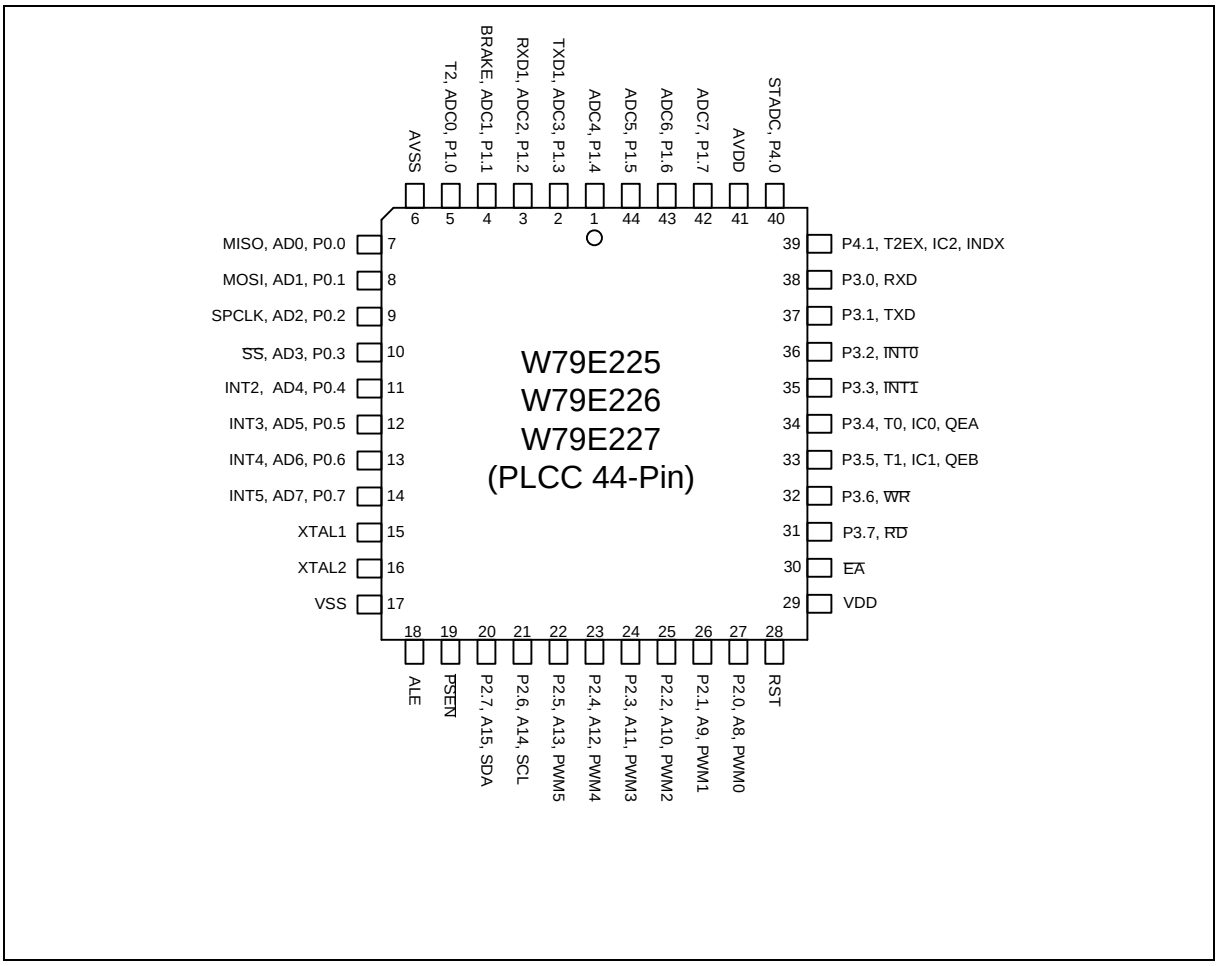
Welcome to E-XFL.COM

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	44-LCC (J-Lead)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/nuvoton-technology-corporation-america/w79e226apg


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

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
7-0	PWMxB	0 = The PWM0 output is low, when Brake is asserted. 1 = The PWM0 output is high, when Brake is asserted. **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

<table>
<tr><th>BIT</th><th>NAME</th><th>FUNCTION</th></tr>
<tr><td>7</td><td>ADCEN</td><td>Enable A/D Converter Function. Set ADCEN to logic high to enable ADC block.</td></tr>
<tr><td>6</td><td>-</td><td>Reserved.</td></tr>
<tr><td>5</td><td>ADCEX</td><td>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.</td></tr>
<tr><td>4</td><td>ADCI</td><td>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.</td></tr>
<tr><td>3</td><td>ADCS</td><td>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.
Notes:
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.
2. Software clearing of ADCS will abort conversion in progress.
3. ADC cannot start a new conversion while ADCS or ADCI is high.</td></tr>
<tr><td>2-0</td><td>AADR</td><td>Select and enable analog input channel from ADC0 to ADC7.
<table>
<tr><th>AADR[2:0]</th><th>ADC selected input</th><th>AADR[2:0]</th><th>ADC selected input</th></tr>
<tr><td>000</td><td>ADCCH0 (P1.0)</td><td>100</td><td>ADCCH4 (P1.4)</td></tr>
<tr><td>001</td><td>ADCCH1 (P1.1)</td><td>101</td><td>ADCCH5 (P1.5)</td></tr>
<tr><td>010</td><td>ADCCH2 (P1.2)</td><td>110</td><td>ADCCH6 (P1.6)</td></tr>
<tr><td>011</td><td>ADCCH3 (P1.3)</td><td>111</td><td>ADCCH7 (P1.7)</td></tr>
</table></td></tr>
</table>

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.

