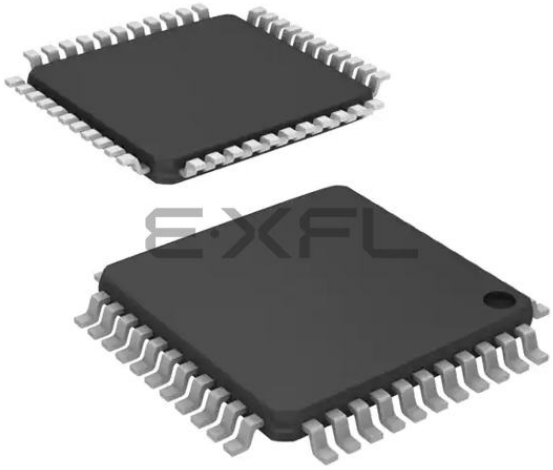


Welcome to E-XFL.COM

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	35
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 11x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	44-TQFP
Supplier Device Package	44-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f46k80-e-pt

6.6 Data Memory and the Extended Instruction Set

Enabling the PIC18 extended instruction set (XINST Configuration bit = 1) significantly changes certain aspects of data memory and its addressing. Using the Access Bank for many of the core PIC18 instructions introduces a new addressing mode for the data memory space. This mode also alters the behavior of Indirect Addressing using FSR2 and its associated operands.

What does not change is just as important. The size of the data memory space is unchanged, as well as its linear addressing. The SFR map remains the same. Core PIC18 instructions can still operate in both Direct and Indirect Addressing mode. Inherent and literal instructions do not change at all. Indirect Addressing with FSR0 and FSR1 also remains unchanged.

6.6.1 INDEXED ADDRESSING WITH LITERAL OFFSET

Enabling the PIC18 extended instruction set changes the behavior of Indirect Addressing using the FSR2 register pair and its associated file operands. Under the proper conditions, instructions that use the Access Bank – that is, most bit-oriented and byte-oriented instructions – can invoke a form of Indexed Addressing using an offset specified in the instruction. This special addressing mode is known as Indexed Addressing with Literal Offset or the Indexed Literal Offset mode.

When using the extended instruction set, this addressing mode requires the following:

- Use of the Access Bank (‘a’ = 0)
- A file address argument that is less than or equal to 5Fh

Under these conditions, the file address of the instruction is not interpreted as the lower byte of an address (used with the BSR in Direct Addressing) or as an 8-bit address in the Access Bank. Instead, the value is interpreted as an offset value to an Address Pointer specified by FSR2. The offset and the contents of FSR2 are added to obtain the target address of the operation.

6.6.2 INSTRUCTIONS AFFECTED BY INDEXED LITERAL OFFSET MODE

Any of the core PIC18 instructions that can use Direct Addressing are potentially affected by the Indexed Literal Offset Addressing mode. This includes all byte-oriented and bit-oriented instructions, or almost one-half of the standard PIC18 instruction set. Instructions that only use Inherent or Literal Addressing modes are unaffected.

Additionally, byte-oriented and bit-oriented instructions are not affected if they do not use the Access Bank (Access RAM bit = 1), or include a file address of 60h or above. Instructions meeting these criteria will continue to execute as before. A comparison of the different possible addressing modes when the extended instruction set is enabled is shown in Figure 6-9.

Those who desire to use byte-oriented or bit-oriented instructions in the Indexed Literal Offset mode should note the changes to assembler syntax for this mode. This is described in more detail in **Section 29.2.1 “Extended Instruction Syntax”**.

TABLE 8-1: REGISTERS ASSOCIATED WITH DATA EEPROM MEMORY

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF
EEADRH	EEPROM Address Register High Byte							
EEADR	EEPROM Address Register Low Byte							
EEDATA	EEPROM Data Register							
EECON2	EEPROM Control Register 2 (not a physical register)							
EECON1	EEPGD	CFGS	—	FREE	WRERR	WREN	WR	RD
IPR4	TMR4IP	EEIP	CMP2IP	CMP1IP	—	CCP5IP	CCP4IP	CCP3IP
PIR4	TMR4IF	EEIF	CMP2IF	CMP1IF	—	CCP5IF	CCP4IF	CCP3IF
PIE4	TMR4IE	EEIE	CMP2IE	CMP1IE	—	CCP5IE	CCP4IE	CCP3IE

Legend: — = unimplemented, read as '0'. Shaded cells are not used during Flash/EEPROM access.

REGISTER 10-5: PIR2: PERIPHERAL INTERRUPT REQUEST (FLAG) REGISTER 2

R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
OSCFIF	—	—	—	BCLIF	HLVDIF	TMR3IF	TMR3GIF
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 **OSCFIF:** Oscillator Fail Interrupt Flag bit
1 = Device oscillator failed, clock input has changed to INTOSC (bit must be cleared in software)
0 = Device clock is operating

bit 6-4 **Unimplemented:** Read as ‘0’

bit 3 **BCLIF:** Bus Collision Interrupt Flag bit
1 = A bus collision occurred (bit must be cleared in software)
0 = No bus collision occurred

bit 2 **HLVDIF:** High/Low-Voltage Detect Interrupt Flag bit
1 = A low-voltage condition occurred (bit must be cleared in software)
0 = The device voltage is above the regulator’s low-voltage trip point

bit 1 **TMR3IF:** TMR3 Overflow Interrupt Flag bit
1 = TMR3 register overflowed (bit must be cleared in software)
0 = TMR3 register did not overflow

bit 0 **TMR3GIF:** TMR3 Gate Interrupt Flag bit
1 = Timer gate interrupt occurred (bit must be cleared in software)
0 = No timer gate interrupt occurred

11.3 PORTB, TRISB and LATB Registers

PORTB is an eight-bit wide, bidirectional port. The corresponding Data Direction and Output Latch registers are TRISB and LATB. All pins on PORTB are digital only.

