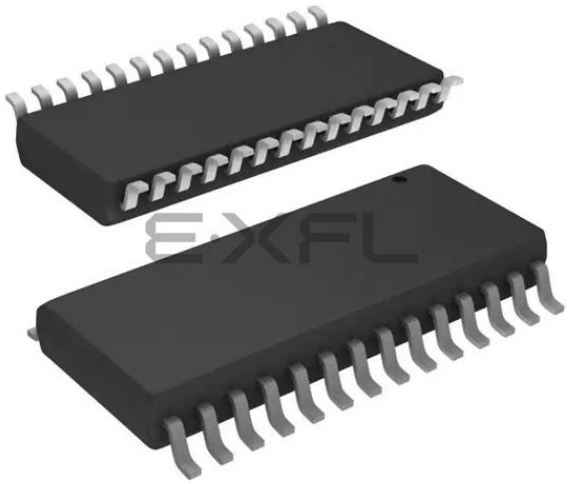


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

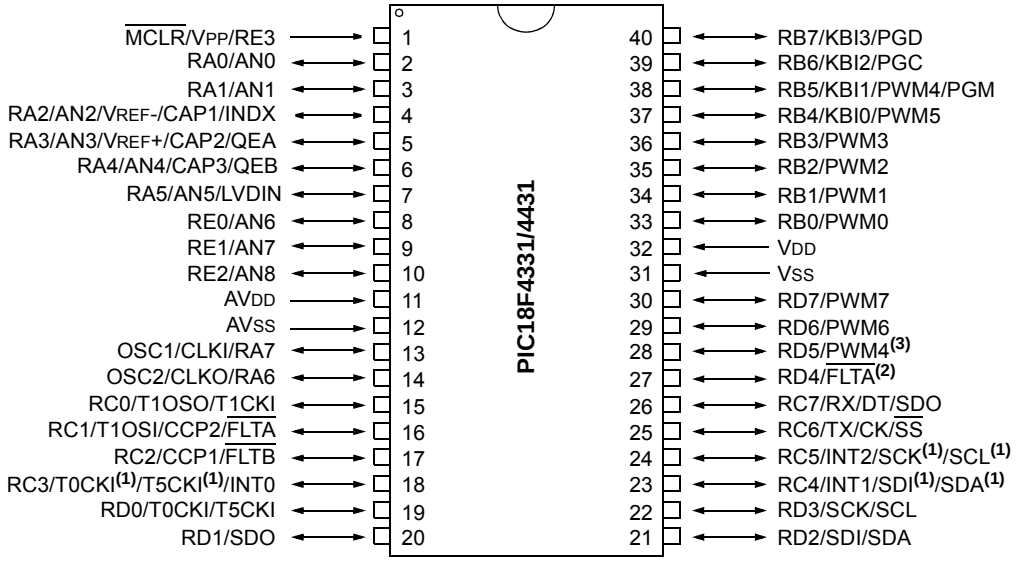
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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	40MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, Power Control PWM, QEI, POR, PWM, WDT
Number of I/O	24
Program Memory Size	16KB (8K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	768 x 8
Voltage - Supply (Vcc/Vdd)	4.2V ~ 5.5V
Data Converters	A/D 5x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°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/pic18f2431-e-so

Pin Diagrams (Continued)

40-Pin PDIP



Note 1: RC3 is the alternate pin for T0CKI/T5CKI; RC4 is the alternate pin for SDI/SDA; RC5 is the alternate pin for SCK/SCL.

2: RD4 is the alternate pin for FLTA.

3: RD5 is the alternate pin for PWM4.

TABLE 1-3: PIC18F4331/4431 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	PDIP	TQFP	QFN			
						PORTC is a bidirectional I/O port.
RC0/T1OSO/T1CKI	15	32	34			
RC0				I/O	ST	Digital I/O.
T1OSO				O	—	Timer1 oscillator output.
T1CKI				I	ST	Timer1 external clock input.
RC1/T1OSI/CCP2/$\overline{\text{FLTA}}$	16	35	35			
RC1				I/O	ST	Digital I/O.
T1OSI				I	CMOS	Timer1 oscillator input.
CCP2				I/O	ST	Capture 2 input, Compare 2 output, PWM2 output.
$\overline{\text{FLTA}}$				I	ST	Fault interrupt input pin.
RC2/CCP1/$\overline{\text{FLTB}}$	17	36	36			
RC2				I/O	ST	Digital I/O.
CCP1				I/O	ST	Capture 1 input/Compare 1 output/PWM1 output.
$\overline{\text{FLTB}}$				I	ST	Fault interrupt input pin.
RC3/T0CKI/T5CKI/INT0	18	37	37			
RC3				I/O	ST	Digital I/O.
T0CKI(1)				I	ST	Timer0 alternate clock input.
T5CKI(1)				I	ST	Timer5 alternate clock input.
INT0				I	ST	External Interrupt 0.
RC4/INT1/SDI/SDA	23	42	42			
RC4				I/O	ST	Digital I/O.
INT1				I	ST	External Interrupt 1.
SDI(1)				I	ST	SPI data in.
SDA(1)				I/O	I^2C	I^2C™ data I/O.
RC5/INT2/SCK/SCL	24	43	43			
RC5				I/O	ST	Digital I/O.
INT2				I	ST	External Interrupt 2.
SCK(1)				I/O	ST	Synchronous serial clock input/output for SPI mode.
SCL(1)				I/O	I^2C	Synchronous serial clock input/output for I^2C mode.
RC6/TX/CK/$\overline{\text{SS}}$	25	44	44			
RC6				I/O	ST	Digital I/O.
TX				O	—	EUSART asynchronous transmit.
CK				I/O	ST	EUSART synchronous clock (see related RX/DT).
$\overline{\text{SS}}$				I	ST	SPI slave select input.
RC7/RX/DT/SDO	26	1	1			
RC7				I/O	ST	Digital I/O.
RX				I	ST	EUSART asynchronous receive.
DT				I/O	ST	EUSART synchronous data (see related TX/CK).
SDO(1)				O	—	SPI data out.

Legend: TTL = TTL compatible input; CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; I = Input; O = Output; P = Power

Note 1: RC3 is the alternate pin for T0CKI/T5CKI; RC4 is the alternate pin for SDI/SDA; RC5 is the alternate pin for SCK/SCL; RC7 is the alternate pin for SDO.

2: RD4 is the alternate pin for $\overline{\text{FLTA}}$.

3: RD5 is the alternate pin for PWM4.

© 2010 Microchip Technology Inc.

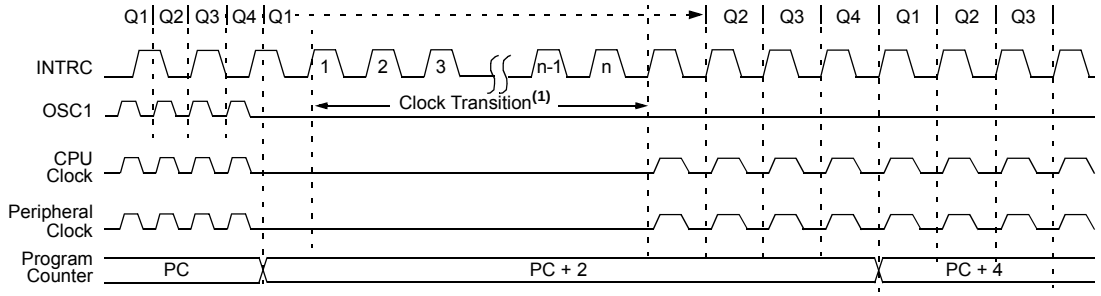
If the IRCF bits and the INTSRC bit are all clear, the INTOSC output is not enabled and the IOFS bit will remain clear; there will be no indication of the current clock source. The INTRC source is providing the device clocks.

