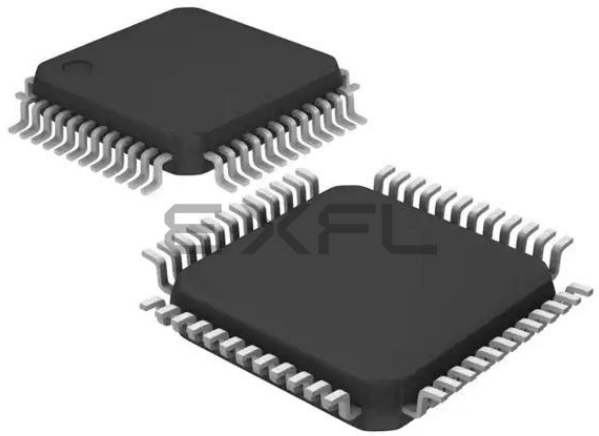


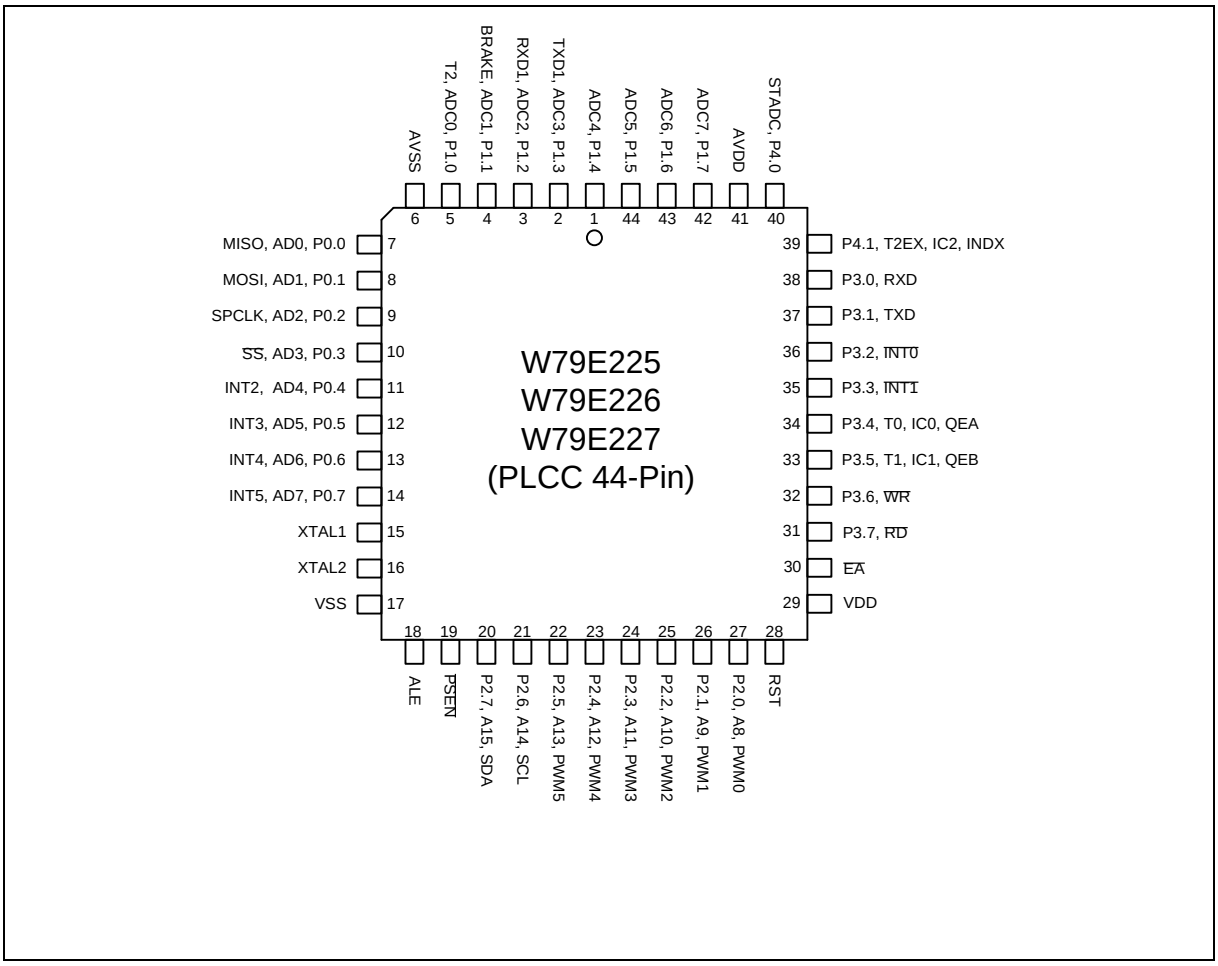
Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	8051/52
Core Size	8-Bit
Speed	40MHz
Connectivity	EBI/EMI, I²C, SPI, UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	38
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	2K x 8
RAM Size	2.25K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/nuvoton-technology-corporation-america/w79e226alg