EXAMPLE 11-2: INITIALIZING PORTB

```
CLRF    PORTB   ; Initialize PORTB by
                ; clearing output
                ; data latches
CLRF    LATB    ; Alternate method
                ; to clear output
                ; data latches
MOVLW   0CFh    ; Value used to
                ; initialize data
                ; direction
MOVWF   TRISB   ; Set RB<3:0> as inputs
                ; RB<5:4> as outputs
                ; RB<7:6> as inputs
```

Each of the PORTB pins has a weak internal pull-up. A single control bit can turn on all the pull-ups. This is performed by clearing bit, $\overline{\text{RBPU}}$ (INTCON2<7>). The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on a Power-on Reset.

Four of the PORTB pins (RB<7:4>) have an interrupt-on-change feature. Only pins configured as inputs can cause this interrupt to occur. Any RB<7:4> pins that are configured as outputs are excluded from the interrupt-on-change comparison.

Comparisons with the input pins (of RB<7:4>) are made with the old value latched on the last read of PORTB. The “mismatch” outputs of RB<7:4> are ORed together to generate the RB Port Change Interrupt with Flag bit, RBIF (INTCON<0>).

This interrupt can wake the device from power-managed modes. To clear the interrupt in the Interrupt Service Routine:

1. Perform any read or write of PORTB (except with the `MOVFF (ANY), PORTB` instruction).
2. Wait one instruction cycle (such as executing a `NOP` instruction).

   This ends the mismatch condition.
3. Clear flag bit, RBIF.

A mismatch condition will continue to set flag bit, RBIF. Reading PORTB will end the mismatch condition and allow flag bit, RBIF, to be cleared after a one T_{CY} delay.

The interrupt-on-change feature is recommended for wake-up on key depression operation and operations where PORTB is only used for the interrupt-on-change feature. Polling of PORTB is not recommended while using the interrupt-on-change feature.

The RB<3:2> pins are multiplexed as CTMU edge inputs. RB5 has an additional function for Timer3 and Timer1. It can be configured for Timer3 clock input or Timer1 external clock gate input.

TABLE 11-7: PORTD FUNCTIONS

Pin Name	Function	TRIS Setting	I/O	I/O Type	Description
RD0/C1INA/ PSP0	RD0	0	O	DIG	LATD<0> data output.
		1	I	ST	PORTD<0> data input.
	C1INA	1	I	ANA	Comparator 1 Input A.
	PSP0	x	I/O	ST	Parallel Slave Port data.
RD1/C1INB/ PSP1	RD1(1)	0	O	DIG	LATD<1> data output.
		1	I	ST	PORTD<1> data input.
	C1INB(1)	1	I	ANA	Comparator 1 Input B.
	PSP1(1)	x	I/O	ST	Parallel Slave Port data.
RD2/C2INA/ PSP2	RD2	0	O	DIG	LATD<2> data output.
		1	I	ST	PORTD<2> data input.
	C2INA	1	I	ANA	Comparator 2 Input A.
	PSP2	x	I/O	ST	Parallel Slave Port data.
RD3/C2INB/ CTMUI/PSP3	RD3	0	O	DIG	LATD<3> data output.
		1	I	ST	PORTD<3> data input.
	C2INB	1	I	ANA	Comparator 2 Input B.
	CTMUI	x	I	—	CTMU pulse generator charger for the C2INB comparator input.
	PSP3	x	I/O	ST	Parallel Slave Port data.
RD4/ECCP1/ P1A/PSP4	RD4	0	O	DIG	LATD<4> data output.
		1	I	ST	PORTD<4> data input.
	ECCP1	0	O	DIG	ECCP1 compare output and ECCP1 PWM output; takes priority over port data.
		1	I	ST	ECCP1 capture input.
	P1A	0	O	DIG	ECCP1 Enhanced PWM output, Channel A. May be configured for tri-state during Enhanced PWM shutdown events; takes priority over port data.
	PSP4	x	I/O	ST	Parallel Slave Port data.
RD5/P1B/PSP5	RD5	0	O	DIG	LATD<5> data output.
		1	I	ST	PORTD<5> data input.
	P1B	0	O	DIG	ECCP1 Enhanced PWM output, Channel B. May be configured for tri-state during Enhanced PWM shutdown events; takes priority over port data.
	PSP5	x	I/O	ST	Parallel Slave Port data.
RD6/TX2/CK2 P1C/PSP6	RD6	0	O	DIG	LATD<6> data output.
		1	I	ST	PORTD<6> data input.
	TX2(1)	0	O	DIG	Asynchronous serial data output (EUSARTx module); takes priority over port data.
	CK2(1)	0	O	DIG	Synchronous serial clock output (EUSARTx module); user must configure as an input.
		1	I	ST	Synchronous serial clock input (EUSARTx module); user must configure as an input.
	P1C	0	O	DIG	ECCP1 Enhanced PWM output, Channel C. May be configured for tri-state during Enhanced PWM.
	PSP6	x	I/O	ST	Parallel Slave Port data.

Legend: O = Output; I = Input; ANA = Analog Signal; DIG = CMOS Output; ST = Schmitt Trigger Buffer Input; x = Don't care (TRIS bit does not affect port direction or is overridden for this option)

Note 1: This is the pin assignment for 40 and 44-pin devices (PIC18F4XK80).

16.5 Timer3 Gates

Timer3 can be configured to count freely or the count can be enabled and disabled using the Timer3 gate circuitry. This is also referred to as the Timer3 gate count enable.

The Timer3 gate can also be driven by multiple selectable sources.

16.5.1 TIMER3 GATE COUNT ENABLE

The Timer3 Gate Enable mode is enabled by setting the TMR3GE bit (TxGCON<7>). The polarity of the Timer3 Gate Enable mode is configured using the T3GPOL bit (T3GCON<6>).

