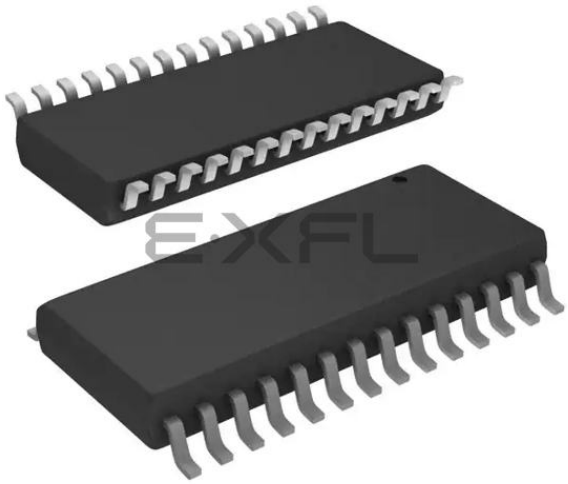


Welcome to E-XFL.COM

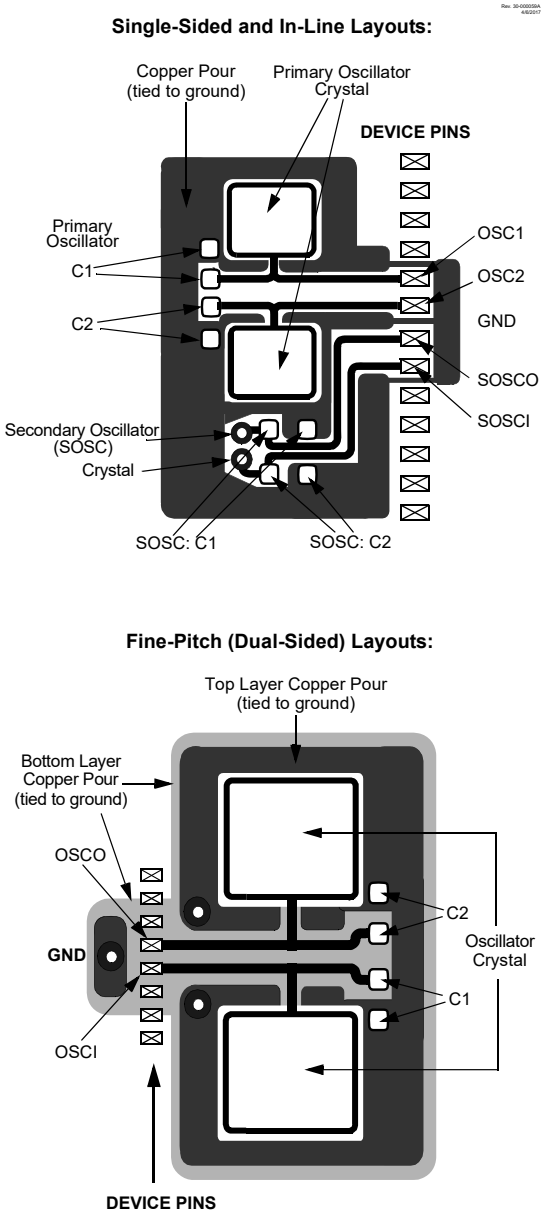
What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	64MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	25
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
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-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f25q10-i-so

Figure 2-3. Suggested Placement of the Oscillator Circuit



In planning the application's routing and I/O assignments, ensure that adjacent port pins, and other signals in close proximity to the oscillator, are benign (i.e., free of high frequencies, short rise and fall times, and other similar noise).

For additional information and design guidance on oscillator circuits, refer to these Microchip Application Notes, available at the corporate website (www.microchip.com):

- AN826, *"Crystal Oscillator Basics and Crystal Selection for rfPIC™ and PICmicro® Devices"*
- AN849, *"Basic PICmicro® Oscillator Design"*
- AN943, *"Practical PICmicro® Oscillator Analysis and Design"*
- AN949, *"Making Your Oscillator Work"*

Related Links

4. Oscillator Module (with Fail-Safe Clock Monitor)

Bits 9:8 – BORV[1:0] Brown-out Reset Voltage Selection bit

Value	Description
11	Brown-out Reset Voltage (V_{BOR}) set to 1.90V
10	Brown-out Reset Voltage (V_{BOR}) set to 2.45V
01	Brown-out Reset Voltage (V_{BOR}) set to 2.7V
00	Brown-out Reset Voltage (V_{BOR}) set to 2.85V

Bits 7:6 – BOREN[1:0] Brown-out Reset Enable bits
When enabled, Brown-out Reset Voltage (V_{BOR}) is set by BORV bit

Value	Description
11	Brown-out Reset enabled, SBOREN bit is ignored
10	Brown-out Reset enabled while running, disabled in Sleep; SBOREN is ignored
01	Brown-out Reset enabled according to SBOREN
00	Brown-out Reset disabled

Bit 5 – $\overline{\text{LPBOREN}}$ Low-Power BOR Enable bit

Value	Description
1	Low-Power Brown-out Reset is disabled
0	Low-Power Brown-out Reset is enabled