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.

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

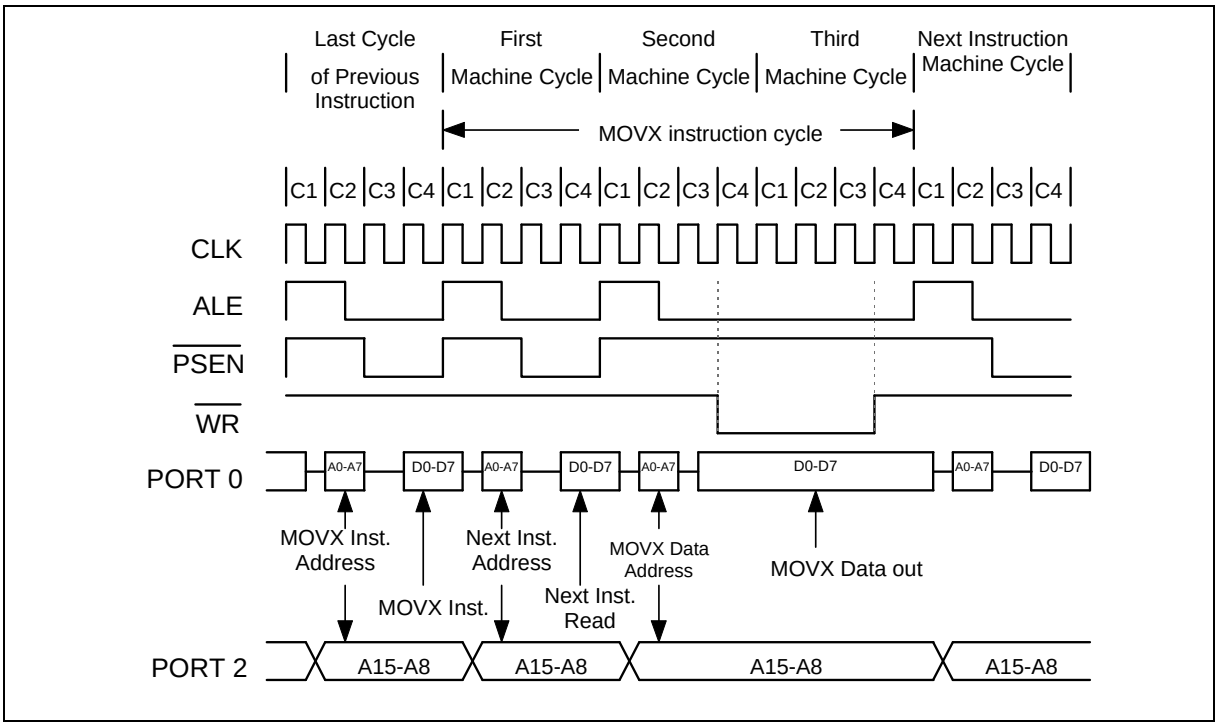


Figure 8-7: Data Memory Write with Stretch Value = 1

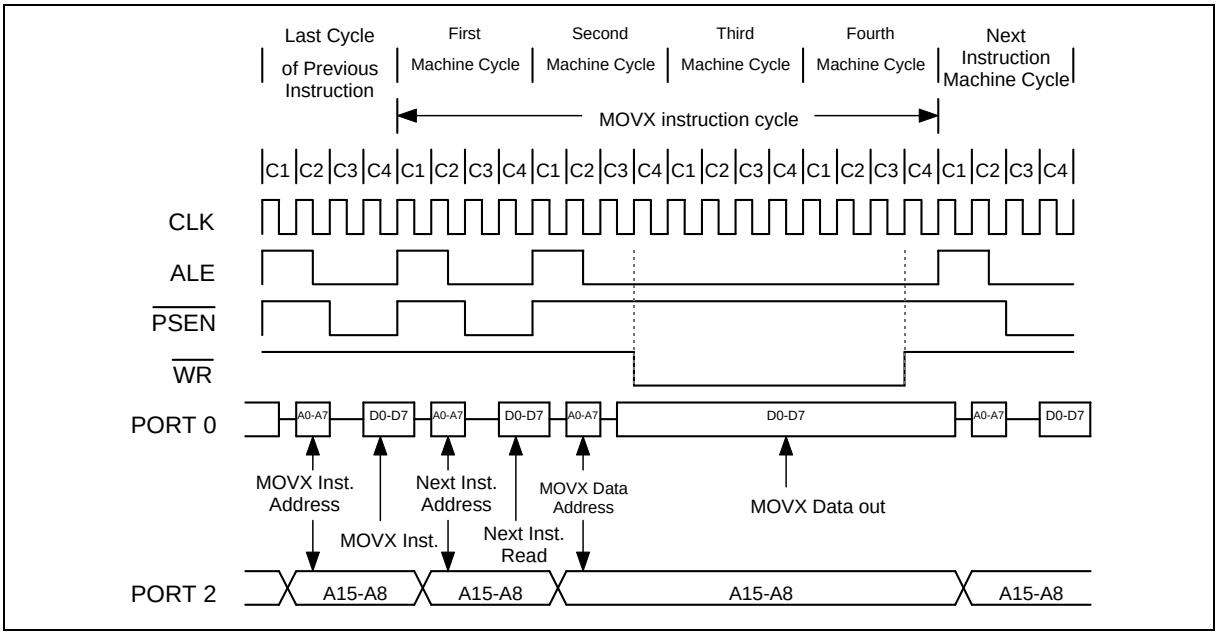


Figure 8-8: Data Memory Write with Stretch Value = 2

