MCS}$ - 20 | nS | $t_{MCS}$ = 0<br>$t_{MCS}$>0 |
| Data Hold after Read | $t_{RHDX}$ | 0 | | nS | |
| Data Float after Read | $t_{RHDZ}$ | | $t_{CLCL}$ - 5<br>$2.0t_{CLCL}$ - 5 | nS | $t_{MCS}$ = 0<br>$t_{MCS}$>0 |
| ALE Low to Valid Data In | $t_{LLDV}$ | | $2.5t_{CLCL}$ - 5<br>$t_{MCS}$ + $2t_{CLCL}$ - 40 | nS | $t_{MCS}$ = 0<br>$t_{MCS}$>0 |
| Port 0 Address to Valid Data In | $t_{AVDV1}$ | | $3.0t_{CLCL}$ - 20<br>$2.0t_{CLCL}$ - 5 | nS | $t_{MCS}$ = 0<br>$t_{MCS}$>0 |
| ALE Low to $\overline{RD}$ or $\overline{WR}$ Low | $t_{LLWL}$ | $0.5t_{CLCL}$ - 5<br>$1.5t_{CLCL}$ - 5 | $0.5t_{CLCL}$ + 5<br>$1.5t_{CLCL}$ + 5 | nS | $t_{MCS}$ = 0<br>$t_{MCS}$>0 |
| Port 0 Address to $\overline{RD}$ or $\overline{WR}$ Low | $t_{AVWL}$ | $t_{CLCL}$ - 5<br>$2.0t_{CLCL}$ - 5 | | nS | $t_{MCS}$ = 0<br>$t_{MCS}$>0 |

MOVX Characteristics Using Stretch Memory Cycle, continud

| PARAMETER | SYMBOL | VARIABLE CLOCK MIN. | VARIABLE CLOCK MAX. | UNITS | STRECH |
|---|---|---|---|---|---|
| Port 2 Address to $\overline{RD}$ or $\overline{WR}$ Low | $t_{AVWL2}$ | $1.5t_{CLCL}$ - 5<br>$2.5t_{CLCL}$ - 5 | | nS | $t_{MCS}$ = 0<br>$t_{MCS}$>0 |
| Data Valid to WR Transition | $t_{QVWX}$ | -5<br>$1.0t_{CLCL}$ - 5 | | nS | $t_{MCS}$ = 0<br>$t_{MCS}$>0 |
| Data Hold after Write | $t_{WHQX}$ | $t_{CLCL}$ - 5<br>$2.0t_{CLCL}$ - 5 | | nS | $t_{MCS}$ = 0<br>$t_{MCS}$>0 |
| $\overline{RD}$ Low to Address Float | $t_{RLAZ}$ | | $0.5t_{CLCL}$ - 5 | nS | |
| $\overline{RD}$ or $\overline{WR}$ high to ALE high | $t_{WHLH}$ | 0<br>$1.0t_{CLCL}$ - 5 | 10<br>$1.0t_{CLCL}$ + 5 | nS | $t_{MCS}$ = 0<br>$t_{MCS}$>0 |

**Note:** $t_{MCS}$ is a time period related to the Stretch memory cycle selection. The following table shows the time period of $t_{MCS}$ for each selection of the Stretch value.

| M2 | M1 | M0 | MOVX CYCLES | $T_{MCS}$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 2 machine cycles | 0 |
| 0 | 0 | 1 | 3 machine cycles | 4 $t_{CLCL}$ |
| 0 | 1 | 0 | 4 machine cycles | 8 $t_{CLCL}$ |
| 0 | 1 | 1 | 5 machine cycles | 12 $t_{CLCL}$ |
| 1 | 0 | 0 | 6 machine cycles | 16 $t_{CLCL}$ |
| 1 | 0 | 1 | 7 machine cycles | 20 $t_{CLCL}$ |
| 1 | 1 | 0 | 8 machine cycles | 24 $t_{CLCL}$ |
| 1 | 1 | 1 | 9 machine cycles | 28 $t_{CLCL}$ |

Explanation of Logics Symbols

In order to maintain compatibility with the original 8051 family, this device specifies the same parameter for each device, using the same symbols. The explanation of the symbols is as follows.

| | | | |
|---|---|---|---|
| t | Time | A | Address |
| C | Clock | D | Input Data |
| H | Logic level high | L | Logic level low |
| I | Instruction | P | $\overline{PSEN}$ |
| Q | Output Data | R | $\overline{RD}$ signal |
| V | Valid | W | $\overline{WR}$ signal |
| X | No longer a valid state | Z | Tri-state |

## 24.4 The ADC Converter DC ELECTRICAL CHARACTERISTICS

($V_{DD}$–$V_{SS}$ = 3.0~5V±10%, $T_A$ = -40~85°C, Fosc = 20MHz, unless otherwise specified.)

| PARAMETER | SYMBOL | SPECIFICATION | | | |
|---|---|---|---|---|---|
| | | MIN. | TYP. | MAX. | UNIT |
| Analog input | AVin | $V_{SS}$-0.2 | | $V_{DD}$+0.2 | V |
| ADC clock | ADCCLK | 200KHz | - | 5MHz | Hz |
| Conversion time | $t_C$ | | $52t_{ADC}$[1] | | us |
| Differential non-linearity | DNL | -1 | - | +1 | LSB |
| Integral non-linearity | INL | -2 | - | +2 | LSB |
| Offset error | Ofe | -1 | - | +1 | LSB |
| Gain error | Ge | -1 | - | +1 | % |
| Absolute voltage error | Ae | -3 | - | +3 | LSB |