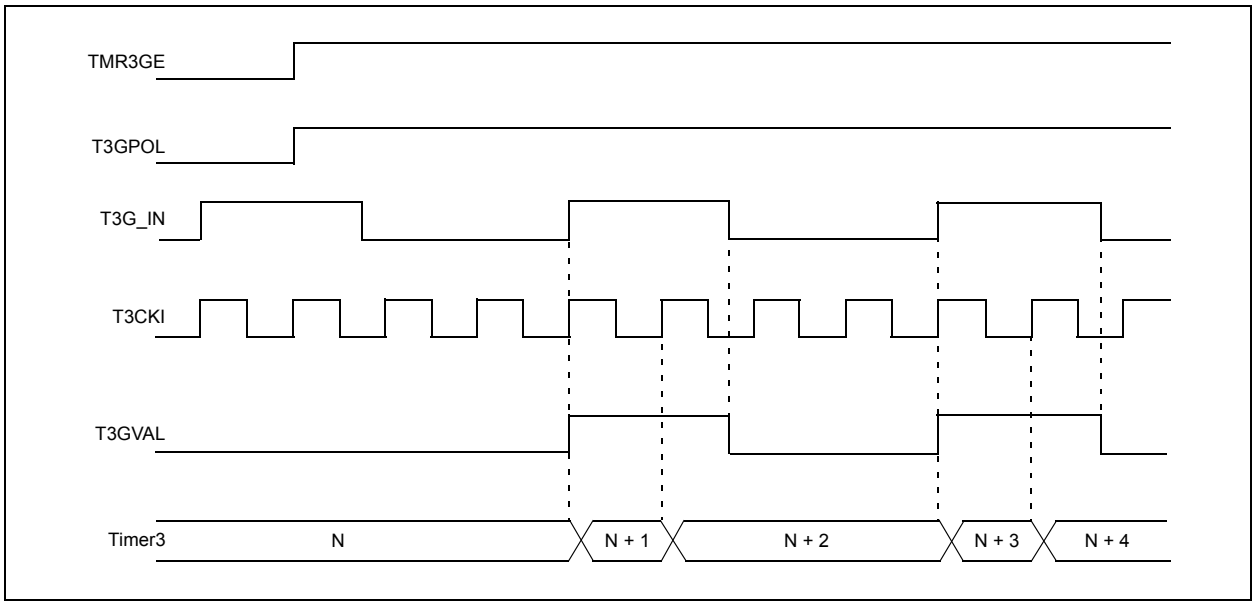
When Timer3 Gate Enable mode is enabled, Timer3 will increment on the rising edge of the Timer3 clock source. When Timer3 Gate Enable mode is disabled, no incrementing will occur and Timer3 will hold the current count. See Figure 16-2 for timing details.

TABLE 16-1: TIMER3 GATE ENABLE SELECTIONS

T3CLK(†)	T3GPOL (T3GCON<6>)	T3G Pin	Timer3 Operation
↑	0	0	Counts
↑	0	1	Holds Count
↑	1	0	Holds Count
↑	1	1	Counts

† The clock on which TMR3 is running. For more information, see T3CLK in Figure 16-1.

FIGURE 16-2: TIMER3 GATE COUNT ENABLE MODE



20.2.2 TIMER1/2/3/4 MODE SELECTION

The timers that are to be used with the capture feature (Timer1 2, 3 or 4) must be running in Timer mode or Synchronized Counter mode. In Asynchronous Counter mode, the capture operation may not work. The timer to be used with each ECCP module is selected in the CCPTMRS register (Register 20-2).

20.2.3 SOFTWARE INTERRUPT

When the Capture mode is changed, a false capture interrupt may be generated. The user should keep the CCP1IE interrupt enable bit clear to avoid false interrupts. The interrupt flag bit, CCP1IF, should also be cleared following any such change in operating mode.

20.2.4 ECCP PRESCALER

There are four prescaler settings in Capture mode; they are specified as part of the operating mode selected by the mode select bits (CCP1M<3:0>). Whenever the ECCP module is turned off, or Capture mode is disabled, the prescaler counter is cleared. This means that any Reset will clear the prescaler counter.

Switching from one capture prescaler to another may generate an interrupt. Also, the prescaler counter will not be cleared; therefore, the first capture may be from a non-zero prescaler. Example 20-1 provides the recommended method for switching between capture prescalers. This example also clears the prescaler counter and will not generate the “false” interrupt.

EXAMPLE 20-1: CHANGING BETWEEN CAPTURE PRESCALERS

```
CLRF   CCP1CON     ; Turn ECCP module off
MOVLW  NEW_CAPT_PS ; Load WREG with the
                   ; new prescaler mode
                   ; value and ECCP ON
MOVWF  CCP1CON     ; Load ECCP1CON with
                   ; this value
```