If the IRCF bits are changed from all clear (thus, enabling the INTOSC output), or if INTSRC is set, the IOFS bit becomes set after the INTOSC output becomes stable. Clocks to the device continue while the INTOSC source stabilizes, after an interval of TIOBST.

If the IRCF bits were previously at a non-zero value, or if INTSRC was set before setting SCS1 and the INTOSC source was already stable, the IOFS bit will remain set.

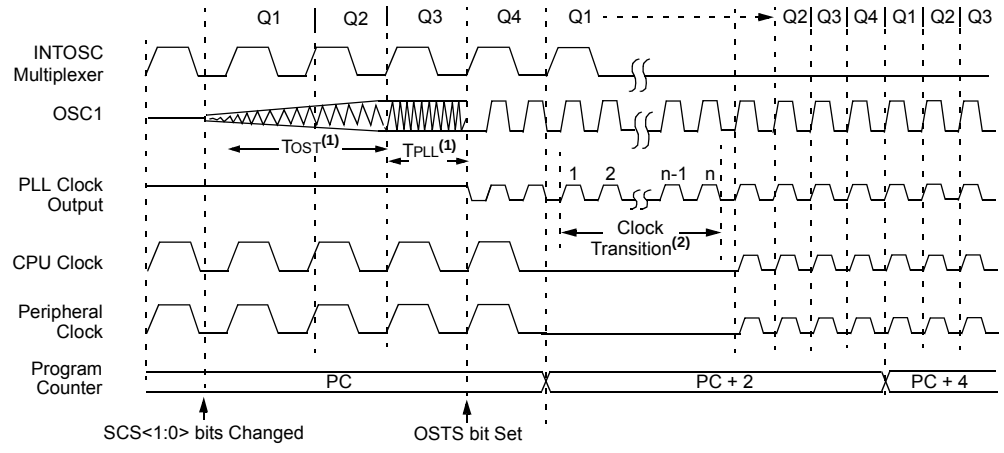
On transitions from RC_RUN mode to PRI_RUN mode, the device continues to be clocked from the INTOSC multiplexer while the primary clock is started. When the primary clock becomes ready, a clock switch to the primary clock occurs (see Figure 4-4). When the clock switch is complete, the IOFS bit is cleared, the OSTS bit is set and the primary clock is providing the device clock. The IDLEN and SCS bits are not affected by the switch. The INTRC source will continue to run if either the WDT or the Fail-Safe Clock Monitor is enabled.

FIGURE 4-3: TRANSITION TIMING TO RC_RUN MODE



Note 1: Clock transition typically occurs within 2-4 TOSC.

FIGURE 4-4: TRANSITION TIMING FROM RC_RUN MODE TO PRI_RUN MODE



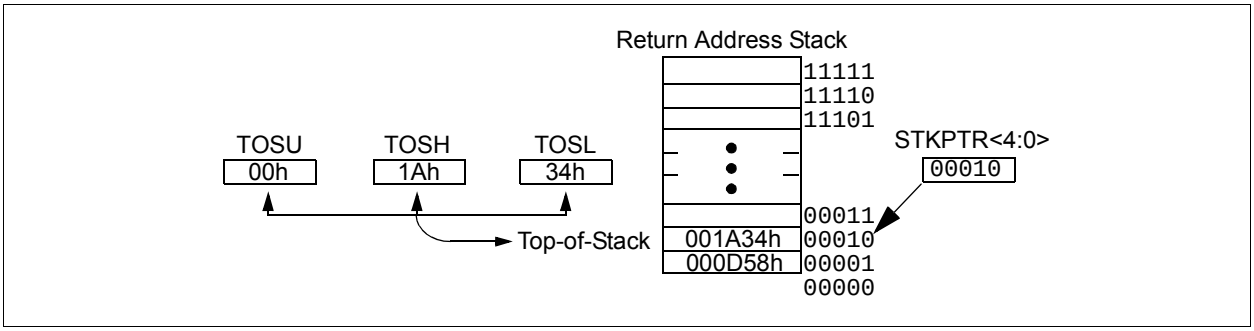
Note 1: TOST = 1024 TOSC; TPLL = 2 ms (approx). These intervals are not shown to scale.
2: Clock transition typically occurs within 2-4 TOSC.

© 2010 Microchip Technology Inc.

When the stack has been popped enough times to unload the stack, the next pop will return a value of zero to the PC and set the STKUNF bit, while the Stack Pointer remains at zero. The STKUNF bit will remain set until cleared by software or a POR occurs.

Note:	Returning a value of zero to the PC on an underflow has the effect of vectoring the program to the Reset vector, where the stack conditions can be verified and appropriate actions can be taken. This is not the same as a Reset as the contents of the SFRs are not affected.

FIGURE 6-3: RETURN ADDRESS STACK AND ASSOCIATED REGISTERS



REGISTER 6-1: STKPTR: STACK POINTER REGISTER

R/C-0	R/C-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
STKFUL(1)	STKUNF(1)	—	SP4	SP3	SP2	SP1	SP0
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 **STKFUL:** Stack Full Flag bit(1)
1 = Stack became full or overflowed
0 = Stack has not become full or overflowed

bit 6 **STKUNF:** Stack Underflow Flag bit(1)
1 = Stack underflow occurred
0 = Stack underflow did not occur

bit 5 **Unimplemented:** Read as ‘0’

bit 4-0 **SP<4:0>:** Stack Pointer Location bits

Note 1: Bit 7 and bit 6 are cleared by user software or by a POR.

REGISTER 10-2: INTCON2: INTERRUPT CONTROL REGISTER 2

R/W-1	R/W-1	R/W-1	R/W-1	U-0	R/W-1	U-0	R/W-1
$\overline{\text{RBPU}}$	INTEDG0	INTEDG1	INTEDG2	—	TMR0IP	—	RBIP
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 **$\overline{\text{RBPU}}$:** PORTB Pull-up Enable bit
1 = All PORTB pull-ups are disabled
0 = PORTB pull-ups are enabled by individual port latch values

bit 6 **INTEDG0:** External Interrupt 0 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge

bit 5 **INTEDG1:** External Interrupt 1 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge

bit 4 **INTEDG2:** External Interrupt 2 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge

bit 3 **Unimplemented:** Read as ‘0’

bit 2 **TMR0IP:** TMR0 Overflow Interrupt Priority bit
1 = High priority
0 = Low priority

bit 1 **Unimplemented:** Read as ‘0’

bit 0 **RBIP:** RB Port Change Interrupt Priority bit
1 = High priority
0 = Low priority

Note: Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the global enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt. This feature allows for software polling.

© 2010 Microchip Technology Inc.

11.5 PORTE, TRISE and LATE Registers

Note: PORTE is only available on PIC18F4331/4431 devices.

