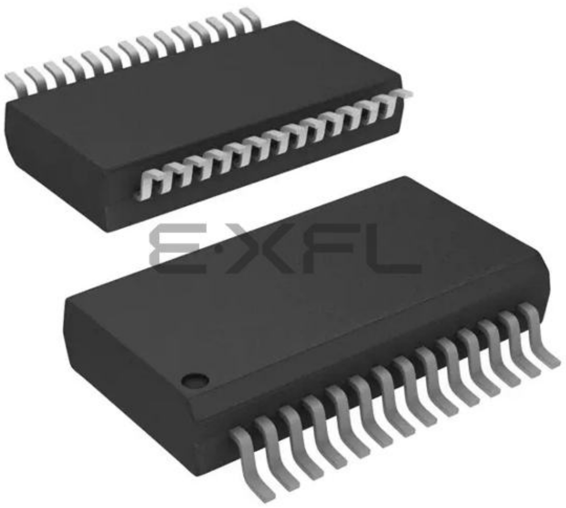


Welcome to E-XFL.COM

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	ECANbus, I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	24
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 ~ 5.5V
Data Converters	A/D 8x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°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/pic18f26k80-e-ss

TABLE 1-5: PIC18F4XK80 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number		Pin Type	Buffer Type	Description
	PDIP	QFN/ TQFP			
					PORTB is a bidirectional I/O port.
RB0/AN10/FLT0/INT0	33	8			
RB0			I/O	ST/ CMOS	Digital I/O.
AN10			I	Analog	Analog Input 10.
FLT0			I	ST	Enhanced PWM Fault input for ECCP1.
INT0			I	ST	External Interrupt 0.
RB1/AN8/CTDIN/INT1	34	9			
RB1			I/O	ST/ CMOS	Digital I/O.
AN8			I	Analog	Analog Input 8.
CTDIN			I	ST	CTMU pulse delay input.
INT1			I	ST	External Interrupt 1.
RB2/CANTX/CTED1/ INT2	35	10			
RB2			I/O	ST/ CMOS	Digital I/O.
CANTX			O	CMOS	CAN bus TX.
CTED1			I	ST	CTMU Edge 1 input.
INT2			I	ST	External Interrupt 2.
RB3/CANRX/CTED2/ INT3	36	11			
RB3			I/O	ST/ CMOS	Digital I/O.
CANRX			I	ST	CAN bus RX.
CTED2			I	ST	CTMU Edge 2 input.
INT3			I	ST	External Interrupt 3.
RB4/AN9/CTPLS/KBI0	37	14			
RB4			I/O	ST/ CMOS	Digital I/O.
AN9			I	Analog	Analog Input 9.
CTPLS			O	ST	CTMU pulse generator output.
KBI0			I	ST	Interrupt-on-change pin.
RB5/T0CKI/T3CKI/CCP5/ KBI1	38	15			
RB5			I/O	ST/ CMOS	Digital I/O.
T0CKI			I	ST	Timer0 external clock input.
T3CKI			I	ST	Timer3 external clock input.
CCP5			I/O	ST	Capture 5 input/Compare 5 output/PWM5 output.
KBI1			I	ST	Interrupt-on-change pin.

Legend: I^2C™ = I^2C/SMBus input buffer; CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; Analog = Analog input; I = Input; O = Output; P = Power

TABLE 5-4: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices			Power-on Reset, Brown-out Reset	MCLR Resets, WDT Reset, RESET Instruction, Stack Resets	Wake-up via WDT or Interrupt
B2D1	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	xxxx xxxx	uuuu uuuu	uuuu uuuu
B2D0	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	xxxx xxxx	uuuu uuuu	uuuu uuuu
B2DLC	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	-xxx xxxx	-uuu uuuu	-uuu uuuu
B2EIDL	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	xxxx xxxx	uuuu uuuu	uuuu uuuu
B2EIDH	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	xxxx xxxx	uuuu uuuu	uuuu uuuu
B2SIDL	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	xxxx x-xx	uuuu u-uu	uuuu u-uu
B2SIDH	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	xxxx xxxx	uuuu uuuu	uuuu uuuu
B2CON	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	0000 0000	0000 0000	uuuu uuuu
CANCON_RO8	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	1000 0000	1000 0000	uuuu uuuu
B1D7	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	xxxx xxxx	uuuu uuuu	uuuu uuuu
B1D6	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	xxxx xxxx	uuuu uuuu	uuuu uuuu
B1D5	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	xxxx xxxx	uuuu uuuu	uuuu uuuu
B1D4	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	xxxx xxxx	uuuu uuuu	uuuu uuuu
B1D3	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	xxxx xxxx	uuuu uuuu	uuuu uuuu
B1D2	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	xxxx xxxx	uuuu uuuu	uuuu uuuu
B1D1	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	xxxx xxxx	uuuu uuuu	uuuu uuuu
B1D0	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	xxxx xxxx	uuuu uuuu	uuuu uuuu
B1DLC	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	-xxx xxxx	-uuu uuuu	-uuu uuuu
B1EIDL	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	xxxx xxxx	uuuu uuuu	uuuu uuuu
B1EIDH	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	xxxx xxxx	uuuu uuuu	uuuu uuuu
B1SIDL	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	xxxx x-xx	uuuu u-uu	uuuu u-uu
B1SIDH	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	xxxx xxxx	uuuu uuuu	uuuu uuuu
B1CON	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	0000 0000	0000 0000	uuuu uuuu
CANCON_RO9	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	1000 0000	1000 0000	uuuu uuuu
CANSTAT_RO	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	1000 0000	1000 0000	uuuu uuuu
B0D7	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	xxxx xxxx	uuuu uuuu	uuuu uuuu
B0D6	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	xxxx xxxx	uuuu uuuu	uuuu uuuu
B0D5	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	xxxx xxxx	uuuu uuuu	uuuu uuuu
B0D4	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	xxxx xxxx	uuuu uuuu	uuuu uuuu
B0D3	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	xxxx xxxx	uuuu uuuu	uuuu uuuu
B0D2	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	xxxx xxxx	uuuu uuuu	uuuu uuuu
B0D1	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	xxxx xxxx	uuuu uuuu	uuuu uuuu
B0D0	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	xxxx xxxx	uuuu uuuu	uuuu uuuu
B0DLC	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	-xxx xxxx	-uuu uuuu	-uuu uuuu
B0EIDL	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	xxxx xxxx	uuuu uuuu	uuuu uuuu

Legend: u = unchanged; x = unknown; - = unimplemented bit, read as ‘0’; q = value depends on condition. Shaded cells indicate conditions do not apply for the designated device.

Note 1: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).

2: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).

3: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the TOSU, TOSH and TOSL are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.

4: See Table 5-3 for Reset value for specific conditions.

5: Bits 6 and 7 of PORTA, LATA and TRISA are enabled, depending on the oscillator mode selected. When not enabled as PORTA pins, they are disabled and read as ‘0’.

6.4.3.1 FSR Registers and the INDF Operand

At the core of Indirect Addressing are three sets of registers: FSR0, FSR1 and FSR2. Each represents a pair of 8-bit registers: FSRnH and FSRnL. The four upper bits of the FSRnH register are not used, so each FSR pair holds a 12-bit value. This represents a value that can address the entire range of the data memory in a linear fashion. The FSR register pairs, then, serve as pointers to data memory locations.

Indirect Addressing is accomplished with a set of Indirect File Operands, INDF0 through INDF2. These can be thought of as "virtual" registers. The operands are mapped in the SFR space, but are not physically implemented. Reading or writing to a particular INDF register actually accesses its corresponding FSR register pair. A read from INDF1, for example, reads the data at the address indicated by FSR1H:FSR1L.

