Section 12. Timers

HIGHLIGHTS

This section of the manual contains the following major topics:

12.1 Introduction

Depending on the specific variant, the dsPIC30F device family offers several 16-bit timers. These timers are designated as Timer1, Timer2, Timer3, ..., etc.

Each timer module is a 16-bit timer/counter consisting of the following readable/writable registers:

- TMRx: 16-bit timer count register
- PRx: 16-bit period register associated with the timer
- TxCON: 16-bit control register associated with the timer

Each timer module also has the associated bits for interrupt control:

- Interrupt Enable Control bit (TxIE)
- Interrupt Flag Status bit (TxIF)
- Interrupt Priority Control bits (TxIP<2:0>)

With certain exceptions, all of the 16-bit timers have the same functional circuitry. The 16-bit timers are classified into three types to account for their functional differences:

- Type A time base
- Type B time base
- Type C time base

Some 16-bit timers can be combined to form a 32-bit timer.

This section does not describe the dedicated timers that are associated with peripheral devices. For example, this includes the time bases associated with the Motor Control PWM module and the Quadrature Encoder Interface (QEI) module.

12.2 Timer Variants

All 16-bit timers available on the dsPIC30F devices are functionally identical with certain exceptions. The 16-bit timers are classified into three functional types; Type A timers, Type B timers and Type C timers.

Note: Please refer to the device data sheet for the available timers and the type of each.

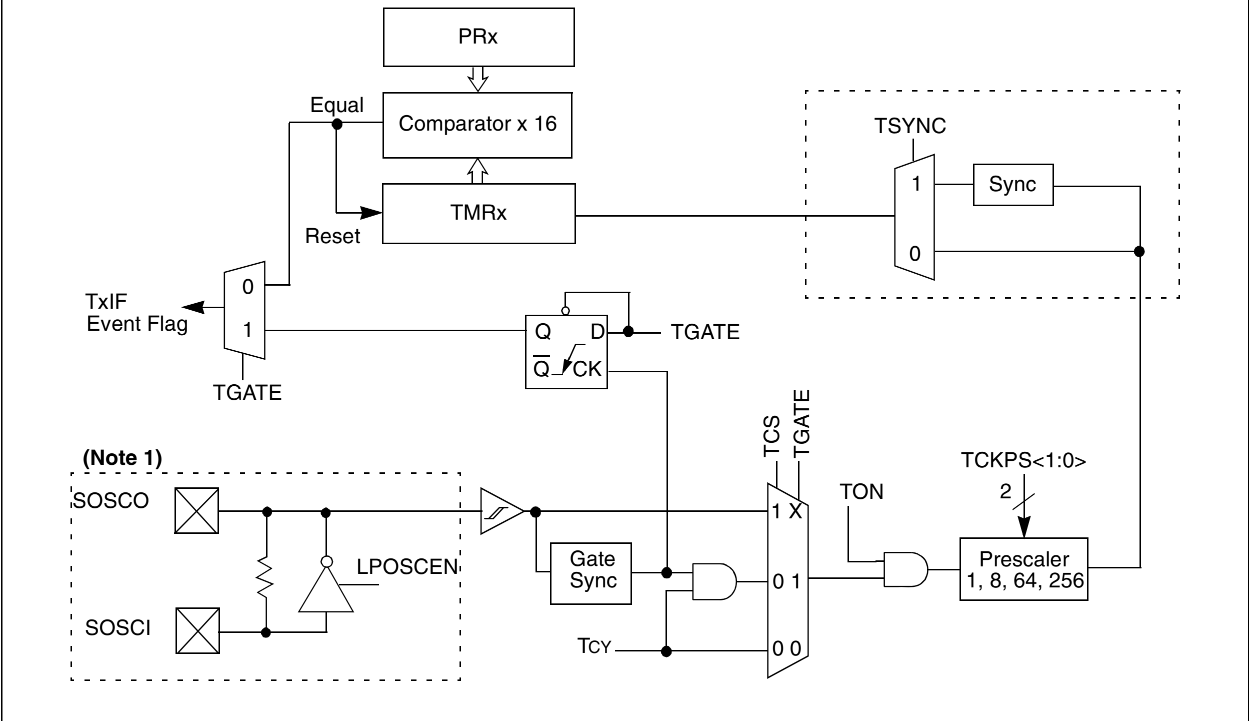
12.2.1 Type A Timer

At least one Type A timer is available on most dsPIC30F devices. For most dsPIC30F devices, Timer1 is a Type A timer. A Type A timer has the following unique features over other types:

- can be operated from the device Low Power 32 kHz Oscillator
- can be operated in an Asynchronous mode from an external clock source

In particular, the unique features of a Type A timer allow it to be used for Real-Time Clock (RTC) applications. A block diagram of the Type A timer is shown in Figure 12-1.

