

Welcome to [E-XFL.COM](https://www.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	I ² C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	25
Program Memory Size	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	3.6K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 24x10b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SSOP (0.209", 5.30mm Width)
Supplier Device Package	28-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf26k40t-i-ss

2.3 Master Clear ($\overline{\text{MCLR}}$) Pin

The $\overline{\text{MCLR}}$ pin provides two specific device functions: Device Reset, and Device Programming and Debugging. If programming and debugging are not required in the end application, a direct connection to V_{DD} may be all that is required. The addition of other components, to help increase the application's resistance to spurious Resets from voltage sags, may be beneficial. A typical configuration is shown in Figure 2-1. Other circuit designs may be implemented, depending on the application's requirements.

During programming and debugging, the resistance and capacitance that can be added to the pin must be considered. Device programmers and debuggers drive the $\overline{\text{MCLR}}$ pin. Consequently, specific voltage levels (V_{IH} and V_{IL}) and fast signal transitions must not be adversely affected. Therefore, specific values of $R1$ and $C1$ will need to be adjusted based on the application and PCB requirements. For example, it is recommended that the capacitor, $C1$, be isolated from the $\overline{\text{MCLR}}$ pin during programming and debugging operations by using a jumper (Figure 2-2). The jumper is replaced for normal run-time operations.

Any components associated with the $\overline{\text{MCLR}}$ pin should be placed within 0.25 inch (6 mm) of the pin.

2.4 ICSP™ Pins

The PGC and PGD pins are used for In-Circuit Serial Programming™ (ICSP™) and debugging purposes. It is recommended to keep the trace length between the ICSP connector and the ICSP pins on the device as short as possible. If the ICSP connector is expected to experience an ESD event, a series resistor is recommended, with the value in the range of a few tens of ohms, not to exceed 100Ω.

Pull-up resistors, series diodes and capacitors on the PGC and PGD pins are not recommended as they will interfere with the programmer/debugger communications to the device. If such discrete components are an application requirement, they should be removed from the circuit during programming and debugging. Alternatively, refer to the AC/DC characteristics and timing requirements information in the respective device Flash programming specification for information on capacitive loading limits, and pin input voltage high (V_{IH}) and input low (V_{IL}) requirements.

For device emulation, ensure that the “Communication Channel Select” (i.e., PGCx/PGDx pins), programmed into the device, matches the physical connections for the ICSP to the Microchip debugger/emulator tool.

For more information on available Microchip development tools connection requirements, refer to **Section 36.0 “Development Support”**.

FIGURE 2-2: EXAMPLE OF $\overline{\text{MCLR}}$ PIN CONNECTIONS

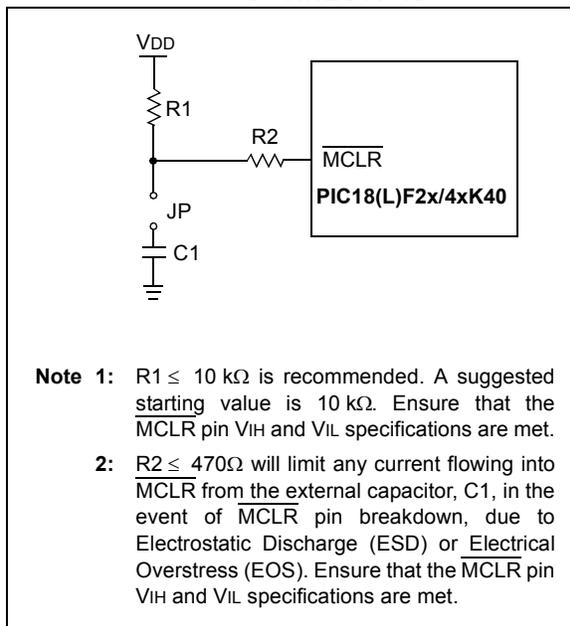
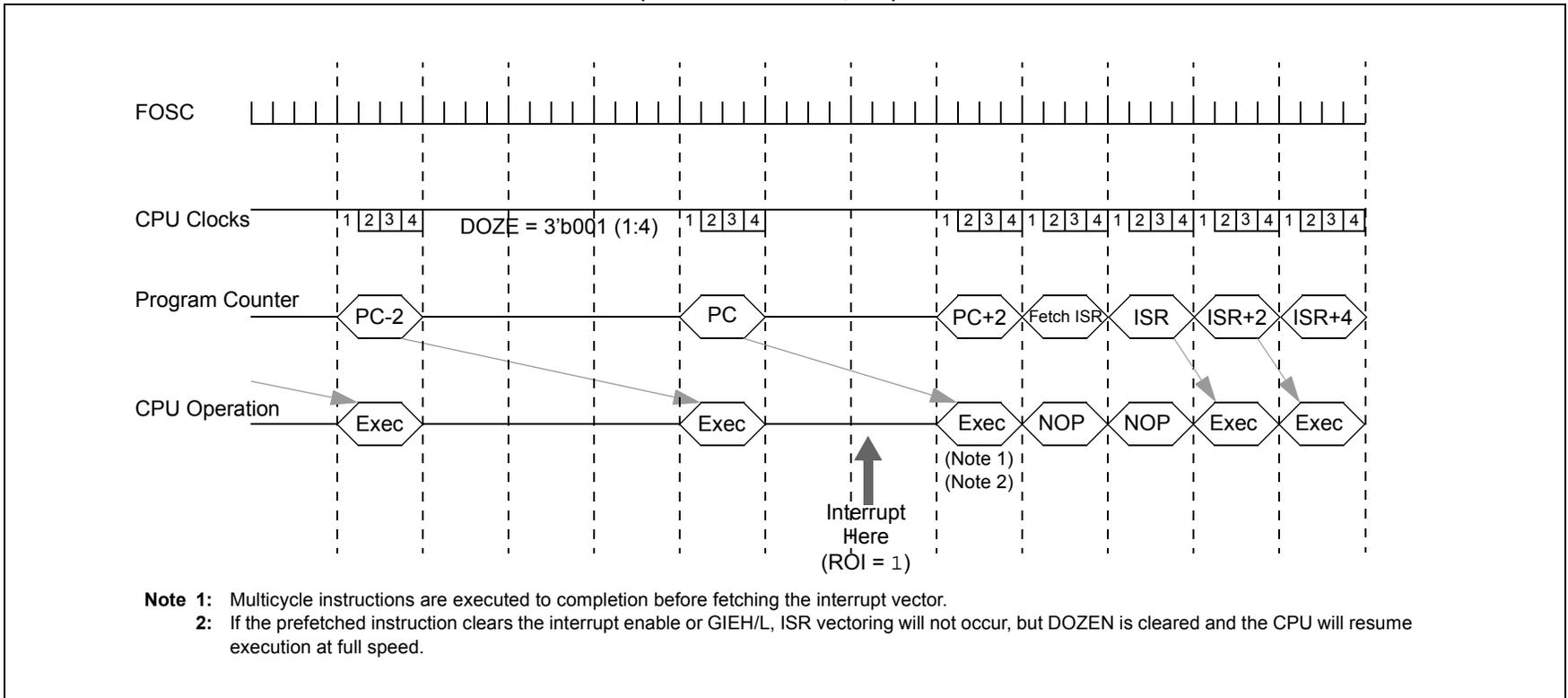


FIGURE 6-1: DOZE MODE OPERATION EXAMPLE (DOZE<2:0> = 001, 1:4)



9.2 Independent Clock Source

The WWDT can derive its time base from either the 31 kHz LFINTOSC or 31.25 kHz MFINTOSC internal oscillators, depending on the value of WDTE<1:0> Configuration bits.