Instructions that use the INDF registers as operands actually use the contents of their corresponding FSR as a pointer to the instruction's target. The INDF operand is just a convenient way of using the pointer.

Because Indirect Addressing uses a full 12-bit address, data RAM banking is not necessary. Thus, the current contents of the BSR and the Access RAM bit have no effect on determining the target address.

FIGURE 6-8: INDIRECT ADDRESSING

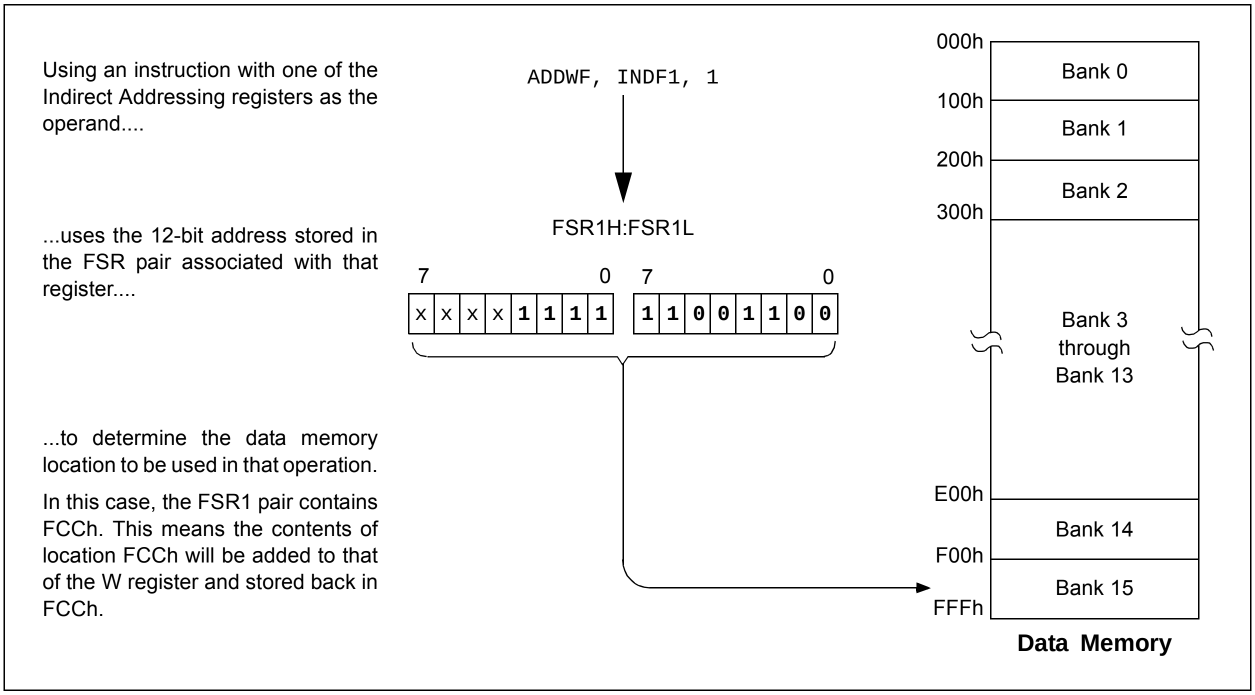


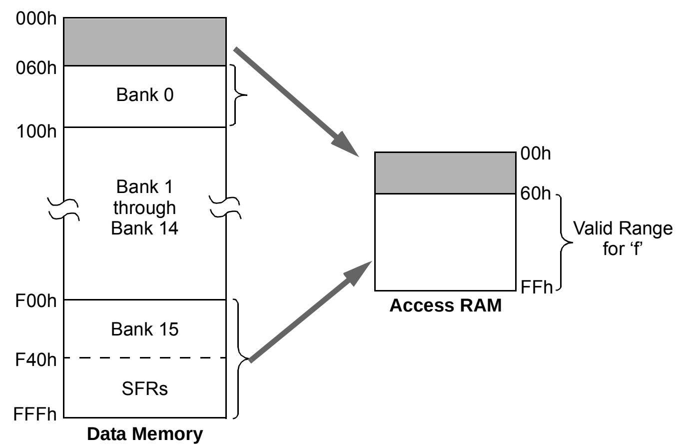
FIGURE 6-9: COMPARING ADDRESSING OPTIONS FOR BIT-ORIENTED AND BYTE-ORIENTED INSTRUCTIONS (EXTENDED INSTRUCTION SET ENABLED)

EXAMPLE INSTRUCTION: `ADDWF, f, d, a` (Opcode: `0010 01da ffff ffff`)

When a = `0` and f ≥ 60h:

The instruction executes in Direct Forced mode. 'f' is interpreted as a location in the Access RAM between 060h and FFFh. This is the same as locations, F60h to FFFh, (Bank 15) of data memory.

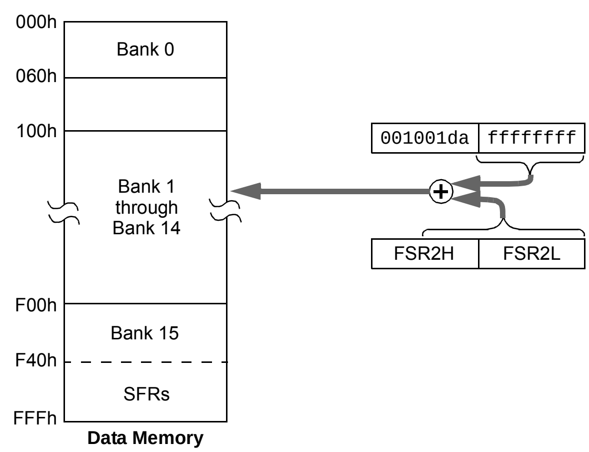
Locations below 060h are not available in this addressing mode.



When a = `0` and f ≤ 5Fh:

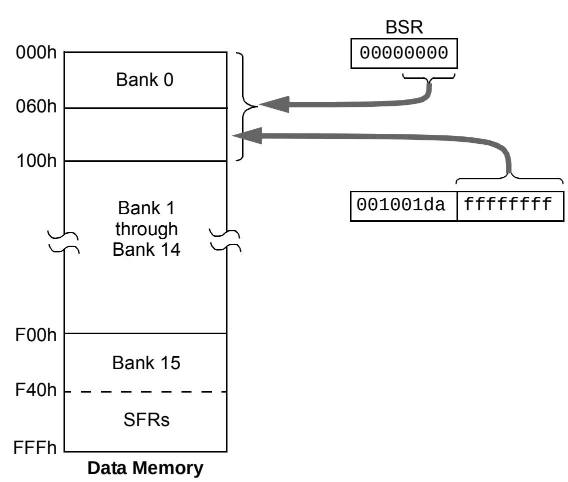
The instruction executes in Indexed Literal Offset mode. 'f' is interpreted as an offset to the address value in FSR2. The two are added together to obtain the address of the target register for the instruction. The address can be anywhere in the data memory space.

Note that in this mode, the correct syntax is now:
`ADDWF [k], d`
where 'k' is the same as 'f'.



When a = `1` (all values of f):

The instruction executes in Direct mode (also known as Direct Long mode). 'f' is interpreted as a location in one of the 16 banks of the data memory space. The bank is designated by the Bank Select Register (BSR). The address can be in any implemented bank in the data memory space.



