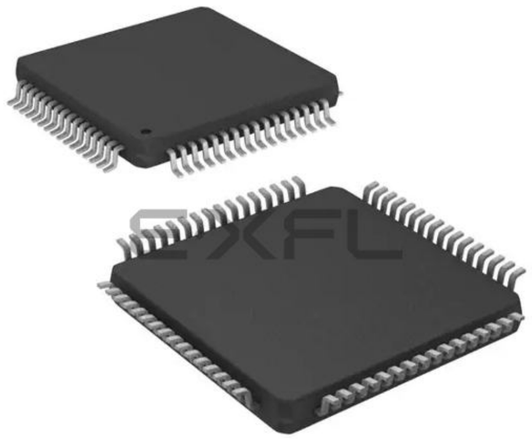


Welcome to E-XFL.COM

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.

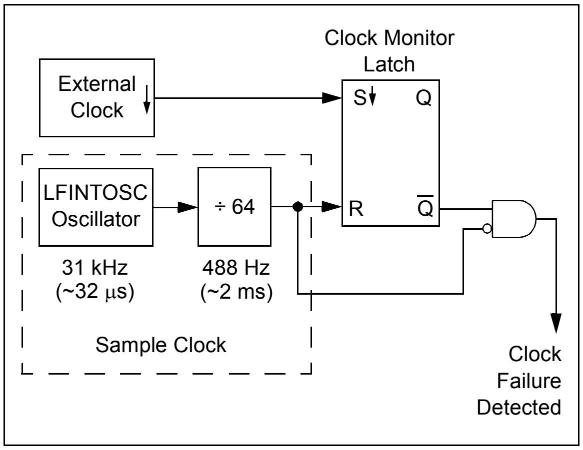
Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	64MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	60
Program Memory Size	128KB (64K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	3.5K x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 47x10b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f67k40-i-pt

4.5 Fail-Safe Clock Monitor

The Fail-Safe Clock Monitor (FSCM) allows the device to continue operating should the external oscillator fail. The FSCM is enabled by setting the FCMEN bit in the Configuration Words. The FSCM is applicable to all external Oscillator modes (LP, XT, HS, ECL/M/H and Secondary Oscillator).

FIGURE 4-9: FSCM BLOCK DIAGRAM



4.5.1 FAIL-SAFE DETECTION

The FSCM module detects a failed oscillator by comparing the external oscillator to the FSCM sample clock. The sample clock is generated by dividing the LFINTOSC by 64. See Figure 4-9. Inside the fail detector block is a latch. The external clock sets the latch on each falling edge of the external clock. The sample clock clears the latch on each rising edge of the sample clock. A failure is detected when an entire half-cycle of the sample clock elapses before the external clock goes low.

4.5.2 FAIL-SAFE OPERATION

When the external clock fails, the FSCM overwrites the COSC bits to select HFINTOSC (`3'b110`). The frequency of HFINTOSC would be determined by the previous state of the HFFRQ bits and the NDIV/CDIV bits. The bit flag OSCFIF of the PIR1 register is set. Setting this flag will generate an interrupt if the OSCFIE bit of the PIE1 register is also set. The device firmware can then take steps to mitigate the problems that may arise from a failed clock. The system clock will continue to be sourced from the internal clock source until the device firmware successfully restarts the external oscillator and switches back to external operation, by writing to the NOSC and NDIV bits of the OSCCON1 register.

4.5.3 FAIL-SAFE CONDITION CLEARING

The Fail-Safe condition is cleared after a Reset, executing a `SLEEP` instruction or changing the NOSC and NDIV bits of the OSCCON1 register. When switching to the external oscillator or PLL, the OST is restarted. While the OST is running, the device continues to operate from the INTOSC selected in OSCCON1. When the OST times out, the Fail-Safe condition is cleared after successfully switching to the external clock source. The OSCFIF bit should be cleared prior to switching to the external clock source. If the Fail-Safe condition still exists, the OSCFIF flag will again become set by hardware.

REGISTER 13-6: CRCACCL: CRC ACCUMULATOR LOW BYTE REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
ACC<7:0>							
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-0 **ACC<7:0>**: CRC Accumulator Register bits
Writing to this register writes to the CRC accumulator register through the CRC write bus. Reading from this register reads the CRC accumulator.

REGISTER 13-7: CRCSHIFTH: CRC SHIFT HIGH BYTE REGISTER

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
SHIFT<15:8>							
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-0 **SHIFT<15:8>**: CRC Shifter Register bits
Reading from this register reads the CRC Shifter.

REGISTER 13-8: CRCSHIFTL: CRC SHIFT LOW BYTE REGISTER

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
SHIFT<7:0>							
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-0 **SHIFT<7:0>**: CRC Shifter Register bits
Reading from this register reads the CRC Shifter.

TABLE 13-5: SUMMARY OF REGISTERS ASSOCIATED WITH CRC

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
CRCACCH	ACC<15:8>								155
CRCACCL	ACC<7:0>								156
CRCCON0	EN	GO	BUSY	ACCM	—	—	SHIFTM	FULL	154
CRCCON1	DLEN<3:0>				PLEN<3:0>				154
CRCDATH	DATA<15:8>								155
CRCDATL	DATA<7:0>								155
CRCSHIFTH	SHIFT<15:8>								156
CRCSHIFTL	SHIFT<7:0>								156
CRCXORH	X<15:8>								157
CRCXORL	X<7:1>							—	157
PMD0	SYSCMD	FVRMD	HLVDMD	CRCMD	SCANMD	NVMMD	CLKRMD	IOCMD	66
SCANCON0	SCANEN	SCANGO	BUSY	INVALID	INTM	—	MODE<1:0>		158
SCANHADRU	—	—	HADR<21:16>						160
SCANHADRH	HADR<15:8>								161
SCANHADRL	HADR<7:0>								161
SCANLADRU	—	—	LADR<21:16>						159
SCANLADRH	LADR<15:8>								159
SCANLADRL	LADR<7:0>								160
SCANTRIG	—	—	—	—	TSEL<3:0>				162
INTCON	GIE/GIEH	PEIE/GIEL	IPEN	—	INT3EDG	INT2EDG	INT1EDG	INT0EDG	173
PIR8	SCANIF	CRCIF	NVMIF	—	—	—	—	CWG1IF	183
PIE8	SCANIE	CRCIE	NVMIE	—	—	—	—	CWG1IE	193
IPR8	SCANIP	CRCIP	NVMIP	—	—	—	—	CWG1IP	203