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.

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.

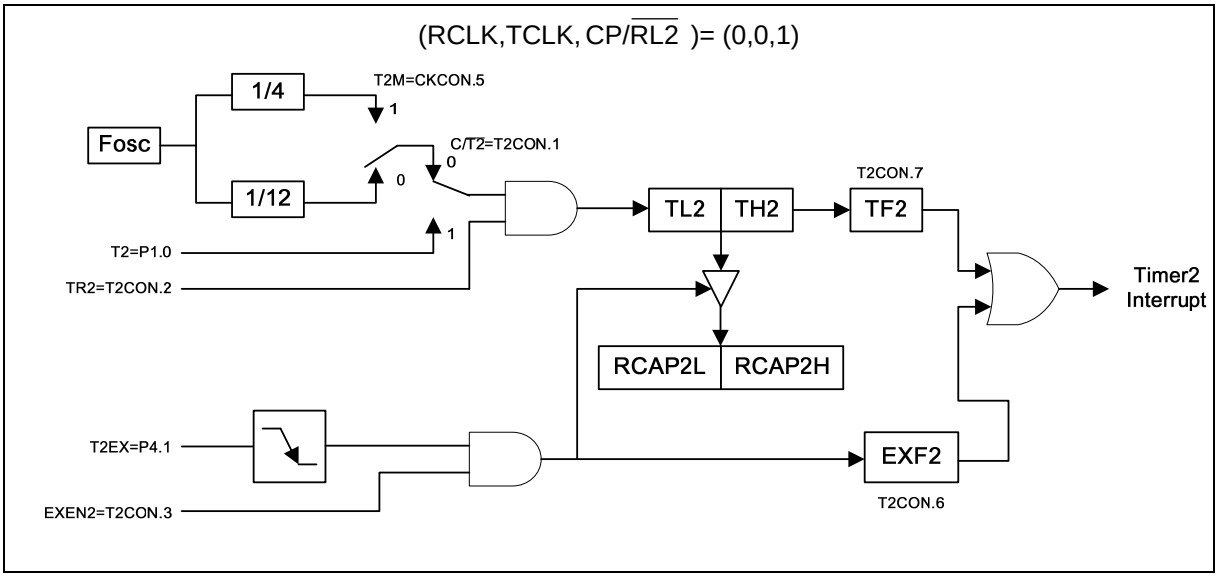


Figure 12-4: Timer2 16-Bit Capture Mode

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.