**Notes:**1. $t_{ADC}$: The period time of ADC input clock.

## 24.5 I2C Bus Timing Characteristics

| PARAMETER | SYMBOL | STANDARD MODE I2C BUS | | UNIT |
|---|---|---|---|---|
| | | MIN. | MAX. | |
| SCL clock frequency | $f_{SCL}$ | 0 | 100 | kHz |
| bus free time between a STOP and START condition | $t_{BUF}$ | 4.7 | - | uS |
| Hold time (repeated) START condition. After this period, the first clock pulse is generated | $t_{Hd;STA}$ | 4.0 | - | uS |
| Low period of the SCL clock | $t_{LOW}$ | 4.7 | - | uS |
| HIGH period of the SCL clock | $t_{HIGH}$ | 4.0 | - | uS |
| Set-up time for a repeated START condition | $t_{SU;STA}$ | 4.7 | - | uS |
| Data hold time | $t_{HD;DAT}$ | 5.0 | - | uS |
| Data set-up time | $t_{SU;DAT}$ | 250 | - | nS |
| Rise time of both SDA and SCL signals | $t_r$ | - | 1000 | nS |
| Fall time of both SDA and SCL signals | $t_f$ | - | 300 | nS |
| Set-up time for STOP condition | $t_{SU;STO}$ | 4.0 | - | uS |
| Capacitive load for each bus line | $C_b$ | - | 400 | pF |

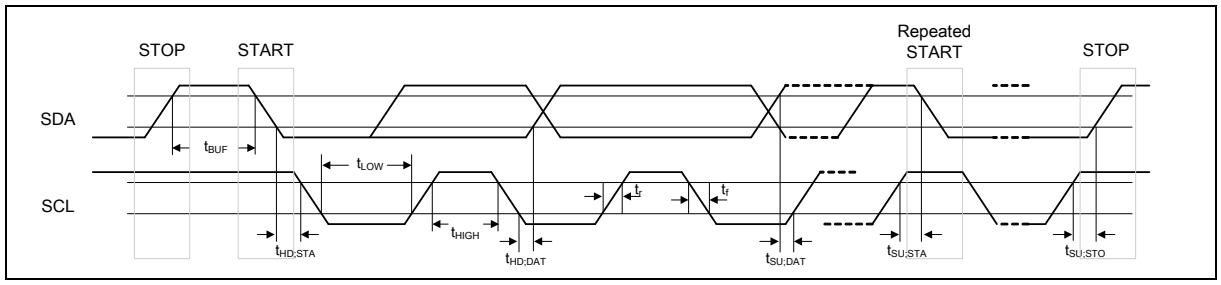
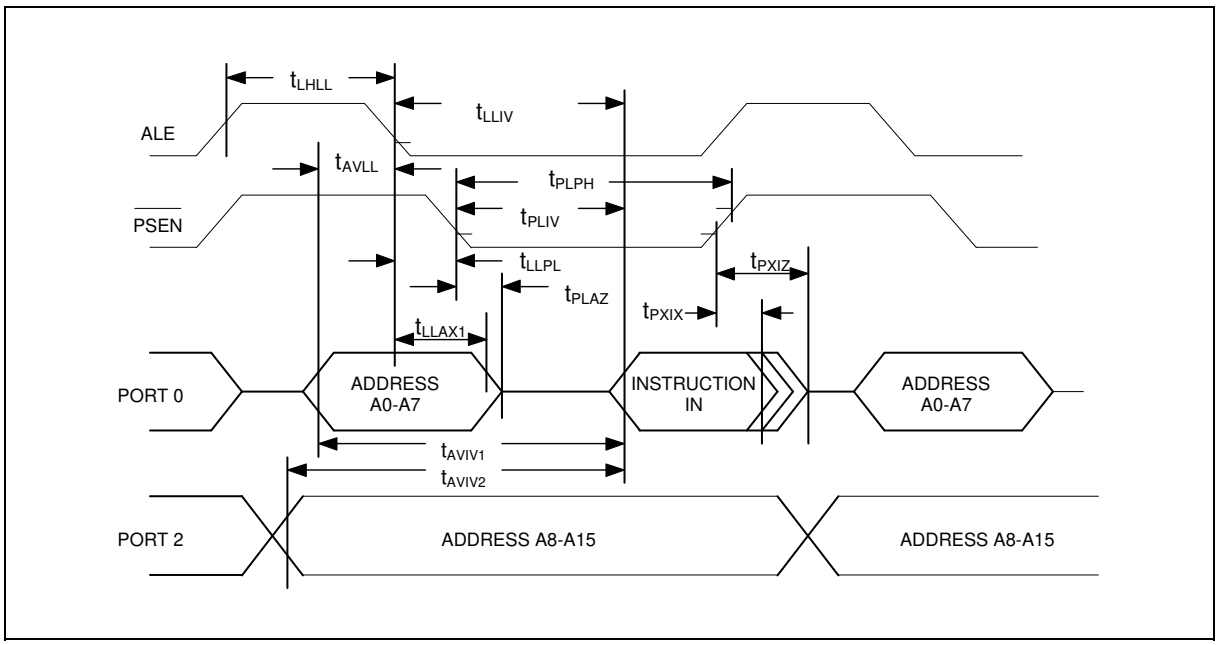
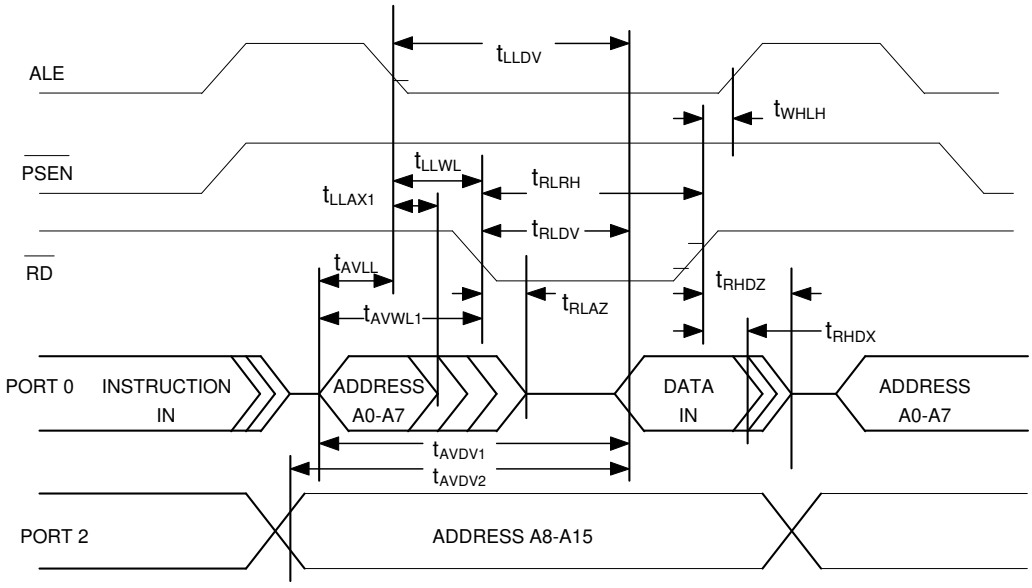