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 P4.1, T2EX, IC2, INDX
13 P3.0, RXD
14 P3.1, TXD
15 P3.2, INT0
16 P3.3, INT1
17 P3.4, T0, IC0, QEA
18 P3.5, T1, IC1, QEB
19 P3.6, WR
20 P3.7, RD
21 EA
22 VDD
23 RST
24 P2.0, A8, PWM0
25 P2.1, A9, PWM1
26 P2.2, A10, PWM2
27 P2.3, A11, PWM3
28 P2.4, A12, PWM4
29 P2.5, A13, PWM5
30 P2.6, A14, SCL
31 P2.7, A15, SDA
32 PSEN
33 ALE
34 XTAL2
35 XTAL1
36 VSS
37 MISO, AD0, P0.0
38 MOSI, AD1, P0.1
39 SPCLK, AD2, P0.2
40 SS, AD3, P0.3
41 INT2, AD4, P0.4
42 INT3, AD5, P0.5
43 INT4, AD6, P0.6
44 INT5, AD7, P0.7

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.

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: 1: divide-by-4 clock. 0: divide-by-12 clock.
4	T1M	Timer 1 clock select: 1: divide-by-4 clock. 0: divide-by-12 clock.
3	T0M	Timer 0 clock select: 1: divide-by-4 clock. 0: divide-by-12 clock.
2	MD2	Stretch MOVX select bit 2: 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. (**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.

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. 1 = Enable. 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. 1 = Enable. 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. 1 = Enable. 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. 0 = Timer 3 as reload mode, TF3 indicates the overflow flag 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	PWMEOM	PWM Channel 0, 2 and 4 Output Mode. 0 = Disable PWM channels 0, 2 and 4 to pwm output pins. 1 = Enable PWM channels 0, 2 and 4 to pwm output pins.
6	PWMOOM	PWM Channel 1, 3 and 5 Output Mode. 0 = Disable PWM channels 1, 3 and 5 to pwm output pins. 1 = Enable PWM channels 1, 3 and 5 to pwm output pins.
5	PWM6OM	PWM Channel 6 Output Mode. 0 = Disable PWM channel 6 to pwm output pin. 1 = Enable PWM channel 6 to pwm output pin.
4	PWM7OM	PWM Channel 7 Output Mode. 0 = Disable PWM channel 7 to pwm output pin. 1 = Enable PWM channel 7 to pwm output pin.
3-1	-	Reserved.
0	BKF	The External Brake Pin Flag. 0 = The PWM is not brake. 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.

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. 0 = the clock frequency is coherent to the system clock Fosc. 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

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. 0 = No overrun. 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.

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

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).