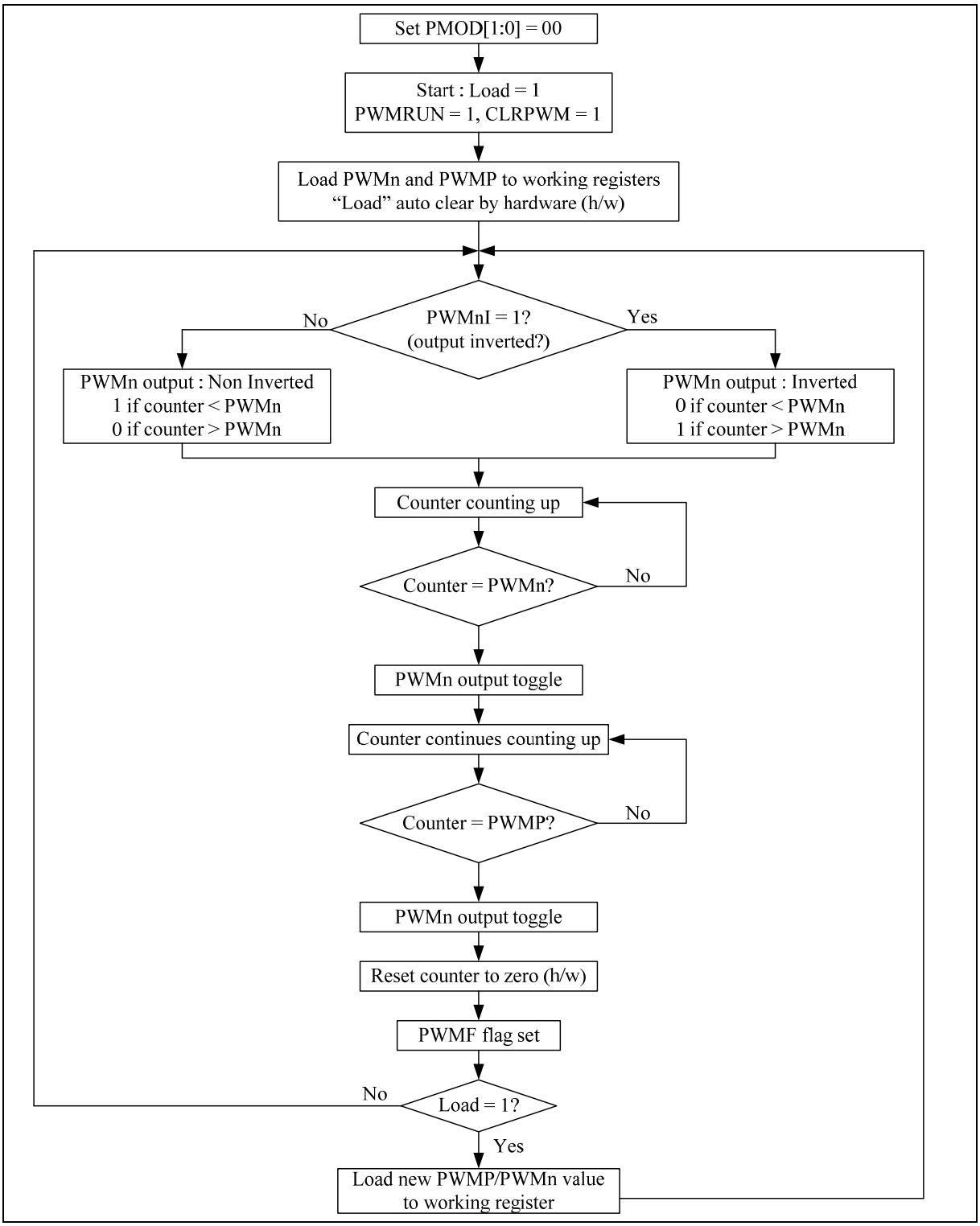


Figure 14-17: Edge-Aligned Flow Diagram

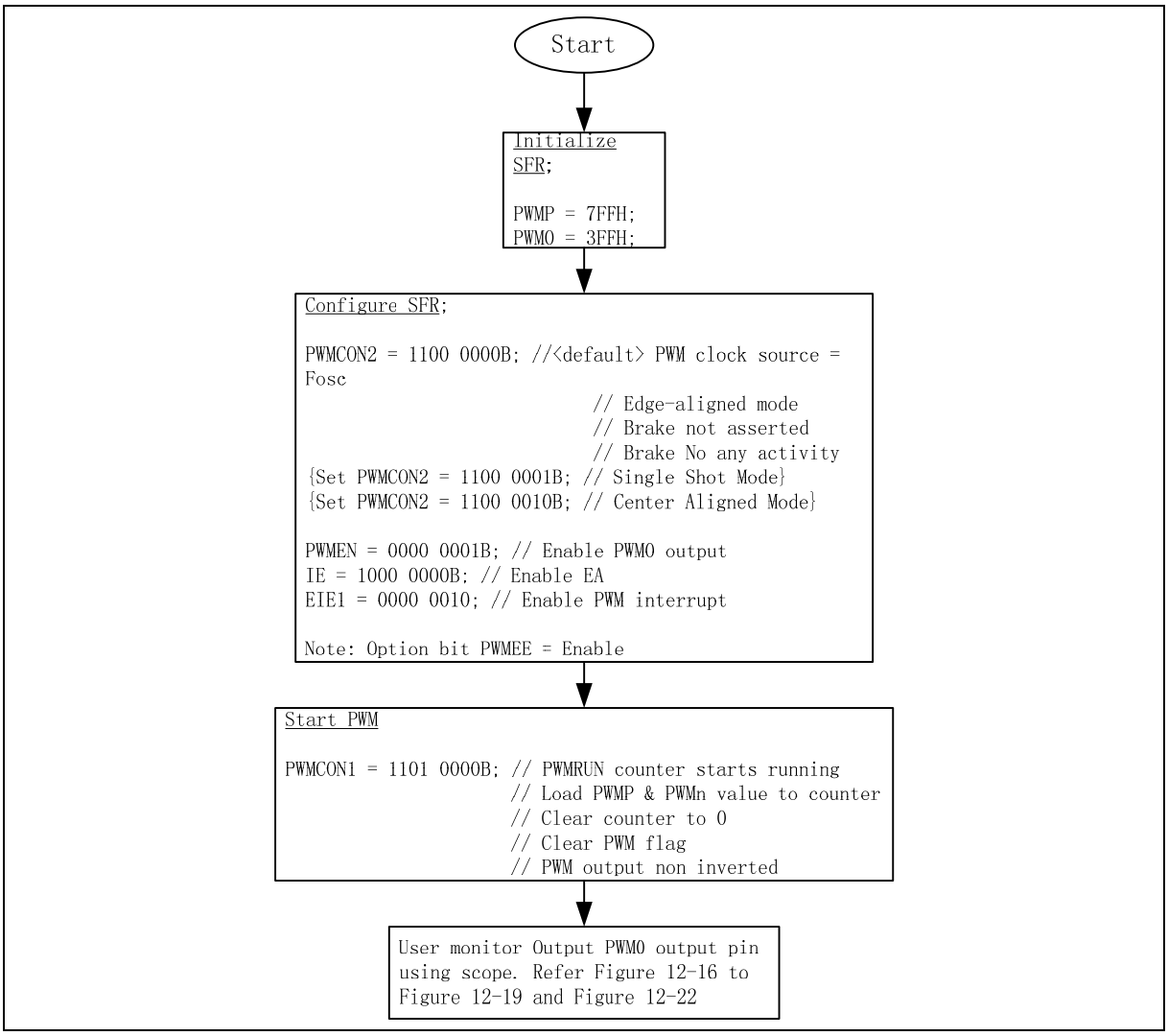


Figure 14-18: Program Flow for Edge-Aligned mode

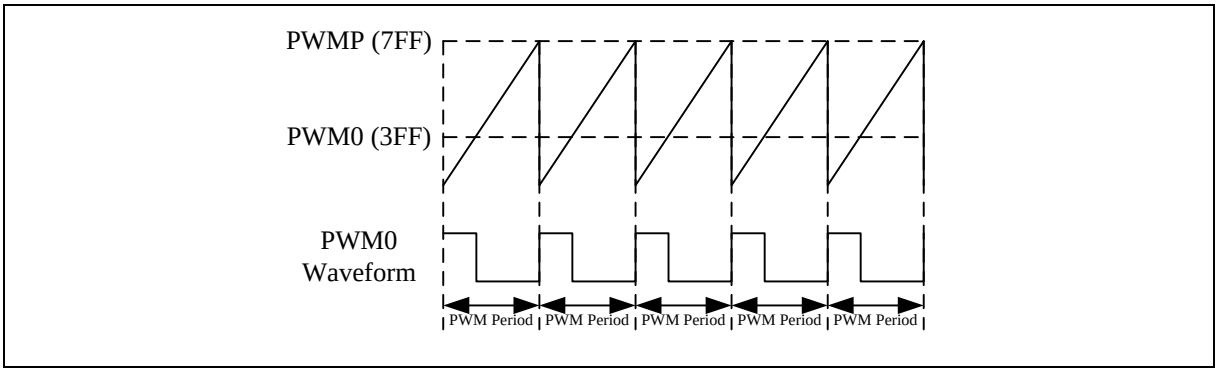


Figure 14-19: PWM0 Edge Aligned Waveform Output