Bit 1 – $\overline{\text{PWRTE}}$ Power-up Timer Enable bit

Value	Description
1	PWRT disabled
0	PWRT enabled

Bit 0 – MCLRE Master Clear (MCLR) Enable bit

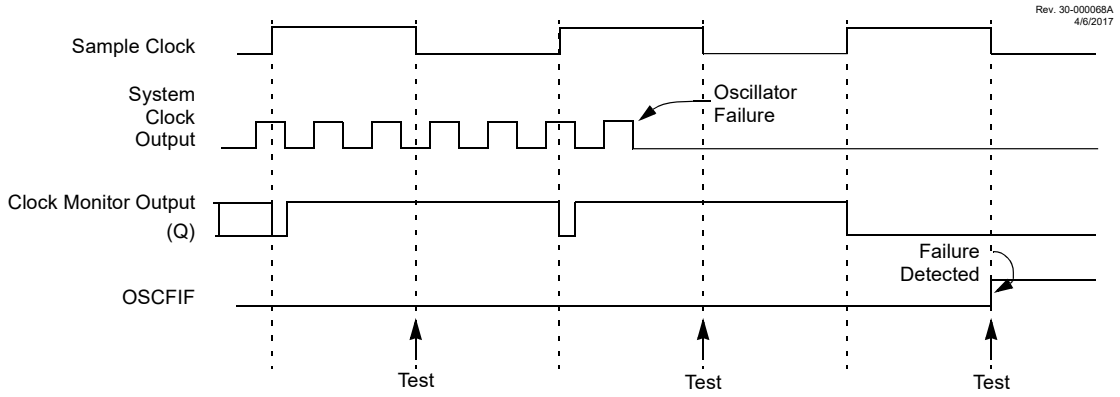
Value	Condition	Description
x	If LVP = 1	RE3 pin function is $\overline{\text{MCLR}}$
1	If LVP = 0	$\overline{\text{MCLR}}$ pin is $\overline{\text{MCLR}}$
0	If LVP = 0	$\overline{\text{MCLR}}$ pin function is port defined function

Note: BORV - The higher voltage setting is recommended for operation at or above 16 MHz.

Related Links

7.4.3 PMD2

Figure 4-10. FSCM Timing Diagram



Note: The system clock is normally at a much higher frequency than the sample clock. The relative frequencies in this example have been chosen for clarity.

4.6.5 OSCFRQ

Name: OSCFRQ
Address: 0xED9

HFINTOSC Frequency Selection Register

Bit	7	6	5	4	3	2	1	0
					HFFRQ[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					q	q	q	q

Bits 3:0 – HFFRQ[3:0] HFINTOSC Frequency Selection bits

HFFRQ	Nominal Freq (MHz)
1001	Reserved
1010	
1111	
1110	
1101	
1100	
1011	
1000(1)	64
0111	48
0110	32
0101	16
0100	12
0011	8
0010(1)	4
0001	2
0000	1

Note:

1. Refer to Table 4-1 for more information.

Figure 10-11. Comparing Addressing Options for Bit-Oriented and Byte-Oriented Instructions (Extended Instruction Set Enabled)

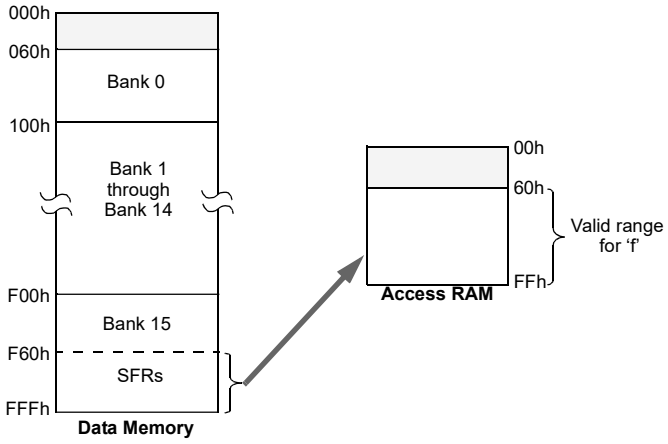
Rev. 30-000110A
4/18/2017

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 0FFh. This is the same as locations F60h to FFFh (Bank 15) of data memory.

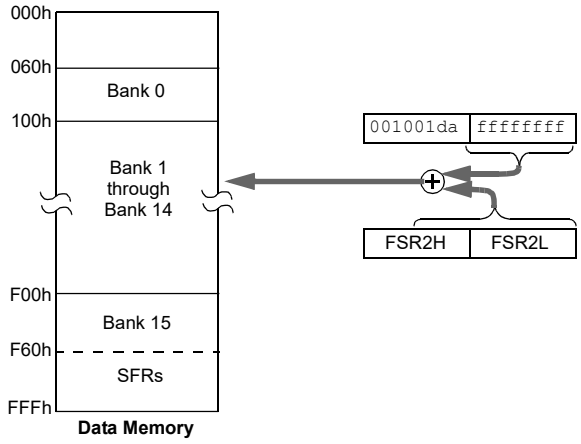
Locations below 60h are not available in this addressing mode.