Figure 24-1: I2C Bus Timing

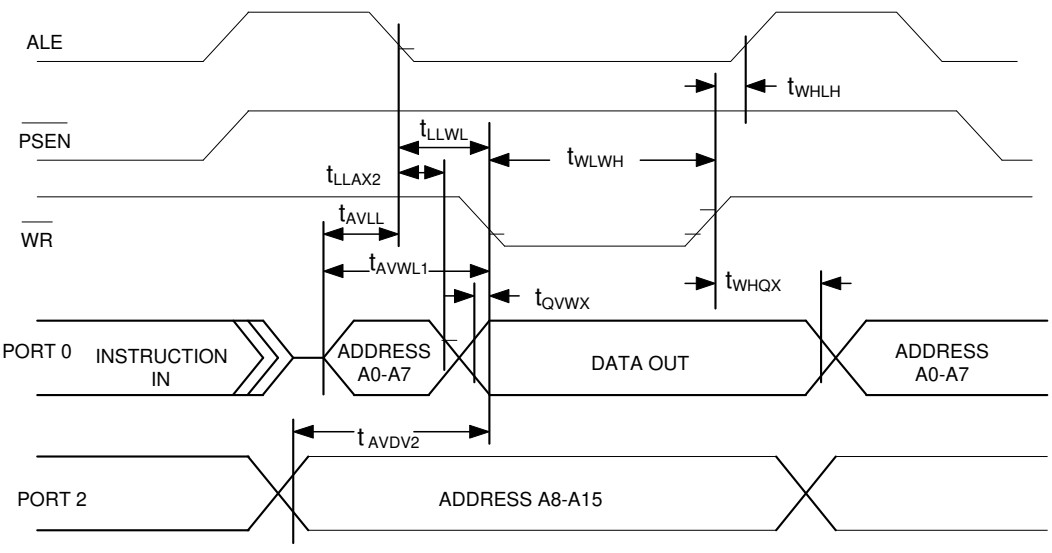
## 24.6 Program Memory Read Cycle



## 24.7 Data Memory Read Cycle




## 24.8 Data Memory Write Cycle



# 25. TYPICAL APPLICATION CIRCUITS

## 25.1 Expanded External Program Memory and Crystal

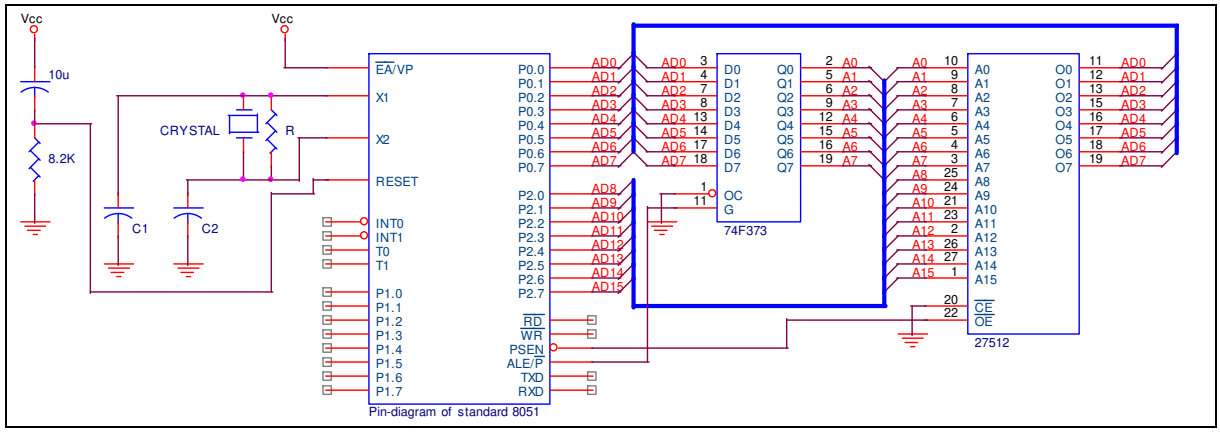



Figure 25-1: Typical External Program Memory and Crystal connections

| CRYSTAL | C1 | C2 | R |