If WDTE = 2'b1x, then the clock source will be enabled depending on the WDTCCS<2:0> Configuration bits.

If WDTE = 2'b01, the SEN bit should be set by software to enable WWDT, and the clock source is enabled by the WDTCS bits in the WDTCON1 register.

Time intervals in this chapter are based on a minimum nominal interval of 1 ms. See **Section 37.0 “Electrical Specifications”** for LFINTOSC and MFINTOSC tolerances.

9.3 WWDT Operating Modes

The Windowed Watchdog Timer module has four operating modes controlled by the WDTE<1:0> bits in Configuration Words. See Table 9-1.

9.3.1 WWDT IS ALWAYS ON

When the WDTE bits of Configuration Words are set to '11', the WWDT is always on.

WWDT protection is active during Sleep.

9.3.2 WWDT IS OFF IN SLEEP

When the WDTE bits of Configuration Words are set to '10', the WWDT is on, except in Sleep.

WWDT protection is not active during Sleep.

9.3.3 WWDT CONTROLLED BY SOFTWARE

When the WDTE bits of Configuration Words are set to '01', the WWDT is controlled by the SEN bit of the WDTCON0 register.

WWDT protection is unchanged by Sleep. See Table 9-1 for more details.

TABLE 9-1: WWDT OPERATING MODES

WDTE<1:0>	SEN	Device Mode	WWDT Mode
11	X	X	Active
10	X	Awake	Active
		Sleep	Disabled
01	1	X	Active
	0	X	Disabled
00	X	X	Disabled

9.4 Time-out Period

If the WDTCCPS<4:0> Configuration bits default to 5'b11111, then the WDTPS bits of the WDTCON0 register set the time-out period from 1 ms to 256 seconds (nominal). If any value other than the default value is assigned to WDTCCPS<4:0> Configuration bits, then the timer period will be based on the WDTCCPS<4:0> bits in the CONFIG3L register. After a Reset, the default time-out period is 2s.

9.5 Watchdog Window

The Windowed Watchdog Timer has an optional Windowed mode that is controlled by the WDTCCWS<2:0> Configuration bits and WINDOW<2:0> bits of the WDTCON1 register. In the Windowed mode, the CLRWDT instruction must occur within the allowed window of the WDT period. Any CLRWDT instruction that occurs outside of this window will trigger a window violation and will cause a WWDT Reset, similar to a WWDT time out. See Figure 9-2 for an example.

The window size is controlled by the WINDOW<2:0> Configuration bits, or the WINDOW<2:0> bits of WDTCON1, if WDTCCWS<2:0> = 111.

The five Most Significant bits of the WDTTMR register are used to determine whether the window is open, as defined by the WINDOW<2:0> bits of the WDTCON1 register.

In the event of a window violation, a Reset will be generated and the WDTWV bit of the PCON0 register will be cleared. This bit is set by a POR or can be set in firmware.

9.6 Clearing the WWDT

The WWDT is cleared when any of the following conditions occur:

- Any Reset
- Valid CLRWDT instruction is executed
- Device enters Sleep
- Exit Sleep by Interrupt
- WWDT is disabled
- Oscillator Start-up Timer (OST) is running
- Any write to the WDTCON0 or WDTCON1 registers

9.6.1 CLRWDT CONSIDERATIONS (WINDOWED MODE)

When in Windowed mode, the WWDT must be armed before a CLRWDT instruction will clear the timer. This is performed by reading the WDTCON0 register. Executing a CLRWDT instruction without performing such an arming action will trigger a window violation regardless of whether the window is open or not.

See Table 9-2 for more information.

10.3 PIC18 Instruction Cycle

10.3.1 CLOCKING SCHEME

The microcontroller clock input, whether from an internal or external source, is internally divided by four to generate four non-overlapping quadrature clocks (Q1, Q2, Q3 and Q4). Internally, the program counter is incremented on every Q1; the instruction is fetched from the program memory and latched into the instruction register during Q4. The instruction is decoded and executed during the following Q1 through Q4. The clocks and instruction execution flow are shown in Figure 10-2.

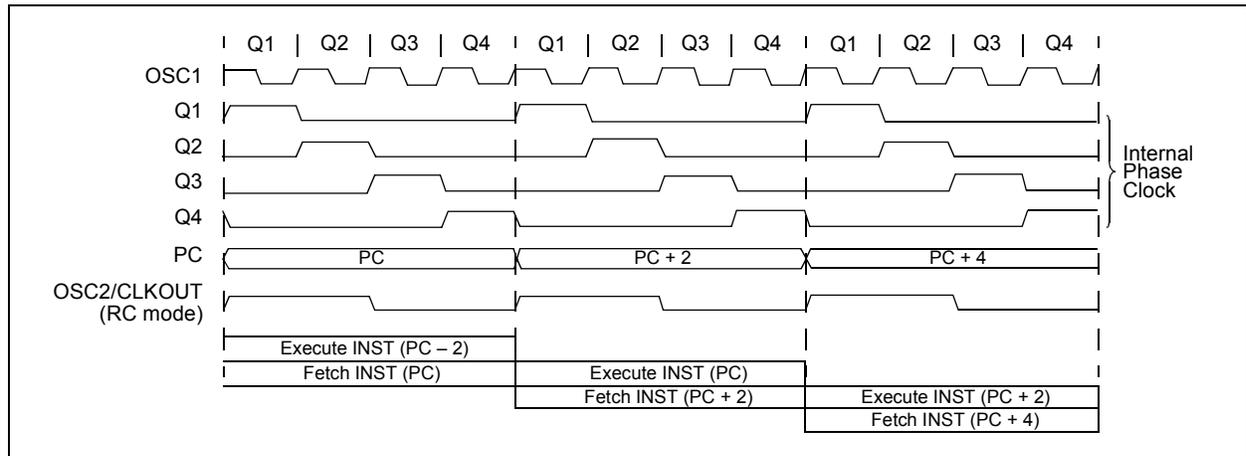
10.3.2 INSTRUCTION FLOW/PIPELINING

An “Instruction Cycle” consists of four Q cycles: Q1 through Q4. The instruction fetch and execute are pipelined in such a manner that a fetch takes one instruction cycle, while the decode and execute take another instruction cycle. However, due to the pipelining, each instruction effectively executes in one cycle. If an instruction causes the program counter to change (e.g., GOTO), then two cycles are required to complete the instruction (Example 10-3).

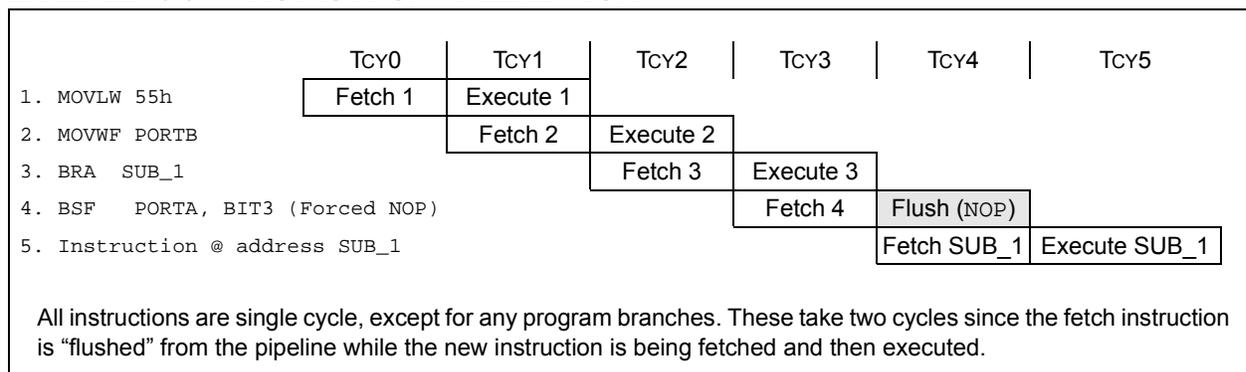
A fetch cycle begins with the Program Counter (PC) incrementing in Q1.