When 'a' = 0 and f ≤ 5Fh:

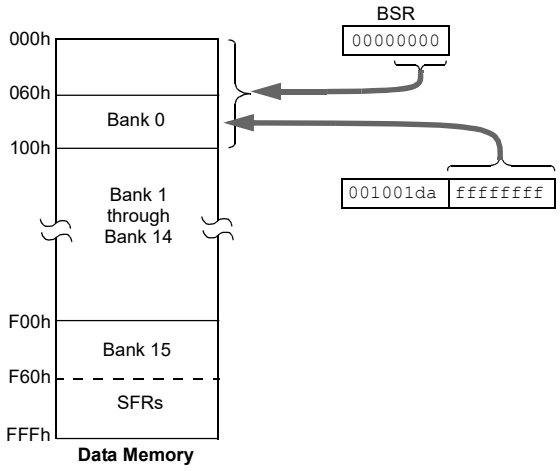
The instruction executes in Indexed Literal Offset mode. 'f' is interpreted as an offset to the address value in FSR2. The two are added together to obtain the address of the target register for the instruction. The address can be anywhere in the data memory space.

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.



10.8.5 STATUS

Name: STATUS
Address: 0xFD8

Status Register

Bit	7	6	5	4	3	2	1	0
		$\overline{TO}$	$\overline{PD}$	N	OV	Z	DC	C
Access		R	R	R/W	R/W	R/W	R/W	R/W
Reset		1	1	0	0	0	0	0

Bit 6 – $\overline{TO}$ Time-Out bit
Reset States: POR/BOR = 1
All Other Resets = q

Value	Description
1	Set at power-up or by execution of `CLRWDT` or `SLEEP` instruction
0	A WDT time-out occurred

Bit 5 – $\overline{PD}$ Power-Down bit
Reset States: POR/BOR = 1
All Other Resets = q

Value	Description
1	Set at power-up or by execution of `CLRWDT` instruction
0	Cleared by execution of the `SLEEP` instruction

Bit 4 – N Negative bit
Used for signed arithmetic (2's complement); indicates if the result is negative, (ALU MSb = `1`).
Reset States: POR/BOR = 0
All Other Resets = u

Value	Description
1	The result is negative
0	The result is positive

Bit 3 – OV Overflow bit
Used for signed arithmetic (2's complement); indicates an overflow of the 7-bit magnitude, which causes the sign bit (bit 7) to change state.
Reset States: POR/BOR = 0
All Other Resets = u

Value	Description
1	Overflow occurred for current signed arithmetic operation
0	No overflow occurred

Bit 2 – Z Zero bit
Reset States: POR/BOR = 0

13. Cyclic Redundancy Check (CRC) Module with Memory Scanner

The Cyclic Redundancy Check (CRC) module provides a software-configurable hardware-implemented CRC checksum generator. This module includes the following features:

- Any standard CRC up to 16 bits can be used
- Configurable Polynomial
- Any seed value up to 16 bits can be used
- Standard and reversed bit order available
- Augmented zeros can be added automatically or by the user
- Memory scanner for fast CRC calculations on program memory user data
- Software loadable data registers for communication CRC's

13.1 CRC Module Overview

The CRC module provides a means for calculating a check value of program memory. The CRC module is coupled with a memory scanner for faster CRC calculations. The memory scanner can automatically provide data to the CRC module. The CRC module can also be operated by directly writing data to SFRs, without using a scanner.

13.2 CRC Functional Overview

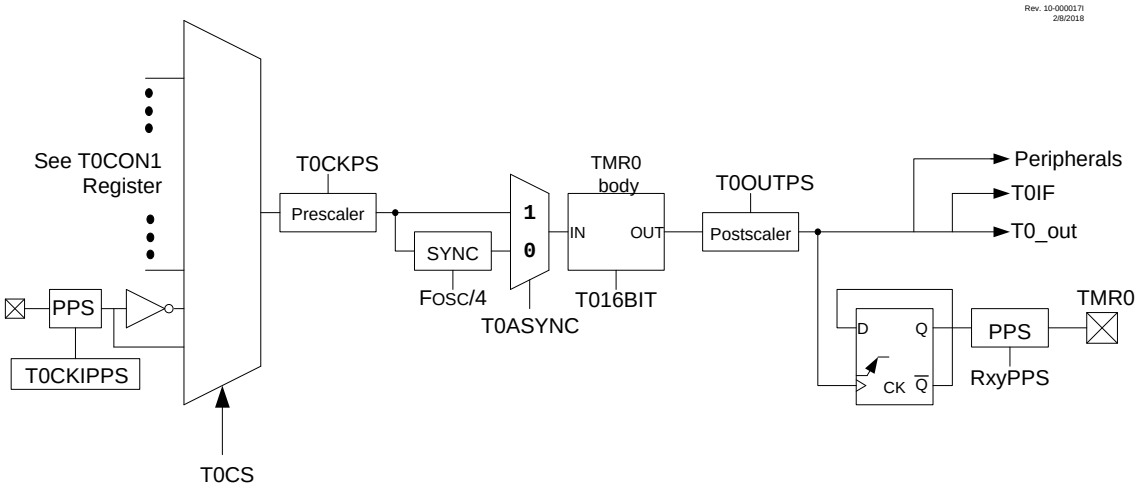
The CRC module can be used to detect bit errors in the Flash memory using the built-in memory scanner or through user input RAM memory. The CRC module can accept up to a 16-bit polynomial with up to a 16-bit seed value. A CRC calculated check value (or checksum) will then be generated into the 13.12.4 CRCACC registers for user storage. The CRC module uses an XOR shift register implementation to perform the polynomial division required for the CRC calculation.