Legend: — = unimplemented location, read as '0'. Shaded cells are not used for the CRC module.

REGISTER 14-26: IPR4: PERIPHERAL INTERRUPT PRIORITY REGISTER 4

U-0	U-0	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
—	—	RC5IP	TX5IP	RC4IP	TX4IP	RC3IP	TX3IP
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5 **RC5IP:** EUSART5 Receive Interrupt Priority bit
1 = High priority
0 = Low priority

bit 4 **TX5IP:** EUSART5 Transmit Interrupt Priority bit
1 = High priority
0 = Low priority

bit 3 **RC4IP:** EUSART4 Receive Interrupt Priority bit
1 = High priority
0 = Low priority

bit 2 **TX4IP:** EUSART4 Transmit Interrupt Priority bit
1 = High priority
0 = Low priority

bit 1 **RC3IP:** EUSART3 Receive Interrupt Priority bit
1 = High priority
0 = Low priority

bit 0 **TX3IP:** EUSART3 Transmit Interrupt Priority bit
1 = High priority
0 = Low priority

REGISTER 17-2: RxyPPS: PIN Rxy OUTPUT SOURCE SELECTION REGISTER

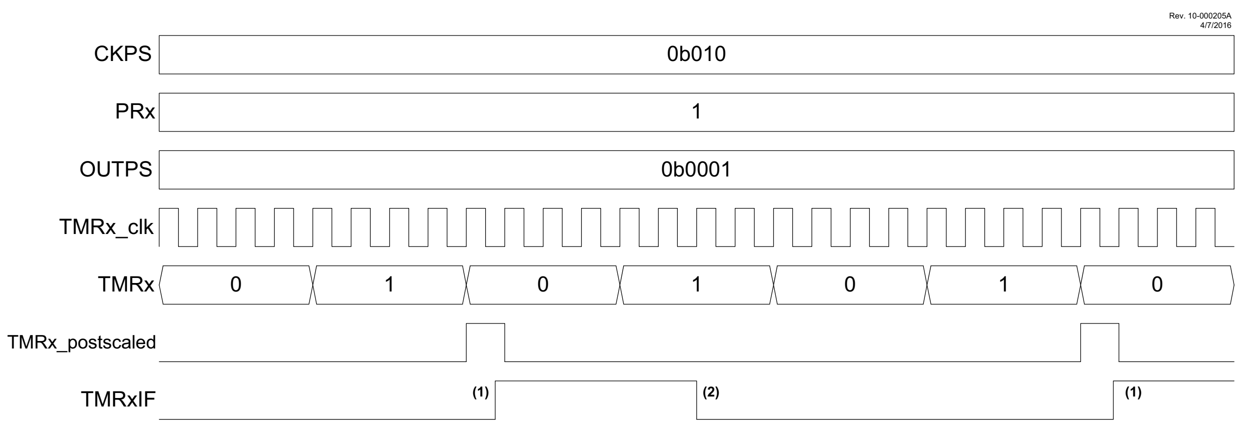
bit 5-0 **RxyPPS<5:0>:** Pin Rxy Output Source Selection bits

RxyPPS<5:0>	Pin Rxy Output Source	Output can be redirected to PORTx							
0x21	ADGRDB	—	—	C	—	—	—	—	H
0x20	ADGRDA	—	—	C	—	—	—	—	H
0x1F	DSM1	—	—	C	—	—	—	—	H
0x1E	CLKR	—	—	C	—	—	—	—	H
0x1D	TMR0	—	B	C	—	—	—	—	—
0x1C	MSSP2 (SDO/SDA)	—	B	—	D	—	—	—	—
0x1B	MSSP2 (SCK/SCL)	—	B	—	D	—	—	—	—
0x1A	MSSP1 (SDO/SDA)	—	B	C	—	—	—	—	—
0x19	MSSP1 (SCK/SCL)	—	B	C	—	—	—	—	—
0x18	CMP3	—	—	—	—	—	F	G	—
0x17	CMP2	—	—	—	—	—	F	G	—
0x16	CMP1	—	—	—	—	—	F	G	—
0x15	EUSART5 (DT)	—	—	—	—	E	—	G	—
0x14	EUSART5 (TX/CK)	—	—	—	—	E	—	G	—
0x13	EUSART4 (DT)	—	B	C	—	—	—	—	—
0x12	EUSART4 (TX/CK)	—	B	C	—	—	—	—	—
0x11	EUSART3 (DT)	—	B	—	—	E	—	—	—
0x10	EUSART3 (TX/CK)	—	B	—	—	E	—	—	—
0xF	EUSART2 (DT)	—	—	—	D	—	—	G	—
0xE	EUSART2 (TX/CK)	—	—	—	D	—	—	G	—
0xD	EUSART1 (DT)	—	—	C	D	—	—	—	—
0xC	EUSART1 (TX/CK)	—	—	C	D	—	—	—	—
0xB	PWM7	—	—	C	—	E	—	—	—
0xA	PWM6	—	—	C	—	E	—	—	—
0x9	CCP5	—	—	—	—	E	—	G	—
0x8	CCP4	—	—	—	—	E	—	G	—
0x7	CCP3	—	—	C	—	E	—	—	—
0x6	CCP2	—	—	C	—	E	—	—	—
0x5	CCP1	—	—	C	—	E	—	—	—
0x4	CWG1D	—	—	—	—	E	—	G	—
0x3	CWG1C	—	—	C	—	E	—	—	—
0x2	CWG1B	—	—	—	—	E	—	G	—
0x1	CWG1A	—	—	C	—	E	—	—	—
0x0	LATxy	A	B	C	D	E	F	G	H