11.9 Parallel Slave Port

PORTD can function as an 8-bit-wide Parallel Slave Port (PSP), or microprocessor port, when control bit, PSPMODE (PSPCON<4>), is set. The port is asynchronously readable and writable by the external world through the $\overline{\text{RD}}$ control input pin (RE0/AN5/$\overline{\text{RD}}$) and $\overline{\text{WR}}$ control input pin (RE1/AN6/C1OUT/$\overline{\text{WR}}$).

Note:	The Parallel Slave Port is available only on 40/44-pin and 64-pin devices.

The PSP can directly interface to an 8-bit microprocessor data bus. The external microprocessor can read or write the PORTD latch as an eight-bit latch.

Setting bit, PSPMODE, enables port pin, RE0/AN5/$\overline{\text{RD}}$, to be the $\overline{\text{RD}}$ input, RE1/AN6/C1OUT/$\overline{\text{WR}}$ to be the $\overline{\text{WR}}$ input and RE2/AN7/C2OUT/$\overline{\text{CS}}$ to be the $\overline{\text{CS}}$ (Chip Select) input. For this functionality, the corresponding data direction bits of the TRISE register (TRISE<2:0>) must be configured as inputs (= `111`).

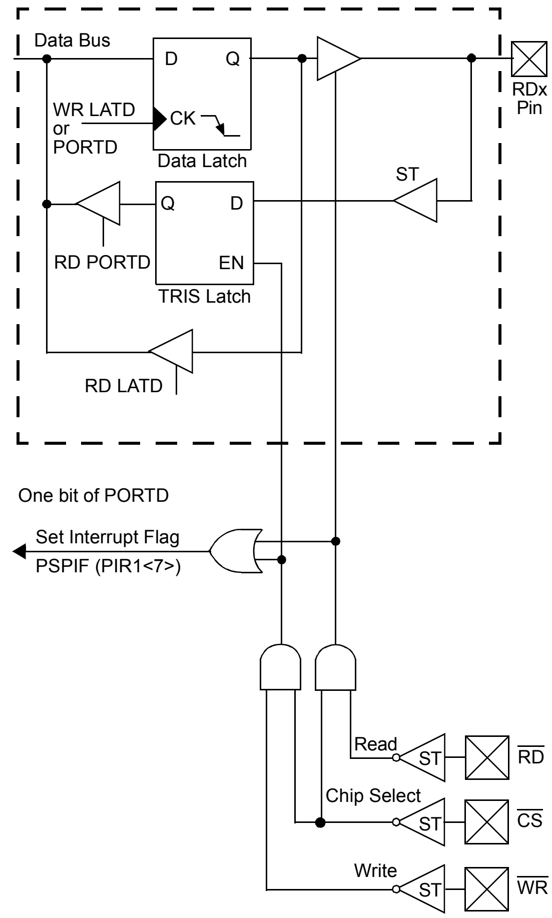
A write to the PSP occurs when both the $\overline{\text{CS}}$ and $\overline{\text{WR}}$ lines are first detected low and ends when either are detected high. The PSPIF and IBF flag bits (PIR1<7> and PSPCON<7>, respectively) are set when the write ends.

A read from the PSP occurs when both the $\overline{\text{CS}}$ and $\overline{\text{RD}}$ lines are first detected low. The data in PORTD is read out and the OBF bit (PSPCON<6>) is set. If the user writes new data to PORTD to set OBF, the data is immediately read out, but the OBF bit is not set.

When either the $\overline{\text{CS}}$ or $\overline{\text{RD}}$ line is detected high, the PORTD pins return to the input state and the PSPIF bit is set. User applications should wait for PSPIF to be set before servicing the PSP. When this happens, the IBF and OBF bits can be polled and the appropriate action taken.

The timing for the control signals in Write and Read modes is shown in Figure 11-4 and Figure 11-5, respectively.

FIGURE 11-3: PORTD AND PORTE BLOCK DIAGRAM (PARALLEL SLAVE PORT)



Note: The I/O pin has protection diodes to V_{DD} and V_{SS}.

19.1 CCP Module Configuration

Each Capture/Compare/PWM module is associated with a control register (generically, CCPxCON) and a data register (CCPRx). The data register, in turn, is comprised of two 8-bit registers: CCPRxL (low byte) and CCPRxH (high byte). All registers are both readable and writable.

19.1.1 CCP MODULES AND TIMER RESOURCES

The CCP modules utilize Timers, 1 through 4, varying with the selected mode. Various timers are available to the CCP modules in Capture, Compare or PWM modes, as shown in Table 19-1.

TABLE 19-1: CCP MODE – TIMER RESOURCE

CCP Mode	Timer Resource
Capture	Timer1 or Timer3
Compare	
PWM	Timer2 or Timer4

The assignment of a particular timer to a module is determined by the Timer to CCP enable bits in the CCPTMRS register (see Register 19-2). All of the modules may be active at once and may share the same timer resource if they are configured to operate in the same mode (Capture/Compare or PWM) at the same time.

The CCPTMRS register selects the timers for CCP modules, 2, 3, 4 and 5. The possible configurations are shown in Table 19-2.

TABLE 19-2: TIMER ASSIGNMENTS FOR CCP MODULES 2, 3, 4 AND 5

CCPTMRS Register											
CCP2			CCP3			CCP4			CCP5		
C2TSEL	Capture/ Compare Mode	PWM Mode	C3TSEL	Capture/ Compare Mode	PWM Mode	C4TSEL	Capture/ Compare Mode	PWM Mode	C5TSEL	Capture/ Compare Mode	PWM Mode
0	TMR1	TMR2	0	TMR1	TMR2	0	TMR1	TMR2	0 0	TMR1	TMR2
1	TMR3	TMR4	1	TMR3	TMR4	1	TMR3	TMR4	0 1	TMR3	TMR4

19.1.2 OPEN-DRAIN OUTPUT OPTION

When operating in Output mode (the Compare or PWM modes), the drivers for the CCPx pins can be optionally configured as open-drain outputs. This feature allows the voltage level on the pin to be pulled to a higher level through an external pull-up resistor and allows the output to communicate with external circuits without the need for additional level shifters.

The open-drain output option is controlled by the CCPxOD bits (ODCON<6:2>). Setting the appropriate bit configures the pin for the corresponding module for open-drain operation.

20.3 Compare Mode

In Compare mode, the 16-bit CCPR1 register value is constantly compared against the Timer register pair value selected in the CCPTMR1 register. When a match occurs, the ECCP1 pin can be:

- Driven high
- Driven low
- Toggled (high-to-low or low-to-high)
- Unchanged (that is, reflecting the state of the I/O latch)

The action on the pin is based on the value of the mode select bits (CCP1M<3:0>). At the same time, the interrupt flag bit, CCP1IF, is set.

20.3.1 ECCP PIN CONFIGURATION

Users must configure the ECCP1 pin as an output by clearing the appropriate TRIS bit.

Note:	Clearing the CCP1CON register will force the ECCP1 compare output latch (depending on device configuration) to the default low level. This is not the port I/O data latch.

20.3.2 TIMER1/2/3/4 MODE SELECTION

Timer1, 2, 3 or 4 must be running in Timer mode or Synchronized Counter mode if the ECCP module is using the compare feature. In Asynchronous Counter mode, the compare operation will not work reliably.

20.3.3 SOFTWARE INTERRUPT MODE

When the Generate Software Interrupt mode is chosen (CCP1M<3:0> = `1010`), the ECCP1 pin is not affected; only the CCP1IF interrupt flag is affected.