18. Timer0 Module

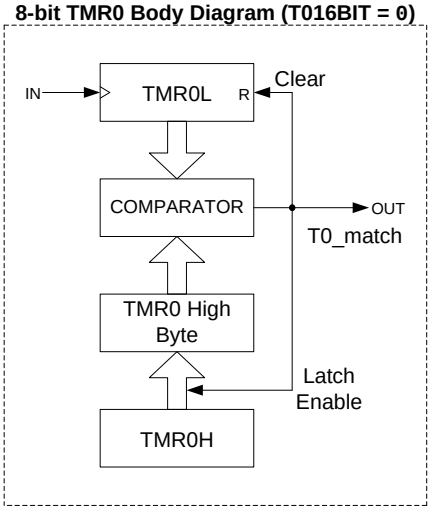
Timer0 module has the following features:

- 8-Bit B\Timer with Programmable Period
- 16-Bit Timer
- Selectable Clock Sources
- Synchronous and Asynchronous Operation
- Programmable Prescaler and Postscaler
- Interrupt on Match or Overflow
- Output on I/O Pin (via PPS) or to Other Peripherals
- Operation During Sleep

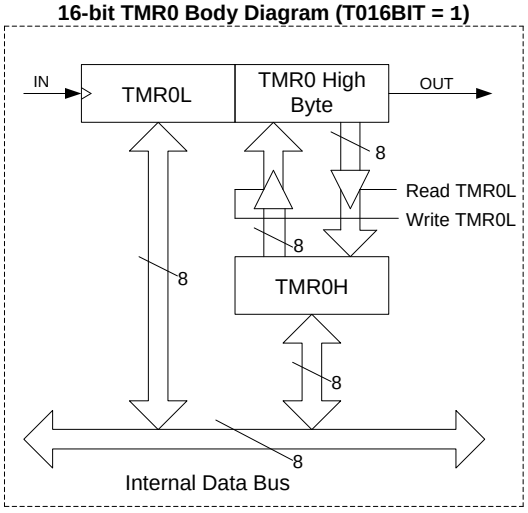
Figure 18-1. Timer0 Block Diagram



8-bit TMR0 Body Diagram (T016BIT = 0)



16-bit TMR0 Body Diagram (T016BIT = 1)



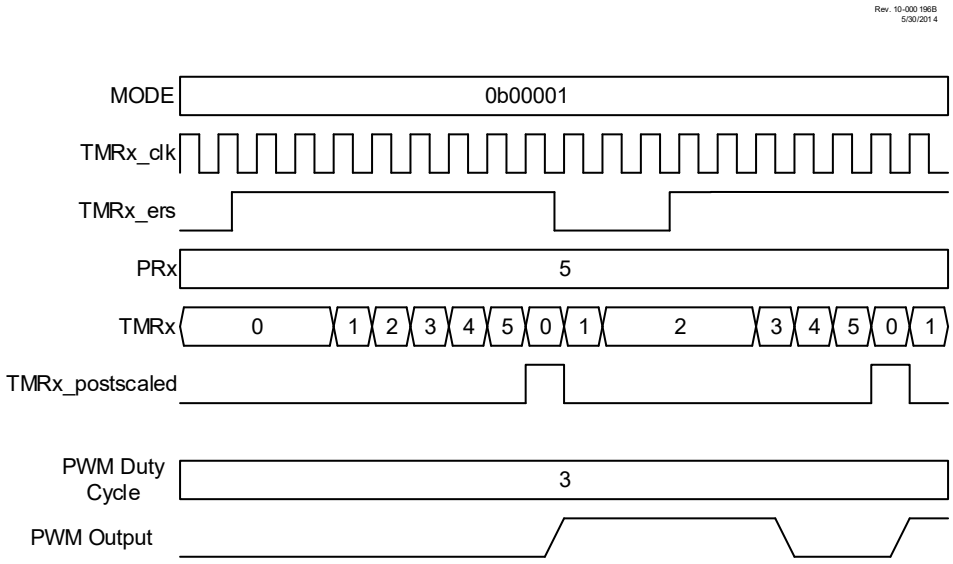
20.6.2 Hardware Gate Mode

The Hardware Gate modes operate the same as the Software Gate mode except the TMRx_ers external signal can also gate the timer. When used with the CCP, the gating extends the PWM period. If the timer is stopped when the PWM output is high, then the duty cycle is also extended.

When MODE<4:0> = 00001 then the timer is stopped when the external signal is high. When MODE<4:0> = 00010, then the timer is stopped when the external signal is low.

Figure 20-4 illustrates the Hardware Gating mode for MODE<4:0> = 00001 in which a high input level starts the counter.