PORTE is a 4-bit wide, bidirectional port. Three pins (RE0/AN6, RE1/AN7 and RE2/AN8) are individually configurable as inputs or outputs. These pins have Schmitt Trigger input buffers. When selected as an analog input, these pins will read as '0's.

The corresponding Data Direction register is TRISE. Setting a TRISE bit (= 1) will make the corresponding PORTE pin an input (i.e., put the corresponding output driver in a high-impedance mode). Clearing a TRISE bit (= 0) will make the corresponding PORTE pin an output (i.e., put the contents of the output latch on the selected pin).

TRISE controls the direction of the RE pins, even when they are being used as analog inputs. The user must make sure to keep the pins configured as inputs when using them as analog inputs.

Note: On a Power-on Reset, RE<2:0> are configured as analog inputs.

The Data Latch register (LATE) is also memory mapped. Read-modify-write operations on the LATE register read and write the latched output value for PORTE.

The fourth pin of PORTE ($\overline{MCLR}$/VPP/RE3) is an input only pin available for PIC18F4331/4431 devices. Its operation is controlled by the MCLRE Configuration bit in Configuration Register 3H (CONFIG3H<7>). When selected as a port pin (MCLRE = 0), it functions as a digital input-only pin. As such, it does not have TRIS or LAT bits associated with its operation. Otherwise, it functions as the device's master clear input. In either configuration, RE3 also functions as the programming voltage input during programming.

Note: On a Power-on Reset, RE3 is enabled as a digital input only if Master Clear functionality is disabled.

EXAMPLE 11-5: INITIALIZING PORTE

```
CLRF    PORTE       ; Initialize PORTE by
                    ; clearing output
                    ; data latches
CLRF    LATE        ; Alternate method
                    ; to clear output
                    ; data latches
MOVLW   0x3F        ; Configure A/D
MOVWF   ANSEL0      ; for digital inputs
BCF     ANSEL1, 0   ;
MOVLW   0x03        ; Value used to
                    ; initialize data
                    ; direction
MOVWF   TRISE       ; Set RE<0> as input
                    ; RE<1> as output
                    ; RE<2> as input
```

11.5.1 PORTE IN 28-PIN DEVICES

For PIC18F2331/2431 devices, PORTE is not available. It is only available for PIC18F4331/4431 devices.

REGISTER 11-1: TRISE REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-1	R/W-1
—	—	—	—	—	TRISE2	TRISE1	TRISE0
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 7-3 **Unimplemented:** Read as '0'

bit 2 **TRISE2**: RE2 Direction Control bit
1 = Input
0 = Output

bit 1 **TRISE1**: RE1 Direction Control bit
1 = Input
0 = Output

bit 0 **TRISE0**: RE0 Direction Control bit
1 = Input
0 = Output

© 2010 Microchip Technology Inc.

15.0 TIMER5 MODULE

The Timer5 module implements these features:

- 16-bit timer/counter operation
- Synchronous and Asynchronous Counter modes
- Continuous Count and Single-Shot Operating modes
- Four programmable prescaler values (1:1 to 1:8)
- Interrupt generated on period match
- Special Event Trigger Reset function
- Double-buffered registers
- Operation during Sleep
- CPU wake-up from Sleep
- Selectable hardware Reset input with a wake-up feature

Timer5 is a general purpose timer/counter that incorporates additional features for use with the Motion Feedback Module (see **Section 17.0 "Motion Feedback Module"**). It may also be used as a general purpose timer or a Special Event Trigger delay timer. When used as a general purpose timer, it can be configured to generate a delayed Special Event Trigger (e.g., an ADC Special Event Trigger) using a preprogrammed period delay.

Timer5 is controlled through the Timer5 Control register (T5CON), shown in Register 15-1. The timer can be enabled or disabled by setting or clearing the control bit TMR5ON (T5CON<0>).

A block diagram of Timer5 is shown in Figure 15-1.

REGISTER 15-1: T5CON: TIMER5 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
T5SEN	$\overline{\text{RESEN}}$(1)	T5MOD	T5PS1	T5PS0	$\overline{\text{T5SYNC}}$(2)	TMR5CS	TMR5ON
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 **T5SEN:** Timer5 Sleep Enable bit
1 = Timer5 is enabled during Sleep
0 = Timer5 is disabled during Sleep

bit 6 **$\overline{\text{RESEN}}$:** Special Event Trigger Reset Enable bit(1)
1 = Special Event Trigger Reset is disabled
0 = Special Event Trigger Reset is enabled

bit 5 **T5MOD:** Timer5 Mode bit
1 = Single-Shot mode is enabled
0 = Continuous Count mode is enabled

bit 4-3 **T5PS<1:0>:** Timer5 Input Clock Prescale Select bits
11 = 1:8
10 = 1:4
01 = 1:2
00 = 1:1

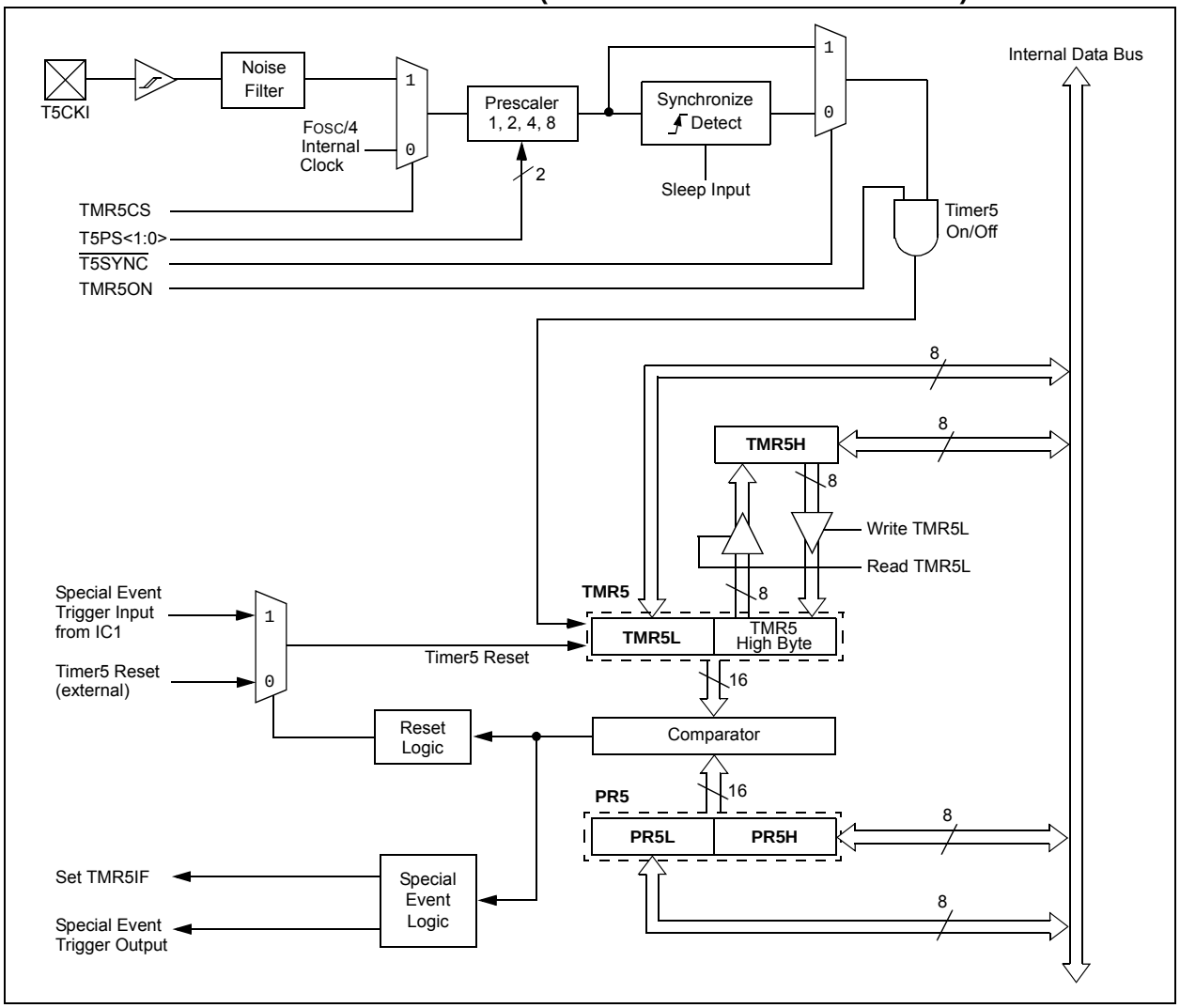
bit 2 **$\overline{\text{T5SYNC}}$:** Timer5 External Clock Input Synchronization Select bit(2)
When TMR5CS = 1:
1 = Do not synchronize external clock input
0 = Synchronize external clock input
When TMR5CS = 0:
This bit is ignored. Timer5 uses the internal clock when TMR5CS = 0.