FIGURE 20-1: CAPTURE MODE OPERATION BLOCK DIAGRAM

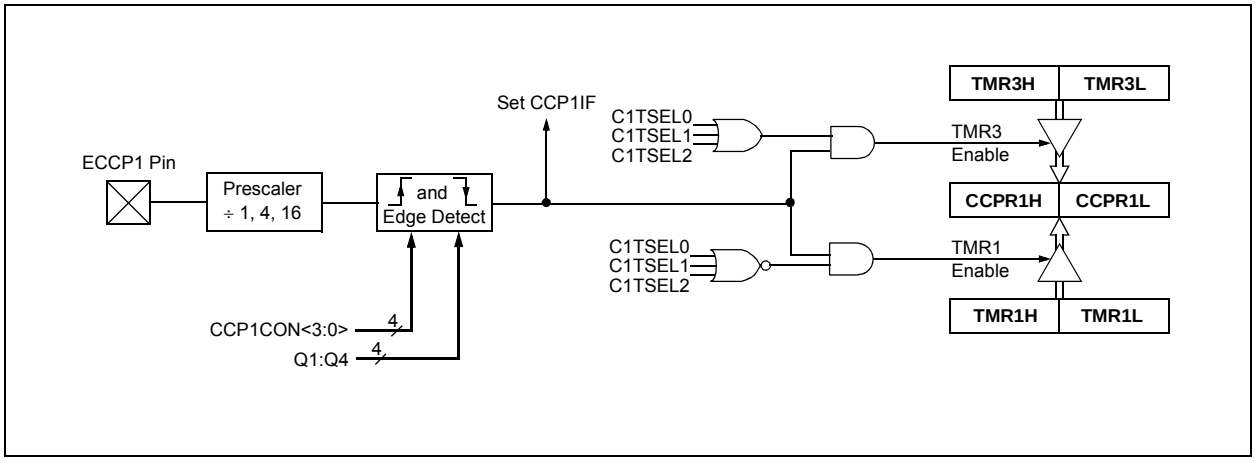


FIGURE 21-5: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 0)

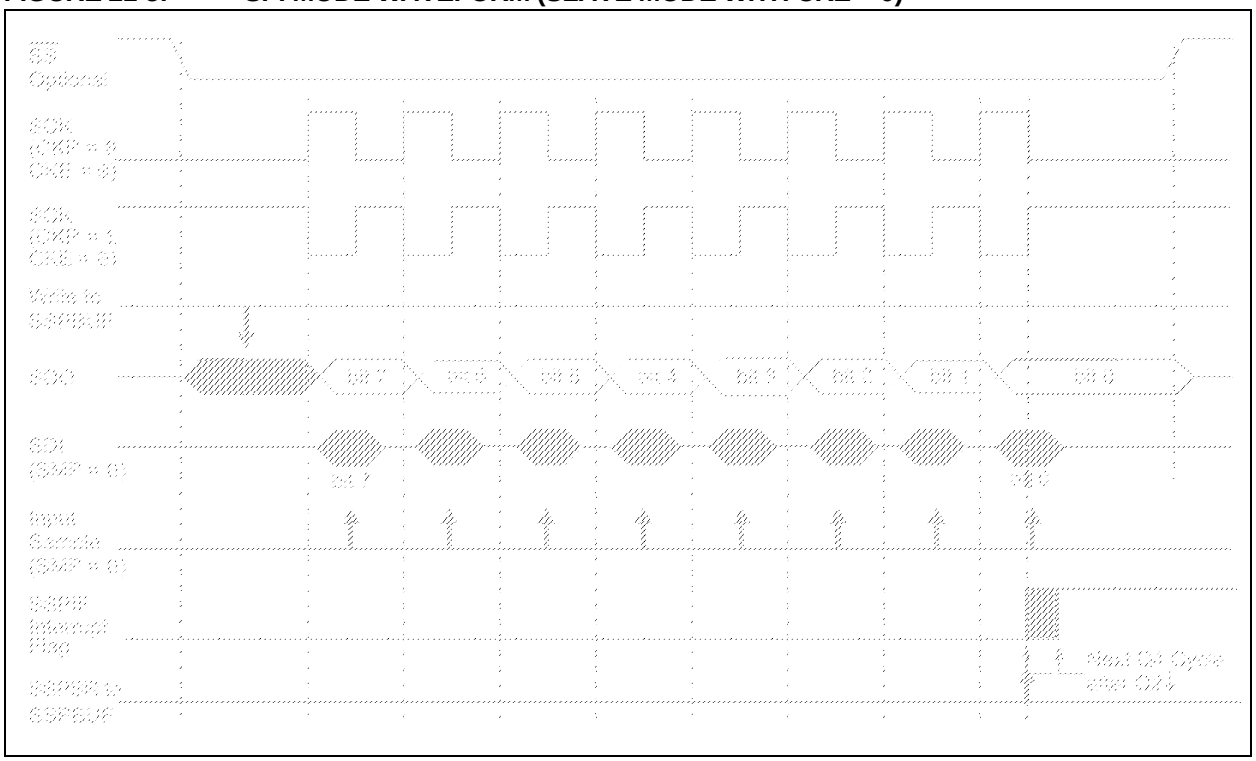
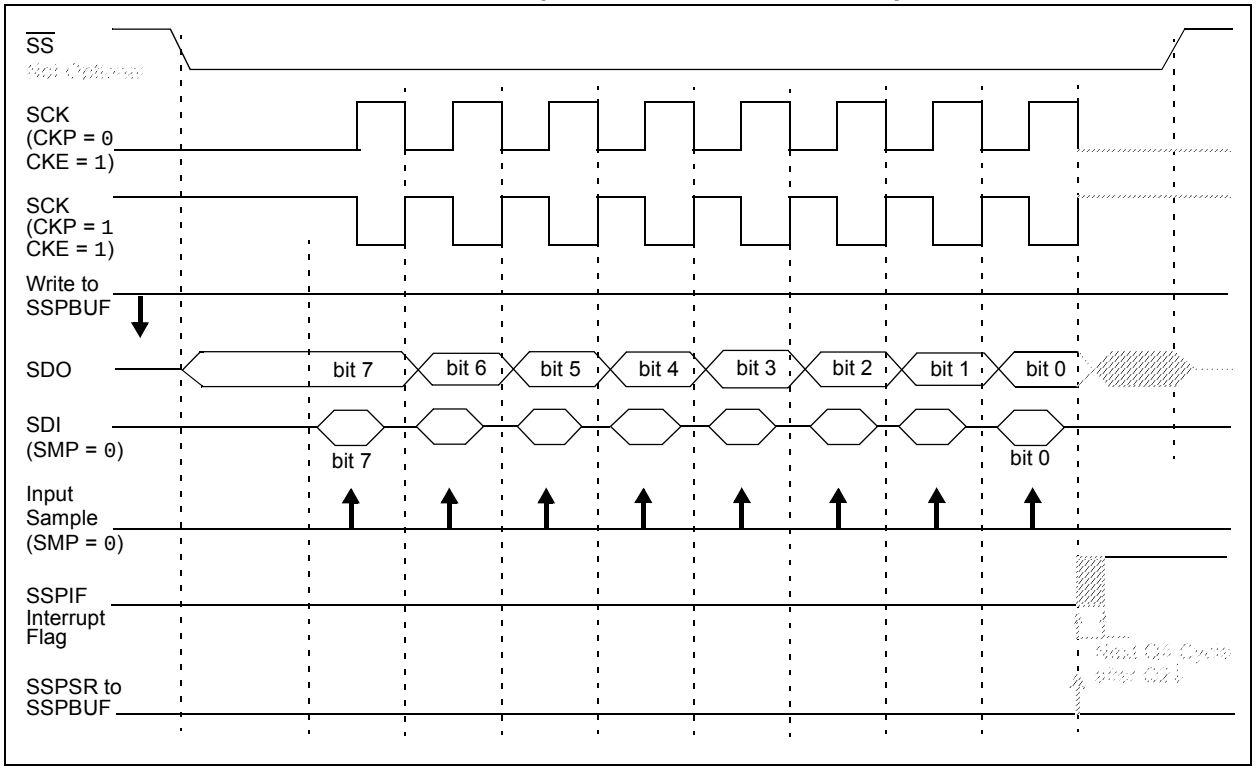