In the execution cycle, the fetched instruction is latched into the Instruction Register (IR) in cycle Q1. This instruction is then decoded and executed during the Q2, Q3 and Q4 cycles. Data memory is read during Q2 (operand read) and written during Q4 (destination write).

FIGURE 10-2: CLOCK/INSTRUCTION CYCLE



EXAMPLE 10-3: INSTRUCTION PIPELINE FLOW



EXAMPLE 11-4: WRITING TO PROGRAM FLASH MEMORY

```

        MOVLW    D'64'                ; number of bytes in erase block
        MOVWF   COUNTER
        MOVLW   BUFFER_ADDR_HIGH     ; point to buffer
        MOVWF   FSR0H
        MOVLW   BUFFER_ADDR_LOW
        MOVWF   FSR0L
        MOVLW   CODE_ADDR_UPPER      ; Load TBLPTR with the base
        MOVWF   TBLPTRU              ; address of the memory block
        MOVLW   CODE_ADDR_HIGH
        MOVWF   TBLPTRH
        MOVLW   CODE_ADDR_LOW
        MOVWF   TBLPTRL

READ_BLOCK
        TBLRD*+                       ; read into TABLAT, and inc
        MOVF    TABLAT, W             ; get data
        MOVWF   POSTINC0             ; store data
        DECFSZ  COUNTER              ; done?
        BRA     READ_BLOCK           ; repeat

MODIFY_WORD
        MOVLW   BUFFER_ADDR_HIGH     ; point to buffer
        MOVWF   FSR0H
        MOVLW   BUFFER_ADDR_LOW
        MOVWF   FSR0L
        MOVLW   NEW_DATA_LOW         ; update buffer word
        MOVWF   POSTINC0
        MOVLW   NEW_DATA_HIGH
        MOVWF   INDF0

ERASE_BLOCK
        MOVLW   CODE_ADDR_UPPER      ; load TBLPTR with the base
        MOVWF   TBLPTRU              ; address of the memory block
        MOVLW   CODE_ADDR_HIGH
        MOVWF   TBLPTRH
        MOVLW   CODE_ADDR_LOW
        MOVWF   TBLPTRL
        BCF     NVMCON1, NVMREG0     ; point to Program Flash Memory
        BSF     NVMCON1, NVMREG1     ; point to Program Flash Memory
        BSF     NVMCON1, WREN        ; enable write to memory
        BSF     NVMCON1, FREE        ; enable Erase operation
        BCF     INTCON, GIE          ; disable interrupts
        MOVLW   55h
        MOVWF   NVMCON2              ; write 55h
        MOVLW   AAh
        MOVWF   NVMCON2              ; write 0AAh
        BSF     NVMCON1, WR          ; start erase (CPU stall)
        BSF     INTCON, GIE          ; re-enable interrupts
        TBLRD*-                       ; dummy read decrement
        MOVLW   BUFFER_ADDR_HIGH     ; point to buffer
        MOVWF   FSR0H
        MOVLW   BUFFER_ADDR_LOW
        MOVWF   FSR0L
WRITE_BUFFER_BACK
        MOVLW   BlockSize            ; number of bytes in holding register
        MOVWF   COUNTER
        MOVLW   D'64'/BlockSize     ; number of write blocks in 64 bytes
        MOVWF   COUNTER2
    
```

15.0 I/O PORTS

TABLE 15-1: PORT AVAILABILITY PER DEVICE

Device	PORTA	PORTB	PORTC	PORTD	PORTE
PIC18(L)F26K40	•	•	•		•
PIC18(L)F45/46K40	•	•	•	•	•

Each port has eight registers to control the operation. These registers are:

- PORTx registers (reads the levels on the pins of the device)
- LATx registers (output latch)
- TRISx registers (data direction)
- ANSELx registers (analog select)
- WPUx registers (weak pull-up)
- INLVLx (input level control)
- SLRCONx registers (slew rate control)
- ODCONx registers (open-drain control)

Most port pins share functions with device peripherals, both analog and digital. In general, when a peripheral is enabled on a port pin, that pin cannot be used as a general purpose output; however, the pin can still be read.

The Data Latch (LATx registers) is useful for read-modify-write operations on the value that the I/O pins are driving.

A write operation to the LATx register has the same effect as a write to the corresponding PORTx register. A read of the LATx register reads of the values held in the I/O PORT latches, while a read of the PORTx register reads the actual I/O pin value.

Ports that support analog inputs have an associated ANSELx register. When an ANSELx bit is set, the digital input buffer associated with that bit is disabled.

Disabling the input buffer prevents analog signal levels on the pin between a logic high and low from causing excessive current in the logic input circuitry. A simplified model of a generic I/O port, without the interfaces to other peripherals, is shown in Figure 15-1.