|---|---|---|---|
| 16 MHz | 0P~20P | 0P~20P | - |
| 24 MHz | 0P~12P | 0P~12P | - |
| 33 MHz | 10P | 10P | 10K~1K |
| 40 MHz | 1P | 1P | 10K~1K |

The above table shows the reference values for crystal applications.

**Note:** C1, C2, R components refer to Figure above.

## 25.2 Expanded External Data Memory and Oscillator

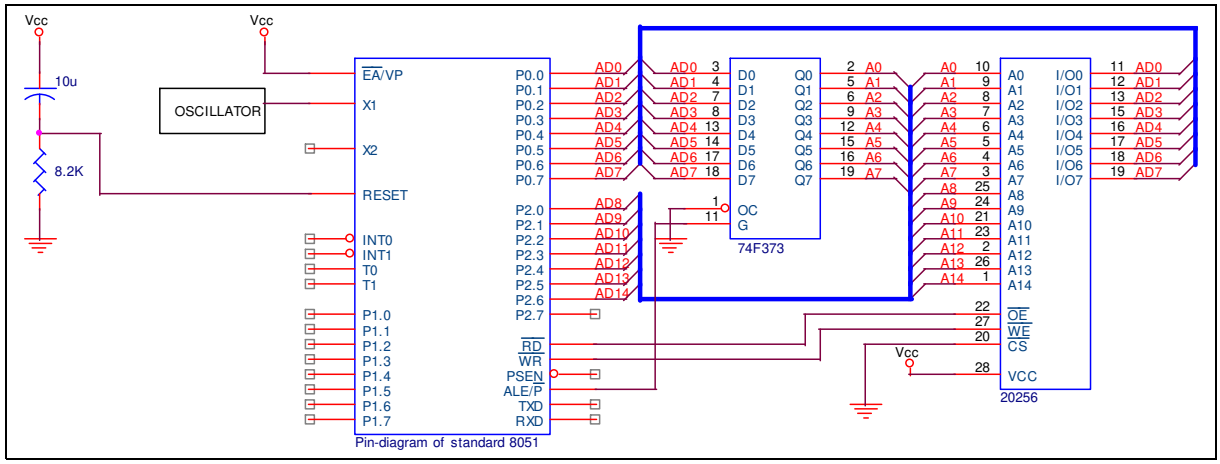
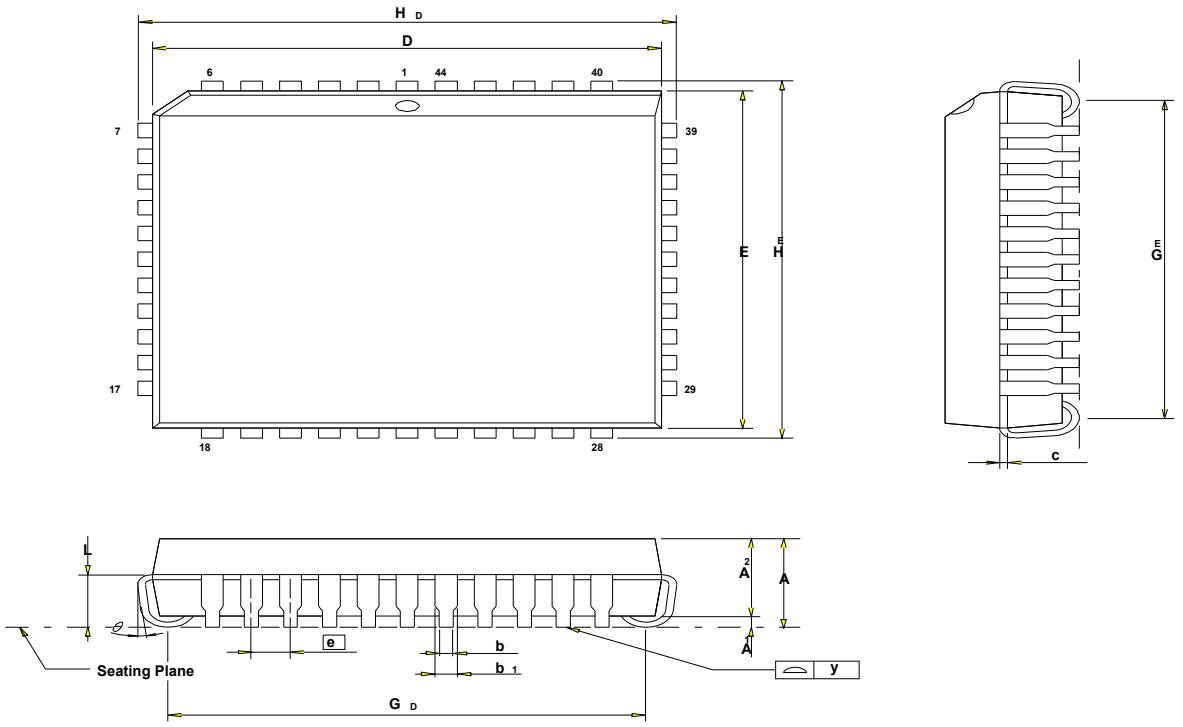