20.4 Timer2 Interrupt

Timer2 can also generate a device interrupt. The interrupt is generated when the postscaler counter matches one of 16 postscale options (from 1:1 through 1:16), which are selected with the postscaler control bits, OUTPS<3:0> of the T2CON register. The interrupt is enabled by setting the TMR2IE interrupt enable bit of the PIE5 register. Interrupt timing is illustrated in Figure 20-3.

FIGURE 20-3: TIMER2 PRESCALER, POSTSCALER, AND INTERRUPT TIMING DIAGRAM



Note 1: Setting the interrupt flag is synchronized with the instruction clock. Synchronization may take as many as 2 instruction cycles

2: Cleared by software.

REGISTER 21-2: CCPTMRS0: CCP TIMERS CONTROL REGISTER 0

R/W-0/0	R/W-1/1	R/W-0/0	R/W-1/1	R/W-0/0	R/W-1/1	R/W-0/0	R/W-1/1
C4TSEL<1:0>		C3TSEL<1:0>		C2TSEL<1:0>		C1TSEL<1:0>	
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7-6 **C4TSEL<1:0>:** CCP4 Timer Selection bits
- `11` = CCP4 is based off Timer7 in Capture/Compare mode and Timer8 in PWM mode
- `10` = CCP4 is based off Timer5 in Capture/Compare mode and Timer6 in PWM mode
- `01` = CCP4 is based off Timer3 in Capture/Compare mode and Timer4 in PWM mode
- `00` = CCP4 is based off Timer1 in Capture/Compare mode and Timer2 in PWM mode

bit 5-4 **C3TSEL<1:0>:** CCP3 Timer Selection bits
- `11` = CCP3 is based off Timer7 in Capture/Compare mode and Timer8 in PWM mode
- `10` = CCP3 is based off Timer5 in Capture/Compare mode and Timer6 in PWM mode
- `01` = CCP3 is based off Timer3 in Capture/Compare mode and Timer4 in PWM mode
- `00` = CCP3 is based off Timer1 in Capture/Compare mode and Timer2 in PWM mode

bit 3-2 **C2TSEL<1:0>:** CCP2 Timer Selection bits
- `11` = CCP2 is based off Timer7 in Capture/Compare mode and Timer8 in PWM mode
- `10` = CCP2 is based off Timer5 in Capture/Compare mode and Timer6 in PWM mode
- `01` = CCP2 is based off Timer3 in Capture/Compare mode and Timer4 in PWM mode
- `00` = CCP2 is based off Timer1 in Capture/Compare mode and Timer2 in PWM mode

bit 1-0 **C1TSEL<1:0>:** CCP1 Timer Selection bits
- `11` = CCP1 is based off Timer7 in Capture/Compare mode and Timer8 in PWM mode
- `10` = CCP1 is based off Timer5 in Capture/Compare mode and Timer6 in PWM mode
- `01` = CCP1 is based off Timer3 in Capture/Compare mode and Timer4 in PWM mode
- `00` = CCP1 is based off Timer1 in Capture/Compare mode and Timer2 in PWM mode

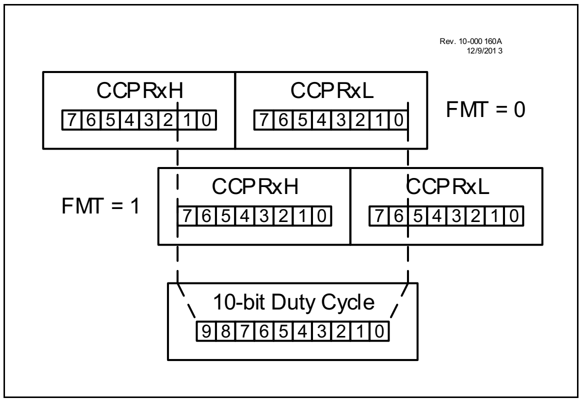
21.5.5 PWM DUTY CYCLE

The PWM duty cycle is specified by writing a 10-bit value to the CCPRxH:CCPRxL register pair. The alignment of the 10-bit value is determined by the FMT bit of the CCPxCON register (see Figure 21-5). The CCPRxH:CCPRxL register pair can be written to at any time; however the duty cycle value is not latched into the 10-bit buffer until after a match between PR2 and TMR2.

Equation 21-2 is used to calculate the PWM pulse width.

Equation 21-3 is used to calculate the PWM duty cycle ratio.

FIGURE 21-5: PWM 10-BIT ALIGNMENT



EQUATION 21-2: PULSE WIDTH

$$Pulse\ Width = (CCPRxH{:}CCPRxL\ register\ pair) \bullet T_{OSC} \bullet (TMR2\ Prescale\ Value)$$

EQUATION 21-3: DUTY CYCLE RATIO

$$Duty\ Cycle\ Ratio = \frac{(CCPRxH{:}CCPRxL\ register\ pair)}{4(PR2 + 1)}$$

CCPRxH:CCPRxL register pair are used to double buffer the PWM duty cycle. This double buffering is essential for glitchless PWM operation.