FIGURE 15-1: GENERIC I/O PORT OPERATION

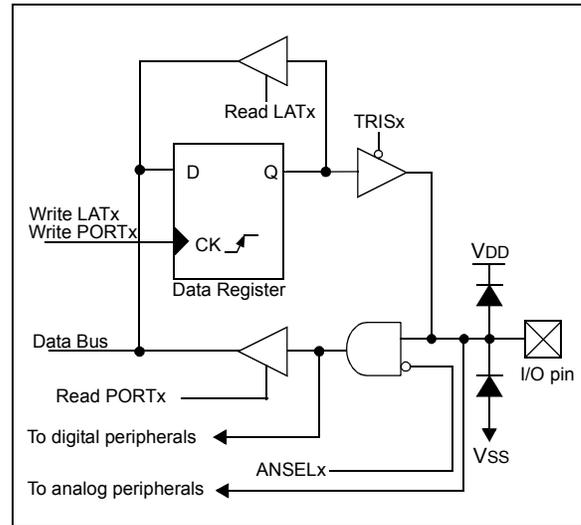
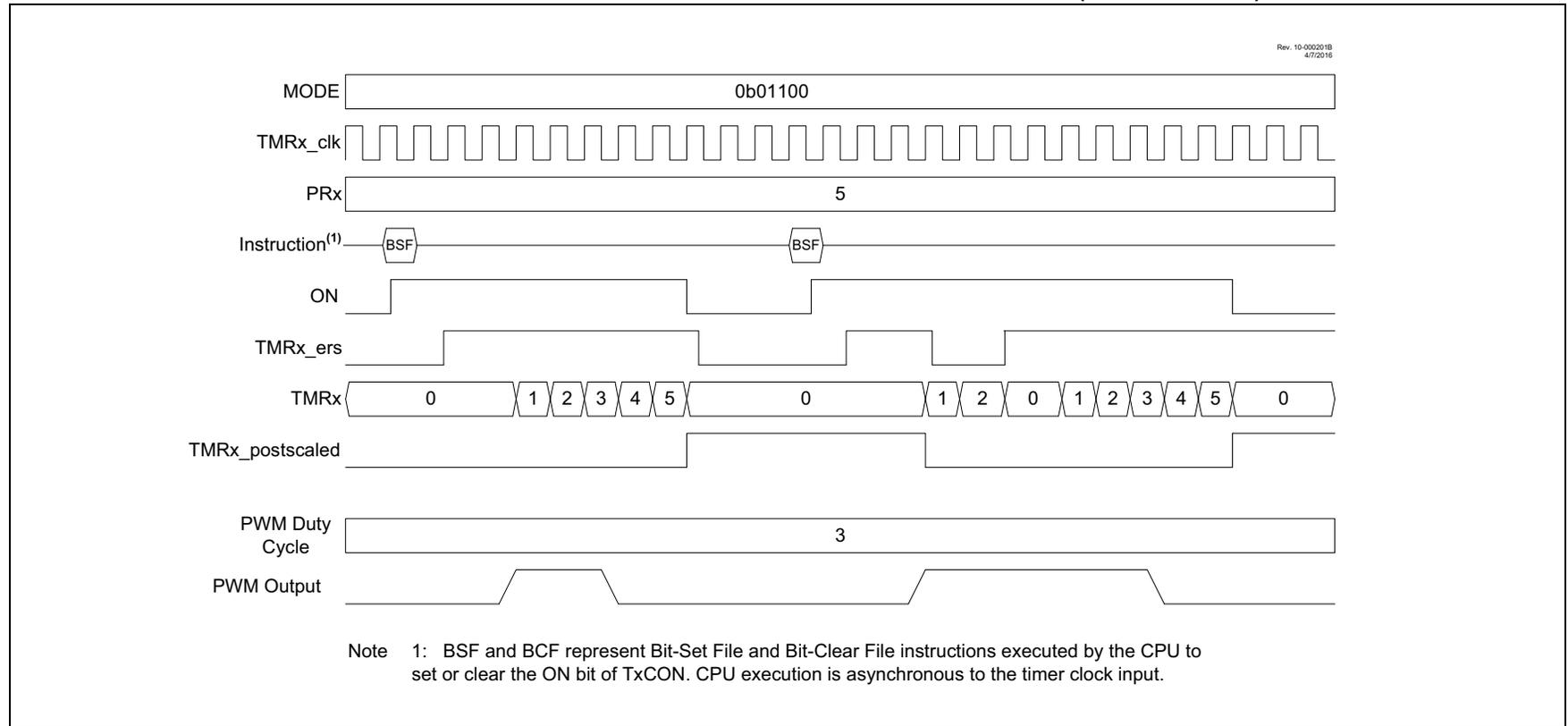


FIGURE 20-10: EDGE-TRIGGERED HARDWARE LIMIT ONE-SHOT MODE TIMING DIAGRAM (MODE = 01100)



PIC18LF26/45/46K40

REGISTER 21-5: CCPRxH: CCPx REGISTER HIGH BYTE

R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	
CCPRx<15:8>								
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 7-0

MODE = Capture Mode:

CCPRxH<7:0>: MSB of captured TMR1 value

MODE = Compare Mode:

CCPRxH<7:0>: MSB compared to TMR1 value

MODE = PWM Mode && FMT = 0:

CCPRxH<7:2>: Not used

CCPRxH<1:0>: CCPW<9:8> – Pulse-Width MS 2 bits

MODE = PWM Mode && FMT = 1:

CCPRxH<7:0>: CCPW<9:2> – Pulse-Width MS 8 bits

23.2 ZCD Logic Output

The ZCD module includes a Status bit, which can be read to determine whether the current source or sink is active. The ZCDOUT bit of the ZCDCON register is set when the current sink is active, and cleared when the current source is active. The ZCDOUT bit is affected by the polarity bit.

The ZCDOUT signal can also be used as input to other modules. This is controlled by the registers of the corresponding module. ZCDOUT can be used as follows:

- Gate source for TMR1/3/5
- Clock source for TMR2/4/6
- Reset source for TMR2/4/6

23.3 ZCD Logic Polarity

The ZCDPOL bit of the ZCDCON register inverts the ZCDOUT bit relative to the current source and sink output. When the ZCDPOL bit is set, a ZCDOUT high indicates that the current source is active, and a low output indicates that the current sink is active.

The ZCDPOL bit affects the ZCD interrupts.

23.4 ZCD Interrupts

An interrupt will be generated upon a change in the ZCD logic output when the appropriate interrupt enables are set. A rising edge detector and a falling edge detector are present in the ZCD for this purpose.

The ZCDIF bit of the PIR2 register will be set when either edge detector is triggered and its associated enable bit is set. The ZCDINTP enables rising edge interrupts and the ZCDINTN bit enables falling edge interrupts. Both are located in the ZCDCON register. Priority of the interrupt can be changed if the IPEN bit of the INTCON register is set. The ZCD interrupt can be made high or low priority by setting or clearing the ZCDIP bit of the IPR2 register.

To fully enable the interrupt, the following bits must be set:

- ZCDIE bit of the PIE2 register
- ZCDINTP bit of the ZCDCON register (for a rising edge detection)
- ZCDINTN bit of the ZCDCON register (for a falling edge detection)
- PEIE and GIE bits of the INTCON register

Changing the ZCDPOL bit will cause an interrupt, regardless of the level of the ZCDSEN bit.

The ZCDIF bit of the PIR2 register must be cleared in software as part of the interrupt service. If another edge is detected while this flag is being cleared, the flag will still be set at the end of the sequence.

23.5 Correcting for VCPINV offset

The actual voltage at which the ZCD switches is the reference voltage at the non-inverting input of the ZCD op amp. For external voltage source waveforms other than square waves, this voltage offset from zero causes the zero-cross event to occur either too early or too late.

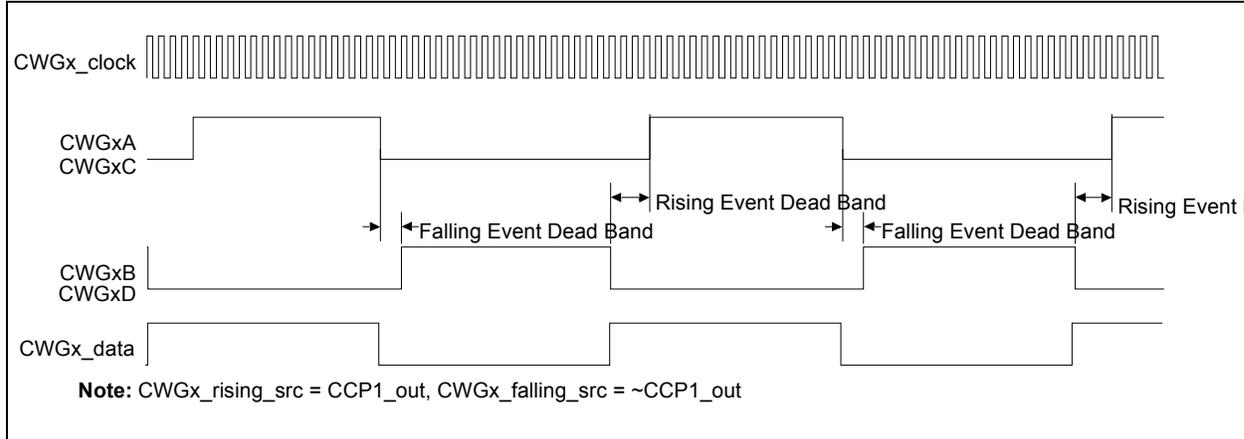
23.5.1 CORRECTION BY AC COUPLING

When the external voltage source is sinusoidal, the effects of the ZCPINV offset can be eliminated by isolating the external voltage source from the ZCD pin with a capacitor, in addition to the voltage reducing resistor. The capacitor will cause a phase shift resulting in the ZCD output switch in advance of the actual zero-crossing event. The phase shift will be the same for both rising and falling zero crossings, which can be compensated for by either delaying the CPU response to the ZCD switch by a timer or other means, or selecting a capacitor value large enough that the phase shift is negligible.