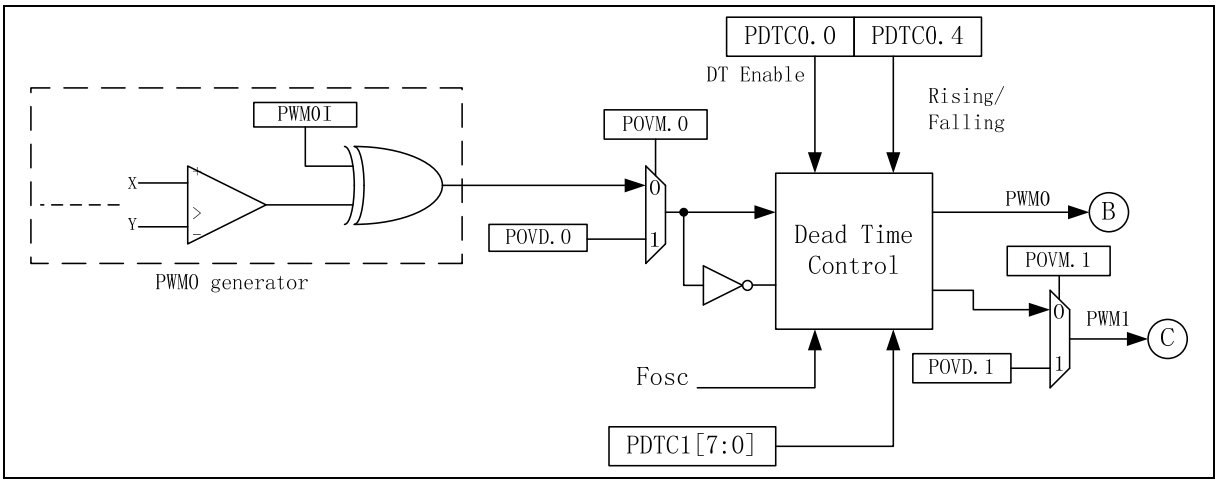


Figure 14-9: Complementary PWM with Dead-time and Override functions

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.

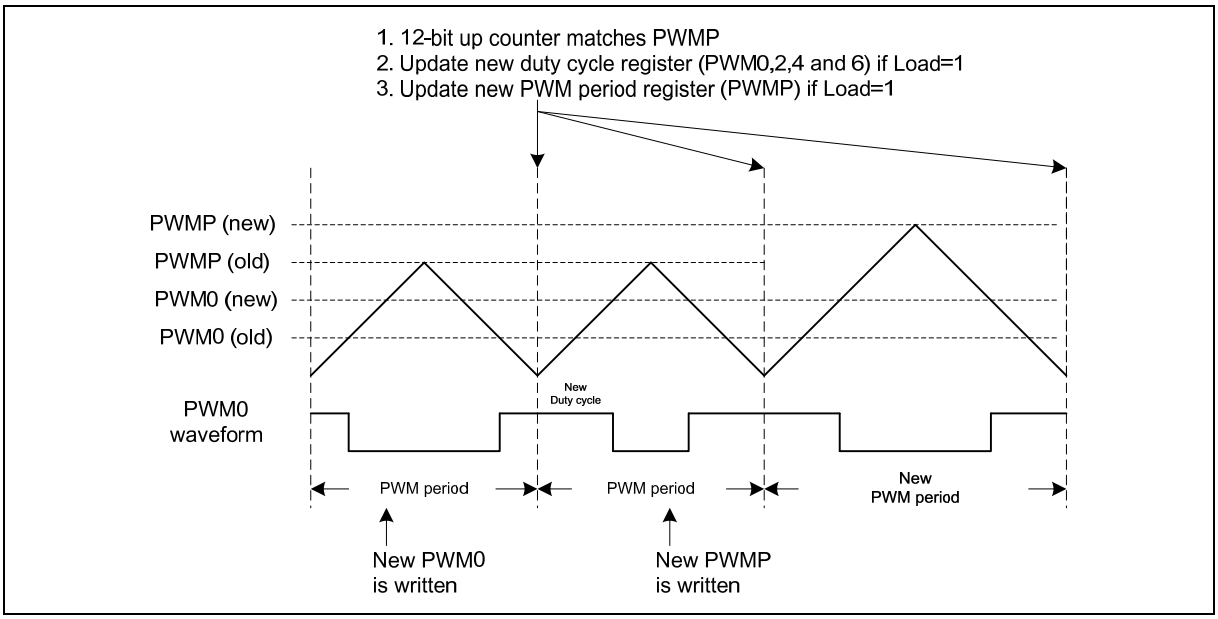


Figure 14-20: Center-Aligned Mode

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 3rd 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.