14.8 Center Aligned PWM (up/down counter)

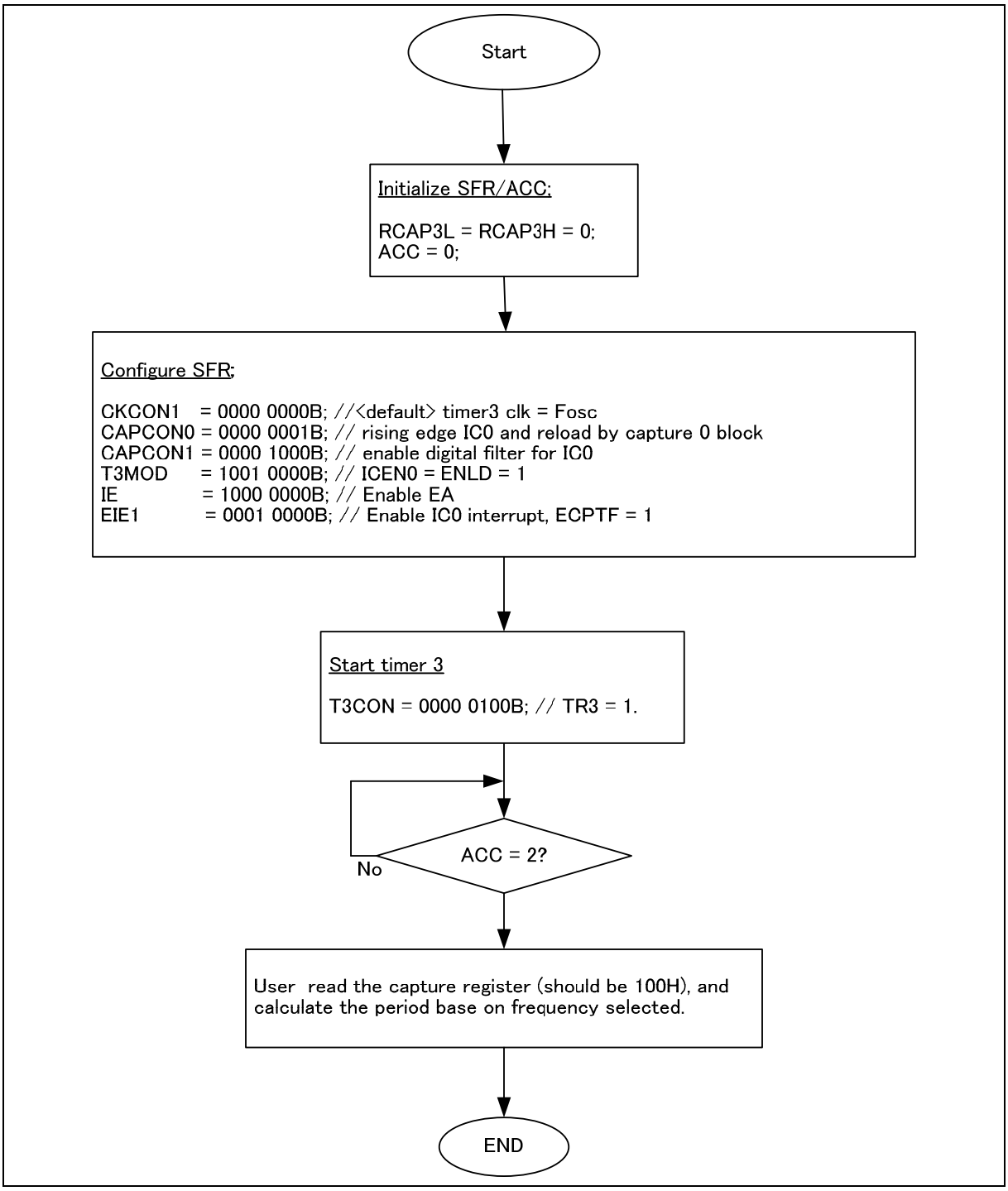


Figure 15-6: Program flow for measurement with IC0 between pulses with rising edge detection (ACC is incremented in interrupt service routine).