Figure 12-1: Type A Timer Block Diagram



Note 1: Refer to **Section 7. "Oscillator"** for information on enabling the LP Oscillator.

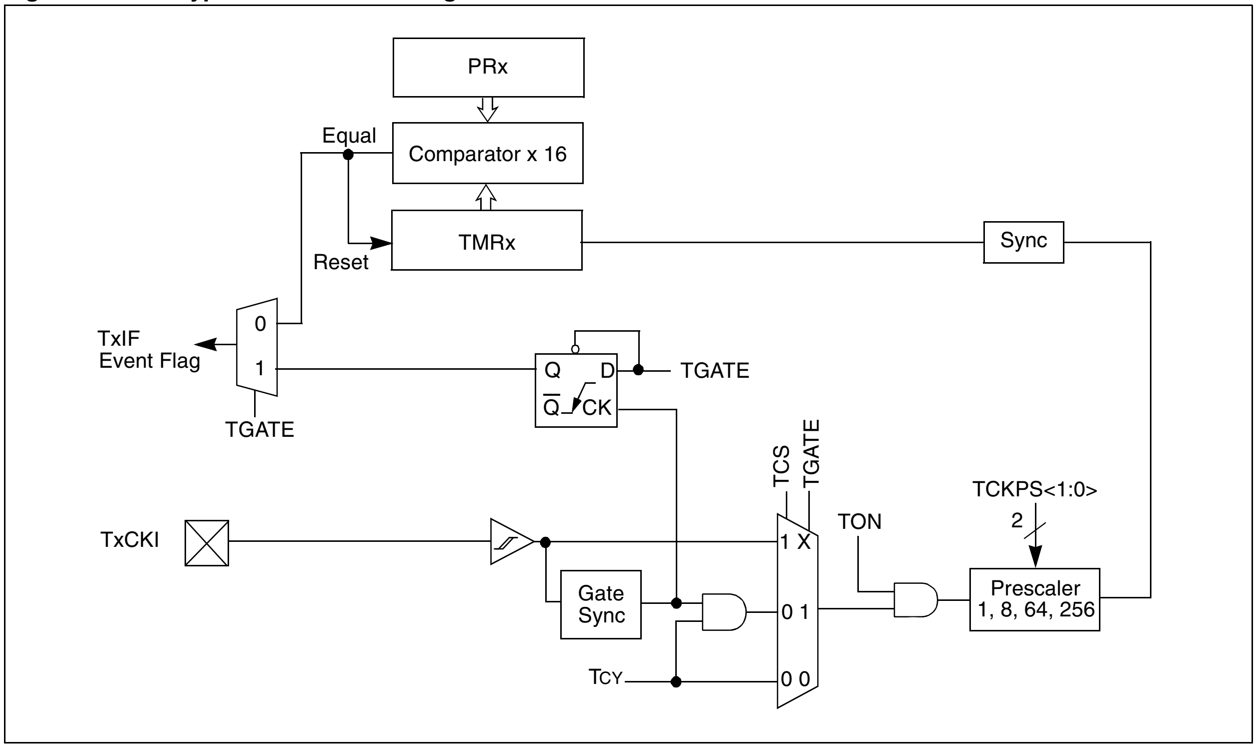
12.2.2 Type B Timer

Timer2 and Timer4, if present, are Type B timers on most dsPIC30F devices. A Type B timer has the following unique features over other types of timers:

- A Type B timer can be concatenated with a Type C timer to form a 32-bit timer. The TxCON register for a Type B timer has the T32 control bits to enable the 32-bit timer function.
- The clock synchronization for a Type B timer is performed after the prescale logic.

A block diagram of the Type B timer is shown in Figure 12-2.