Figure 20-4. Hardware Gate Mode Timing Diagram (MODE = 00001)



Related Links

21.4 PWM Overview

22. (PWM) Pulse-Width Modulation

20.6.3 Edge-Triggered Hardware Limit Mode

In Hardware Limit mode, the timer can be reset by the TMRx_ers external signal before the timer reaches the period count. Three types of Resets are possible:

- Reset on rising or falling edge (MODE<4:0>= 00011)
- Reset on rising edge (MODE<4:0> = 00100)
- Reset on falling edge (MODE<4:0> = 00101)

When the timer is used in conjunction with the CCP in PWM mode then an early Reset shortens the period and restarts the PWM pulse after a two clock delay. Refer to Figure 20-5.

20.9.2 TxPR

Name: TxPR
Address: 0xFBB,0xFB5,0xFAF

Timer Period Register

Bit	7	6	5	4	3	2	1	0
	TxPR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 7:0 – TxPR[7:0] Timer Period Register bits

Value	Description
0 - 255	The timer restarts at '0' when TxTMR reaches TxPR value

4. Write ‘55h’ to TXxREG to load the Sync character into the transmit FIFO buffer.
5. After the Break has been sent, the SENDB bit is reset by hardware and the Sync character is then transmitted.

When the TXxREG becomes empty, as indicated by the TXxIF, the next data byte can be written to TXxREG.

27.2.5 Receiving a Break Character

The EUSART module can receive a Break character in two ways.

The first method to detect a Break character uses the FERR bit of the RCxSTA register and the received data as indicated by RCxREG. The Baud Rate Generator is assumed to have been initialized to the expected baud rate.

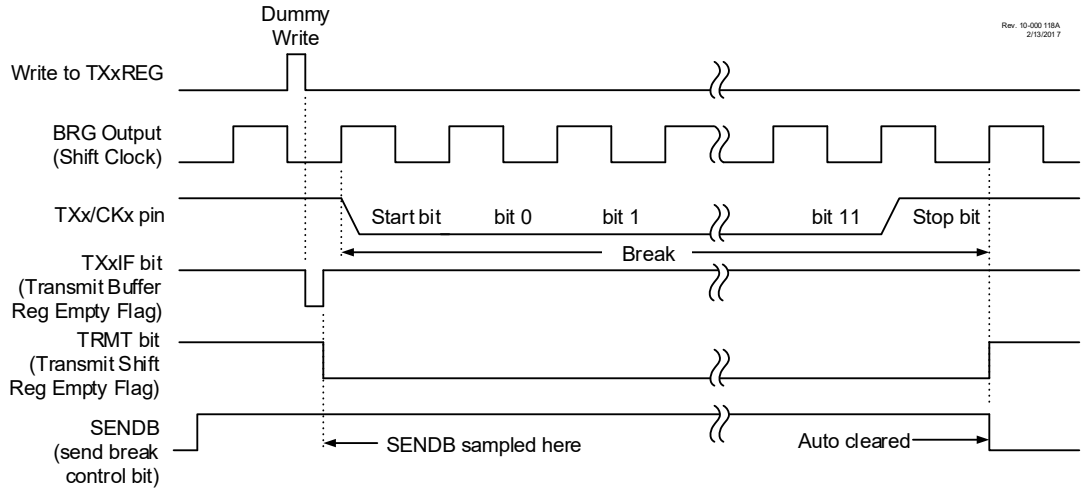
A Break character has been received when all three of the following conditions are true:

- RCxIF bit is set
- FERR bit is set
- RCxREG = 00h

The second method uses the Auto-Wake-up feature described in 27.2.3 Auto-Wake-up on Break. By enabling this feature, the EUSART will sample the next two transitions on RX/DT, cause an RCxIF interrupt, and receive the next data byte followed by another interrupt.

Note that following a Break character, the user will typically want to enable the Auto-Baud Detect feature. For both methods, the user can set the ABDEN bit of the BAUDxCON register before placing the EUSART in Sleep mode.

Figure 27-9. Send Break Character Sequence



27.3 EUSART Synchronous Mode

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

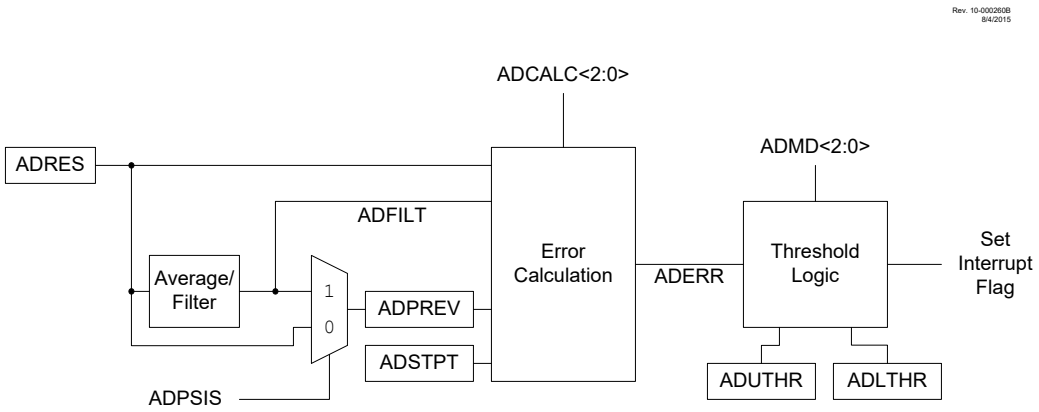
31.4.5 Additional Sample and Hold Capacitance