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.

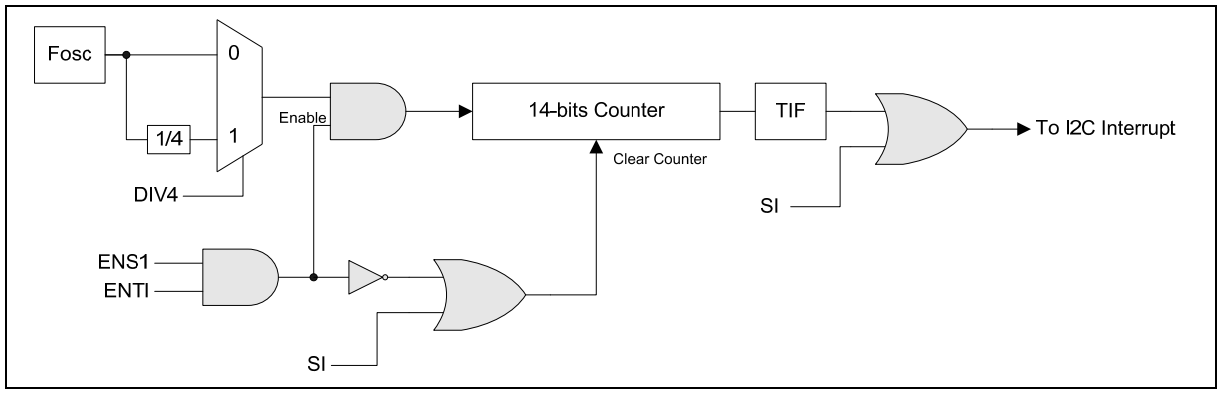


Figure 17-3: I2C Time-out Block Diagram

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 9th 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