Figure 25-2: Typical External Data Memory and Oscillator connections

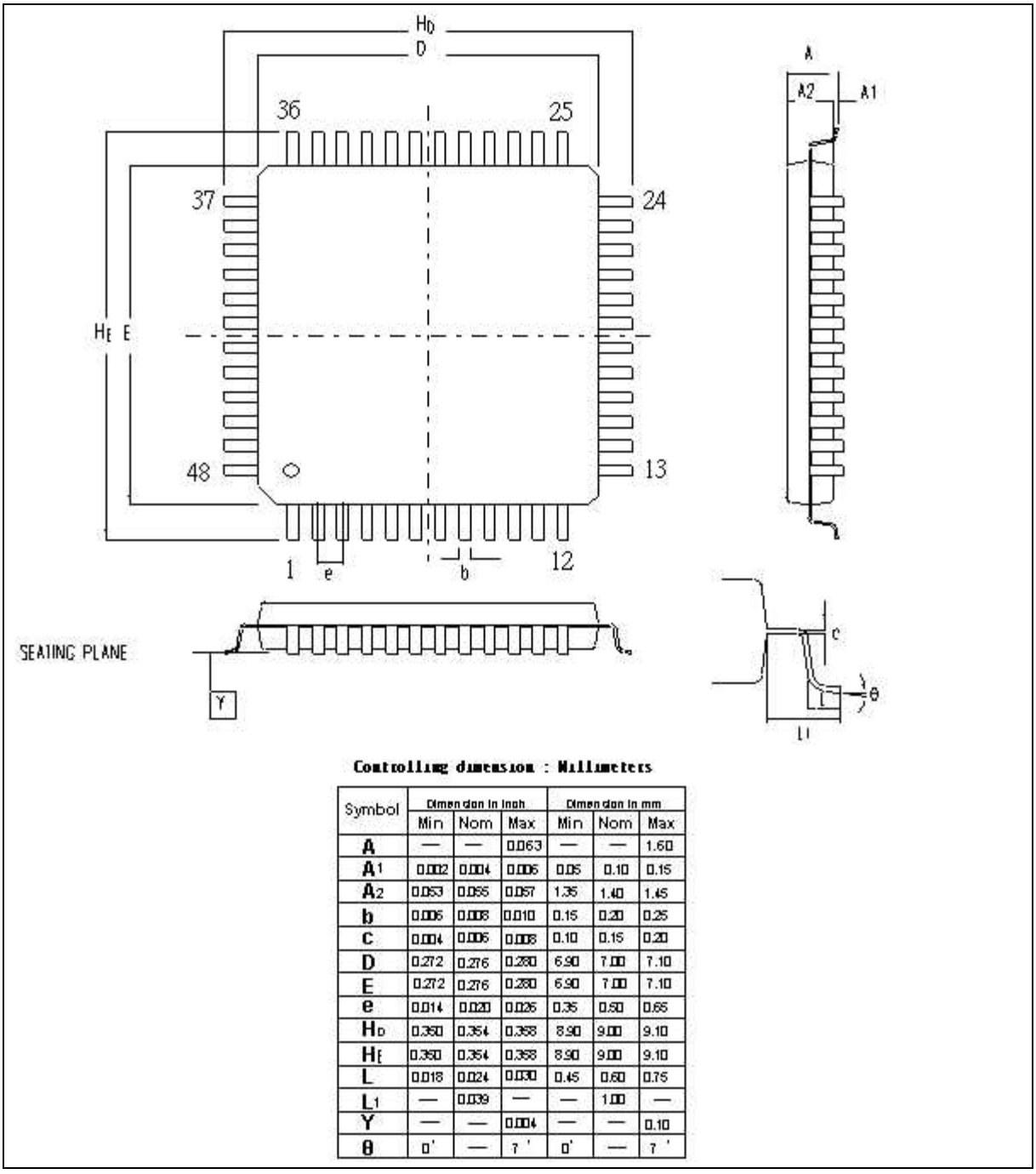
## 26. PACKAGE DIMENSION

### 26.1 44L PLCC




| Symbol | Dimension in inch | | | Dimension in mm | | |
|---|---|---|---|---|---|---|
| | Min | Nom | Max | Min | Nom | Max |
| A | —— | —— | 0.185 | —— | —— | 4.70 |
| $A_1$ | 0.020 | —— | —— | 0.51 | —— | —— |
| $A_2$ | 0.145 | 0.150 | 0.155 | 3.68 | 3.81 | 3.94 |
| $b_1$ | 0.026 | 0.028 | 0.032 | 0.66 | 0.71 | 0.81 |
| b | 0.016 | 0.018 | 0.022 | 0.41 | 0.46 | 0.56 |
| c | 0.008 | 0.010 | 0.014 | 0.20 | 0.25 | 0.36 |
| D | 0.648 | 0.653 | 0.658 | 16.46 | 16.59 | 16.71 |
| E | 0.648 | 0.653 | 0.658 | 16.46 | 16.59 | 16.71 |
| e | 0.050 BSC | | | 1.27 BSC | | |
| $G_D$ | 0.590 | 0.610 | 0.630 | 14.99 | 15.49 | 16.00 |
| $G_E$ | 0.590 | 0.610 | 0.630 | 14.99 | 15.49 | 16.00 |
| $H_D$ | 0.680 | 0.690 | 0.700 | 17.27 | 17.53 | 17.78 |
| $H_E$ | 0.680 | 0.690 | 0.700 | 17.27 | 17.53 | 17.78 |
| L | 0.090 | 0.100 | 0.110 | 2.29 | 2.54 | 2.79 |
| y | —— | —— | 0.004 | —— | —— | 0.10 |

## 26.2 48L LQFP (7x7x1.4mm footprint 2.0mm)




Controlling dimension : Millimeters

| Symbol | Dimension in inch | | | Dimension in mm | | |
|---|---|---|---|---|---|---|
| | Min | Nom | Max | Min | Nom | Max |
| A | — | — | 0.063 | — | — | 1.60 |
| A1 | 0.002 | 0.004 | 0.006 | 0.05 | 0.10 | 0.15 |
| A2 | 0.053 | 0.055 | 0.057 | 1.35 | 1.40 | 1.45 |
| b | 0.006 | 0.008 | 0.010 | 0.15 | 0.20 | 0.25 |
| c | 0.004 | 0.006 | 0.008 | 0.10 | 0.15 | 0.20 |
| D | 0.272 | 0.276 | 0.280 | 6.90 | 7.00 | 7.10 |
| E | 0.272 | 0.276 | 0.280 | 6.90 | 7.00 | 7.10 |
| e | 0.014 | 0.020 | 0.026 | 0.35 | 0.50 | 0.65 |
| HD | 0.350 | 0.354 | 0.358 | 8.90 | 9.00 | 9.10 |
| HE | 0.350 | 0.354 | 0.358 | 8.90 | 9.00 | 9.10 |
| L | 0.018 | 0.024 | 0.030 | 0.45 | 0.60 | 0.75 |
| L1 | — | 0.039 | — | — | 1.00 | — |
| Y | — | — | 0.004 | — | — | 0.10 |
| θ | 0° | — | 7° | 0° | — | 7° |

## 27. APPLICATION NOTE

### In-system Programming Software Examples

This application note illustrates the in-system programmability of the Winbond W79E22X SERIES Flash EPROM microcontroller. In this example, microcontroller will boot from APFlash bank and waiting for a key to enter in-system programming mode for re-programming the contents of 64 KB APFlash. While entering in-system programming mode, microcontroller executes the loader program in 4KB LDFlash bank. The loader program erases the 64 KB APFlash then reads the new code data from external SRAM buffer (or through other interfaces) to update the APFlash.

If the customer uses the reboot mode to update his program, please enable this b3 or b4 of security bits from the writer. Please refer security bits for detail description.