The 8-bit timer TMR2 register is concatenated with either the 2-bit internal system clock (FOSC), or two bits of the prescaler, to create the 10-bit time base. The system clock is used if the Timer2 prescaler is set to 1:1.

When the 10-bit time base matches the CCPRxH:CCPRxL register pair, then the CCPx pin is cleared (see Figure 21-4).

21.5.6 PWM RESOLUTION

The resolution determines the number of available duty cycles for a given period. For example, a 10-bit resolution will result in 1024 discrete duty cycles, whereas an 8-bit resolution will result in 256 discrete duty cycles.

The maximum PWM resolution is ten bits when PR2 is 255. The resolution is a function of the PR2 register value as shown by Equation 21-4.

EQUATION 21-4: PWM RESOLUTION

$$Resolution = \frac{\log[4(PR2 + 1)]}{\log(2)}\ bits$$

Note:	If the pulse-width value is greater than the period the assigned PWM pin(s) will remain unchanged.

FIGURE 24-14: CWG SHUTDOWN BLOCK DIAGRAM

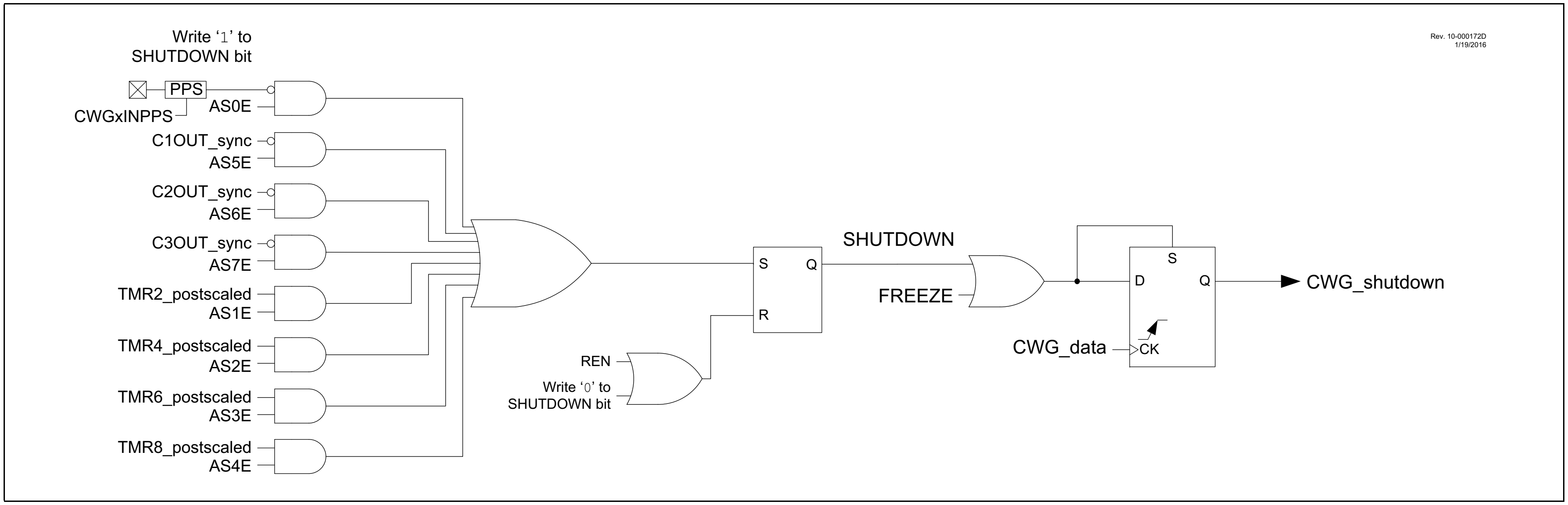


FIGURE 24-15: SHUTDOWN FUNCTIONALITY, AUTO-RESTART DISABLED (REN = 0, LSAC = 01, LSBD = 01)