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

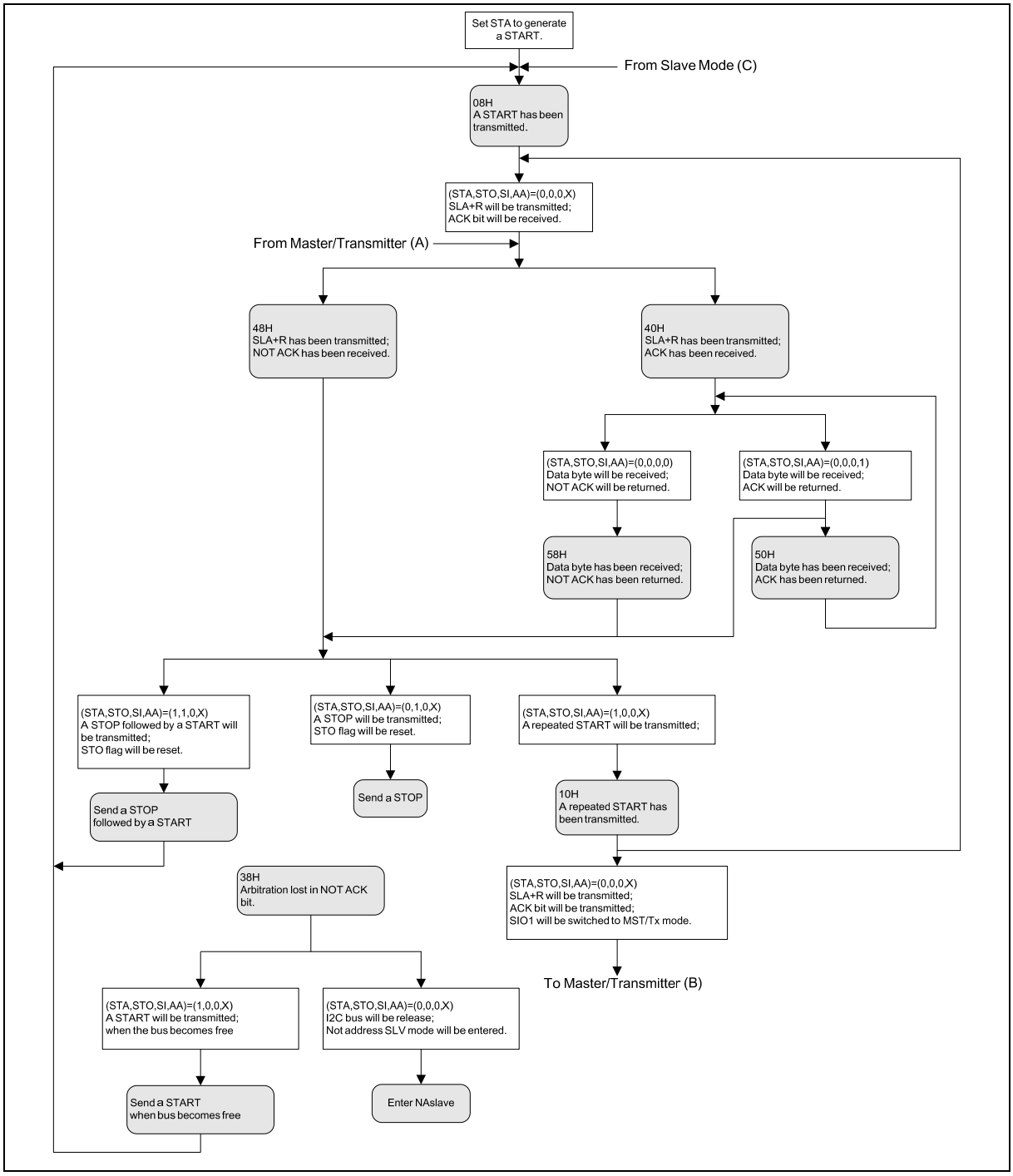


Figure 17-6: Master Receiver Mode

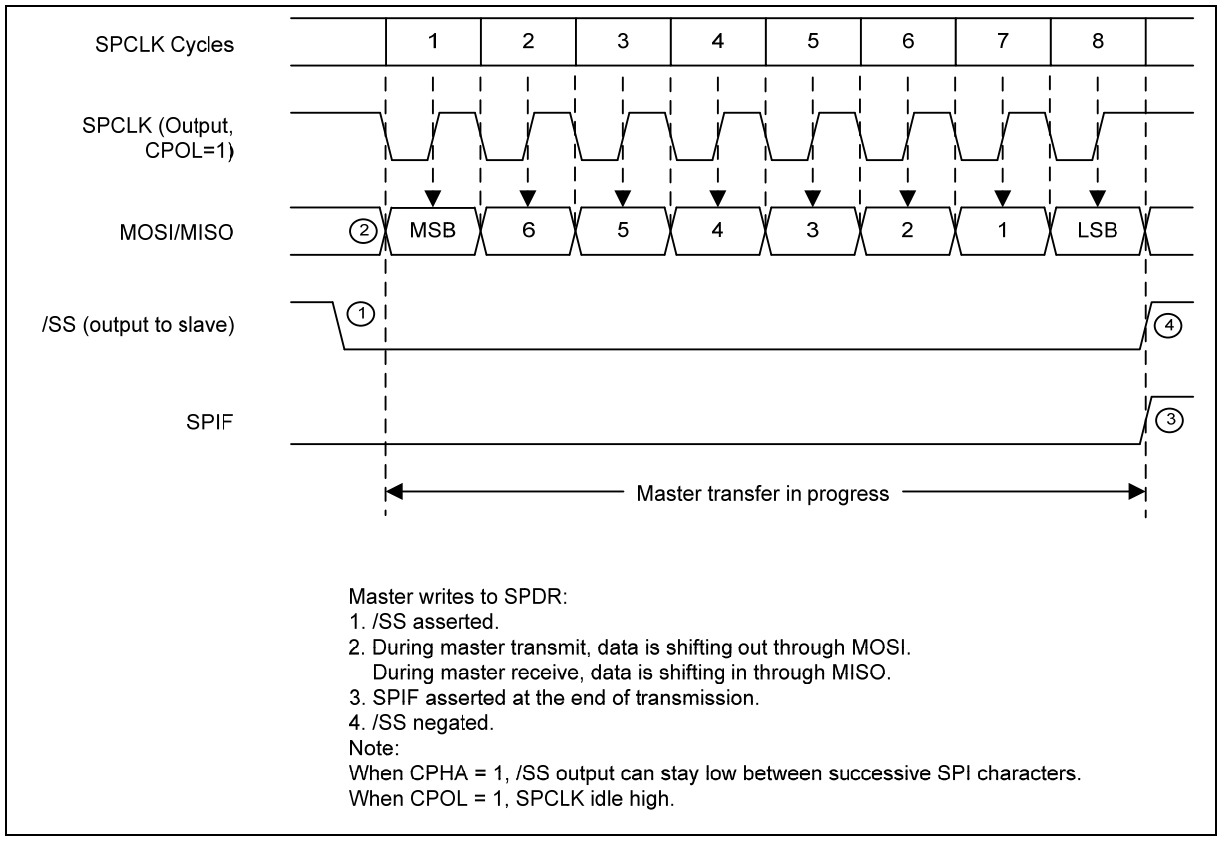


Figure 18-5: Master Mode Transmission (CPOL = 1, CPHA = 1)

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.