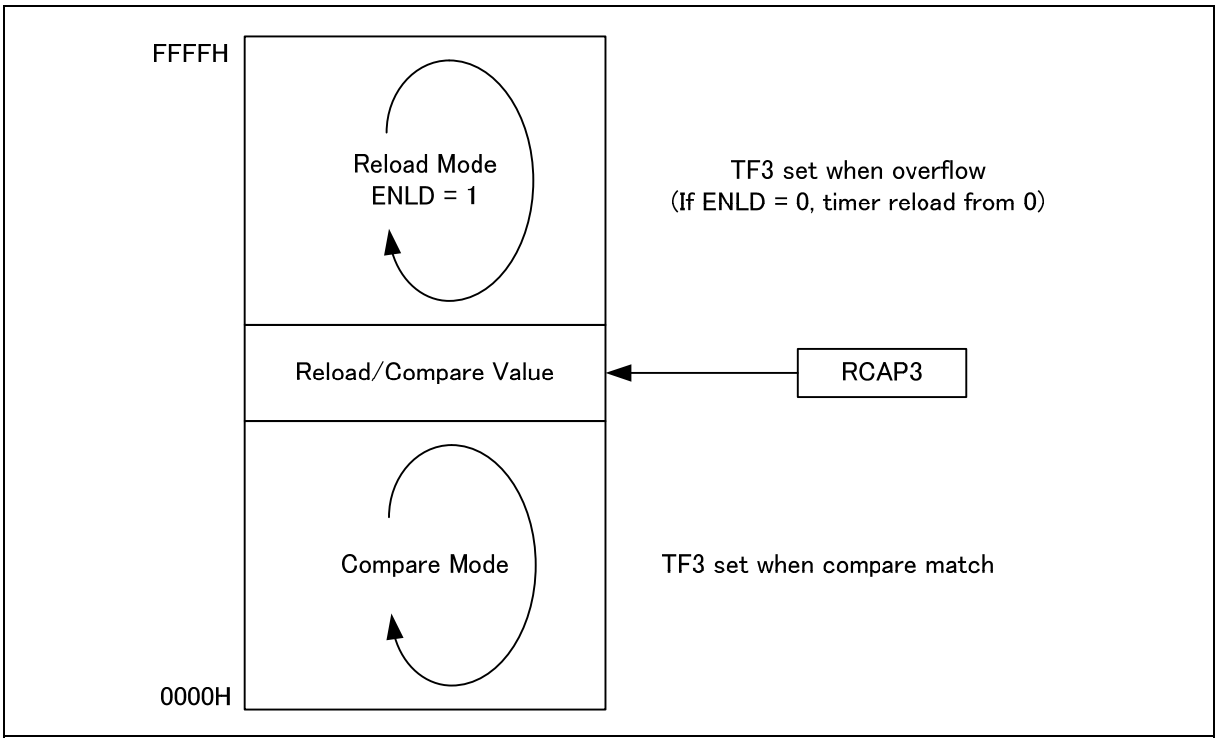


Figure 15-8: Compare/Reload Function

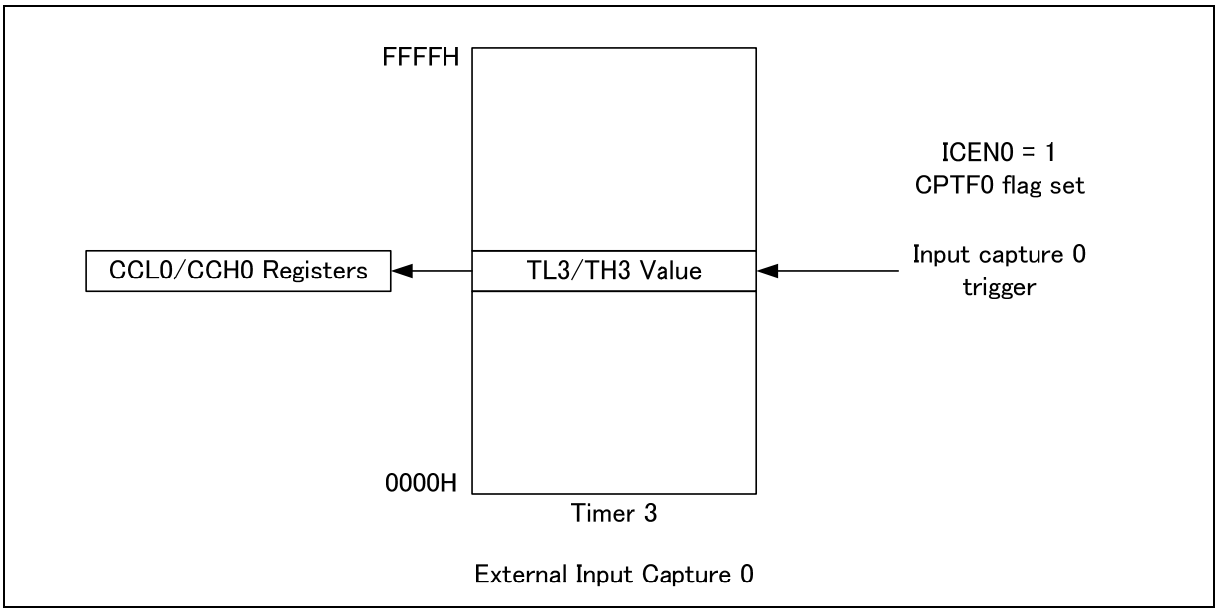


Figure 15-9: Input Capture 0 Triggers

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,