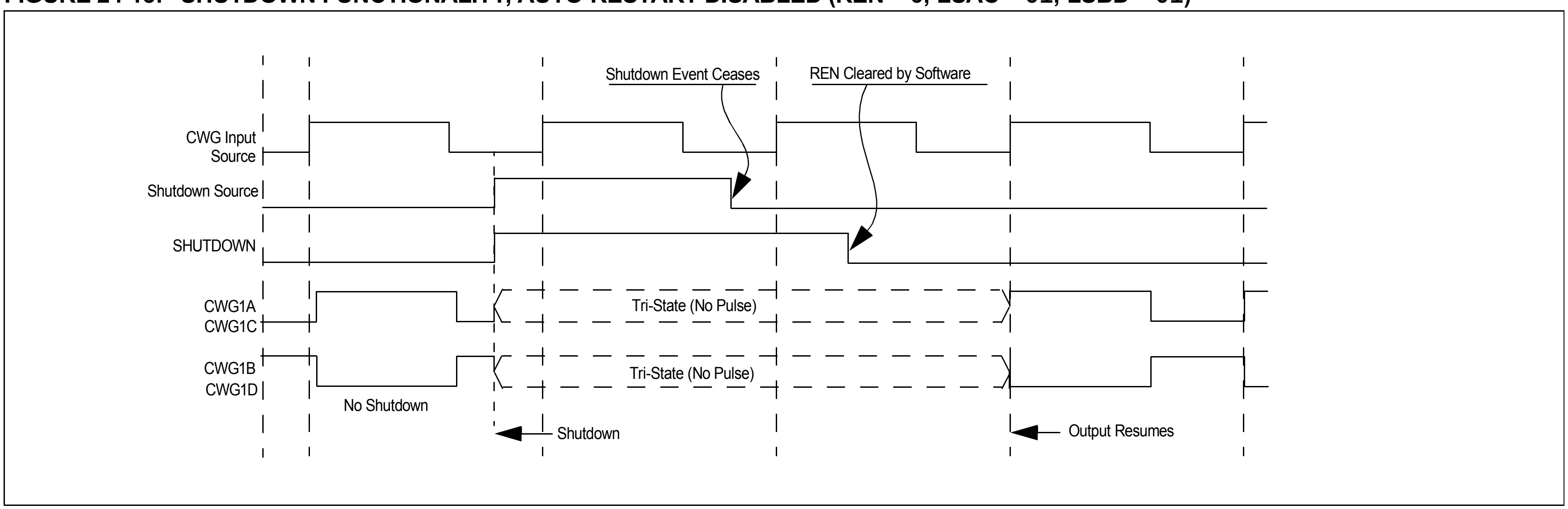
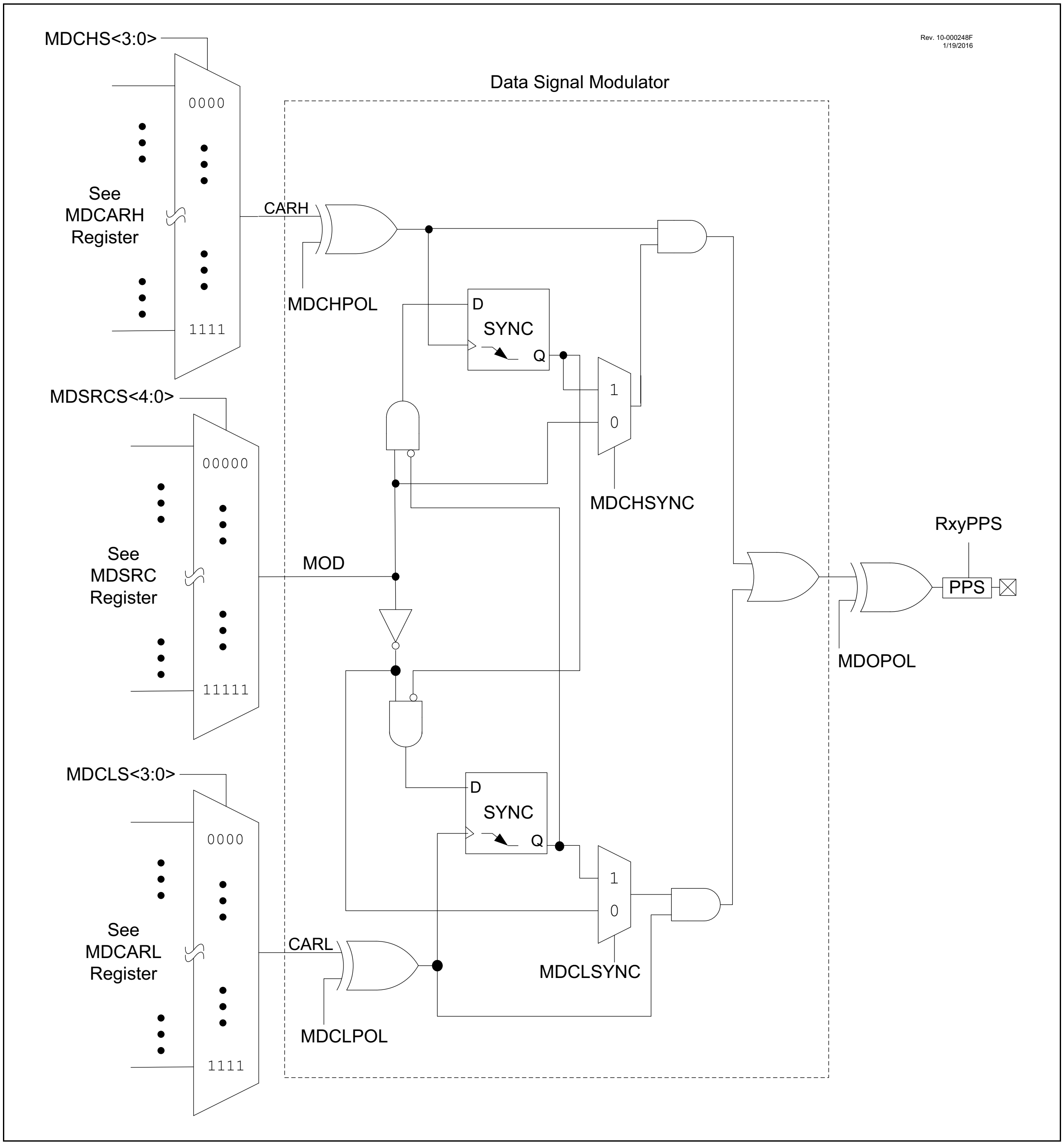


FIGURE 26-1: SIMPLIFIED BLOCK DIAGRAM OF THE DATA SIGNAL MODULATOR



REGISTER 26-2: MDCON1: MODULATION CONTROL REGISTER 1

U-0	U-0	R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0
—	—	CHPOL	CHSYNC	—	—	CLPOL	CLSYNC
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5 **CHPOL:** Modulator High Carrier Polarity Select bit
1 = Selected high carrier signal is inverted
0 = Selected high carrier signal is not inverted

bit 4 **CHSYNC:** Modulator High Carrier Synchronization Enable bit
1 = Modulator waits for a falling edge on the high time carrier signal before allowing a switch to the low time carrier
0 = Modulator output is not synchronized to the high time carrier signal[(1)]

bit 3-2 **Unimplemented:** Read as ‘0’

bit 1 **CLPOL:** Modulator Low Carrier Polarity Select bit
1 = Selected low carrier signal is inverted
0 = Selected low carrier signal is not inverted

bit 0 **CLSYNC:** Modulator Low Carrier Synchronization Enable bit
1 = Modulator waits for a falling edge on the low time carrier signal before allowing a switch to the high time carrier
0 = Modulator output is not synchronized to the low time carrier signal[(1)]

Note 1: Narrowed carrier pulse widths or spurs may occur in the signal stream if the carrier is not synchronized.