bit 1 **TMR5CS:** Timer5 Clock Source Select bit
1 = External clock from the T5CKI pin
0 = Internal clock (T_{CY})

bit 0 **TMR5ON:** Timer5 On bit
1 = Timer5 is enabled
0 = Timer5 is disabled

Note 1: These bits are not implemented on PIC18F2331/2431 devices and read as '0'.
2: For Timer5 to operate during Sleep mode, $\overline{\text{T5SYNC}}$ must be set.

FIGURE 15-1: TIMER5 BLOCK DIAGRAM (16-BIT READ/WRITE MODE SHOWN)



15.1 Timer5 Operation

Timer5 combines two 8-bit registers to function as a 16-bit timer. The TMR5L register is the actual low byte of the timer; it can be read and written to directly. The high byte is contained in an unmapped register; it is read and written to through TMR5H, which serves as a buffer. Each register increments from 00h to FFh.

A second register pair, PR5H and PR5L, serves as the Period register; it sets the maximum count for the TMR5 register pair. When TMR5 reaches the value of PR5, the timer rolls over to 00h and sets the TMR5IF interrupt flag. A simplified block diagram of the Timer5 module is shown in Figure 2-1.

Note:	The Timer5 may be used as a general purpose timer and as the time base resource to the Motion Feedback Module (Input Capture or Quadrature Encoder Interface).

Timer5 supports three configurations:

- 16-Bit Synchronous Timer
- 16-Bit Synchronous Counter
- 16-Bit Asynchronous Counter

In Synchronous Timer configuration, the timer is clocked by the internal device clock. The optional Timer5 prescaler divides the input by 2, 4, 8 or not at all (1:1). The TMR5 register pair increments on Q1. Clearing TMR5CS (= 0) selects the internal device clock as the timer sampling clock.

© 2010 Microchip Technology Inc.

16.4 Compare Mode

In Compare mode, the 16-bit CCPR1 (CCPR2) register value is constantly compared against the TMR1 register pair value. When a match occurs, the RC2/CCP1 (RC1/CCP2) pin:

- is driven high
- is driven low
- toggles output (high-to-low or low-to-high)
- remains unchanged (interrupt only)

The action on the pin is based on the value of control bits, CCP1M<3:0> (CCP2M<3:0>). At the same time, interrupt flag bit CCP1IF (CCP2IF) is set.

16.4.1 CCP PIN CONFIGURATION

The user must configure the CCP1 pin as an output by clearing the appropriate TRISC bit.

Note:	Clearing the CCPxCON register will force the RC1 or RC2 compare output latch to the default low level. This is not the PORTC I/O data latch.

16.4.2 TIMER1 MODE SELECTION

Timer1 must be running in Timer mode or Synchronized Counter mode if the CCP module is using the compare feature. In Asynchronous Counter mode, the compare operation may not work.

16.4.3 SOFTWARE INTERRUPT MODE

When generate software interrupt is chosen, the CCP1 pin is not affected. Only a CCP interrupt is generated (if enabled).

16.4.4 SPECIAL EVENT TRIGGER

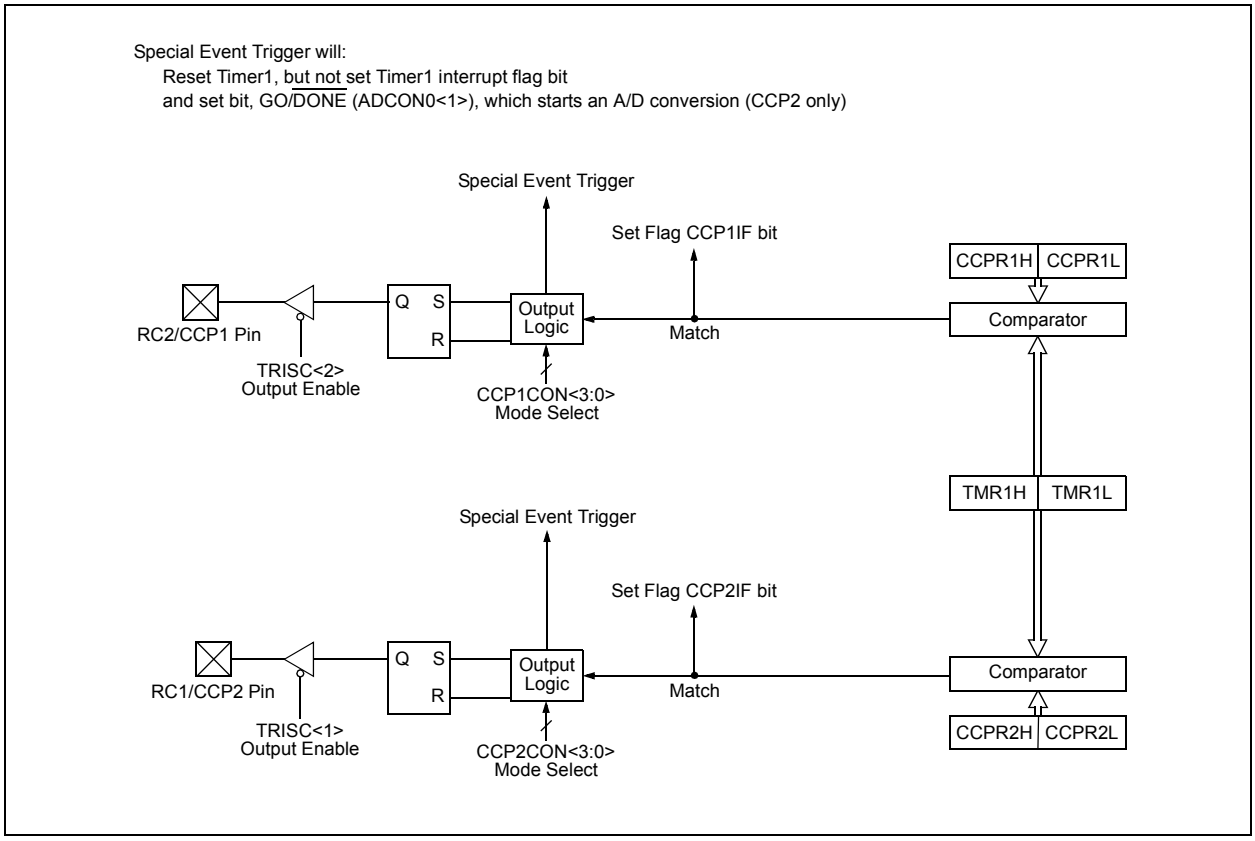
In this mode, an internal hardware trigger is generated which may be used to initiate an action.