To determine the series resistor and capacitor values for this configuration, start by computing the impedance, Z , to obtain a peak current of $300\ \mu\text{A}$. Next, arbitrarily select a suitably large non-polar capacitor and compute its reactance, X_c , at the external voltage source frequency. Finally, compute the series resistor, capacitor peak voltage, and phase shift by the formulas shown in Equation 23-2.

When this technique is used and the input signal is not present, the ZCD will tend to oscillate. To avoid this oscillation, connect the ZCD pin to V_{DD} or GND with a high-impedance resistor such as 200K.

FIGURE 24-2: CWG1 HALF-BRIDGE MODE OPERATION



24.2.2 PUSH-PULL MODE

In Push-Pull mode, two output signals are generated, alternating copies of the input as illustrated in Figure 24-4. This alternation creates the push-pull effect required for driving some transformer-based power supply designs. Steering modes are not used in Push-Pull mode. A basic block diagram for the Push-Pull mode is shown in Figure 24-3.

The push-pull sequencer is reset whenever EN = 0 or if an auto-shutdown event occurs. The sequencer is clocked by the first input pulse, and the first output appears on CWG1A.

The unused outputs CWG1C and CWG1D drive copies of CWG1A and CWG1B, respectively, but with polarity controlled by the POLC and POLD bits of the CWG1CON1 register, respectively.

FIGURE 24-11: SIMPLIFIED CWG BLOCK DIAGRAM (OUTPUT STEERING MODES)

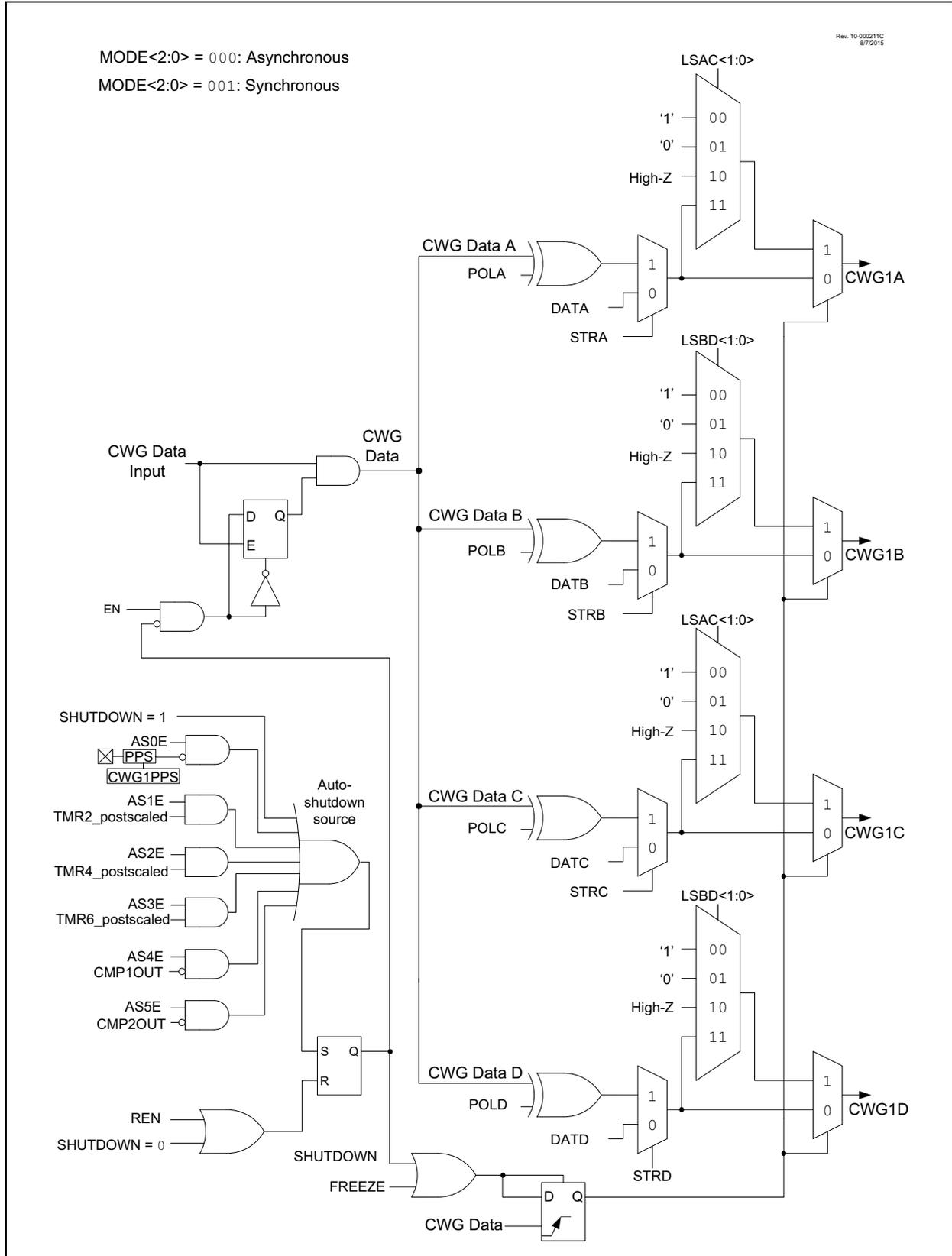
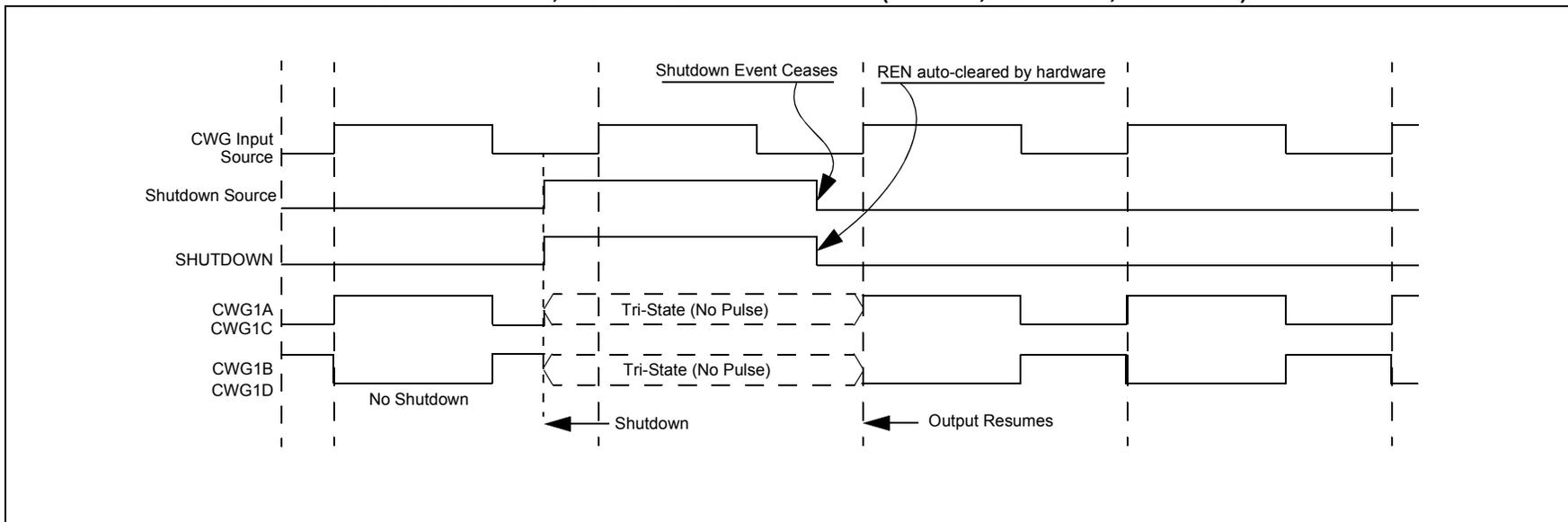