28.5 EUSART Synchronous Mode

Synchronous serial communications are typically used in systems with a single master and one or more slaves. The master device contains the necessary circuitry for baud rate generation and supplies the clock for all devices in the system. Slave devices can take advantage of the master clock by eliminating the internal clock generation circuitry.

There are two signal lines in Synchronous mode: a bidirectional data line and a clock line. Slaves use the external clock supplied by the master to shift the serial data into and out of their respective receive and transmit shift registers. Since the data line is bidirectional, synchronous operation is half-duplex only. Half-duplex refers to the fact that master and slave devices can receive and transmit data but not both simultaneously. The EUSART can operate as either a master or slave device.

Start and Stop bits are not used in synchronous transmissions.

28.5.1 SYNCHRONOUS MASTER MODE

The following bits are used to configure the EUSART for synchronous master operation:

- SYNC = `1`
- CSRC = `1`
- SREN = `0` (for transmit); SREN = `1` (for receive)
- CREN = `0` (for transmit); CREN = `1` (for receive)
- SPEN = `1`

Setting the SYNC bit of the TXxSTA register configures the device for synchronous operation. Setting the CSRC bit of the TXxSTA register configures the device as a master. Clearing the SREN and CREN bits of the RCxSTA register ensures that the device is in the Transmit mode, otherwise the device will be configured to receive. Setting the SPEN bit of the RCxSTA register enables the EUSART.

28.5.1.1 Master Clock

Synchronous data transfers use a separate clock line, which is synchronous with the data. A device configured as a master transmits the clock on the TX/CK line. The TXx/CKx pin output driver is automatically enabled when the EUSART is configured for synchronous transmit or receive operation. Serial data bits change on the leading edge to ensure they are valid at the trailing edge of each clock. One clock cycle is generated for each data bit. Only as many clock cycles are generated as there are data bits.

28.5.1.2 Clock Polarity

A clock polarity option is provided for Microwire compatibility. Clock polarity is selected with the SCKP bit of the BAUDxCON register. Setting the SCKP bit sets the clock Idle state as high. When the SCKP bit is set, the data changes on the falling edge of each clock. Clearing the SCKP bit sets the Idle state as low. When the SCKP bit is cleared, the data changes on the rising edge of each clock.

28.5.1.3 Synchronous Master Transmission

Data is transferred out of the device on the RXx/DTx pin. The RXx/DTx and TXx/CKx pin output drivers are automatically enabled when the EUSART is configured for synchronous master transmit operation.

A transmission is initiated by writing a character to the TXxREG register. If the TSR still contains all or part of a previous character the new character data is held in the TXxREG until the last bit of the previous character has been transmitted. If this is the first character, or the previous character has been completely flushed from the TSR, the data in the TXxREG is immediately transferred to the TSR. The transmission of the character commences immediately following the transfer of the data to the TSR from the TXxREG.

Each data bit changes on the leading edge of the master clock and remains valid until the subsequent leading clock edge.

Note:	The TSR register is not mapped in data memory, so it is not available to the user.

28.5.1.4 Synchronous Master Transmission Setup:

1. Initialize the SPxBRGH, SPxBRGL register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see **Section 28.4 “EUSART Baud Rate Generator (BRG)”**).
2. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC.
3. Disable Receive mode by clearing bits SREN and CREN.
4. Enable Transmit mode by setting the TXEN bit.
5. If 9-bit transmission is desired, set the TX9 bit.
6. If interrupts are desired, set the TXxIE bit of the PIE3/4 registers and the GIE and PEIE bits of the INTCON register.
7. If 9-bit transmission is selected, the ninth bit should be loaded in the TX9D bit.
8. Start transmission by loading data to the TXxREG register.

TABLE 28-9: SUMMARY OF REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE TRANSMISSION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
BAUDxCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	451
INTCON	GIE/GIEH	PEIE/GIEL	IPEN	—	INT3EDG	INT2EDG	INT1EDG	INT0EDG	173
PIE3	RC2IE	TX2IE	RC1IE	TX1IE	BCL2IE	SSP2IE	BCL1IE	SSP1IE	188
PIR3	RC2IF	TX2IF	RC1IF	TX1IF	BCL2IF	SSP2IF	BCL1IF	SSP1IF	177
IPR3	RC2IP	TX2IP	RC1IP	TX1IP	BCL2IP	SSP2IP	BCL1IP	SSP1IP	198
PIE4	—	—	RC5IE	TX5IE	RC4IE	TX4IE	RC3IE	TX3IE	189
PIR4	—	—	RC5IF	TX5IF	RC4IF	TX4IF	RC3IF	TX3IF	178
IPR4	—	—	RC5IP	TX5IP	RC4IP	TX4IP	RC3IP	TX3IP	199
RCxSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	450
RxyPPS	—	—	RxyPPS<5:0>						228
TXxPPS	—	—	TXPPS<5:0>						225
TXxREG	EUSARTx Transmit Data Register								452*
TXxSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	449

Legend: — = unimplemented location, read as ‘0’. Shaded cells are not used for synchronous slave transmission.
* Page provides register information.