The Special Event Trigger output of CCP1 resets the TMR1 register pair. This allows the CCPR1 register to effectively be a 16-bit programmable period register for Timer1.

The Special Event Trigger output of CCP2 resets the TMR1 register pair. Additionally, the CCP2 Special Event Trigger will start an A/D conversion if the A/D module is enabled.

Note:	The Special Event Trigger from the CCP2 module will not set the Timer1 interrupt flag bit.

FIGURE 16-2: COMPARE MODE OPERATION BLOCK DIAGRAM



REGISTER 18-8: FLTCONFIG: FAULT CONFIGURATION 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
BRFEN	FLTBS(1)	FLTBMOD(1)	FLTBEN(1)	FLTCON(2)	FLTAS	FLTAMOD	FLTAEN
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 **BRFEN:** Breakpoint Fault Enable bit
- 1 = Enable Fault condition on a breakpoint (i.e., only when PWMPIN = 1)
- 0 = Disable Fault condition

bit 6 **FLTBS:** Fault B Status bit(1)
- 1 = $\overline{\text{FLTB}}$ is asserted:
  if FLTBMOD = 0, cleared by the user;
  if FLTBMOD = 1, cleared automatically at beginning of the new period when $\overline{\text{FLTB}}$ is deasserted
- 0 = No Fault

bit 5 **FLTBMOD:** Fault B Mode bit(1)
- 1 = Cycle-by-Cycle mode: Pins are inactive for the remainder of the current PWM period or until $\overline{\text{FLTB}}$ is deasserted; FLTBS is cleared automatically when $\overline{\text{FLTB}}$ is inactive (no Fault present)
- 0 = Inactive mode: Pins are deactivated (catastrophic failure) until $\overline{\text{FLTB}}$ is deasserted and FLTBS is cleared by the user only

bit 4 **FLTBEN:** Fault B Enable bit(1)
- 1 = Enable Fault B
- 0 = Disable Fault B

bit 3 **FLTCON:** Fault Configuration bit(2)
- 1 = $\overline{\text{FLTA}}$, $\overline{\text{FLTB}}$ or both deactivates all PWM outputs
- 0 = $\overline{\text{FLTA}}$ or $\overline{\text{FLTB}}$ deactivates PWM<5:0>

bit 2 **FLTAS:** Fault A Status bit
- 1 = $\overline{\text{FLTA}}$ is asserted:
  if FLTAMOD = 0, cleared by the user;
  if FLTAMOD = 1, cleared automatically at beginning of the new period when $\overline{\text{FLTA}}$ is deasserted
- 0 = No Fault

bit 1 **FLTAMOD:** Fault A Mode bit
- 1 = Cycle-by-Cycle mode: Pins are inactive for the remainder of the current PWM period or until $\overline{\text{FLTA}}$ is deasserted; FLTAS is cleared automatically
- 0 = Inactive mode: Pins are deactivated (catastrophic failure) until $\overline{\text{FLTA}}$ is deasserted and FLTAS is cleared by the user only

bit 0 **FLTAEN:** Fault A Enable bit
- 1 = Enable Fault A
- 0 = Disable Fault A

Note 1: Unimplemented in PIC18F2331/2431 devices; maintain these bits clear.

2: PWM<7:6> are implemented only on PIC18F4331/4431 devices. On PIC18F2331/2431 devices, setting or clearing FLTCON has no effect.

18.13 PWM Update Lockout

For a complex PWM application, the user may need to write up to four Duty Cycle registers and the PWM Time Base Period register, PTPER, at a given time. In some applications, it is important that all buffer registers be written before the new duty cycle and period values are loaded for use by the module.

A PWM update lockout feature may optionally be enabled so the user may specify when new duty cycle buffer values are valid. The PWM update lockout feature is enabled by setting the control bit, UDIS, in the PWMCON1 register. This bit affects all Duty Cycle Buffer registers and the PWM Time Base Period register, PTPER.

To perform a PWM update lockout:

1. Set the UDIS bit.
2. Write all Duty Cycle registers and PTPER, if applicable.
3. Clear the UDIS bit to re-enable updates.
4. With this, when UDIS bit is cleared, the buffer values will be loaded to the actual registers. This makes a synchronous loading of the registers.

18.14 PWM Special Event Trigger

The PWM module has a Special Event Trigger capability that allows A/D conversions to be synchronized to the PWM time base. The A/D sampling and conversion time may be programmed to occur at any point within the PWM period. The Special Event Trigger allows the user to minimize the delay between the time when A/D conversion results are acquired and the time when the duty cycle value is updated.

The PWM 16-bit Special Event Trigger register, SEVTCMP (high and low), and five control bits in the PWMCON1 register are used to control its operation.

The PTMR value for which a Special Event Trigger should occur is loaded into the SEVTCMP register pair. The SEVTDIR bit in the PWMCON1 register specifies the counting phase when the PWM time base is in a Continuous Up/Down Count mode.

If the SEVTDIR bit is cleared, the Special Event Trigger will occur on the upward counting cycle of the PWM time base. If SEVTDIR is set, the Special Event Trigger will occur on the downward count cycle of the PWM time base. The SEVTDIR bit has effect only when the PWM timer is in the Continuous Up/Down Count mode.

18.14.1 SPECIAL EVENT TRIGGER ENABLE

The PWM module will always produce Special Event Trigger pulses. This signal may optionally be used by the A/D module. Refer to **Section 21.0 “10-Bit High-Speed Analog-to-Digital Converter (A/D) Module”** for details.

18.14.2 SPECIAL EVENT TRIGGER POSTSCALER

The PWM Special Event Trigger has a postscaler that allows a 1:1 to 1:16 postscale ratio. The postscaler is configured by writing the SEVOPS<3:0> control bits in the PWMCON1 register.

The Special Event Trigger output postscaler is cleared on any write to the SEVTCMP register pair, or on any device Reset.

© 2010 Microchip Technology Inc.