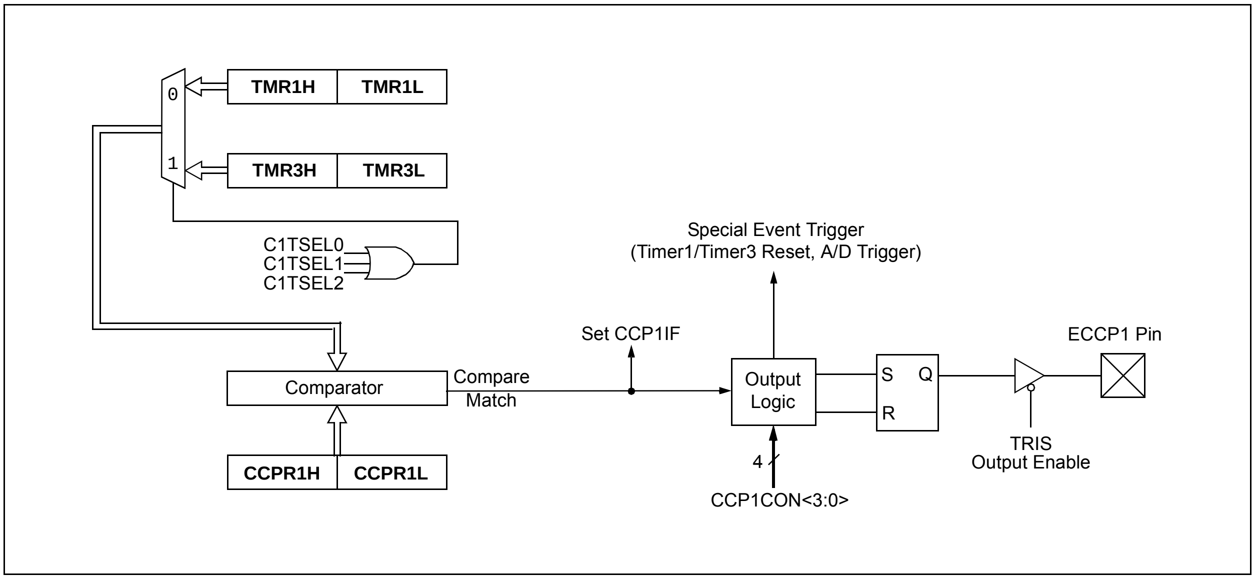
20.3.4 SPECIAL EVENT TRIGGER

The ECCP module is equipped with a Special Event Trigger. This is an internal hardware signal generated in Compare mode to trigger actions by other modules. The Special Event Trigger is enabled by selecting the Compare Special Event Trigger mode (CCP1M<3:0> = `1011`).

The Special Event Trigger resets the Timer register pair for whichever timer resource is currently assigned as the module’s time base. This allows the CCPR1 registers to serve as a programmable Period register for either timer.

The Special Event Trigger can also start an A/D conversion. In order to do this, the A/D Converter must already be enabled.

FIGURE 20-2: COMPARE MODE OPERATION BLOCK DIAGRAM



REGISTER 20-4: ECCP1DEL: ENHANCED PWM CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
P1RSEN	P1DC6	P1DC5	P1DC4	P1DC3	P1DC2	P1DC1	P1DC0
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 **P1RSEN:** PWM Restart Enable bit

1 = Upon auto-shutdown, the ECCP1ASE bit clears automatically once the shutdown event goes away; the PWM restarts automatically

0 = Upon auto-shutdown, ECCP1ASE must be cleared by software to restart the PWM

bit 6-0 **P1DC<6:0>:** PWM Delay Count bits

P1DCn = Number of FOSC/4 (4 * TOSC) cycles between the scheduled time when a PWM signal **should** transition active and the **actual** time it does transition active.

20.4.7 PULSE STEERING MODE

In Single Output mode, pulse steering allows any of the PWM pins to be the modulated signal. Additionally, the same PWM signal can simultaneously be available on multiple pins.

Once the Single Output mode is selected (CCP1M<3:2> = 11 and P1M<1:0> = 00 of the CCP1CON register), the user firmware can bring out the same PWM signal to one, two, three or four output pins by setting the appropriate STR<D:A> bits (PSTR1CON<3:0>), as provided in Table 20-2.

> **Note:** The associated TRIS bits must be set to output (‘0’) to enable the pin output driver in order to see the PWM signal on the pin.

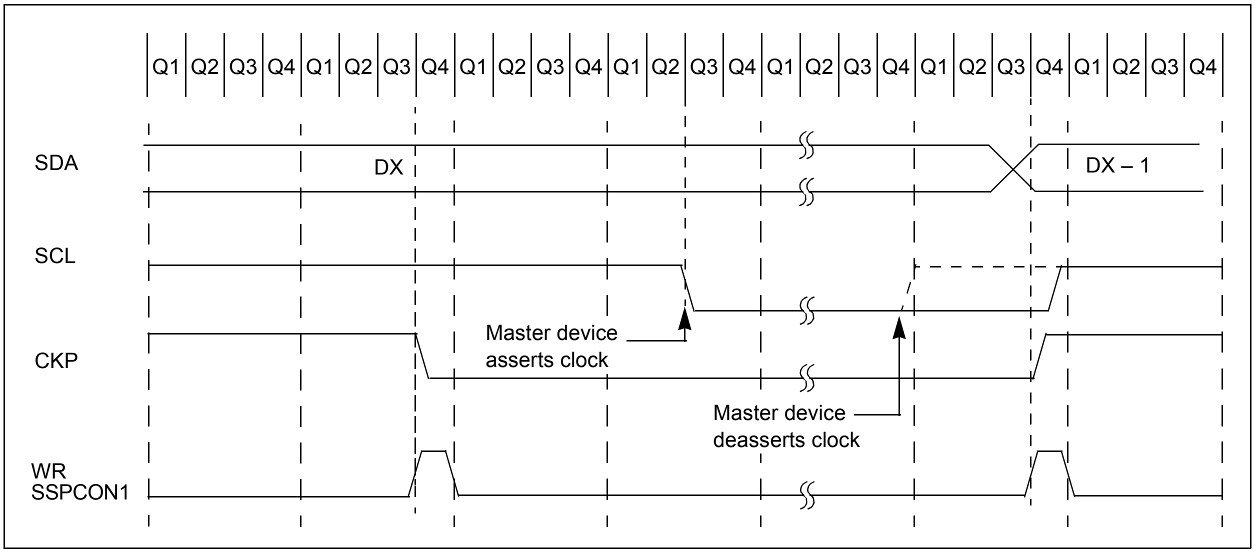
While the PWM Steering mode is active, the CCP1M<1:0> bits (CCP1CON<1:0>) select the PWM output polarity for the P1<D:A> pins.

The PWM auto-shutdown operation also applies to the PWM Steering mode, as described in **Section 20.4.4 “Enhanced PWM Auto-shutdown mode”**. An auto-shutdown event will only affect pins that have PWM outputs enabled.

21.4.4.5 Clock Synchronization and the CKP bit

When the CKP bit is cleared, the SCL output is forced to '0'. However, clearing the CKP bit will not assert the SCL output low until the SCL output is already sampled low. Therefore, the CKP bit will not assert the SCL line until an external I²C master device has already asserted the SCL line. The SCL output will remain low until the CKP bit is set and all other devices on the I²C bus have deasserted SCL. This ensures that a write to the CKP bit will not violate the minimum high time requirement for SCL (see Figure 21-14).