Figure 12-2: Type B Timer Block Diagram



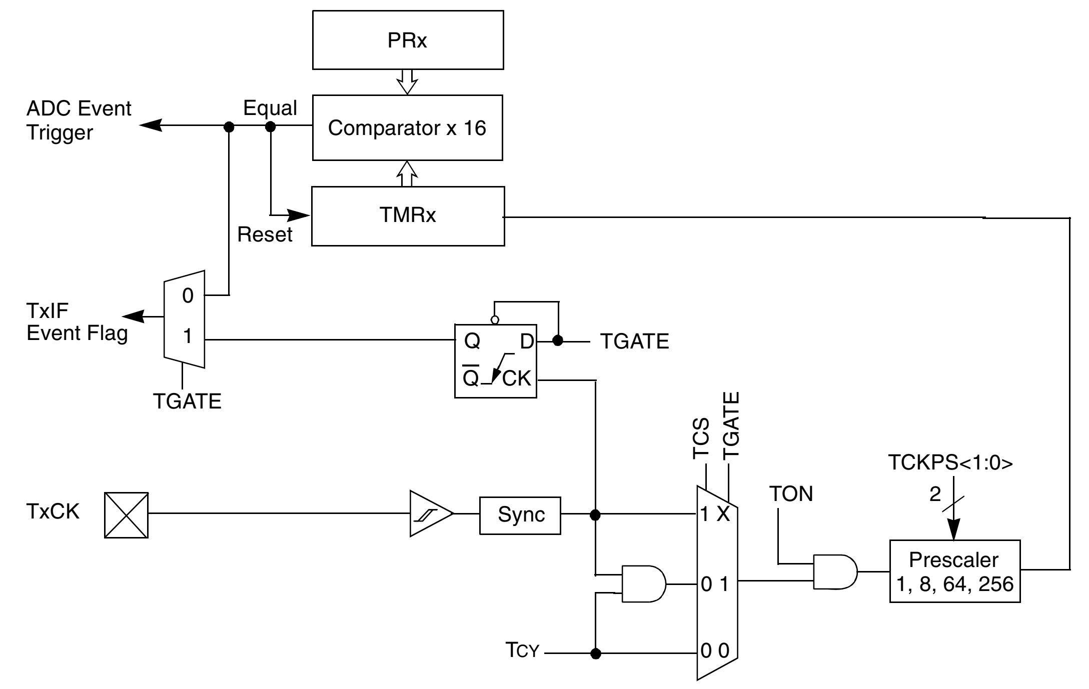
12.2.3 Type C Timer

Timer3 and Timer5 are Type C timers on most dsPIC30F devices. A Type C timer has the following unique features over other types of timers:

- A Type C timer can be concatenated with a Type B timer to form a 32-bit timer.
- On a given device, at least one Type C timer has the ability to trigger an A/D conversion.

A block diagram of the Type C timer is shown in Figure 12-3.

Figure 12-3: Type C Timer Block Diagram



Note: In certain variants of the dsPIC30F family, the TxCK pin may not be available. Refer to the device data sheet for the I/O pin details. In such cases, the timer must use the system clock (FOSC/4) as its input clock, unless it is configured for 32-bit operation.

12.3 Control Registers

Register 12-1: TxCON: Type A Time Base Register

Upper Byte:							
R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
TON	—	TSIDL	—	—	—	—	—
bit 15							bit 8

Lower Byte:							
U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	U-0
—	TGATE	TCKPS<1:0>		—	TSYNC	TCS	—
bit 7							bit 0

bit 15	**TON:** Timer On Control bit 1 = Starts the timer 0 = Stops the timer
bit 14	**Unimplemented:** Read as '0'
bit 13	**TSIDL:** Stop in Idle Mode bit 1 = Discontinue timer operation when device enters Idle mode 0 = Continue timer operation in Idle mode
bit 12-7	**Unimplemented:** Read as '0'
bit 6	**TGATE:** Timer Gated Time Accumulation Enable bit 1 = Gated time accumulation enabled 0 = Gated time accumulation disabled (TCS must be set to '0' when TGATE = 1. Reads as '0' if TCS = 1)
bit 5-4	**TCKPS<1:0>:** Timer Input Clock Prescale Select bits 11 = 1:256 prescale value 10 = 1:64 prescale value 01 = 1:8 prescale value 00 = 1:1 prescale value
bit 3	**Unimplemented:** Read as '0'
bit 2	**TSYNC:** Timer External Clock Input Synchronization Select bit When TCS = 1: 1 = Synchronize external clock input 0 = Do not synchronize external clock input When TCS = 0: This bit is ignored. Read as '0'. Timer1 uses the internal clock when TCS = 0.
bit 1	**TCS:** Timer Clock Source Select bit 1 = External clock from pin TxCK 0 = Internal clock (FOSC/4)
bit 0	**Unimplemented:** Read as '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