FIGURE 21-6: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 1)



23.3 A/D Acquisition Requirements

For the A/D Converter to meet its specified accuracy, the Charge Holding (C_{HOLD}) capacitor must be allowed to fully charge to the input channel voltage level. The analog input model is shown in Figure 23-5. The source impedance (R_S) and the internal sampling switch (R_{SS}) impedance directly affect the time required to charge the capacitor C_{HOLD}. The sampling switch (R_{SS}) impedance varies over the device voltage (V_{DD}).

The source impedance affects the offset voltage at the analog input (due to pin leakage current). **The maximum recommended impedance for analog sources is 2.5 kΩ.** After the analog input channel is selected or changed, the channel must be sampled for at least the minimum acquisition time before starting a conversion.

Note:	When the conversion is started, the holding capacitor is disconnected from the input pin.

To calculate the minimum acquisition time, Equation 23-1 can be used. This equation assumes that 1/2 LSb error is used (1,024 steps for the A/D). The 1/2 LSb error is the maximum error allowed for the A/D to meet its specified resolution.

Equation 23-3 shows the calculation of the minimum required acquisition time, T_{ACQ}. This calculation is based on the following application system assumptions:

C_{HOLD}	=	25 pF
Rs	=	2.5 kΩ
Conversion Error	≤	1/2 LSb
V_{DD}	=	3V → Rss = 2 kΩ
Temperature	=	85°C (system max.)

EQUATION 23-1: ACQUISITION TIME

$$T_{ACQ} = \text{Amplifier Settling Time} + \text{Holding Capacitor Charging Time} + \text{Temperature Coefficient}$$
$$= T_{AMP} + T_C + T_{COFF}$$

EQUATION 23-2: A/D MINIMUM CHARGING TIME

$$V_{HOLD} = (V_{REF} - (V_{REF}/2048)) \bullet (1 - e^{(-T_C/C_{HOLD}(R_{IC} + R_{SS} + R_S))})$$
or
$$T_C = -(C_{HOLD})(R_{IC} + R_{SS} + R_S)\ln(1/2048)$$

EQUATION 23-3: CALCULATING THE MINIMUM REQUIRED ACQUISITION TIME

$T_{ACQ} = T_{AMP} + T_C + T_{COFF}$

$T_{AMP} = 0.2\ \mu s$

$T_{COFF} = (\text{Temp} - 25°C)(0.02\ \mu s/°C)$
$(85°C - 25°C)(0.02\ \mu s/°C)$
$1.2\ \mu s$

Temperature coefficient is only required for temperatures > 25°C. Below 25°C, T_{COFF} = 0 ms.

$T_C = -(C_{HOLD})(R_{IC} + R_{SS} + R_S)\ln(1/2048)\ \mu s$
$-(25\ pF)(1\ k\Omega + 2\ k\Omega + 2.5\ k\Omega)\ln(0.0004883)\ \mu s$
$1.05\ \mu s$

$T_{ACQ} = 0.2\ \mu s + 1.05\ \mu s + 1.2\ \mu s$
$2.45\ \mu s$

24.0 COMPARATOR MODULE

The analog comparator module contains two comparators that can be independently configured in a variety of ways. The inputs can be selected from the analog inputs and two internal voltage references. The digital outputs are available at the pin level and can also be read through the control register. Multiple output and interrupt event generation are also available. A generic single comparator from the module is shown in Figure 24-1.