FIGURE 21-14: CLOCK SYNCHRONIZATION TIMING



21.4.9 I^2C™ MASTER MODE REPEATED START CONDITION TIMING

A Repeated Start condition occurs when the RSEN bit (SSPCON2<1>) is programmed high and the I^2C logic module is in the Idle state. When the RSEN bit is set, the SCL pin is asserted low. When the SCL pin is sampled low, the Baud Rate Generator is loaded with the contents of SSPADD<5:0> and begins counting. The SDA pin is released (brought high) for one Baud Rate Generator count (TBRG). When the Baud Rate Generator times out, and if SDA is sampled high, the SCL pin will be deasserted (brought high). When SCL is sampled high, the Baud Rate Generator is reloaded with the contents of SSPADD<6:0> and begins counting. SDA and SCL must be sampled high for one TBRG. This action is then followed by assertion of the SDA pin (SDA = 0) for one TBRG while SCL is high. Following this, the RSEN bit (SSPCON2<1>) will be automatically cleared and the Baud Rate Generator will not be reloaded, leaving the SDA pin held low. As soon as a Start condition is detected on the SDA and SCL pins, the S bit (SSPSTAT<3>) will be set. The SSPIF bit will not be set until the Baud Rate Generator has timed out.

Note 1: If RSEN is programmed while any other event is in progress, it will not take effect.

2: A bus collision during the Repeated Start condition occurs if:
- SDA is sampled low when SCL goes from low-to-high.
- SCL goes low before SDA is asserted low. This may indicate that another master is attempting to transmit a data '1'.

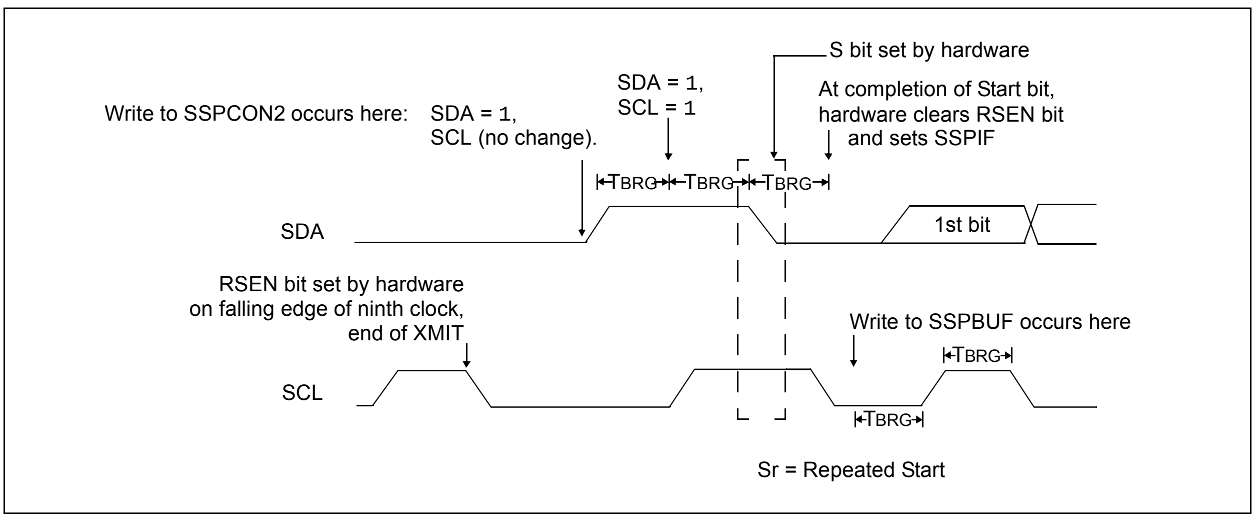
Immediately following the SSPIF bit getting set, the user may write the SSPBUF with the 7-bit address in 7-bit mode or the default first address in 10-bit mode. After the first eight bits are transmitted and an ACK is received, the user may then transmit an additional eight bits of address (10-bit mode) or eight bits of data (7-bit mode).

21.4.9.1 WCOL Status Flag

If the user writes the SSPBUF when a Repeated Start sequence is in progress, the WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

Note: Because queueing of events is not allowed, writing of the lower 5 bits of SSPCON2 is disabled until the Repeated Start condition is complete.

FIGURE 21-22: REPEATED START CONDITION WAVEFORM



21.4.10 I²C™ MASTER MODE TRANSMISSION

Transmission of a data byte, a 7-bit address or the other half of a 10-bit address, is accomplished by simply writing a value to the SSPBUF register. This action will set the Buffer Full flag bit, BF, and allow the Baud Rate Generator to begin counting and start the next transmission. Each bit of address/data will be shifted out onto the SDA pin after the falling edge of SCL is asserted (see data hold time specification Parameter 106). SCL is held low for one Baud Rate Generator rollover count (T_{BRG}). Data should be valid before SCL is released high (see data setup time specification Parameter 107). When the SCL pin is released high, it is held that way for T_{BRG}. The data on the SDA pin must remain stable for that duration and some hold time after the next falling edge of SCL. After the eighth bit is shifted out (the falling edge of the eighth clock), the BF flag is cleared and the master releases SDA. This allows the slave device being addressed to respond with an $\overline{\text{ACK}}$ bit during the ninth bit time if an address match occurred, or if data was received properly. The status of $\overline{\text{ACK}}$ is written into the ACKDT bit on the falling edge of the ninth clock. If the master receives an Acknowledge, the Acknowledge Status bit, ACKSTAT, is cleared; if not, the bit is set. After the ninth clock, the SSPIF bit is set and the master clock (Baud Rate Generator) is suspended until the next data byte is loaded into the SSPBUF, leaving SCL low and SDA unchanged (Figure 21-23).

After the write to the SSPBUF, each bit of the address will be shifted out on the falling edge of SCL until all seven address bits and the R/$\overline{\text{W}}$ bit are completed. On the falling edge of the eighth clock, the master will deassert the SDA pin, allowing the slave to respond with an Acknowledge. On the falling edge of the ninth clock, the master will sample the SDA pin to see if the address was recognized by a slave. The status of the $\overline{\text{ACK}}$ bit is loaded into the ACKSTAT status bit (SSPCON2<6>). Following the falling edge of the ninth clock transmission of the address, the SSPIF flag is set, the BF flag is cleared and the Baud Rate Generator is turned off until another write to the SSPBUF takes place, holding SCL low and allowing SDA to float.

21.4.10.1 BF Status Flag

In Transmit mode, the BF bit (SSPSTAT<0>) is set when the CPU writes to SSPBUF and is cleared when all 8 bits are shifted out.

21.4.10.2 WCOL Status Flag

If the user writes the SSPBUF when a transmit is already in progress (i.e., SSPSR is still shifting out a data byte), the WCOL bit is set and the contents of the buffer are unchanged (the write doesn't occur) after 2 T_{CY} after the SSPBUF write. If SSPBUF is rewritten within 2 T_{CY}, the WCOL bit is set and SSPBUF is updated. This may result in a corrupted transfer.

The user should verify that the WCOL bit is clear after each write to SSPBUF to ensure the transfer is correct. In all cases, WCOL must be cleared in software.

21.4.10.3 ACKSTAT Status Flag

In Transmit mode, the ACKSTAT bit (SSPCON2<6>) is cleared when the slave has sent an Acknowledge ($\overline{\text{ACK}}$ = 0) and is set when the slave does not Acknowledge ($\overline{\text{ACK}}$ = 1). A slave sends an Acknowledge when it has recognized its address (including a general call), or when the slave has properly received its data.

21.4.11 I²C™ MASTER MODE RECEPTION

Master mode reception is enabled by programming the Receive Enable bit, RCEN (SSPCON2<3>).