REGISTER 32-2: ADCON1: ADC CONTROL REGISTER 1

R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0	R/W-0/0
ADDPOL	ADIPEN	ADGPOL	—	—	—	—	ADDSEN
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7 **ADDPOL:** Precharge Polarity bit

If ADPRE>0x00:

ADPPOL	Action During 1st Precharge Stage	
	External (selected analog I/O pin)	Internal (AD sampling capacitor)
1	Shorted to AVDD	C_{HOLD} shorted to VSS
0	Shorted to VSS	C_{HOLD} shorted to AVDD

Otherwise:
The bit is ignored

bit 6 **ADIPEN:** A/D Inverted Precharge Enable bit

If ADDSEN = 1

1 = The precharge and guard signals in the second conversion cycle are the opposite polarity of the first cycle

0 = Both Conversion cycles use the precharge and guards specified by ADPPOL and ADGPOL

Otherwise:
The bit is ignored

bit 5 **ADGPOL:** Guard Ring Polarity Selection bit

1 = ADC guard Ring outputs start as digital high during Precharge stage

0 = ADC guard Ring outputs start as digital low during Precharge stage

bit 4-1 **Unimplemented:** Read as '0'

bit 0 **ADDSEN:** Double-sample enable bit

1 = Two conversions are performed on each trigger. Data from the first conversion appears in ADPREV

0 = One conversion is performed for each trigger

34.2 HLVD Setup

To set up the HLVD module:

1. Select the desired HLVD trip point by writing the value to the HLVDSEL<3:0> bits of the HLVDCON1 register.
2. Depending on the application to detect high-voltage peaks or low-voltage drops or both, set the HLVDINTH or HLVDINTL bit appropriately.
3. Enable the HLVD module by setting the HLVDEN bit.
4. Clear the HLVD interrupt flag (PIR2 register), which may have been set from a previous interrupt.
5. If interrupts are desired, enable the HLVD interrupt by setting the HLVDIE in the PIE2 register and GIE bits.

   An interrupt will not be generated until the HLVDRDY bit is set.

Note:	Before changing any module settings (HLVDINTH, HLVDINTL, HLVDSEL<3:0>), first disable the module (HLVDEN = `0`), make the changes and re-enable the module. This prevents the generation of false HLVD events.

34.3 Current Consumption

When the module is enabled, the HLVD comparator and voltage divider are enabled and consume static current. The total current consumption, when enabled, is specified in electrical specification Parameter **D206** (Table 38-3Table 38-3).

Depending on the application, the HLVD module does not need to operate constantly. To reduce current requirements, the HLVD circuitry may only need to be enabled for short periods where the voltage is checked. After such a check, the module could be disabled.

34.4 HLVD Start-up Time

The internal reference voltage of the HLVD module, specified in electrical specification (Table 38-17), may be used by other internal circuitry, such as the programmable Brown-out Reset. If the HLVD or other circuits using the voltage reference are disabled to lower the device's current consumption, the reference voltage circuit will require time to become stable before a low or high-voltage condition can be reliably detected. This start-up time, T_{FVRST}, is an interval that is independent of device clock speed. It is specified in electrical specification (Table 38-17).

The HLVD interrupt flag is not enabled until T_{FVRST} has expired and a stable reference voltage is reached. For this reason, brief excursions beyond the set point may not be detected during this interval (see Figure 34-2 or Figure 34-3).

BNC — Branch if Not Carry

Syntax:	BNC n
Operands:	-128 ≤ n ≤ 127
Operation:	if CARRY bit is '0' (PC) + 2 + 2n → PC
Status Affected:	None
Encoding:	`1110` `0011` `nnnn` `nnnn`
Description:	If the CARRY bit is '0', then the program will branch. The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 2 + 2n. This instruction is then a 2-cycle instruction.
Words:	1
Cycles:	1(2)

Q Cycle Activity:

If Jump:

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	Write to PC
No operation	No operation	No operation	No operation

If No Jump:

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	No operation

Example: `HERE BNC Jump`

Before Instruction
PC = address `(HERE)`
After Instruction
If CARRY = 0;
PC = address `(Jump)`
If CARRY = 1;
PC = address `(HERE + 2)`

BNN — Branch if Not Negative

Syntax:	BNN n
Operands:	-128 ≤ n ≤ 127
Operation:	if NEGATIVE bit is '0' (PC) + 2 + 2n → PC
Status Affected:	None
Encoding:	`1110` `0111` `nnnn` `nnnn`
Description:	If the NEGATIVE bit is '0', then the program will branch. The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 2 + 2n. This instruction is then a 2-cycle instruction.
Words:	1
Cycles:	1(2)

Q Cycle Activity:

If Jump:

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	Write to PC
No operation	No operation	No operation	No operation

If No Jump:

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	No operation

Example: `HERE BNN Jump`

Before Instruction
PC = address `(HERE)`
After Instruction
If NEGATIVE = 0;
PC = address `(Jump)`
If NEGATIVE = 1;
PC = address `(HERE + 2)`

GOTO	Unconditional Branch
Syntax:	GOTO k
Operands:	$0 \le k \le 1048575$
Operation:	$k \rightarrow$ PC<20:1>
Status Affected:	None

Encoding:

1st word (k<7:0>)	1110	1111	$k_7$kkk	kkkk$_0$
2nd word(k<19:8>)	1111	k_{19}kkk	kkkk	kkkk$_8$

Description: GOTO allows an unconditional branch anywhere within entire 2-Mbyte memory range. The 20-bit value 'k' is loaded into PC<20:1>. GOTO is always a 2-cycle instruction.