TABLE 18-6: REGISTERS ASSOCIATED WITH THE POWER CONTROL PWM MODULE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	54
IPR3	—	—	—	PTIP	IC3DRIP	IC2QEIP	IC1IP	TMR5IP	56
PIE3	—	—	—	PTIE	IC3DRIE	IC2QEIE	IC1IE	TMR5IE	56
PIR3	—	—	—	PTIF	IC3DRIF	IC2QEIF	IC1IF	TMR5IF	56
PTCON0	PTOPS3	PTOPS2	PTOPS1	PTOPS0	PTCKPS1	PTCKPS0	PTMOD1	PTMOD0	58
PTCON1	PTEN	PTDIR	—	—	—	—	—	—	58
PTMRL(1)	PWM Time Base Register (lower 8 bits)								58
PTMRH(1)	**UNUSED**				PWM Time Base Register (upper 4 bits)				58
PTPERL(1)	PWM Time Base Period Register (lower 8 bits)								58
PTPERH(1)	**UNUSED**				PWM Time Base Period Register (upper 4 bits)				58
SEVTCMPL(1)	PWM Special Event Compare Register (lower 8 bits)								58
SEVTCMPH(1)	**UNUSED**				PWM Special Event Compare Register (upper 4 bits)				58
PWMCON0	—	PWMEN2	PWMEN1	PWMEN0	PMOD3(2)	PMOD2	PMOD1	PMOD0	58
PWMCON1	SEVOPS3	SEVOPS2	SEVOPS1	SEVOPS0	SEVTDIR	—	UDIS	OSYNC	58
DTCON	DTPS1	DTPS0	DT5	DT4	DT3	DT2	DT1	DT0	58
FLTCONFIG	BRFEN	FLTBS(2)	FLTBMOD(2)	FLTBEN(2)	FLTCON	FLTAS	FLTAMOD	FLTAEN	58
OVDCOND	POVD7(2)	POVD6(2)	POVD5	POVD4	POVD3	POVD2	POVD1	POVD0	58
OVDCONS	POUT7(2)	POUT6(2)	POUT5	POUT4	POUT3	POUT2	POUT1	POUT0	58
PDC0L(1)	PWM Duty Cycle #0L Register (lower 8 bits)								58
PDC0H(1)	**UNUSED**		PWM Duty Cycle #0H Register (upper 6 bits)						58
PDC1L(1)	PWM Duty Cycle #1L register (lower 8 bits)								58
PDC1H(1)	**UNUSED**		PWM Duty Cycle #1H Register (upper 6 bits)						58
PDC2L(1)	PWM Duty Cycle #2L Register (lower 8 bits)								58
PDC2H(1)	**UNUSED**		PWM Duty Cycle #2H Register (upper 6 bits)						58
PDC3L(1,2)	PWM Duty Cycle #3L Register (lower 8 bits)								58
PDC3H(1,2)	**UNUSED**		PWM Duty Cycle #3H Register (upper 6 bits)						58

Legend: — = Unimplemented, read as '0'. Shaded cells are not used with the power control PWM.

Note 1: Double-buffered register pairs. Refer to text for explanation of how these registers are read and written to.

2: Unimplemented in PIC18F2331/2431 devices; maintain these bits clear. Reset values shown are for PIC18F4331/4431 devices.

REGISTER 19-2: SSPCON: SYNCHRONOUS SERIAL PORT CONTROL REGISTER (CONTINUED)

bit 3-0 **SSPM<3:0>**: Synchronous Serial Port Mode Select bits(3)

0000 = SPI Master mode, Clock = FOSC/4
0001 = SPI Master mode, Clock = FOSC/16
0010 = SPI Master mode, Clock = FOSC/64
0011 = SPI Master mode, Clock = TMR2 output/2
0100 = SPI Slave mode, Clock = SCK pin, $\overline{SS}$ pin control enabled
0101 = SPI Slave mode, Clock = SCK pin, $\overline{SS}$ pin control disabled, $\overline{SS}$ can be used as I/O pin
0110 = I^2C Slave mode, 7-bit address
0111 = I^2C Slave mode, 10-bit address
1011 = I^2C Firmware Controlled Master mode (slave Idle)
1110 = I^2C Slave mode, 7-bit address with Start and Stop bit interrupts enabled
1111 = I^2C Slave mode, 10-bit address with Start and Stop bit interrupts enabled

Note 1: In Master mode, the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPBUF register.

2: When enabled, these pins must be properly configured as inputs or outputs.

3: Bit combinations not specifically listed here are either reserved or implemented in I^2C™ mode only.

© 2010 Microchip Technology Inc.

20.0 ENHANCED UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (EUSART)

The Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART) module is one of the two serial I/O modules available in the PIC18F2331/2431/4331/4431 family of microcontrollers. EUSART is also known as a Serial Communications Interface or SCI.

The EUSART can be configured as a full-duplex asynchronous system that can communicate with peripheral devices, such as CRT terminals and personal computers. It can also be configured as a half-duplex synchronous system that can communicate with peripheral devices, such as A/D or D/A integrated circuits, serial EEPROMs, etc.

The EUSART module implements additional features, including automatic baud rate detection and calibration, automatic wake-up on Sync Break reception and 12-bit Break character transmit. These features make it ideally suited for use in Local Interconnect Network (LIN/J2602) bus systems.

The EUSART can be configured in the following modes:

- Asynchronous (full-duplex) with:
  - Auto-wake-up on character reception
  - Auto-baud calibration
  - 12-bit Break character transmission
- Synchronous – Master (half-duplex) with selectable clock polarity
- Synchronous – Slave (half-duplex) with selectable clock polarity

In order to configure pins, TX and RX, as the Enhanced Universal Synchronous Asynchronous Receiver Transmitter:

- SPEN (RCSTA<7>) bit must be set (= 1),
- TRISC<6> bit must be set (= 1), and
- TRISC<7> bit must be set (= 1).

Note:	The EUSART control will automatically reconfigure the pin from input to output as needed.

The operation of the Enhanced USART module is controlled through three registers:

- Transmit Status and Control (TXSTA)
- Receive Status and Control (RCSTA)
- Baud Rate Control (BAUDCON)

These are detailed on the following pages in Register 20-1, Register 20-2 and Register 20-3, respectively.

20.1 Asynchronous Operation in Power-Managed Modes

The EUSART may operate in Asynchronous mode while the peripheral clocks are being provided by the internal oscillator block. This makes it possible to remove the crystal or resonator that is commonly connected as the primary clock on the OSC1 and OSC2 pins.

The factory calibrates the internal oscillator block output (INTOSC) for 8 MHz (see Table 26-6). However, this frequency may drift as V_{DD} or temperature changes, and this directly affects the asynchronous baud rate. Two methods may be used to adjust the baud rate clock, but both require a reference clock source of some kind.

The first (preferred) method uses the OSCTUNE register to adjust the INTOSC output back to 8 MHz. Adjusting the value in the OSCTUNE register allows for fine resolution changes to the system clock source (see **Section 3.6.4 “INTOSC Frequency Drift”** for more information).

The other method adjusts the value in the Baud Rate Generator (BRG). There may not be fine enough resolution when adjusting the Baud Rate Generator to compensate for a gradual change in the peripheral clock frequency.

21.1 Configuring the A/D Converter

The A/D Converter has two types of conversions, two modes of operation and eight different Sequencing modes. These features are controlled by the ACONV bit (ADCON0<5>), ACSCH bit (ADCON0<4>) and ACMOD<1:0> bits (ADCON0<3:2>). In addition, the A/D channels are divided into four groups as defined in the ADCHS register. Table 21-1 shows the sequence configurations as controlled by the ACSCH and ACMOD<1:0> bits.