Note:	The MSSP module must be in an inactive state before the RCEN bit is set or the RCEN bit will be disregarded.

The Baud Rate Generator begins counting, and on each rollover, the state of the SCL pin changes (high-to-low/low-to-high) and data is shifted into the SSPSR. After the falling edge of the eighth clock, the receive enable flag is automatically cleared, the contents of the SSPSR are loaded into the SSPBUF, the BF flag bit is set, the SSPIF flag bit is set and the Baud Rate Generator is suspended from counting, holding SCL low. The MSSP is now in Idle state awaiting the next command. When the buffer is read by the CPU, the BF flag bit is automatically cleared. The user can then send an Acknowledge bit at the end of reception by setting the Acknowledge Sequence Enable bit, ACKEN (SSPCON2<4>).

21.4.11.1 BF Status Flag

In receive operation, the BF bit is set when an address or data byte is loaded into SSPBUF from SSPSR. It is cleared when the SSPBUF register is read.

21.4.11.2 SSPOV Status Flag

In receive operation, the SSPOV bit is set when 8 bits are received into the SSPSR and the BF flag bit is already set from a previous reception.

21.4.11.3 WCOL Status Flag

If the user writes the SSPBUF when a receive is already in progress (i.e., SSPSR is still shifting in a data byte), the WCOL bit is set and the contents of the buffer are unchanged (the write doesn't occur).

27.2.3 DEDICATED CAN RECEIVE BUFFER REGISTERS

This section shows the dedicated CAN Receive Buffer registers with their associated control registers.

REGISTER 27-13: RXB0CON: RECEIVE BUFFER 0 CONTROL REGISTER

Mode 0	R/C-0	R/W-0	R/W-0	U-0	R-0	R/W-0	R-0	R-0
	RXFUL[(1)]	RXM1	RXM0	—	RXRTRRO	RXB0DBEN	JTOFF[(2)]	FILHIT0

Mode 1,2	R/C-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
	RXFUL[(1)]	RXM1	RTRRO	FILHITF4	FILHIT3	FILHIT2	FILHIT1	FILHIT0
	bit 7							bit 0

Legend:	C = Clearable bit		
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 **RXFUL:** Receive Full Status bit[(1)]
1 = Receive buffer contains a received message
0 = Receive buffer is open to receive a new message

bit 6,6-5 Mode 0:
RXM<1:0>: Receive Buffer Mode bit 1 (combines with RXM0 to form RXM<1:0> bits, see bit 5)
11 = Receive all messages (including those with errors); filter criteria is ignored
10 = Receive only valid messages with extended identifier; EXIDEN in RXFnSIDL must be ‘1’
01 = Receive only valid messages with standard identifier; EXIDEN in RXFnSIDL must be ‘0’
00 = Receive all valid messages as per the EXIDEN bit in the RXFnSIDL register
Mode 1, 2:
RXM1: Receive Buffer Mode bit 1
1 = Receive all messages (including those with errors); acceptance filters are ignored
0 = Receive all valid messages as per acceptance filters

bit 5 Mode 0:
RXM0: Receive Buffer Mode bit 0 (combines with RXM1 to form RXM<1:0>bits, see bit 6)
Mode 1, 2:
RTRRO: Remote Transmission Request bit for Received Message (read-only)
1 = A remote transmission request is received
0 = A remote transmission request is not received

bit 4 Mode 0:
Unimplemented: Read as ‘0’
Mode 1, 2:
FILHIT<4:0>: Filter Hit bit 4
This bit combines with other bits to form filter acceptance bits<4:0>.

bit 3 Mode 0:
RXRTRRO: Remote Transmission Request bit for Received Message (read-only)
1 = A remote transmission request is received
0 = A remote transmission request is not received
Mode 1, 2:
FILHIT<4:0>: Filter Hit bit 3
This bit combines with other bits to form filter acceptance bits<4:0>.

Note 1: This bit is set by the CAN module upon receiving a message and must be cleared by software after the buffer is read. As long as RXFUL is set, no new message will be loaded and the buffer will be considered full. After clearing the RXFUL flag, the PIR5 bit, RXB0IF, can be cleared. If RXB0IF is cleared, but RXFUL is not cleared, then RXB0IF is set again.

2: This bit allows the same filter jump table for both RXB0CON and RXB1CON.

REGISTER 27-42: RXMnSIDL: RECEIVE ACCEPTANCE MASK 'n' STANDARD IDENTIFIER MASK REGISTERS, LOW BYTE [0 ≤ n ≤ 1]

R/W-x	R/W-x	R/W-x	U-0	R/W-0	U-0	R/W-x	R/W-x
SID2	SID1	SID0	—	EXIDEN(1)	—	EID17	EID16
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 7-5 **SID<2:0>:** Standard Identifier Mask bits or Extended Identifier Mask bits (EID<20:18>)

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

bit 3 Mode 0:
Unimplemented: Read as '0'
Mode 1, 2:
EXIDEN: Extended Identifier Filter Enable Mask bit(1)
1 = Messages selected by the EXIDEN bit in RXFnSIDL will be accepted
0 = Both standard and extended identifier messages will be accepted

bit 2 **Unimplemented:** Read as '0'

bit 1-0 **EID<17:16>:** Extended Identifier Mask bits

Note 1: This bit is available in Mode 1 and 2 only.

REGISTER 27-43: RXMnEIDH: RECEIVE ACCEPTANCE MASK 'n' EXTENDED IDENTIFIER MASK REGISTERS, HIGH BYTE [0 ≤ n ≤ 1]

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8
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 **EID<15:8>:** Extended Identifier Mask bits

REGISTER 27-44: RXMnEIDL: RECEIVE ACCEPTANCE MASK 'n' EXTENDED IDENTIFIER MASK REGISTERS, LOW BYTE [0 ≤ n ≤ 1]

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0
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 **EID<7:0>:** Extended Identifier Mask bits

27.2.4 CAN BAUD RATE REGISTERS

This section describes the CAN Baud Rate registers.

Note:	These registers are writable in Configuration mode only.

REGISTER 27-52: BRGCON1: BAUD RATE CONTROL REGISTER 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SJW1	SJW0	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0
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 **SJW<1:0>:** Synchronized Jump Width bits

11 = Synchronization jump width time = 4 x T_Q
10 = Synchronization jump width time = 3 x T_Q
01 = Synchronization jump width time = 2 x T_Q
00 = Synchronization jump width time = 1 x T_Q

bit 5-0 **BRP<5:0>:** Baud Rate Prescaler bits

111111 = T_Q = (2 x 64)/F_{OSC}
111110 = T_Q = (2 x 63)/F_{OSC}
:
:
000001 = T_Q = (2 x 2)/F_{OSC}
000000 = T_Q = (2 x 1)/F_{OSC}