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.

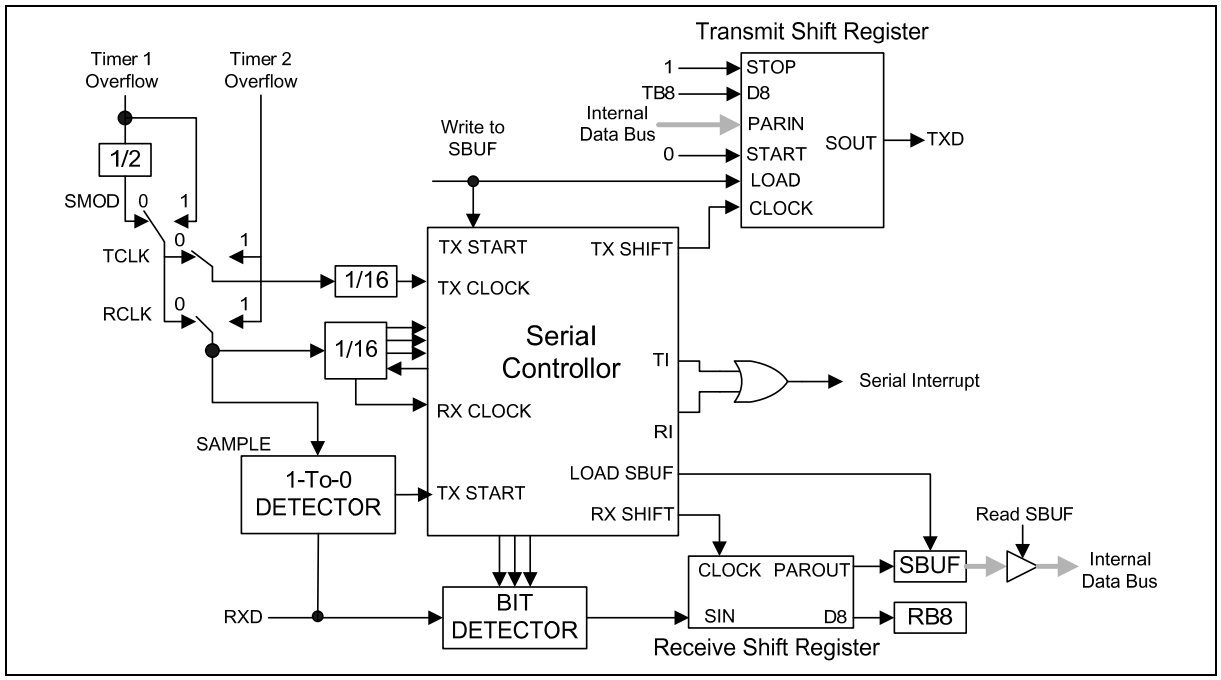


Figure 16-4 Serial Port Mode 3

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.