21.1.1 CONVERSION TYPE

Two types of conversions exist in the high-speed 10-bit A/D Converter module that are selected using the ACONV bit. Single-Shot mode allows a single conversion or sequence to be enabled when ACONV = 0. At the end of the sequence, the GO/DONE bit will be automatically cleared and the interrupt flag, ADIF, will be set. When using Single-Shot mode and configured for Simultaneous mode, STNM2, acquisition time must be used to ensure proper conversion of the analog input signals.

Continuous Loop mode allows the defined sequence to be executed in a continuous loop when ACONV = 1. In this mode, either the user can trigger the start of conversion by setting the GO/DONE bit, or one of the A/D triggers can start the conversion. The interrupt flag, ADIF, is set based on the configuration of the bits, ADRS<1:0> (ADCON3<7:6>). In Simultaneous modes, STNM1 and STNM2 acquisition time must be configured to ensure proper conversion of the analog input signals.

21.1.2 CONVERSION MODE

The ACSCH bit (ADCON0<4>) controls how many channels are used in the configured sequence. When clear, the A/D is configured for single channel conversion and will convert the group selected by the ACMOD<1:0> bits and the channel selected by the GxSEL<1:0> bits (ADCHS register). When ACSCH = 1, the A/D is configured for multiple channel conversion and the sequence is defined by ACMOD<1:0>.

TABLE 21-1: AUTO-CONVERSION SEQUENCE CONFIGURATIONS

Mode	ACSCH	ACMOD<1:0>	Description
Multi-Channel Sequential Mode 1 (SEQM1)	1	00	Groups A and B are sampled and converted sequentially.
Multi-Channel Sequential Mode 2 (SEQM2)	1	01	Groups A, B, C and D are sampled and converted sequentially.
Multi-Channel Simultaneous Mode 1 (STNM1)	1	10	Groups A and B are sampled simultaneously and converted sequentially.
Multi-Channel Simultaneous Mode 2 (STNM2)	1	11	Groups A and B are sampled simultaneously, then converted sequentially. Then, Group C and D are sampled simultaneously, then converted sequentially.
Single Channel Mode 1 (SCM1)	0	00	Group A is sampled and converted.
Single Channel Mode 2 (SCM2)	0	01	Group B is sampled and converted.
Single Channel Mode 3 (SCM3)	0	10	Group C is sampled and converted.
Single Channel Mode 4 (SCM4)	0	11	Group D is sampled and converted.

24.0 INSTRUCTION SET SUMMARY

The PIC18 instruction set adds many enhancements to the previous PIC® instruction sets, while maintaining an easy migration from these PIC instruction sets.

Most instructions are a single program memory word (16 bits), but there are three instructions that require two program memory locations.

Each single-word instruction is a 16-bit word divided into an opcode, which specifies the instruction type and one or more operands, which further specify the operation of the instruction.

The instruction set is highly orthogonal and is grouped into four basic categories:

- **Byte-oriented** operations
- **Bit-oriented** operations
- **Literal** operations
- **Control** operations

The PIC18 instruction set summary in Table 24-2 lists **byte-oriented**, **bit-oriented**, **literal** and **control** operations. Table 24-1 shows the opcode field descriptions.

Most **byte-oriented** instructions have three operands:

1. The file register (specified by 'f')
2. The destination of the result (specified by 'd')
3. The accessed memory (specified by 'a')

The file register designator, 'f', specifies which file register is to be used by the instruction.

The destination designator, 'd', specifies where the result of the operation is to be placed. If 'd' is '0', the result is placed in the WREG register. If 'd' is '1', the result is placed in the file register specified in the instruction.

All **bit-oriented** instructions have three operands:

1. The file register (specified by 'f')
2. The bit in the file register (specified by 'b')
3. The accessed memory (specified by 'a')

The bit field designator, 'b', selects the number of the bit affected by the operation, while the file register designator, 'f', represents the number of the file in which the bit is located.

The **literal** instructions may use some of the following operands:

- A literal value to be loaded into a file register (specified by 'k')
- The desired FSR register to load the literal value into (specified by 'f')
- No operand required (specified by '—')

The **control** instructions may use some of the following operands:

- A program memory address (specified by 'n')
- The mode of the call or return instructions (specified by 's')
- The mode of the table read and table write instructions (specified by 'm')
- No operand required (specified by '—')

All instructions are a single word, except for three double-word instructions. These three instructions were made double word instructions so that all the required information is available in these 32 bits. In the second word, the 4 MSbs are '1's. If this second word is executed as an instruction (by itself), it will execute as a NOP.

All single-word instructions are executed in a single instruction cycle, unless a conditional test is true or the program counter is changed as a result of the instruction. In these cases, the execution takes two instruction cycles with the additional instruction cycle(s) executed as a NOP.

The double word instructions execute in two instruction cycles.

One instruction cycle consists of four oscillator periods. Thus, for an oscillator frequency of 4 MHz, the normal instruction execution time is 1 μs. If a conditional test is true or the program counter is changed as a result of an instruction, the instruction execution time is 2 μs. Two-word branch instructions (if true) would take 3 μs.

Figure 24-1 shows the general formats that the instructions can have.

All examples use the format 'nnh' to represent a hexadecimal number, where 'h' signifies a hexadecimal digit.

The Instruction Set Summary, shown in Table 24-2, lists the instructions recognized by the Microchip Assembler (MPASM™ Assembler). **Section 24.2 "Instruction Set"** provides a description of each instruction.

24.1 Read-Modify-Write Operations

Any instruction that specifies a file register as part of the instruction performs a Read-Modify-Write (R-M-W) operation. The register is read, the data is modified, and the result is stored according to either the instruction or the destination designator, 'd'. A read operation is performed on a register even if the instruction writes to that register.

For example, a "BCF PORTB, 1" instruction will read PORTB, clear bit 1 of the data, then write the result back to PORTB. The read operation would have the unintended result that any condition that sets the RBIF flag would be cleared. The R-M-W operation may also copy the level of an input pin to its corresponding output latch.

BRA **Unconditional Branch**

Syntax: [*label*] 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: [*label*] 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', Access Bank will be selected, overriding the BSR value. If 'a' = 1, then the bank will be selected as per the BSR value.