**EXAMPLE 1:**

```
;*****************************************************************************************************************
;* Example of APFlash program: Program will scan the P1.0. If P1.0 = 0, enters in-system
;* programming mode for updating the content of APFlash code else executes the current ROM code.
;* XTAL = 24 MHz
;*****************************************************************************************************************
.chip 8052
.RAMCHK OFF
.symbols

CHPCON      EQU         9FH
TA          EQU         C7H
SFRAL       EQU         ACH
SFRAH       EQU         ADH
SFRFD       EQU         AEH
SFRCN       EQU         AFH

ORG    0H
LJMP   100H                          ; JUMP TO MAIN PROGRAM
   ;*************************************************************************
   ;*  TIMER0 SERVICE VECTOR ORG = 000BH
   ;*************************************************************************
ORG    00BH
CLR    TR0                           ; TR0 = 0, STOP TIMER0
MOV    TL0, R6
MOV    TH0,R7
RETI
   ;*************************************************************************
   ;*  APFlash MAIN PROGRAM
   ;*************************************************************************
ORG    100H
```

```
MAIN_APFlash:
MOV A, P1                          ; SCAN P1.0
ANL A, #01H
CJNE A, #01H, PROGRAM_APFlash      ; IF P1.0 = 0, ENTER IN-SYSTEM PROGRAMMING MODE
JMP NORMAL_MODE

PROGRAM_64:
        MOV TA, #AAH               ; CHPCON register is written protect by TA register.
        MOV TA, #55H
        MOV CHPCON, #03H           ; CHPCON = 03H, ENTER IN-SYSTEM PROGRAMMING
        MODE
        MOV SFRCN, #0H
        MOV TCON, #00H             ; TR = 0 TIMER0 STOP
        MOV IP, #00H               ; IP = 00H
        MOV IE, #82H               ; TIMER0 INTERRUPT ENABLE FOR WAKE-UP FROM
IDLE MODE
        MOV R6, #F0H               ; TL0 = F0H
        MOV R7, #FFH               ; TH0 = FFH
        MOV TL0, R6
        MOV TH0, R7
        MOV TMOD, #01H             ; TMOD = 01H, SET TIMER0 A 16-BIT TIMER
        MOV TCON, #10H             ; TCON = 10H, TR0 = 1, GO
        MOV PCON, #01H             ; ENTER IDLE MODE FOR LAUNCHING THE IN-SYSTEM
PROGRAMMING
;************** ********************************************************************
;* Normal mode APFlashB APFlash program: depending user's application
;*************************************************************************************
NORMAL_MODE:
                                   .
                                                ; User's application program
                                   .
                                   .
                                   .
```