FIGURE 24-16: SHUTDOWN FUNCTIONALITY, AUTO-RESTART ENABLED (REN = 1, LSAC = 01, LSBD = 01)



PIC18LF26/45/46K40

REGISTER 24-6: CWG1AS0: CWG AUTO-SHUTDOWN CONTROL REGISTER 0

R/W/HS/HC-0/0	R/W-0/0	R/W-0/0	R/W-1/1	R/W-0/0	R/W-1/1	U-0	U-0
SHUTDOWN	REN	LSBD<1:0>		LSAC<1:0>		—	—
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	HS/HC = Bit is set/cleared by hardware
q = Value depends on condition		

bit 7 **SHUTDOWN:** Auto-Shutdown Event Status bit^(1,2)

- 1 = An auto-shutdown state is in effect
- 0 = No auto-shutdown event has occurred

bit 6 **REN:** Auto-Restart Enable bit

- 1 = Auto-restart is enabled
- 0 = Auto-restart is disabled

bit 5-4 **LSBD<1:0>:** CWG1B and CWG1D Auto-Shutdown State Control bits

- 11 = A logic '1' is placed on CWG1B/D when an auto-shutdown event occurs.
- 10 = A logic '0' is placed on CWG1B/D when an auto-shutdown event occurs.
- 01 = Pin is tri-stated on CWG1B/D when an auto-shutdown event occurs.
- 00 = The inactive state of the pin, including polarity, is placed on CWG1B/D after the required dead-band interval when an auto-shutdown event occurs.

bit 3-2 **LSAC<1:0>:** CWG1A and CWG1C Auto-Shutdown State Control bits

- 11 = A logic '1' is placed on CWG1A/C when an auto-shutdown event occurs.
- 10 = A logic '0' is placed on CWG1A/C when an auto-shutdown event occurs.
- 01 = Pin is tri-stated on CWG1A/C when an auto-shutdown event occurs.
- 00 = The inactive state of the pin, including polarity, is placed on CWG1A/C after the required dead-band interval when an auto-shutdown event occurs.

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

- Note 1:** This bit may be written while EN = 0 (Register 24-1), to place the outputs into the shutdown configuration.
- Note 2:** The outputs will remain in auto-shutdown state until the next rising edge of the CWG data input after this bit is cleared.

PIC18LF26/45/46K40

REGISTER 25-3: MDCARH: MODULATION HIGH CARRIER CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	—	CHS<2:0> ⁽¹⁾		
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 7-3 **Unimplemented:** Read as '0'
bit 2-0 **CHS<2:0>:** Modulator Carrier High Selection bits
See Table 25-2 for signal list

REGISTER 25-4: MDCARL: MODULATION LOW CARRIER CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	—	CLS<2:0> ⁽¹⁾		
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 7-3 **Unimplemented:** Read as '0'
bit 2-0 **CLS<2:0>:** Modulator Carrier Low Input Selection bits
See Table 25-2 for signal list

TABLE 25-2: MDCARH/MDCARL SELECTION MUX CONNECTIONS

MDCARH			MDCARL		
CHS<2:0>		Connection	CLS<2:0>		Connection
111	7	PWM4 OUT	111	7	PWM4 OUT
110	6	PWM3 OUT	110	6	PWM3 OUT
101	5	CCP2 OUT	101	5	CCP2 OUT
100	4	CCP1 OUT	100	4	CCP1 OUT
011	3	CLKREF output	011	3	CLKREF output
010	2	HFINTOSC	010	2	HFINTOSC
001	1	FOSC (system clock)	001	1	FOSC (system clock)
000	0	Pin selected by MDCARHPPS	000	0	Pin selected by MDCARLPPS

FIGURE 26-18: I²C SLAVE, 7-BIT ADDRESS, TRANSMISSION (AHEN = 0)

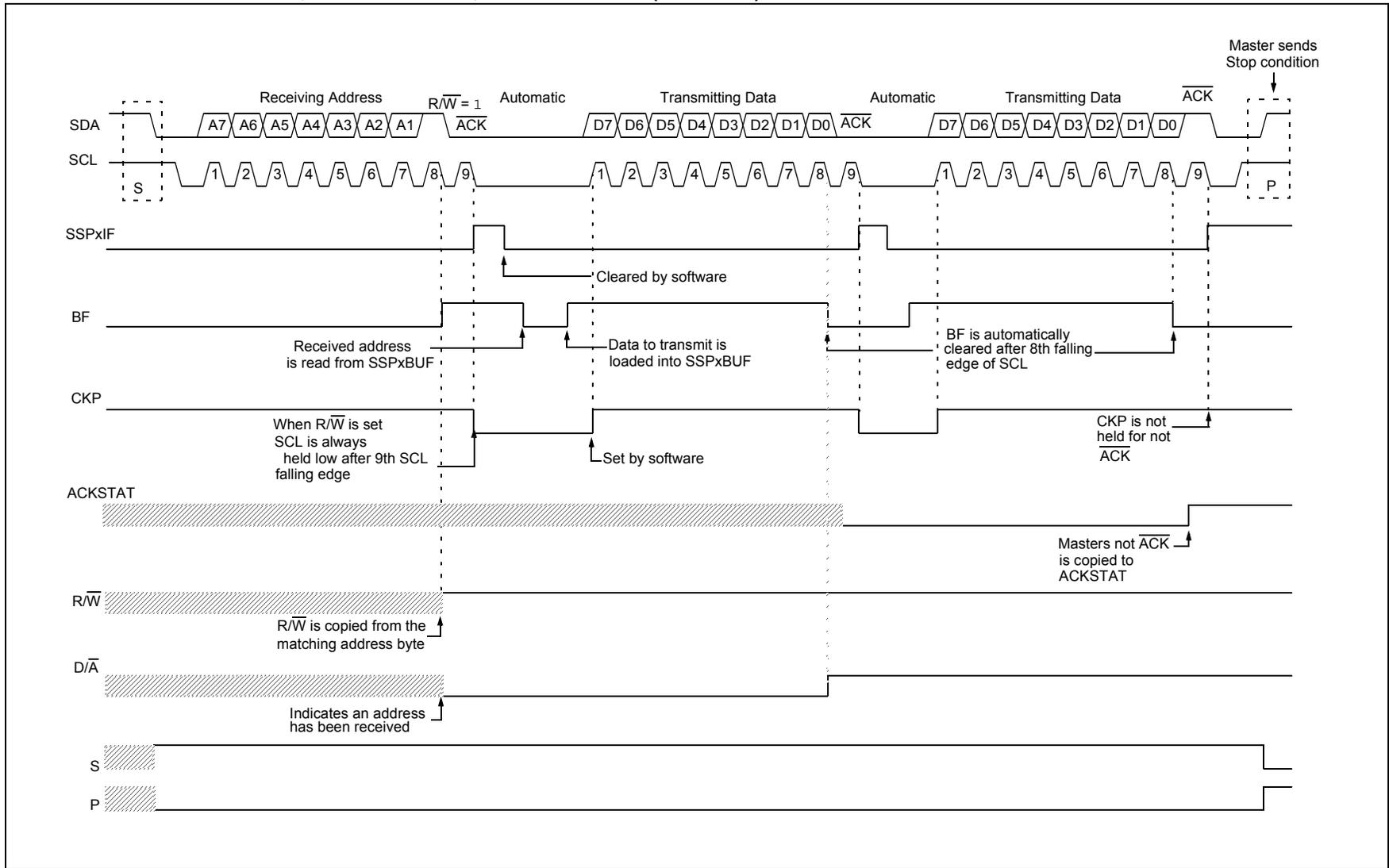
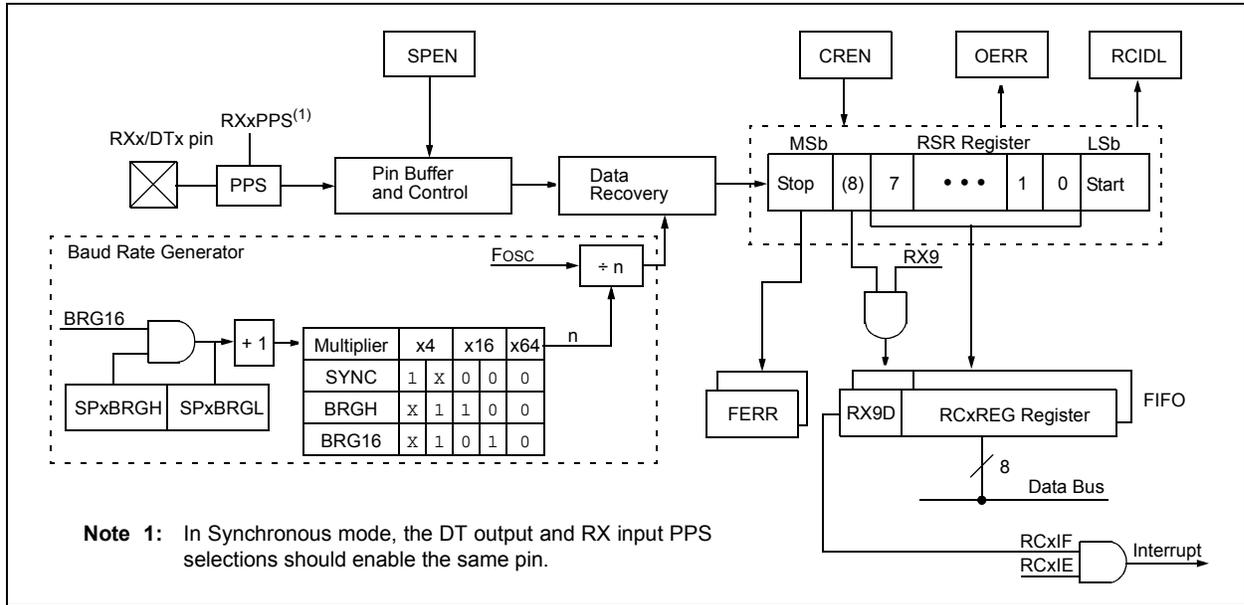