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.

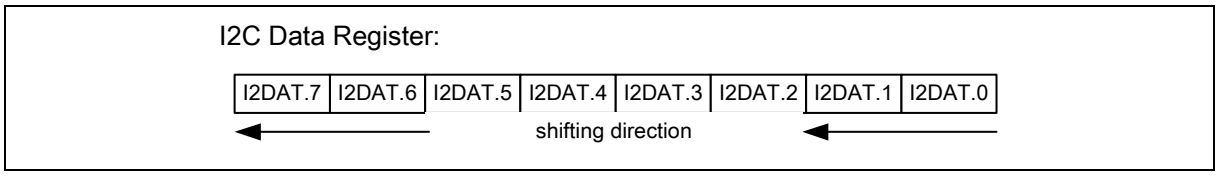


Figure 17-2: I2C Data Shift

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.

17.4.4 Slave/Receiver Mode

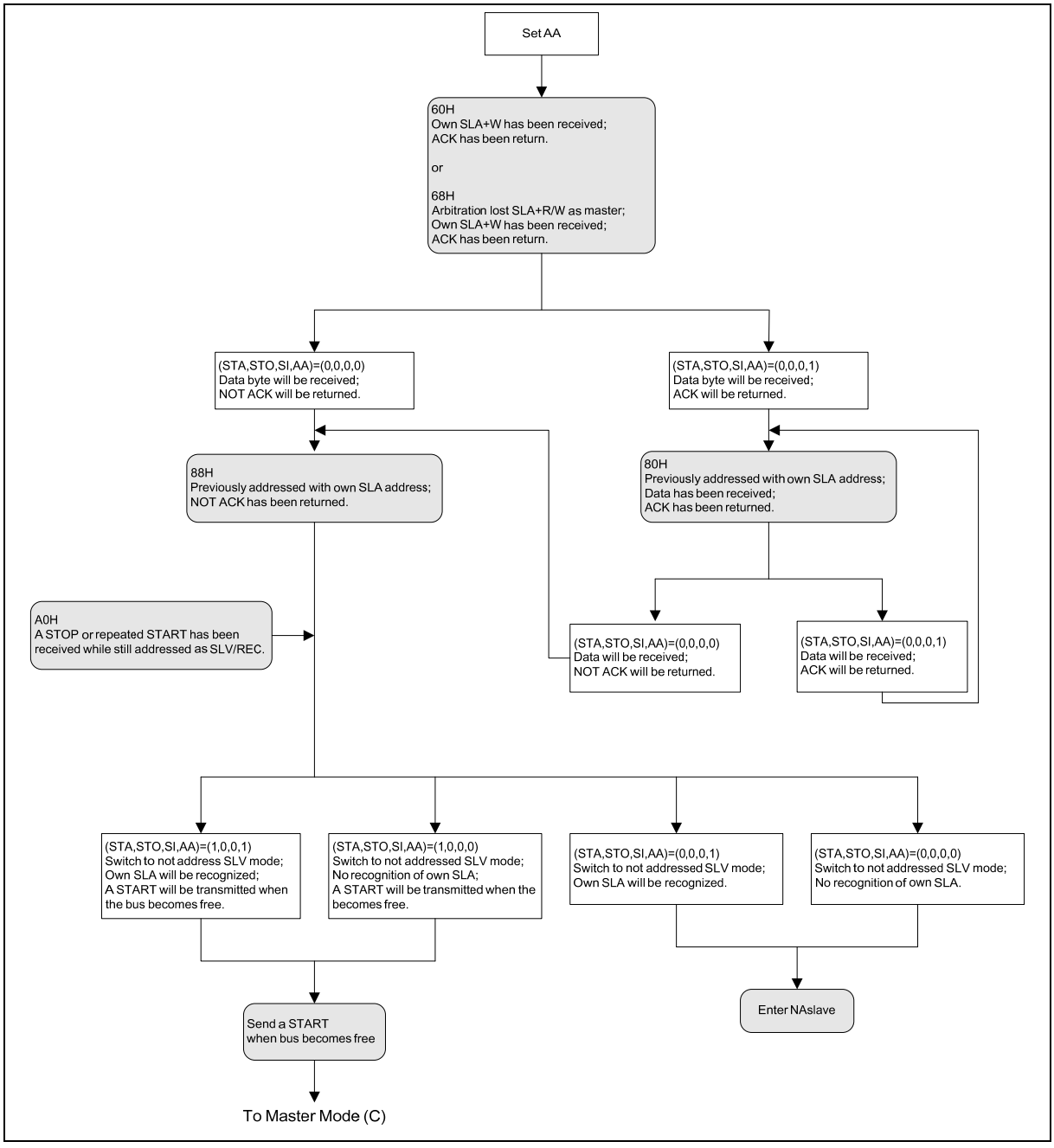


Figure 17-8: Slave Receiver Mode