**EXAMPLE 2:**

```
;*************************************************************************************************************************
;* Example of 4KB LDFlash program: This loader program will erase the APFlashB APFlash first, then
reads the new ;* code from external SRAM and program them into APFlashB APFlash bank. XTAL =
24 MHz
;*************************************************************************************************************************
.chip 8052
.RAMCHK OFF
.symbols

CHPCON      EQU         9FH
```

```
TA              EQU             C7H
SFRAL           EQU             ACH
SFRAH           EQU             ADH
SFRFD           EQU             AEH
SFRCN           EQU             AFH

ORG 000H
LJMP 100H                               ; JUMP TO MAIN PROGRAM
;*****************************************************************************
;* 1. TIMER0 SERVICE VECTOR ORG = 0BH
;*****************************************************************************
ORG 000B

CLR   TR0                               ; TR0 = 0, STOP TIMER0
MOV TL0, R6
MOV TH0, R7
RETI
;*****************************************************************************
;*  4KB LDFlash MAIN PROGRAM
;*****************************************************************************
ORG 100H

MAIN_4K:
MOV TA, #AAH
MOV TA, #55H
MOV CHPCON, #03H                        ; CHPCON = 03H, ENABLE IN-SYSTEM PROGRAMMING.

MOV SFRCN, #0H
MOV TCON, #00H                          ; TCON = 00H, TR = 0 TIMER0 STOP
MOV TMOD, #01H                          ; TMOD = 01H, SET TIMER0 A 16BIT TIMER
MOV IP, #00H                            ; IP = 00H
MOV IE, #82H                            ; IE = 82H, TIMER0 INTERRUPT ENABLED
MOV R6, #F0H
MOV R7, #FFH
MOV TL0, R6
MOV TH0, R7
MOV TCON, #10H                          ; TCON = 10H, TR0 = 1, GO
MOV PCON, #01H                          ; ENTER IDLE MODE

UPDATE_APFlash:
MOV TCON, #00H                          ; TCON = 00H, TR = 0 TIM0 STOP
MOV IP, #00H                            ; IP = 00H
MOV IE, #82H                            ; IE = 82H, TIMER0 INTERRUPT ENABLED
```

```
MOV TMOD, #01H                  ; TMOD = 01H, MODE1
MOV  R6,#D0H                    ; SET WAKE-UP TIME FOR ERASE OPERATION, ABOUT
15 ms DEPENDING ON USER'S SYSTEM CLOCK RATE.
MOV R7, #8AH
MOV TL0, R6
MOV TH0, R7

ERASE_P_4K:
MOV SFRCN, #22H                 ; SFRCN = 22H, ERASE APFlash APFlash0
                                ; SFRCN = A2H, ERASE APFlash1
MOV TCON, #10H                  ; TCON = 10H, TR0 = 1, GO
MOV PCON, #01H                  ; ENTER IDLE MODE (FOR ERASE OPERATION)

;*********************************************************************
;* BLANK CHECK
;*********************************************************************
MOV SFRCN, #0H                  ; SFRCN = 00H, READ APFlashB APFlash0
                                ; SFRCN = 80H, READ APFlashB APFlash1
MOV SFRAH, #0H                  ; START ADDRESS = 0H
MOV SFRAL, #0H
MOV R6, #FDH                    ; SET TIMER FOR READ OPERATION, ABOUT 1.5 μS.
MOV R7, #FFH
MOV TL0, R6
MOV TH0, R7

blank_check_loop:
SETB TR0                        ; Enable TIMER 0
MOV PCON, #01H                  ; Enter idle mode
MOV A, SFRFD                    ; Read one byte
CJNE A, #FFH, blank_check_error
INC SFRAL                       ; Next address
MOV A, SFRAL
JNZ   blank_check_loop
INC  SFRAH
MOV A, SFRAH
CJNE A, #0H, blank_check_loop   ; End address = FFFFH
JMP PROGRAM_APFlashROM

blank_check_error:
JMP $

;*****************************************************************************
;* RE-PROGRAMMING APFlashB APFlash BANK
```

```
;*******************************************************************************
PROGRAM_APFlashROM:
MOV R2, #00H                    ; Target low byte address
MOV R1, #00H                    ; TARGET HIGH BYTE ADDRESS
MOV DPTR, #0H
MOV SFRAH, R1                   ; SFRAH, Target high address
MOV SFRCN, #21H                 ; SFRCN = 21H, PROGRAM APFlash0
                                ; SFRCN = A1H, PROGRAM APFlash1
MOV R6, #9CH                    ; SET TIMER FOR PROGRAMMING, ABOUT 50 μS.
MOV R7, #FFH
MOV TL0, R6
MOV TH0, R7

PROG_D_APFlash:
        MOV SFRAL, R2           ; SFRAL = LOW BYTE ADDRESS
        CALL GET_BYTE_FROM_PC_TO_ACC    ; THIS PROGRAM IS BASED ON USER'S
CIRCUIT.
        MOVX @DPTR, A           ; SAVE DATA INTO SRAM TO VERIFY CODE.
        MOV SFRFD, A            ; SFRFD = data IN
        MOV TCON, #10H          ; TCON = 10H, TR0 = 1,GO
        MOV PCON, #01H          ; ENTER IDLE MODE (PRORGAMMING)
        INC    DPTR
        INC R2
        CJNE R2, #0H, PROG_D_APFlash
        INC R1
        MOV SFRAH, R1
        CJNE R1, #0H, PROG_D_APFlash

;*******************************************************************************
; * VERIFY APFlashB APFlash BANK
;*******************************************************************************
        MOV R4, #03H            ; ERROR COUNTER
        MOV R6, #FDH            ; SET TIMER FOR READ VERIFY, ABOUT 1.5 μS.
        MOV R7, #FFH
        MOV TL0, R6
        MOV TH0, R7
        MOV DPTR, #0H           ; The start address of sample code
        MOV R2, #0H             ; Target low byte address
        MOV R1, #0H             ; Target high byte address
        MOV SFRAH, R1           ; SFRAH, Target high address
        MOV SFRCN, #00H         ; SFRCN = 00H, Read APFlash0
                                ; SFRCN = 80H, Read APFlash1
```

```
READ_VERIFY_APFlash:
        MOV SFRAL,R2                ; SFRAL = LOW ADDRESS
        MOV TCON,#10H               ; TCON = 10H, TR0 = 1,GO
        MOV PCON,#01H
        INC R2
        MOVX A,@DPTR
        INC DPTR
        CJNE A,SFRFD,ERROR_APFlash
        CJNE R2,#0H,READ_VERIFY_APFlash
        INC R1
        MOV SFRAH,R1
        CJNE R1,#0H,READ_VERIFY_APFlash

;*****************************************************************************
;* PROGRAMMING COMPLETLY, SOFTWARE RESET CPU
;*****************************************************************************
        MOV TA, #AAH
        MOV TA, #55H
        MOV CHPCON, #83H            ; SOFTWARE RESET. CPU will restart from APFlash0

ERROR_APFlash:
        DJNZ R4, UPDATE_APFlash     ; IF ERROR OCCURS, REPEAT 3 TIMES.
.                                   ; IN-SYST PROGRAMMING FAIL, USER'S PROCESS TO
DEAL WITH IT.
                        .
                        .
                        .
```