Additional capacitance can be added in parallel with the internal sample and hold capacitor (C_{HOLD}) by using the ADCAP register. This register selects a digitally programmable capacitance which is added to the ADC conversion bus, increasing the effective internal capacitance of the sample and hold capacitor in the ADC module. This is used to improve the match between internal and external capacitance for a better sensing performance. The additional capacitance does not affect analog performance of the ADC because it is not connected during conversion. See Figure 31-11.

31.5 Computation Operation

The ADC module hardware is equipped with post conversion computation features. These features provide data post-processing functions that can be operated on the ADC conversion result, including digital filtering/averaging and threshold comparison functions.

Figure 31-11. Computational Features Simplified Block Diagram



The operation of the ADC computational features is controlled by the 31.7.3.4 ADMD bits.

The module can be operated in one of five modes:

- Basic: This is a legacy mode. In this mode, ADC conversion occurs on single (ADDSEN = 0) or double (ADDSEN = 1) samples. ADIF is set after each conversion is complete.
- Accumulate: With each trigger, the ADC conversion result is added to the accumulator and ADCNT increments. ADIF is set after each conversion. ADTIF is set according to the calculation mode.
- Average: With each trigger, the ADC conversion result is added to the accumulator. When the ADRPT number of samples have been accumulated, a threshold test is performed. Upon the next trigger, the accumulator is cleared. For the subsequent tests, additional ADRPT samples are required to be accumulated.
- Burst Average: At the trigger, the accumulator is cleared. The ADC conversion results are then collected repetitively until ADRPT samples are accumulated and finally the threshold is tested.
- Low-Pass Filter (LPF): With each trigger, the ADC conversion result is sent through a filter. When ADRPT samples have occurred, a threshold test is performed. Every trigger after that the ADC conversion result is sent through the filter and another threshold test is performed.

The five modes are summarized in the following table.

31.7.12 ADRPT

Name: ADRPT
Address: 0xF61

ADC Repeat Setting Register

Bit	7	6	5	4	3	2	1	0
	ADRPT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – ADRPT[7:0] ADC Repeat Threshold bits
Determines the number of times that the ADC is triggered for a threshold check. When ADCNT reaches this value the error threshold is checked. Used when the computation mode is Low-pass Filter, Burst Average, or Average. See Table 31-4 for more details.

Address	Name	Bit Pos.								
0x0EF4	RC2PPS	7:0				PPS[4:0]				
0x0EF5	RC3PPS	7:0				PPS[4:0]				
0x0EF6	RC4PPS	7:0				PPS[4:0]				
0x0EF7	RC5PPS	7:0				PPS[4:0]				
0x0EF8	RC6PPS	7:0				PPS[4:0]				
0x0EF9	RC7PPS	7:0				PPS[4:0]				
0x0EFA ... 0x0F04	Reserved									
0x0F05	IOCAF	7:0	IOCAF7	IOCAF6	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0
0x0F06	IOCAN	7:0	IOCAN7	IOCAN6	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0
0x0F07	IOCAP	7:0	IOCAP7	IOCAP6	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0
0x0F08	INLVLA	7:0	INLVLA7	INLVLA6	INLVLA5	INLVLA4	INLVLA3	INLVLA2	INLVLA1	INLVLA0
0x0F09	SLRCONA	7:0	SLRA7	SLRA6	SLRA5	SLRA4	SLRA3	SLRA2	SLRA1	SLRA0
0x0F0A	ODCONA	7:0	ODCA7	ODCA6	ODCA5	ODCA4	ODCA3	ODCA2	ODCA1	ODCA0
0x0F0B	WPUA	7:0	WPUA7	WPUA6	WPUA5	WPUA4	WPUA3	WPUA2	WPUA1	WPUA0
0x0F0C	ANSELA	7:0	ANSELA7	ANSELA6	ANSELA5	ANSELA4	ANSELA3	ANSELA2	ANSELA1	ANSELA0
0x0F0D	IOCBF	7:0	IOCBF7	IOCBF6	IOCBF5	IOCBF4	IOCBF3	IOCBF2	IOCBF1	IOCBF0
0x0F0E	IOCBN	7:0	IOCBN7	IOCBN6	IOCBN5	IOCBN4	IOCBN3	IOCBN2	IOCBN1	IOCBN0
0x0F0F	IOCBP	7:0	IOCBP7	IOCBP6	IOCBP5	IOCBP4	IOCBP3	IOCBP2	IOCBP1	IOCBP0
0x0F10	INLVLB	7:0	INLVLB7	INLVLB6	INLVLB5	INLVLB4	INLVLB3	INLVLB2	INLVLB1	INLVLB0
0x0F11	SLRCONB	7:0	SLRB7	SLRB6	SLRB5	SLRB4	SLRB3	SLRB2	SLRB1	SLRB0
0x0F12	ODCONB	7:0	ODCB7	ODCB6	ODCB5	ODCB4	ODCB3	ODCB2	ODCB1	ODCB0
0x0F13	WPUB	7:0	WPUB7	WPUB6	WPUB5	WPUB4	WPUB3	WPUB2	WPUB1	WPUB0
0x0F14	ANSELB	7:0	ANSELB7	ANSELB6	ANSELB5	ANSELB4	ANSELB3	ANSELB2	ANSELB1	ANSELB0
0x0F15	IOCCF	7:0	IOCCF7	IOCCF6	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0
0x0F16	IOCCN	7:0	IOCCN7	IOCCN6	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0
0x0F17	IOCCP	7:0	IOCCP7	IOCCP6	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0
0x0F18	INLVLC	7:0	INLVLC7	INLVLC6	INLVLC5	INLVLC4	INLVLC3	INLVLC2	INLVLC1	INLVLC0
0x0F19	SLRCONC	7:0	SLRC7	SLRC6	SLRC5	SLRC4	SLRC3	SLRC2	SLRC1	SLRC0
0x0F1A	ODCONC	7:0	ODCC7	ODCC6	ODCC5	ODCC4	ODCC3	ODCC2	ODCC1	ODCC0
0x0F1B	WPUC	7:0	WPUC7	WPUC6	WPUC5	WPUC4	WPUC3	WPUC2	WPUC1	WPUC0
0x0F1C	ANSELC	7:0	ANSELC7	ANSELC6	ANSELC5	ANSELC4	ANSELC3	ANSELC2	ANSELC1	ANSELC0
0x0F1D ... 0x0F21	Reserved									
0x0F22	IOCEF	7:0					IOCEF3			
0x0F23	IOCEN	7:0					IOCEN3			
0x0F24	IOCEP	7:0					IOCEP3			
0x0F25	INLVLE	7:0					INLVLE3			
0x0F26 ... 0x0F27	Reserved									
0x0F28	WPUE	7:0					WPUE3			
0x0F29	Reserved									
0x0F2A	HLVDCON0	7:0	EN		OUT	RDY			INTH	INTL