Key features of the module includes:

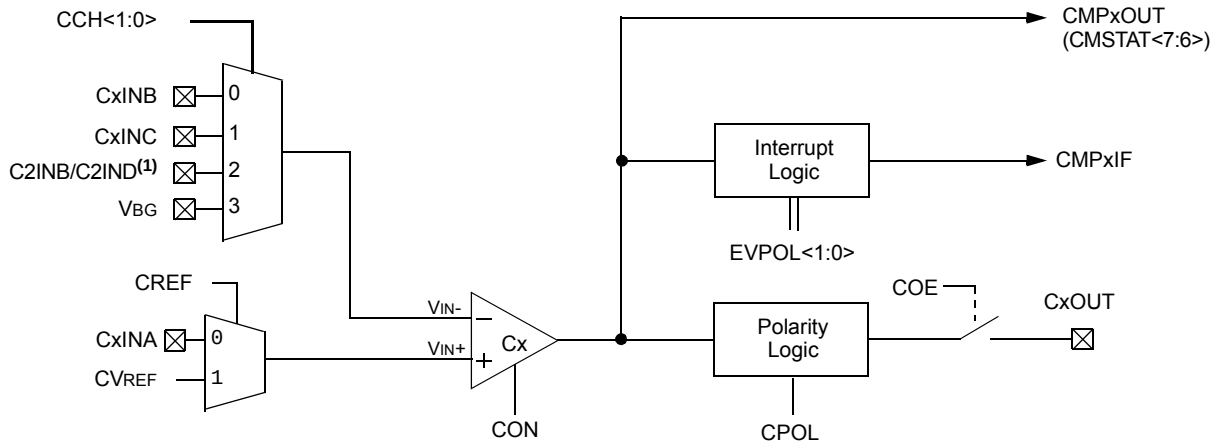
- Independent comparator control
- Programmable input configuration
- Output to both pin and register levels
- Programmable output polarity
- Independent interrupt generation for each comparator with configurable interrupt-on-change

24.1 Registers

The CMxCON registers (CM1CON and CM2CON) select the input and output configuration for each comparator, as well as the settings for interrupt generation (see Register 24-1).

The CMSTAT register (Register 24-2) provides the output results of the comparators. The bits in this register are read-only.

FIGURE 24-1: COMPARATOR SIMPLIFIED BLOCK DIAGRAM



Note 1: Comparator 1 uses C2INB as an input to the inverted terminal.
Comparator 2 uses C1INB as an input to the inverted terminal.

EXAMPLE 27-3: TRANSMITTING A CAN MESSAGE USING BANKED METHOD

```
    ; Need to transmit Standard Identifier message 123h using TXB0 buffer.
    ; To successfully transmit, CAN module must be either in Normal or Loopback mode.
    ; TXB0 buffer is not in access bank. And since we want banked method, we need to make sure
    ; that correct bank is selected.
    BANKSEL TXB0CON                 ; One BANKSEL in beginning will make sure that we are
                                    ; in correct bank for rest of the buffer access.
    ; Now load transmit data into TXB0 buffer.
    MOVLW   MY_DATA_BYTE1           ; Load first data byte into buffer
    MOVWF   TXB0D0                  ; Compiler will automatically set “BANKED” bit
    ; Load rest of data bytes - up to 8 bytes into TXB0 buffer.
    ...
    ; Load message identifier
    MOVLW   60H                     ; Load SID2:SID0, EXIDE = 0
    MOVWF   TXB0SIDL
    MOVLW   24H                     ; Load SID10:SID3
    MOVWF   TXB0SIDH
    ; No need to load TXB0EIDL:TXB0EIDH, as we are transmitting Standard Identifier Message only.

    ; Now that all data bytes are loaded, mark it for transmission.
    MOVLW   B’00001000’             ; Normal priority; Request transmission
    MOVWF   TXB0CON

    ; If required, wait for message to get transmitted
    BTFSC   TXB0CON, TXREQ          ; Is it transmitted?
    BRA     $-2                     ; No. Continue to wait...

    ; Message is transmitted.
```

27.4.3 MODE 2 – ENHANCED FIFO MODE

In Mode 2, two or more receive buffers are used to form the receive FIFO (first in, first out) buffer. There is no one-to-one relationship between the receive buffer and acceptance filter registers. Any filter that is enabled and linked to any FIFO receive buffer can generate acceptance and cause FIFO to be updated.

FIFO length is user-programmable, from 2-8 buffers deep. FIFO length is determined by the very first programmable buffer that is configured as a transmit buffer. For example, if Buffer 2 (B2) is programmed as a transmit buffer, FIFO consists of RXB0, RXB1, B0 and B1, creating a FIFO length of 4. If all programmable buffers are configured as receive buffers, FIFO will have the maximum length of 8.

The following is the list of resources available in Mode 2:

- Three transmit buffers: TXB0, TXB1 and TXB2
- Two receive buffers: RXB0 and RXB1
- Six buffers programmable as TX or RX; receive buffers form FIFO: B0-B5
- Automatic RTR handling on B0-B5
- Sixteen acceptance filters: RXF0-RXF15
- Two dedicated acceptance mask registers; RXF15 programmable as third mask: RXM0-RXM1, RXF15
- Programmable data filter on standard identifier messages: SDFLC, useful for DeviceNet protocol

27.5 CAN Message Buffers

27.5.1 DEDICATED TRANSMIT BUFFERS

The PIC18F66K80 family devices implement three dedicated transmit buffers – TXB0, TXB1 and TXB2. Each of these buffers occupies 14 bytes of SRAM and are mapped into the SFR memory map. These are the only transmit buffers available in Mode 0. Mode 1 and 2 may access these and other additional buffers.

Each transmit buffer contains one Control register (TXBnCON), four Identifier registers (TXBnSIDL, TXBnSIDH, TXBnEIDL, TXBnEIDH), one Data Length Count register (TXBnDLC) and eight Data Byte registers (TXBnDm).

27.5.2 DEDICATED RECEIVE BUFFERS

The PIC18F66K80 family devices implement two dedicated receive buffers: RXB0 and RXB1. Each of these buffers occupies 14 bytes of SRAM and are mapped into SFR memory map. These are the only receive buffers available in Mode 0. Mode 1 and 2 may access these and other additional buffers.

Each receive buffer contains one Control register (RXBnCON), four Identifier registers (RXBnSIDL, RXBnSIDH, RXBnEIDL, RXBnEIDH), one Data Length Count register (RXBnDLC) and eight Data Byte registers (RXBnDm).