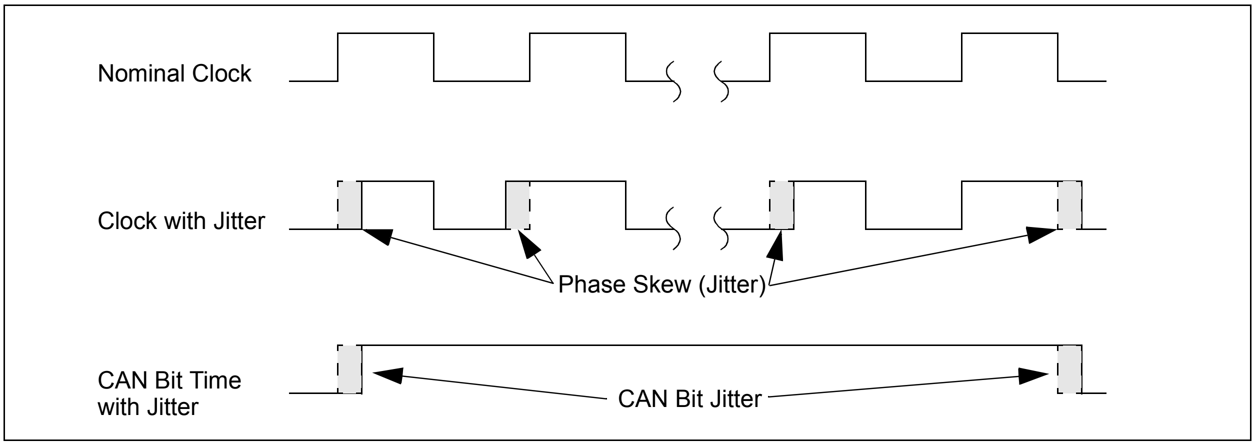
27.9.1 EXTERNAL CLOCK, INTERNAL CLOCK AND MEASURABLE JITTER IN HS-PLL BASED OSCILLATORS

The microcontroller clock frequency generated from a PLL circuit is subject to a jitter, also defined as Phase Jitter or Phase Skew. For its PIC18 Enhanced microcontrollers, Microchip specifies phase jitter (P_{jitter}) as being 2% (Gaussian distribution, within 3 standard deviations, see Parameter F13 in Table 31-7) and Total Jitter (T_{jitter}) as being 2 * P_{jitter}.

The CAN protocol uses a bit-stuffing technique that inserts a bit of a given polarity following five bits with the opposite polarity. This gives a total of 10 bits transmitted without resynchronization (compensation for jitter or phase error).

Given the random nature of the added jitter error, it can be shown that the total error caused by the jitter tends to cancel itself over time. For a period of 10 bits, it is necessary to add only two jitter intervals to correct for jitter induced error: one interval in the beginning of the 10-bit period and another at the end. The overall effect is shown in Figure 27-5.

FIGURE 27-5: EFFECTS OF PHASE JITTER ON THE MICROCONTROLLER CLOCK AND CAN BIT TIME



Once these considerations are taken into account, it is possible to show that the relation between the jitter and the total frequency error can be defined as:

EQUATION 27-4: JITTER AND TOTAL FREQUENCY ERROR

$$\Delta f = \frac{T_{jitter}}{10 \times NBT} = \frac{2 \times P_{jitter}}{10 \times NBT}$$

where jitter is expressed in terms of time and NBT is the Nominal Bit Time.

For example, assume a CAN bit rate of 125 Kb/s, which gives an NBT of 8 µs. For a 16 MHz clock generated from a 4x PLL, the jitter at this clock frequency is:

EQUATION 27-5: 16 MHz CLOCK FROM 4x PLL JITTER:

$$2\% \times \frac{1}{16\text{ MHz}} = \frac{0.02}{16 \times 10^6} = 1.25\text{ns}$$

and resultant frequency error is:

EQUATION 27-6: RESULTANT FREQUENCY ERROR:

$$\frac{2 \times (1.25 \times 10^{-9})}{10 \times (8 \times 10^{-6})} = 3.125 \times 10^{-5} = 0.0031\%$$

REGISTER 28-5: CONFIG3H: CONFIGURATION REGISTER 3 HIGH (BYTE ADDRESS 300005h)

R/P-1	U-0	U-0	U-0	R/P-1	R/P-1	R/P-1	R/P-1
MCLRE	—	—	—	MSSPMSK	T3CKMX(1)	T0CKMX(1)	CANMX
bit 7							bit 0

Legend:	P = Programmable bit		
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 **MCLRE:** $\overline{\text{MCLR}}$ Pin Enable bit
1 = $\overline{\text{MCLR}}$ pin is enabled; RE3 input pin is disabled
0 = RE3 input pin is enabled; $\overline{\text{MCLR}}$ is disabled

bit 6-4 **Unimplemented:** Read as '0'

bit 3 **MSSPMSK:** MSSP V3 7-Bit Address Masking Mode Enable bit
1 = 7-Bit Address Masking mode is enabled
0 = 5-Bit Address Masking mode is enabled

bit 2 **T3CKMX:** Timer3 Clock Input MUX bit(1)
1 = Timer3 gets its clock input from the RG2/T3CKI pin on 64-pin packages
0 = Timer3 gets its clock input from the RB5/T3CKI pin on 64-pin packages

bit 1 **T0CKMX:** Timer0 Clock Input MUX bit(1)
1 = Timer0 gets its clock input from the RB5/T0CKI pin on 64-pin packages
0 = Timer0 gets its clock input from the RG4/T0CKI pin on 64-pin packages

bit 0 **CANMX:** ECAN MUX bit
1 = CANTX and CANRX pins are located on RB2 and RB3, respectively
0 = CANTX and CANRX pins are located on RC6 and RC7, respectively (28-pin and 40/44-pin packages) or on RE4 and RE5, respectively (64-pin package)

Note 1: These bits are implemented only on the 64-pin devices (PIC18F6XK80); maintain as '0' on 28-pin, 40-pin and 44-pin devices.

REGISTER 28-13: DEVID1: DEVICE ID REGISTER 1 FOR THE PIC18F66K80 FAMILY

R	R	R	R	R	R	R	R
DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0
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 7-5 **DEV<2:0>:** Device ID bits

These bits are used with the DEV<10:3> bits in the Device ID Register 2 to identify the part number:
`000` = PIC18F46K80, PIC18LF26K80
`001` = PIC18F26K80, PIC18LF65K80
`010` = PIC18F65K80, PIC18LF45K80
`011` = PIC18F45K80, PIC18LF25K80
`100` = PIC18F25K80
`110` = PIC18LF66K80
`111` = PIC18F66K80, PIC18LF46K80

bit 4-0 **REV<4:0>:** Revision ID bits

These bits are used to indicate the device revision.

REGISTER 28-14: DEVID2: DEVICE ID REGISTER 2 FOR THE PIC18F66K80 FAMILY

R	R	R	R	R	R	R	R
DEV10(1)	DEV9(1)	DEV8(1)	DEV7(1)	DEV6(1)	DEV5(1)	DEV4(1)	DEV3(1)
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 **DEV<10:3>:** Device ID bits(1)

These bits are used with the DEV<2:0> bits in the Device ID Register 1 to identify the part number.

Note 1: These values for DEV<10:3> may be shared with other devices. The specific device is always identified by using the entire DEV<10:0> bit sequence.

SUBFSR **Subtract Literal from FSR**

Syntax: SUBFSR f, k

Operands: 0 ≤ k ≤ 63
f ∈ [0, 1, 2]

Operation: FSRf – k → FSRf

Status Affected: None

Encoding:

1110	1001	ffkk	kkkk

Description: The 6-bit literal 'k' is subtracted from the contents of the FSR specified by 'f'.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example: `SUBFSR 2, 23h`

Before Instruction
FSR2 = 03FFh

After Instruction
FSR2 = 03DCh

SUBULNK **Subtract Literal from FSR2 and Return**

Syntax: SUBULNK k

Operands: 0 ≤ k ≤ 63

Operation: FSR2 – k → FSR2,
(TOS) → PC

Status Affected: None

Encoding:

1110	1001	11kk	kkkk

Description: The 6-bit literal 'k' is subtracted from the contents of the FSR2. A `RETURN` is then executed by loading the PC with the TOS.

The instruction takes two cycles to execute; a `NOP` is performed during the second cycle.