BSF	Bit Set f			
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 ≤ 95 (5Fh). See 36.2.3 Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode for details.			
Words:	1			
Cycles:	1			

Q Cycle Activity:			
Q1	Q2	Q3	Q4
Decode	Read register ‘f’	Process Data	Write register ‘f’

Example:	BSF	FLAG_REG, 7, 1
Before Instruction FLAG_REG = 0Ah After Instruction FLAG_REG = 8Ah		

BTFSC	Bit Test File, Skip if Clear			
Syntax:	BTFSC f, b {,a}			
Operands:	0 ≤ f ≤ 255 0 ≤ b ≤ 7 a ∈ [0,1]			
Operation:	skip if (f<b>) = 0			
Status Affected:	None			
Encoding:	1011	bbba	ffff	ffff
Description:	If bit ‘b’ in register ‘f’ is ‘0’, then the next instruction is skipped. If bit ‘b’ is ‘0’, then the next instruction fetched during the current instruction execution is discarded and a NOP is executed instead, making this a 2-cycle instruction. If ‘a’ is ‘0’, the Access Bank is selected. If ‘a’ is ‘1’, the BSR is used to select the GPR bank. If ‘a’ is ‘0’ and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing			

After Instruction W = 01h C = 1 ; result is positive Z = 0 N = 0		
Example 2:	SUBLW	02h
Before Instruction W = 02h C = ? After Instruction W = 00h C = 1 ; result is zero Z = 1 N = 0		
Example 3:	SUBLW	02h
Before Instruction W = 03h C = ? After Instruction W = FFh ; (2's complement) C = 0 ; result is negative Z = 0 N = 1		

SUBWF	Subtract W from f			
Syntax:	SUBWF f {,d {,a}}			
Operands:	0 ≤ f ≤ 255 d ∈ [0,1] a ∈ [0,1]			
Operation:	(f) – (W) → dest			
Status Affected:	N, OV, C, DC, Z			
Encoding:	0101	11da	ffff	ffff

W = 0 C = 1 ; result is zero Z = 1 N = 0	
Example 3:	`SUBWF REG, 1, 0`
Before Instruction REG = 1 W = 2 C = ? After Instruction REG = FFh ;(2's complement) W = 2 C = 0 ; result is negative Z = 0 N = 1	

SUBWFB	Subtract W from f with Borrow			
Syntax:	SUBWFB f {,d {,a}}			
Operands:	0 ≤ f ≤ 255 d ∈ [0,1] a ∈ [0,1]			
Operation:	(f) – (W) – ($\overline{C}$) → dest			
Status Affected:	N, OV, C, DC, Z			
Encoding:	0101	10da	ffff	ffff
Description:	Subtract W and the CARRY flag (borrow) from register 'f' (2's complement method). If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f' (default). If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank. If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever f ≤ 95 (5Fh). See 36.2.3 Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode for details.			
Words:	1			
Cycles:	1			

The destination argument, ‘d’, functions as before.