There is also a separate Message Assembly Buffer (MAB) which acts as an additional receive buffer. MAB is always committed to receiving the next message from the bus and is not directly accessible to user firmware. The MAB assembles all incoming messages one by one. A message is transferred to appropriate receive buffers only if the corresponding acceptance filter criteria is met.

27.5.3 PROGRAMMABLE TRANSMIT/ RECEIVE BUFFERS

The ECAN module implements six new buffers: B0-B5. These buffers are individually programmable as either transmit or receive buffers. These buffers are available only in Mode 1 and 2. As with dedicated transmit and receive buffers, each of these programmable buffers occupies 14 bytes of SRAM and are mapped into SFR memory map.

Each buffer contains one Control register (BnCON), four Identifier registers (BnSIDL, BnSIDH, BnEIDL, BnEIDH), one Data Length Count register (BnDLC) and eight Data Byte registers (BnDm). Each of these registers contains two sets of control bits. Depending on whether the buffer is configured as transmit or receive, one would use the corresponding control bit set. By default, all buffers are configured as receive buffers. Each buffer can be individually configured as a transmit or receive buffer by setting the corresponding TXENn bit in the BSEL0 register.

When configured as transmit buffers, user firmware may access transmit buffers in any order similar to accessing dedicated transmit buffers. In receive configuration with Mode 1 enabled, user firmware may also access receive buffers in any order required. But in Mode 2, all receive buffers are combined to form a single FIFO. Actual FIFO length is programmable by user firmware. Access to FIFO must be done through the FIFO Pointer bits (FP<4:0>) in the CANCON register. It must be noted that there is no hardware protection against out of order FIFO reads.

BRA **Unconditional Branch**

Syntax: BRA n

Operands: $-1024 \leq n \leq 1023$

Operation: $(PC) + 2 + 2n \rightarrow PC$

Status Affected: None

Encoding:

1101	0nnn	nnnn	nnnn

Description: Add the 2's complement number, '2n', 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 a two-cycle instruction.

Words: 1

Cycles: 2

Q Cycle Activity:

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

Example: HERE BRA Jump

Before Instruction
PC = address (HERE)
After Instruction
PC = address (Jump)

BSF **Bit Set f**

Syntax: BSF f, b {,a}

Operands: $0 \leq f \leq 255$
$0 \leq b \leq 7$
$a \in [0,1]$

Operation: $1 \rightarrow f<b>$

Status Affected: None

Encoding:

1000	bbba	ffff	ffff

Description: Bit 'b' in register 'f' is set.

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 $f \leq 95$ (5Fh). See **Section 29.2.3 "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 register 'f'

Example: BSF FLAG_REG, 7, 1

Before Instruction
FLAG_REG = 0Ah
After Instruction
FLAG_REG = 8Ah

MOVSS **Move Indexed to Indexed**

Syntax: MOVSS [z_s], [z_d]

Operands: $0 \le z_s \le 127$
$0 \le z_d \le 127$

Operation: ((FSR2) + z_s) → ((FSR2) + z_d)

Status Affected: None

Encoding:

1st word (source)	1110	1011	1zzz	$zzzz_s$
2nd word (dest.)	1111	xxxx	xzzz	$zzzz_d$

Description: The contents of the source register are moved to the destination register. The addresses of the source and destination registers are determined by adding the 7-bit literal offsets, 'z_s' or 'z_d', respectively, to the value of FSR2. Both registers can be located anywhere in the 4096-byte data memory space (000h to FFFh).

The MOVSS instruction cannot use the PCL, TOSU, TOSH or TOSL as the destination register.

If the resultant source address points to an Indirect Addressing register, the value returned will be 00h. If the resultant destination address points to an Indirect Addressing register, the instruction will execute as a NOP.

Words: 2

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Determine source addr	Determine source addr	Read source reg
Decode	Determine dest addr	Determine dest addr	Write to dest reg

Example: MOVSS [05h], [06h]

Before Instruction
FSR2 = 80h
Contents of 85h = 33h
Contents of 86h = 11h
After Instruction
FSR2 = 80h
Contents of 85h = 33h
Contents of 86h = 33h

PUSHL **Store Literal at FSR2, Decrement FSR2**

Syntax: PUSHL k

Operands: $0 \le k \le 255$

Operation: k → (FSR2),
FSR2 – 1 → FSR2

Status Affected: None

Encoding:

1111	1010	kkkk	kkkk

Description: The 8-bit literal 'k' is written to the data memory address specified by FSR2. FSR2 is decremented by 1 after the operation.

This instruction allows users to push values onto a software stack.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read 'k'	Process data	Write to destination