Words: 2

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'<7:0>,	No operation	Read literal 'k'<19:8>, Write to PC
No operation	No operation	No operation	No operation

Example: `GOTO THERE`

After Instruction
PC = Address (THERE)

INCF	Increment f
Syntax:	INCF f {,d {,a}}
Operands:	$0 \le f \le 255$ $d \in [0,1]$ $a \in [0,1]$
Operation:	(f) + 1 $\rightarrow$ dest
Status Affected:	C, DC, N, OV, Z

Encoding:

0010	10da	ffff	ffff

Description: The contents of register 'f' are incremented. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f' (default).
If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank.
If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \le 95$ (5Fh). See **Section 36.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write to destination

Example: `INCF CNT, 1, 0`

Before Instruction
CNT = FFh
Z = 0
C = ?
DC = ?

After Instruction
CNT = 00h
Z = 1
C = 1
DC = 1

MULLW — Multiply literal with W

Syntax:	MULLW k
Operands:	$0 \le k \le 255$
Operation:	(W) x k $\rightarrow$ PRODH:PRODL
Status Affected:	None
Encoding:	`0000` `1101` `kkkk` `kkkk`
Description:	An unsigned multiplication is carried out between the contents of W and the 8-bit literal 'k'. The 16-bit result is placed in the PRODH:PRODL register pair. PRODH contains the high byte. W is unchanged. None of the Status flags are affected. Note that neither overflow nor carry is possible in this operation. A zero result is possible but not detected.
Words:	1
Cycles:	1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process Data	Write registers PRODH: PRODL

Example: `MULLW 0C4h`

Before Instruction

W = E2h
PRODH = ?
PRODL = ?

After Instruction

W = E2h
PRODH = ADh
PRODL = 08h

MULWF — Multiply W with f

Syntax:	MULWF f {,a}
Operands:	$0 \le f \le 255$ $a \in [0,1]$
Operation:	(W) x (f) $\rightarrow$ PRODH:PRODL
Status Affected:	None
Encoding:	`0000` `001a` `ffff` `ffff`
Description:	An unsigned multiplication is carried out between the contents of W and the register file location 'f'. The 16-bit result is stored in the PRODH:PRODL register pair. PRODH contains the high byte. Both W and 'f' are unchanged. None of the Status flags are affected. Note that neither overflow nor carry is possible in this operation. A zero result is possible but not detected. If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank. If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \le 95$ (5Fh). See **Section 36.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.
Words:	1
Cycles:	1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write registers PRODH: PRODL

Example: `MULWF REG, 1`

Before Instruction

W = C4h
REG = B5h
PRODH = ?
PRODL = ?

After Instruction

W = C4h
REG = B5h
PRODH = 8Ah
PRODL = 94h

TSTFSZ — Test f, skip if 0

Syntax:	TSTFSZ f {,a}
Operands:	0 ≤ f ≤ 255 a ∈ [0,1]
Operation:	skip if f = 0
Status Affected:	None
Encoding:	`0110` `011a` `ffff` `ffff`
Description:	If 'f' = 0, the next instruction fetched during the current instruction execution is discarded and a NOP is executed, making this a 2-cycle instruction. If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank. If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever f ≤ 95 (5Fh). See **Section 36.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.
Words:	1
Cycles:	1(2) **Note:** 3 cycles if skip and followed by a 2-word instruction.

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	No operation

If skip:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation

If skip and followed by 2-word instruction:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation
No operation	No operation	No operation	No operation

Example:

```
HERE    TSTFSZ  CNT, 1
NZERO   :
ZERO    :
```

Before Instruction
- PC = Address (HERE)

After Instruction
- If CNT = 00h,
- PC = Address (ZERO)
- If CNT ≠ 00h,
- PC = Address (NZERO)

XORLW — Exclusive OR literal with W

Syntax:	XORLW k
Operands:	0 ≤ k ≤ 255
Operation:	(W) .XOR. k → W
Status Affected:	N, Z
Encoding:	`0000` `1010` `kkkk` `kkkk`
Description:	The contents of W are XORed with the 8-bit literal 'k'. The result is placed in W.
Words:	1
Cycles:	1

Q Cycle Activity:

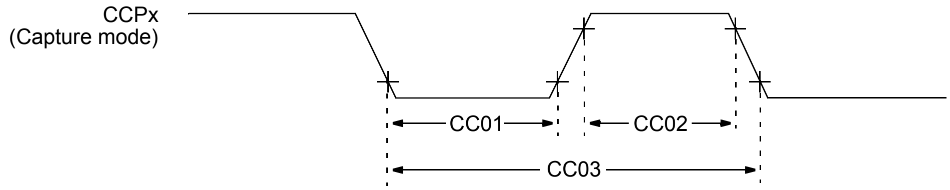
Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process Data	Write to W

Example: `XORLW 0AFh`

Before Instruction
- W = B5h

After Instruction
- W = 1Ah

FIGURE 38-13: CAPTURE/COMPARE/PWM TIMINGS (CCP)



Note: Refer to Figure 38-4 for load conditions.

TABLE 38-20: CAPTURE/COMPARE/PWM REQUIREMENTS (CCP)

Standard Operating Conditions (unless otherwise stated)
Operating Temperature -40°C ≤ TA ≤ +125°C

Param No.	Sym.	Characteristic		Min.	Typ†	Max.	Units	Conditions
CC01*	TccL	CCPx Input Low Time	No Prescaler	0.5TCY + 20	—	—	ns	
			With Prescaler	20	—	—	ns	
CC02*	TccH	CCPx Input High Time	No Prescaler	0.5TCY + 20	—	—	ns	
			With Prescaler	20	—	—	ns	
CC03*	TccP	CCPx Input Period		$\frac{3T_{CY} + 40}{N}$	—	—	ns	N = prescale value

* These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.