This may be thought of as a special case of the `SUBFSR` instruction, where f = 3 (binary '`11`'); it operates only on FSR2.

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write to destination
No Operation	No Operation	No Operation	No Operation

Example: `SUBULNK 23h`

Before Instruction
FSR2 = 03FFh
PC = 0100h

After Instruction
FSR2 = 03DCh
PC = (TOS)

FIGURE 31-14: EXAMPLE SPI SLAVE MODE TIMING (CKE = 1)

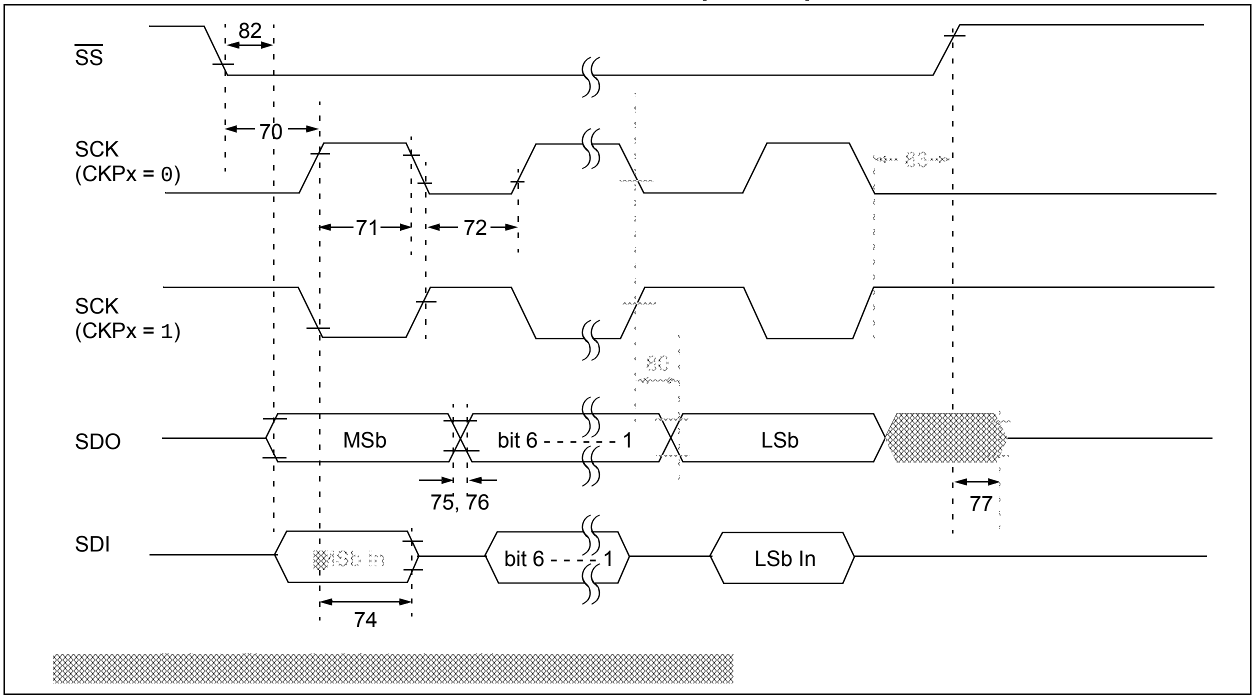


TABLE 31-18: EXAMPLE SPI SLAVE MODE REQUIREMENTS (CKE = 1)

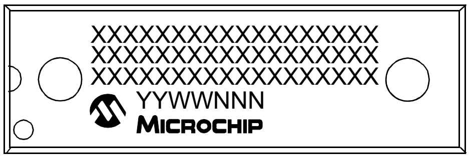
Param No.	Symbol	Characteristic		Min	Max	Units	Conditions
70	TssL2scH, TssL2scL	$\overline{SS}$ ↓ to SCK ↓ or SCK ↑ Input		3 Tcy	—	ns	
70A	TssL2WB	$\overline{SS}$ to write to SSPBUF		3 Tcy	—	ns	
71	TscH	SCK Input High Time (Slave mode)	Continuous	1.25 Tcy + 30	—	ns	
71A			Single Byte	40	—	ns	**(Note 1)**
72	TscL	SCK Input Low Time (Slave mode)	Continuous	1.25 Tcy + 30	—	ns	
72A			Single Byte	40	—	ns	**(Note 1)**
73A	TB2B	Last Clock Edge of Byte 1 to the First Clock Edge of Byte 2		1.5 Tcy + 40	—	ns	**(Note 2)**
74	TscH2diL, TscL2diL	Hold Time of SDI Data Input to SCK Edge		40	—	ns	
75	TdoR	SDO Data Output Rise Time		—	25	ns	
76	TdoF	SDO Data Output Fall Time		—	25	ns	
77	TssH2doZ	$\overline{SS}$ ↑ to SDO Output High-Impedance		10	50	ns	
78	TscR	SCK Output Rise Time (Master mode)		—	25	ns	
79	TscF	SCK Output Fall Time (Master mode)		—	25	ns	
80	TscH2doV, TscL2doV	SDO Data Output Valid after SCK Edge		—	50	ns	
82	TssL2doV	SDO Data Output Valid after $\overline{SS}$ ↓ Edge		—	50	ns	
83	TscH2ssH, TscL2ssH	$\overline{SS}$ ↑ after SCK Edge		1.5 Tcy + 40	—	ns	

Note 1: Requires the use of Parameter 73A.

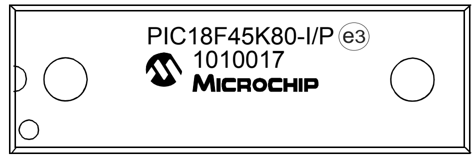
2: Only if Parameter 71A and 72A are used.

32.1 Package Marking Information (Continued)

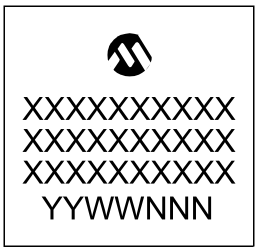
40-Lead PDIP



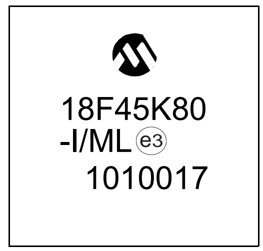
Example



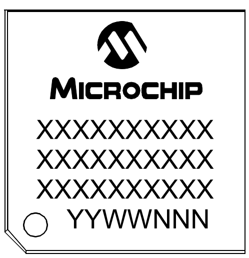
44-Lead QFN



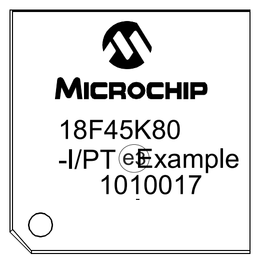
Example



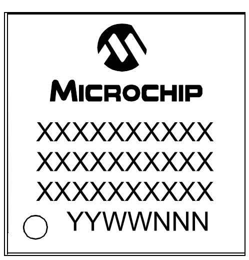
44-Lead TQFP

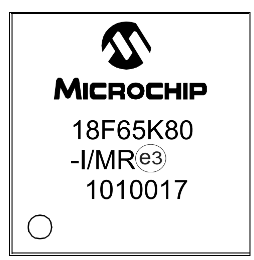


Example

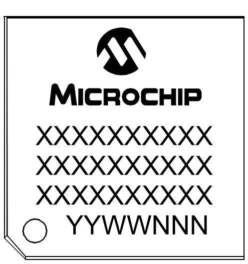


64-Lead QFN





64-Lead TQFP



Example