Example: PUSHL 08h

Before Instruction
FSR2H:FSR2L = 01ECh
Memory (01ECh) = 00h

After Instruction
FSR2H:FSR2L = 01EBh
Memory (01ECh) = 08h

TABLE 31-1: MEMORY PROGRAMMING REQUIREMENTS

DC CHARACTERISTICS			Standard Operating Conditions Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	**Sym**	**Characteristic**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
		Internal Program Memory Programming Specifications(1)					
D110	VPP	Voltage on $\overline{\text{MCLR}}$/VPP/RE5 pin	VDD + 1.5	—	10	V	**(Note 3, Note 4)**
D113	IDDP	Supply Current during Programming	—	—	10	mA	
		Data EEPROM Memory					**(Note 2)**
D120	ED	Byte Endurance	100K	1000K	—	E/W	-40°C to +125°C
D121	VDRW	VDD for Read/Write	1.8	—	5.5	V	Using EECON to read/write PIC18FXXKXX devices
			1.8	—	3.6	V	Using EECON to read/write PIC18LFXXKXX devices
D122	TDEW	Erase/Write Cycle Time	—	4	—	ms	
D123	TRETD	Characteristic Retention	20	—	—	Year	Provided no other specifications are violated
D124	TREF	Number of Total Erase/Write Cycles before Refresh(2)	1M	10M	—	E/W	-40°C to +125°C
		Program Flash Memory					
D130	EP	Cell Endurance	1K	10K	—	E/W	-40°C to +125°C
D131	VPR	VDD for Read	1.8	—	5.5	V	PIC18FXXKXX devices
			1.8	—	3.6	V	PIC18LFXXKXX devices
D132B	VPEW	Voltage for Self-Timed Erase or Write Operations VDD	1.8	—	5.5	V	PIC18FXXKXX devices
D133A	TIW	Self-Timed Write Cycle Time	—	2	—	ms	
D134	TRETD	Characteristic Retention	20	—	—	Year	Provided no other specifications are violated
D135	IDDP	Supply Current during Programming	—	—	10	mA	
D140	TWE	Writes per Erase Cycle	—	—	1		For each physical address

† Data in "Typ" column is at 3.3V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

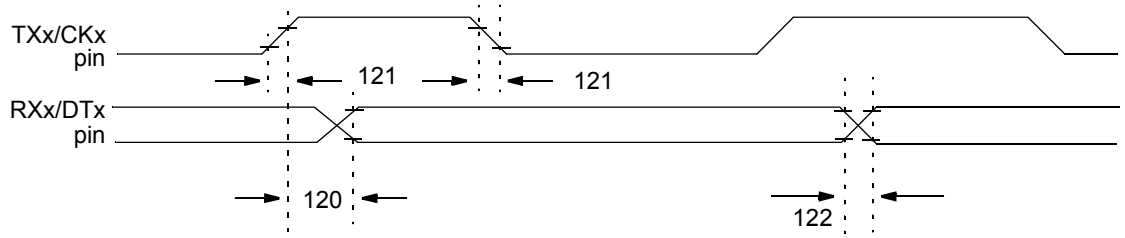
Note 1: These specifications are for programming the on-chip program memory through the use of table write instructions.

2: Refer to **Section 8.8 "Using the Data EEPROM"** for a more detailed discussion on data EEPROM endurance.

3: Required only if Single-Supply Programming is disabled.

4: The MPLAB® ICD 2 does not support variable VPP output. Circuitry to limit the ICD2 VPP voltage must be placed between the ICD 2 and target system when programming or debugging with the ICD2.

FIGURE 31-19: EUSARTx SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING

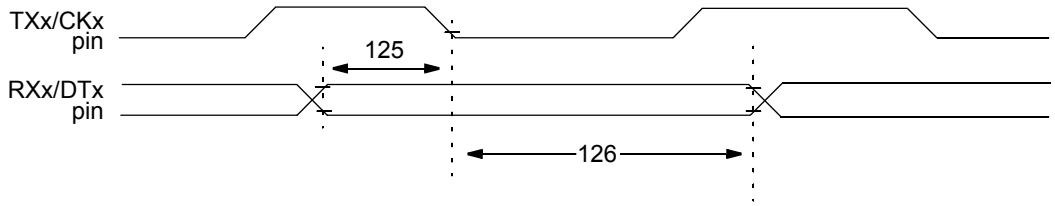


Note: Refer to Figure 31-3 for load conditions.

TABLE 31-23: EUSART/AUSART SYNCHRONOUS TRANSMISSION REQUIREMENTS

Param No.	Symbol	Characteristic	Min	Max	Units	Conditions
120	TCKH2DTV	SYNC XMIT (MASTER and SLAVE) Clock High to Data Out Valid	—	40	ns	
121	TCKRF	Clock Out Rise Time and Fall Time (Master mode)	—	20	ns	
122	TDTRF	Data Out Rise Time and Fall Time	—	20	ns	

FIGURE 31-20: EUSART/AUSART SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING



Note: Refer to Figure 31-3 for load conditions.

TABLE 31-24: EUSART/AUSART SYNCHRONOUS RECEIVE REQUIREMENTS

Param. No.	Symbol	Characteristic	Min	Max	Units	Conditions
125	TDTV2CKL	SYNC RCV (MASTER and SLAVE) Data Hold before CKx ↓ (DTx hold time)	10	—	ns	
126	TCKL2DTL	Data Hold after CKx ↓ (DTx hold time)	15	—	ns	

NOTES:

INDEX

A

B

S

T

PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, such as pricing or delivery, refer to the factory or the listed sales office.

PART NO.	X	/XX	XXX
Device	Temperature Range	Package	Pattern

Device[1,2]	PIC18F25K80, PIC18F26K80, PIC18F45K80, PIC18F46K80, PIC18F65K80, PIC18F66K80 VDD range 1.8V to 5V	
	PIC18LF25K80, PIC18LF26K80, PIC18LF45K80, PIC18LF46K80, PIC18F65K80, PIC18F66K80 VDD range 1.8V to 3.6V	
Temperature Range	I = -40°C to +85°C (Industrial) E = -40°C to +125°C (Extended)	
Package	P = PDIP Plastic Dual In-Line ML = QFN Plastic Quad Flat, No Lead Package SO = SOIC Plastic Small Outline SP = SPDIP Skinny Plastic Dual In-Line SS = SSOP Plastic Shrink Small Outline PT = TQFP Plastic Thin Quad Flatpack	
Pattern	a) QTP, SQTP, Code or Special Requirements (blank otherwise)	

Examples:

a) PIC18F66K80-I/MR 301 = Industrial temp., QFN package, Extended VDD limits, QTP pattern #301.

b) PIC18F66K80-I/PT = Industrial temp., TQFP package, Extended VDD limits.

Note 1: F = Standard Voltage Range
LF = Low Voltage Range

2: T = in tape and reel, TQFP packages only.