## 28. REVISION HISTORY

| REVISION | DATE | PAGE | DESCRIPTION |
|---|---|---|---|
| A1.0 | October 18, 2007 | - | Preliminary version initially issued |
| A1.1 | November 17, 2007 | 8,9 | Incorrect pin number format. Re-alignment. |
| | | 182 | Operating voltage for NVM program/erase min at 3.0V. |
| | | 7 | Added note for minimum NVM program/erase operating voltage. |
| | | 111 | Updated Figure 14-8. Changed label “B” to “C”. |
| A2.0 | December 11, 2007 | 8, 137, 138 | Removed INDX descriptions. |
| | | 99, 100 | Updated diagram for T2EX at P4.1 pin. |
| | | 111 | Updated diagram. Posc replaced with Fosc. |
| | | 182 | Revise the Operating temperature to (-40, +85) °C |
| | | 35 | Revise the content of UART mode select table. (SM0,SM1) is exchanged. |
| A3.0 | March 17, 2008 | 141<br>6, 7 | 1. Modified UART descriptions.<br>2. Add two parts of W79E226 series |
| A4.0 | April 15, 2008 | 68 | Modified CPHA descriptions. |
| A5.0 | April 28, 2008 | 24 | Revise the content of T3CON |
| A6.0 | August 29, 2008 | 6, 7 | 1. Revise part number W79E22XAFG to W79E22XALG |
| | | 18 | 2. Revise On-Chip RC Oscillator 6MHz freq. accuracy to +/- 50% |
| | | 192 | 3. Revise crystal application circuit’s R to 1K~10K ohm |
| A7.0 | December 9, 2008 | 129 | 1. Revise Figure 15-2: Timer3/Capture/Compare/Reload modules |
| A8.0 | March 13, 2009 | 184 | 1. Remove P0~P5 $V_{IL1}$ typical 0.8V item.<br>2. Remove P0~P5 $V_{IH1}$ typical 2.0V item |


### Important Notice

**Nuvoton products are not designed, intended, authorized or warranted for use as components in systems or equipment intended for surgical implantation, atomic energy control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, combustion control instruments, or for other applications intended to support or sustain life. Further more, Nuvoton products are not intended for applications wherein failure of Nuvoton products could result or lead to a situation wherein personal injury, death or severe property or environmental damage could occur.**

**Nuvoton customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Nuvoton for any damages resulting from such improper use or sales.**

Please note that all data and specifications are subject to change without notice.
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.