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

Before Instruction
FLAG_REG = 0x0A
After Instruction
FLAG_REG = 0x8A

FIGURE 26-9: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS

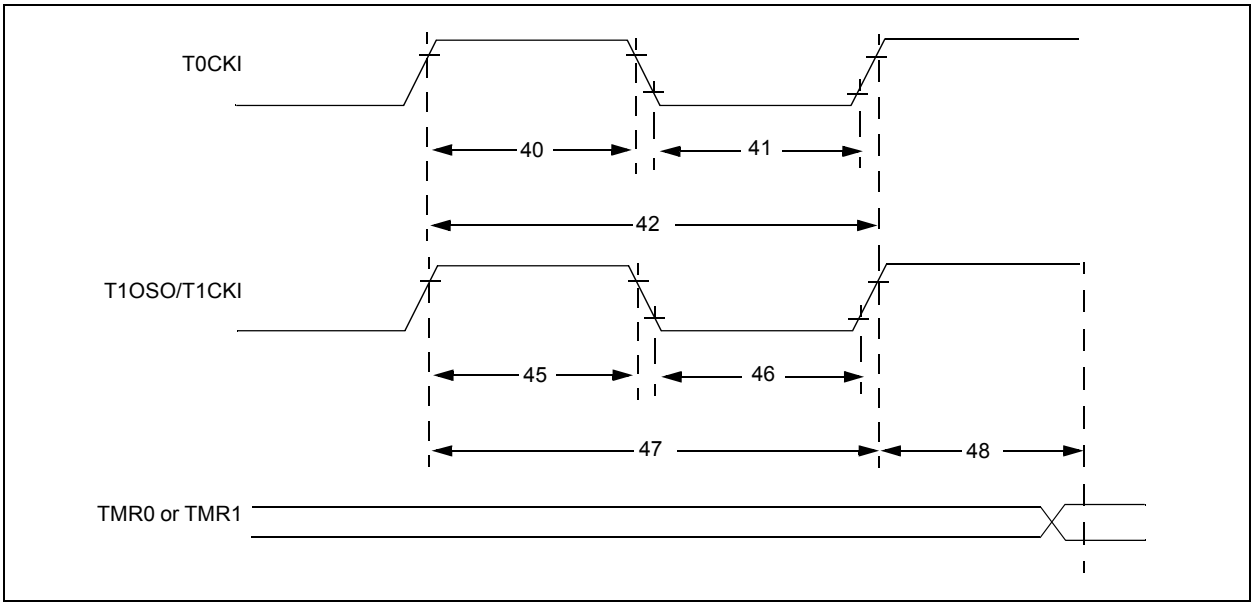


TABLE 26-9: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS

Param No.	Symbol	Characteristic			Min	Max	Units	Conditions
40	Tt0H	T0CKI High Pulse Width		No prescaler	0.5 TCY + 20	—	ns	VDD = 2V
				With prescaler	10	—	ns	
41	Tt0L	T0CKI Low Pulse Width		No prescaler	0.5 TCY + 20	—	ns	
				With prescaler	10	—	ns	
42	Tt0P	T0CKI Period		No prescaler	TCY + 10	—	ns	
				With prescaler	Greater of: 20 ns or $\frac{T_{CY} + 40}{N}$	—	ns	N = prescale value (1, 2, 4,..., 256)
45	Tt1H	T1CKI High Time	Synchronous, no prescaler		0.5 TCY + 20	—	ns	
			Synchronous, with prescaler	PIC18FXX31	10	—	ns	
				PIC18LFXX31	25	—	ns	
			Asynchronous	PIC18FXX31	30	—	ns	
				PIC18LFXX31	50	—	ns	
46	Tt1L	T1CKI Low Time	Synchronous, no prescaler		0.5 TCY + 5	—	ns	
			Synchronous, with prescaler	PIC18FXX31	10	—	ns	
				PIC18LFXX31	25	—	ns	
			Asynchronous	PIC18FXX31	30	—	ns	
				PIC18LFXX31	50	—	ns	
47	Tt1P	T1CKI Input Period	Synchronous		Greater of: 20 ns or $\frac{T_{CY} + 40}{N}$	—	ns	N = prescale value (1, 2, 4, 8)
			Asynchronous		60	—	ns	
	Ft1	T1CKI Oscillator Input Frequency Range			DC	50	kHz	
48	Tcke2tmrl	Delay from External T1CKI Clock Edge to Timer Increment			2 TOSC	7 TOSC	—	

FIGURE 26-17: EUSART SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING

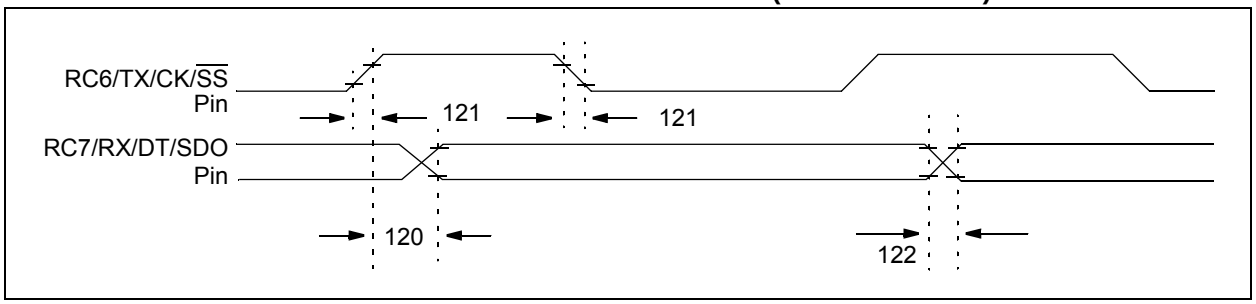


TABLE 26-18: EUSART SYNCHRONOUS TRANSMISSION REQUIREMENTS

Param No.	Symbol	Characteristic		Min	Max	Units	Conditions
120	TckH2dtV	SYNC XMIT (MASTER & SLAVE) Clock High to Data Out Valid	PIC18FXX31	—	40	ns	
			PIC18LFXX31	—	100	ns	
121	Tckrf	Clock Out Rise Time and Fall Time (Master mode)	PIC18FXX31	—	20	ns	
			PIC18LFXX31	—	50	ns	
122	Tdtrf	Data Out Rise Time and Fall Time	PIC18FXX31	—	20	ns	
			PIC18LFXX31	—	50	ns	

FIGURE 26-18: EUSART SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING

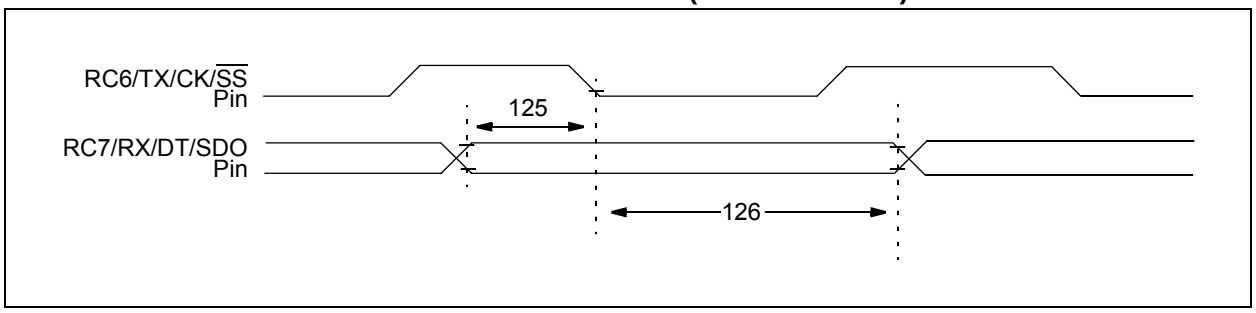


TABLE 26-19: EUSART SYNCHRONOUS RECEIVE REQUIREMENTS

Param. No.	Symbol	Characteristic	Min	Max	Units	Conditions
125	TdtV2ckl	SYNC RCV (MASTER & SLAVE) Data Hold before CK ↓ (DT hold time)	10	—	ns	
126	TckL2dtl	Data Hold after CK ↓ (DT hold time)	15	—	ns	

© 2010 Microchip Technology Inc.

NOTES:

© 2010 Microchip Technology Inc.