In the latest versions of the MPASM™ assembler, language support for the extended instruction set must be explicitly invoked. This is done with either the command line option, /y, or the PE directive in the source listing.

Related Links

10.5 Data Memory and the Extended Instruction Set

36.2.4 Considerations when Enabling the Extended Instruction Set

It is important to note that the extensions to the instruction set may not be beneficial to all users. In particular, users who are not writing code that uses a software stack may not benefit from using the extensions to the instruction set.

Additionally, the Indexed Literal Offset Addressing mode may create issues with legacy applications written to the PIC18 assembler. This is because instructions in the legacy code may attempt to address registers in the Access Bank below 5Fh. Since these addresses are interpreted as literal offsets to FSR2 when the instruction set extension is enabled, the application may read or write to the wrong data addresses.

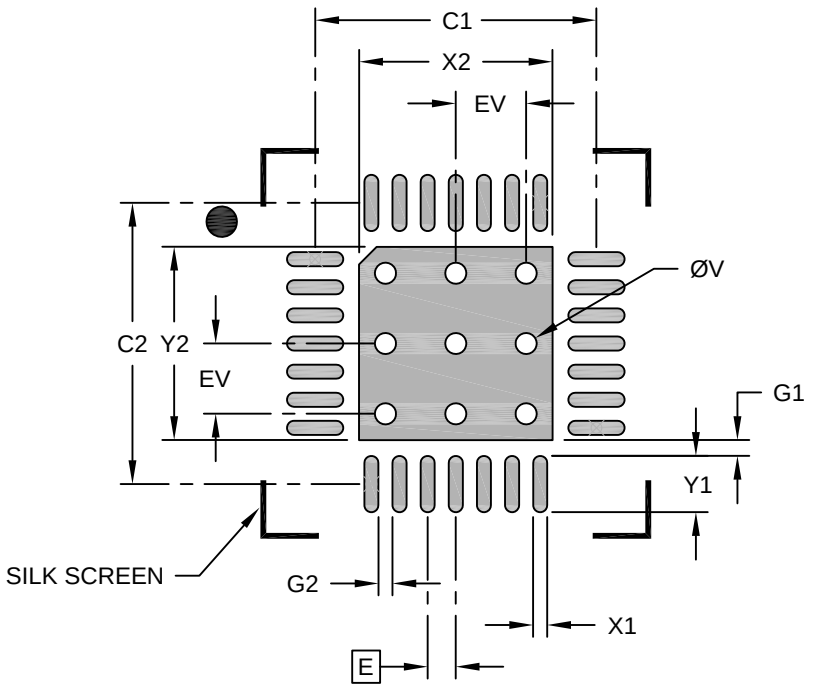
When porting an application to the PIC18F24/25Q10 it is very important to consider the type of code. A large, re-entrant application that is written in ‘C’ and would benefit from efficient compilation will do well when using the instruction set extensions. Legacy applications that heavily use the Access Bank will most likely not benefit from using the extended instruction set.

ADDWF	ADD W to Indexed (Indexed Literal Offset mode)			
Syntax:	ADDWF [k] {,d}			
Operands:	0 ≤ k ≤ 95 d ∈ [0,1]			
Operation:	(W) + ((FSR2) + k) → dest			
Status Affected:	N, OV, C, DC, Z			
Encoding:	0010	01d0	kkkk	kkkk
Description:	The contents of W are added to the contents of the register indicated by FSR2, offset by the value ‘k’. If ‘d’ is ‘0’, the result is stored in W. If ‘d’ is ‘1’, the result is stored back in register ‘f’ (default).			
Words:	1			
Cycles:	1			

Q Cycle Activity:			
Q1	Q2	Q3	Q4
Decode	Read ‘k’	Process Data	Write to destination

28-Lead Very Thin Plastic Quad Flat, No Lead (STX) - 4x4 mm Body [VQFN] With 2.65x2.65 mm Exposed Pad

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.40 BSC	
Optional Center Pad Width	X2			2.75
Optional Center Pad Length	Y2			2.75
Contact Pad Spacing	C1		4.00	
Contact Pad Spacing	C2		4.00	
Contact Pad Width (X28)	X1			0.20
Contact Pad Length (X28)	Y1			0.80
Contact Pad to Center Pad (X28)	G1	0.23		
Contact Pad to Contact Pad (X24)	G2	0.20		
Thermal Via Diameter	V		0.30	
Thermal Via Pitch	EV		1.00	

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

2. For best soldering results, thermal vias, if used, should be filled or tented to avoid solder loss during reflow process

Microchip Technology Drawing C04-2456 Rev A

41. Revision History

Doc Rev.	Date	Comments
A	01/2018	Initial document release.
B	04/2018	Paragraph clarifications: 3.1, 3.2, 3.4, 10.4, 10.4.3 (example), 10.4.3.1, 10.4.3.3, 11.1, 11.1.4 (note), 11.1.4.1, 15.2.6 (related links), 17.3 (note), 33.10.2; Figure updates: 10-9, 10-10, 11-9; Table updates: 38-2, 38-3, 38-7, 38-9, 39-17; Replaced UQFN package with VQFN.