FIGURE 27-2: EUSART RECEIVE BLOCK DIAGRAM



The operation of the EUSART module is controlled through three registers:

- Transmit Status and Control (TXxSTA)
- Receive Status and Control (RCxSTA)
- Baud Rate Control (BAUDxCON)

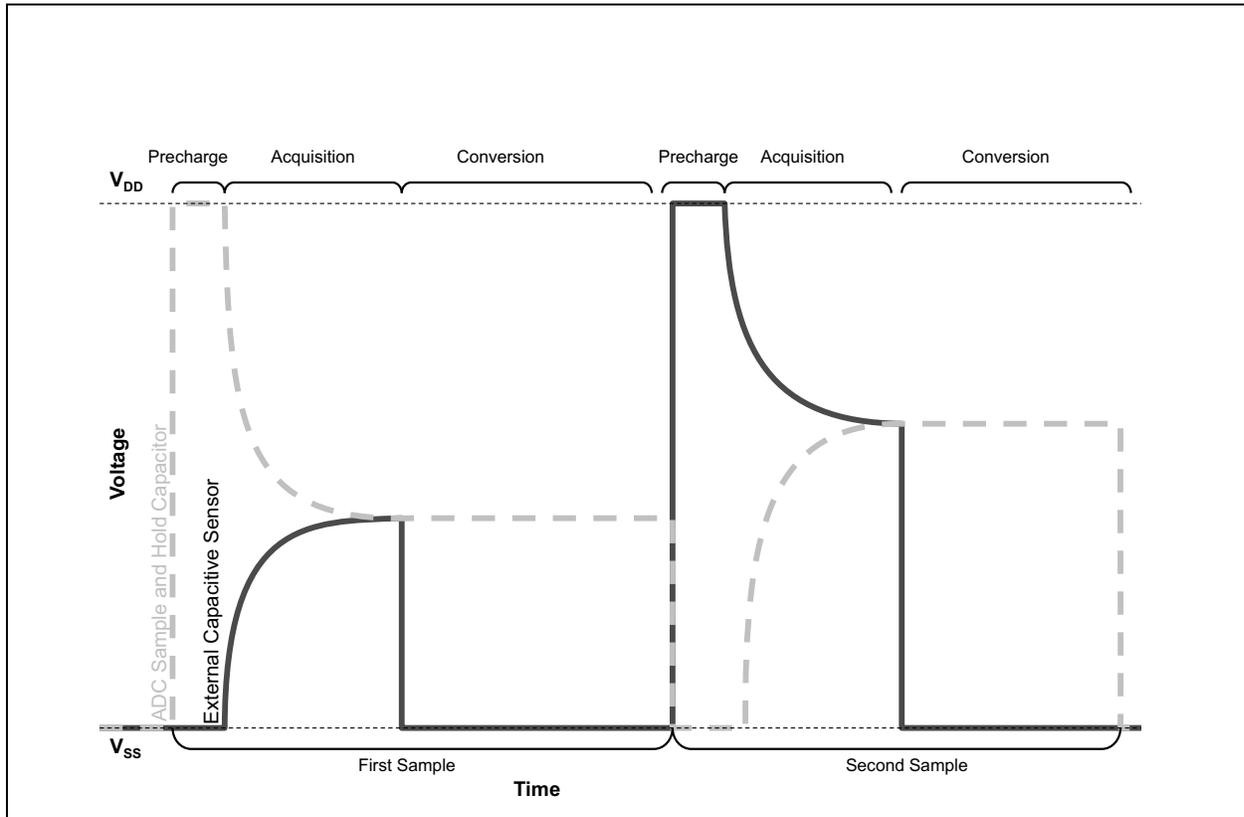
These registers are detailed in Register 27-1, Register 27-2 and Register 27-3, respectively.

The RXx/DTx and TXx/CKx input pins are selected with the RXxPPS and TXxPPS registers, respectively. TXx, CKx, and DTx output pins are selected with each pin's RxyPPS register. Since the RX input is coupled with the DT output in Synchronous mode, it is the user's responsibility to select the same pin for both of these functions when operating in Synchronous mode. The EUSART control logic will control the data direction drivers automatically.