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.

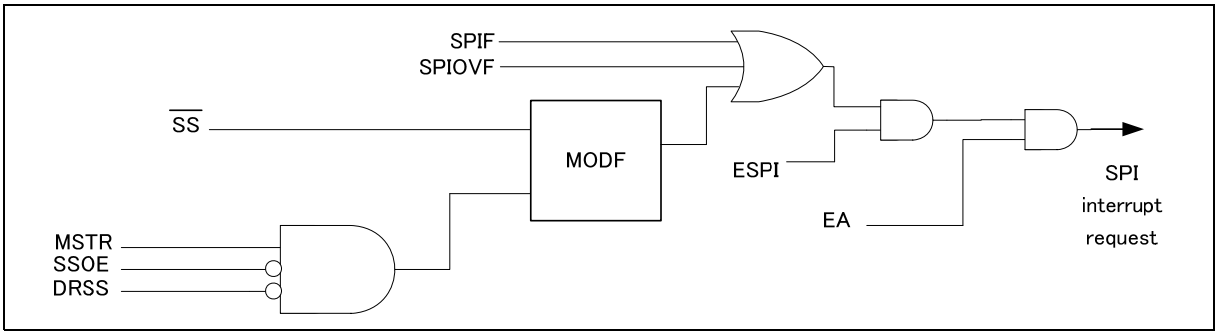


Figure 18-10: SPI interrupt request

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).

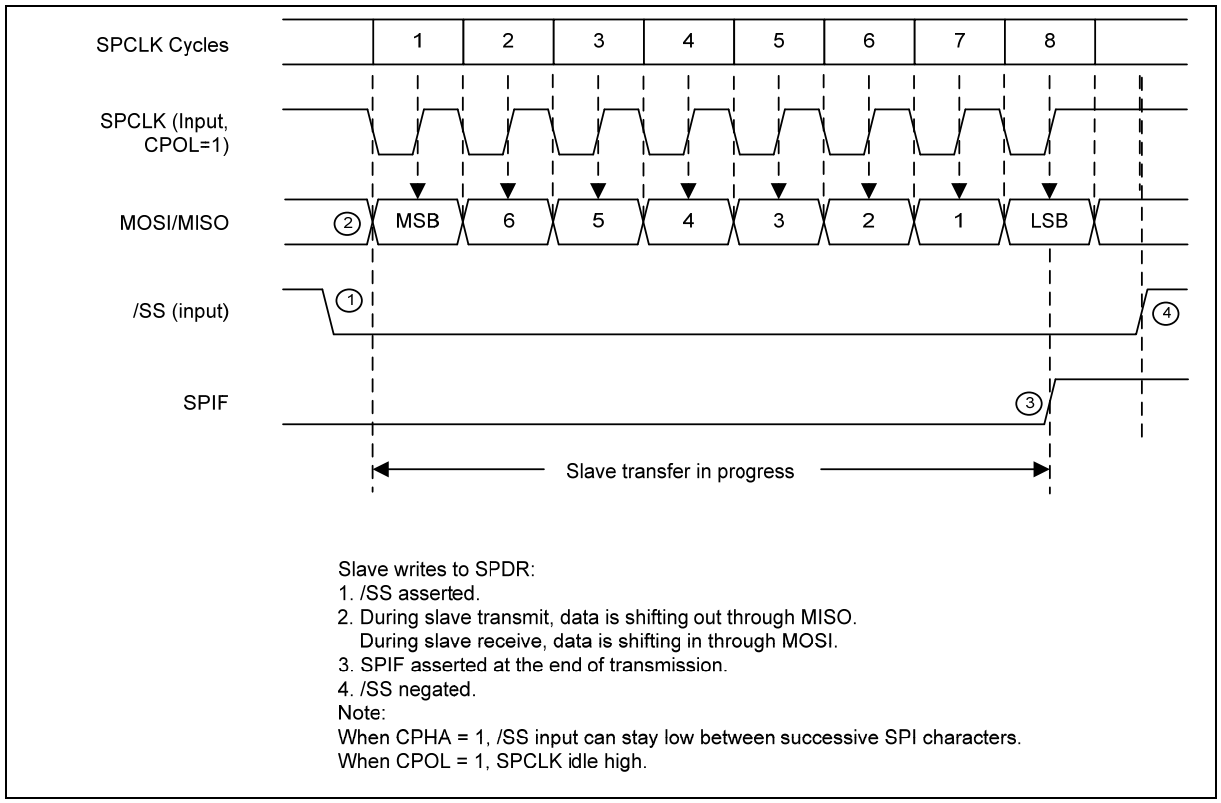


Figure 18-9 show the SPI transfer format, with different CPOL and CPHA. When CPHA = 0, data is sample on the first edge of SPCLK and when CPHA = 1 data is sample on the second edge of the SPCLK. Prior to changing CPOL setting, SPE must be disabled first.

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.

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

```
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
```