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.

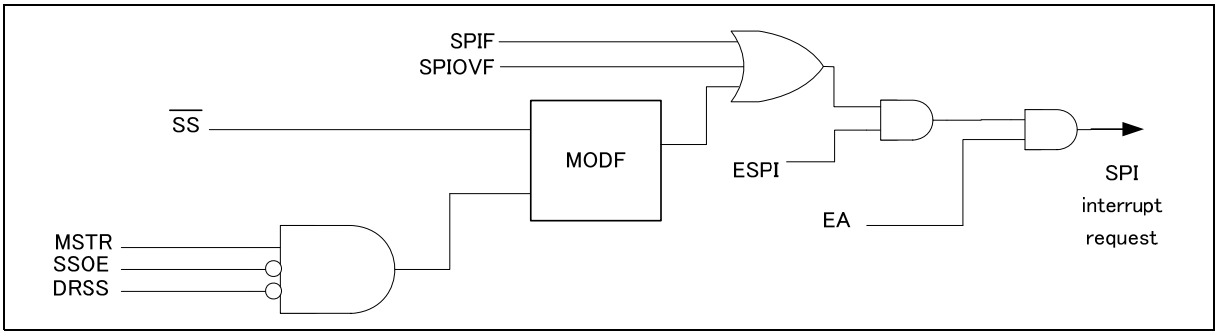


Figure 18-10: SPI interrupt request

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 Input: DRSS=1, SSOE=1
Slave	Output** during /SS = Low 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).