31.4.1 CVD OPERATION

A CVD operation begins with the ADC's internal sample and hold capacitor (C_{HOLD}) being disconnected from the path which connects it to the external capacitive sensor node. While disconnected, C_{HOLD} is precharged to V_{DD} or V_{SS} , while the path to the sensor node is also discharged to V_{DD} or V_{SS} . Typically, this node is discharged to the level opposite that of C_{HOLD} . When the precharge phase is complete, the $V_{\text{DD}}/V_{\text{SS}}$ bias paths for the two nodes are shut off and C_{HOLD} and the path to the external sensor node are re-connected, at which time the acquisition phase of the CVD operation begins. During acquisition, a capacitive voltage divider is formed between the precharged C_{HOLD} and sensor nodes, which results in a final voltage level setting on C_{HOLD} , which is determined by the capacitances and precharge levels of the two nodes. After acquisition, the ADC converts the voltage level on C_{HOLD} . This process is then repeated with the selected precharge levels for both the C_{HOLD} and the inverted sensor nodes. Figure 31-7 shows the waveform for two inverted CVD measurements, which is known as differential CVD measurement.

FIGURE 31-7: DIFFERENTIAL CVD MEASUREMENT WAVEFORM



PIC18(L)F26/45/46K40

REGISTER 31-32: ADOACT: ADC AUTO CONVERSION TRIGGER CONTROL REGISTER

U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	
—	—	—	ADACT<4:0>					
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 7-5	Unimplemented: Read as '0'
bit 4-0	ADACT<4:0>: Auto-Conversion Trigger Select Bits 11111 = Software write to ADPCH 11110 = Reserved, do not use 11101 = Software read of ADRESH 11100 = Software read of ADERRH 11011 = Reserved, do not use • • • 10000 = Reserved, do not use 01111 = Interrupt-on-change Interrupt Flag 01110 = C2_out 01101 = C1_out 01100 = PWM4_out 01011 = PWM3_out 01010 = CCP2_trigger 01001 = CCP1_trigger 01000 = TMR6_postscaled 00111 = TMR5_overflow 00110 = TMR4_postscaled 00101 = TMR3_overflow 00100 = TMR2_postscaled 00011 = TMR1_overflow 00010 = TMR0_overflow 00001 = Pin selected by ADACTPPS 00000 = External Trigger Disabled

PIC18(L)F26/45/46K40

BTG

Bit Toggle f

Syntax:	BTG f, b {,a}								
Operands:	$0 \leq f \leq 255$ $0 \leq b < 7$ $a \in [0,1]$								
Operation:	$(\overline{f\langle b \rangle}) \rightarrow f\langle b \rangle$								
Status Affected:	None								
Encoding:	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="padding: 2px;">0111</td> <td style="padding: 2px;">bbba</td> <td style="padding: 2px;">ffff</td> <td style="padding: 2px;">ffff</td> </tr> </table>	0111	bbba	ffff	ffff				
0111	bbba	ffff	ffff						
Description:	<p>Bit 'b' in data memory location 'f' is inverted.</p> <p>If 'a' is '0', the Access Bank is selected.</p> <p>If 'a' is '1', the BSR is used to select the GPR bank.</p> <p>If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See Section 35.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode" for details.</p>								
Words:	1								
Cycles:	1								
Q Cycle Activity:	<table border="1" style="display: inline-table; border-collapse: collapse; text-align: center;"> <tr> <td style="padding: 2px;">Q1</td> <td style="padding: 2px;">Q2</td> <td style="padding: 2px;">Q3</td> <td style="padding: 2px;">Q4</td> </tr> <tr> <td style="padding: 2px;">Decode</td> <td style="padding: 2px;">Read register 'f'</td> <td style="padding: 2px;">Process Data</td> <td style="padding: 2px;">Write register 'f'</td> </tr> </table>	Q1	Q2	Q3	Q4	Decode	Read register 'f'	Process Data	Write register 'f'
Q1	Q2	Q3	Q4						
Decode	Read register 'f'	Process Data	Write register 'f'						

Example: BTG PORTC, 4, 0

Before Instruction:

PORTC = 0111 0101 [75h]

After Instruction:

PORTC = 0110 0101 [65h]

BOV

Branch if Overflow

Syntax:	BOV n				
Operands:	$-128 \leq n \leq 127$				
Operation:	if OVERFLOW bit is '1' $(PC) + 2 + 2n \rightarrow PC$				
Status Affected:	None				
Encoding:	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="padding: 2px;">1110</td> <td style="padding: 2px;">0100</td> <td style="padding: 2px;">nnnn</td> <td style="padding: 2px;">nnnn</td> </tr> </table>	1110	0100	nnnn	nnnn
1110	0100	nnnn	nnnn		
Description:	<p>If the OVERFLOW bit is '1', then the program will branch.</p> <p>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.</p>				
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 BOV Jump

Before Instruction

PC = address (HERE)

After Instruction

If OVERFLOW = 1;
PC = address (Jump)

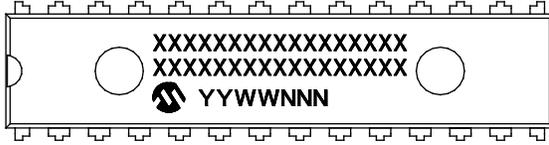
If OVERFLOW = 0;
PC = address (HERE + 2)

PIC18(L)F26/45/46K40

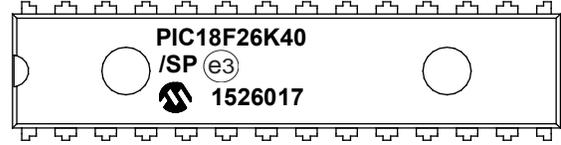
39.0 PACKAGING INFORMATION

Package Marking Information

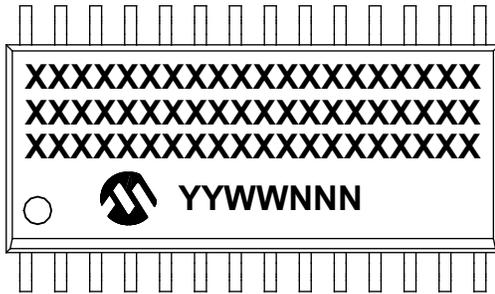
28-Lead SPDIP (.300")



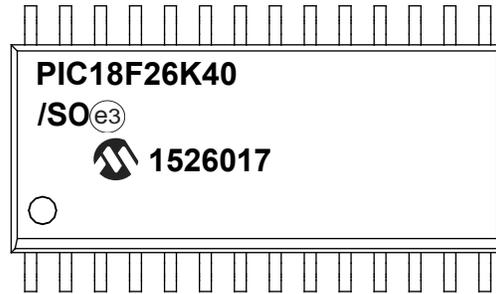
Example



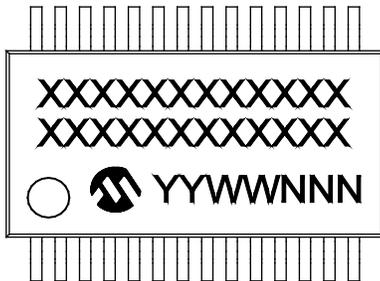
28-Lead SOIC (7.50 mm)



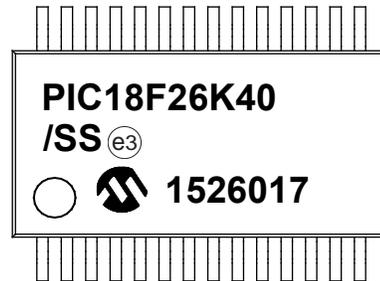
Example



28-Lead SSOP (5.30 mm)



Example



Legend:	XX...X	Customer-specific information or Microchip part number
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week '01')
	NNN	Alphanumeric traceability code
	(e3)	Pb-free JEDEC® designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free JEDEC designator (e3) can be found on the outer packaging for this package.
Note:	In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.	