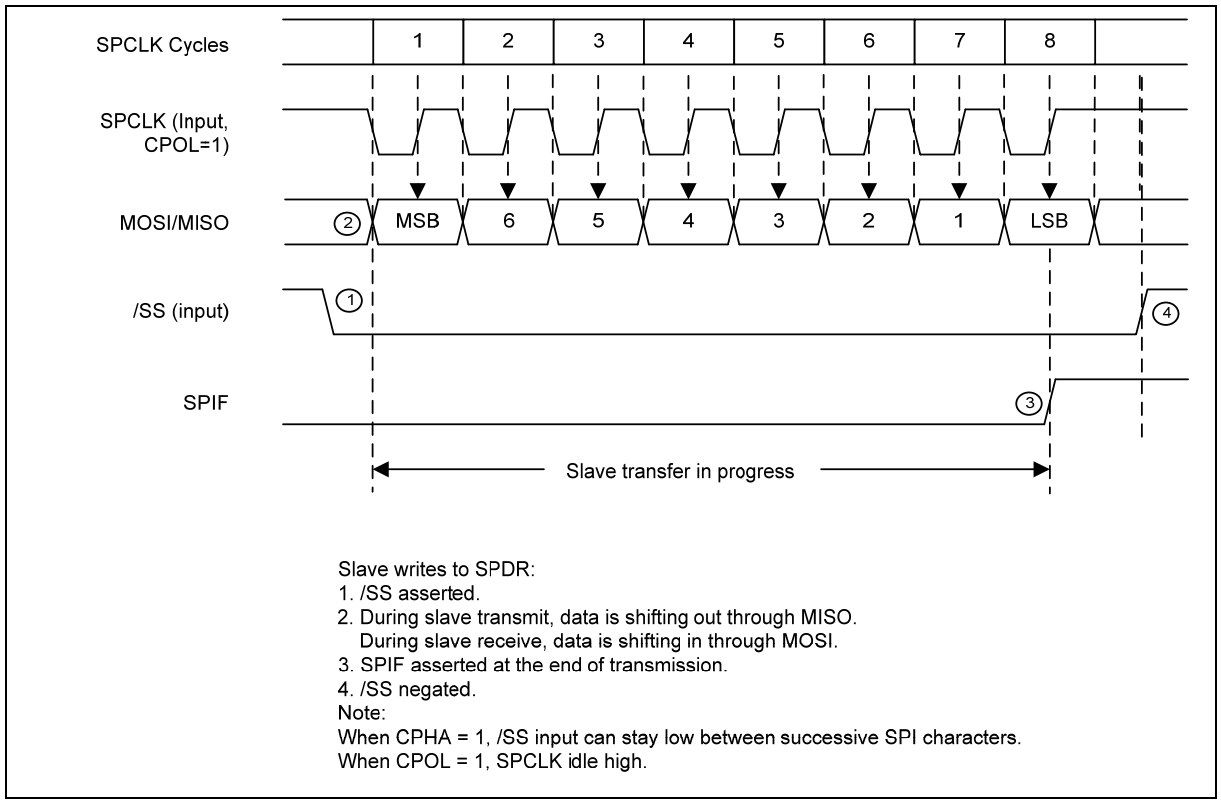


Figure 18-9 show the SPI transfer format, with different CPOL and CPHA. When CPHA = 0, data is sample on the first edge of SPCLK and when CPHA = 1 data is sample on the second edge of the SPCLK. Prior to changing CPOL setting, SPE must be disabled first.

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.

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.

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 $2.0t_{CLCL}$ - 5		nS	t_{MCS} = 0 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 t_{MCS} - 10		nS	t_{MCS} = 0 t_{MCS}>0
$\overline{WR}$ Pulse Width	t_{WLWH}	$2.0t_{CLCL}$ - 5 t_{MCS} - 10		nS	t_{MCS} = 0 t_{MCS}>0
$\overline{RD}$ Low to Valid Data In	t_{RLDV}		$2.0t_{CLCL}$ - 20 t_{MCS} - 20	nS	t_{MCS} = 0 t_{MCS}>0
Data Hold after Read	t_{RHDX}	0		nS	
Data Float after Read	t_{RHDZ}		t_{CLCL} - 5 $2.0t_{CLCL}$ - 5	nS	t_{MCS} = 0 t_{MCS}>0
ALE Low to Valid Data In	t_{LLDV}		$2.5t_{CLCL}$ - 5 t_{MCS} + $2t_{CLCL}$ - 40	nS	t_{MCS} = 0 t_{MCS}>0
Port 0 Address to Valid Data In	t_{AVDV1}		$3.0t_{CLCL}$ - 20 $2.0t_{CLCL}$ - 5	nS	t_{MCS} = 0 t_{MCS}>0
ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	t_{LLWL}	$0.5t_{CLCL}$ - 5 $1.5t_{CLCL}$ - 5	$0.5t_{CLCL}$ + 5 $1.5t_{CLCL}$ + 5	nS	t_{MCS} = 0 t_{MCS}>0
Port 0 Address to $\overline{RD}$ or $\overline{WR}$ Low	t_{AVWL}	t_{CLCL} - 5 $2.0t_{CLCL}$ - 5		nS	t_{MCS} = 0 t_{MCS}>0

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
```