Register 12-2: TxCON: Type B Time Base Register

Upper Byte:

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
TON	—	TSIDL	—	—	—	—	—
bit 15							bit 8

Lower Byte:

U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	U-0
—	TGATE	TCKPS<1:0>		T32	—	TCS	—
bit 7							bit 0

bit 15 **TON:** Timer On bit
<u>When T32 = 1 (in 32-bit Timer mode):</u>
1 = Starts 32-bit TMRx:TMRy timer pair
0 = Stops 32-bit TMRx:TMRy timer pair
<u>When T32 = 0 (in 16-bit Timer mode):</u>
1 = Starts 16-bit timer
0 = Stops 16-bit timer

bit 14 **Unimplemented:** Read as '0'

bit 13 **TSIDL:** Stop in Idle Mode bit
1 = Discontinue timer operation when device enters Idle mode
0 = Continue timer operation in Idle mode

bit 12-7 **Unimplemented:** Read as '0'

bit 6 **TGATE:** Timer Gated Time Accumulation Enable bit
1 = Timer gated time accumulation enabled
0 = Timer gated time accumulation disabled
(TCS must be set to logic '0' when TGATE = 1)

bit 5-4 **TCKPS<1:0>:** Timer Input Clock Prescale Select bits
11 = 1:256 prescale value
10 = 1:64 prescale value
01 = 1:8 prescale value
00 = 1:1 prescale value

bit 3 **T32:** 32-bit Timer Mode Select bits
1 = TMRx and TMRy form a 32-bit timer
0 = TMRx and TMRy form separate 16-bit timer

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

bit 1 **TCS:** Timer Clock Source Select bit
1 = External clock from pin TxCK
0 = Internal clock (FOSC/4)

bit 0 **Unimplemented:** Read as '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

Register 12-3: TxCON: Type C Time Base Register

Upper Byte:							
R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
TON	—	TSIDL	—	—	—	—	—
bit 15							bit 8

Lower Byte:							
U-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	U-0
—	TGATE	TCKPS<1:0>		—	—	TCS	—
bit 7							bit 0

bit 15 **TON:** Timer On bit
1 = Starts 16-bit TMRx
0 = Stops 16-bit TMRx

bit 14 **Unimplemented:** Read as ‘0’

bit 13 **TSIDL:** Stop in Idle Mode bit
1 = Discontinue module operation when device enters Idle mode
0 = Continue module operation in Idle mode

bit 12-7 **Unimplemented:** Read as ‘0’

bit 6 **TGATE:** Timer Gated Time Accumulation Enable bit
1 = Timer gated time accumulation enabled
0 = Timer gated time accumulation disabled (Read as ‘0’ if TCS = 1)
(TCS must be set to logic ‘0’ when TGATE = 1)

bit 5-4 **TCKPS<1:0>:** Timer Input Clock Prescale Select bits
11 = 1:256 prescale value
10 = 1:64 prescale value
01 = 1:8 prescale value
00 = 1:1 prescale value

bit 3-2 **Unimplemented:** Read as ‘0’

bit 1 **TCS:** Timer Clock Source Select bit
1 = External clock from pin TxCK
0 = Internal clock (FOSC/4)

bit 0 **Unimplemented:** Read as ‘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

12.4 Modes of Operation

Each timer module can operate in one of the following modes:

- As a synchronous timer
- As a synchronous counter
- As a gated timer
- As an asynchronous counter (Type A time base only)

The Timer modes are determined by the following bits:

- TCS (TxCON<1>): Timer Clock Source Control bit
- TSYNC (T1CON<2>): Timer Synchronization Control bit (Type A time base only)
- TGATE (TxCON<6>): Timer Gate Control bit

Each timer module is enabled or disabled using the TON Control bit (TxCON <15>).

Note: Only Type A time bases support the External Asynchronous Clock mode.

12.4.1 Timer Mode

All types of timers have the ability to operate in Timer mode. In Timer mode, the input clock to the timer is provided from the internal system clock (FOSC/4). When enabled, the timer increments once per instruction cycle for a 1:1 prescaler setting. The Timer mode is selected by clearing the TCS control bit (TxCON<1>). The Synchronous mode control bit, TSYNC (T1CON<2>), has no effect, since the system clock source is used to generate the timer clock.

Example 12-1: Initialization Code for 16-bit Timer Using System Clock

```
;   The following code example will enable Timer1 interrupts,
;   load the Timer1 Period register and start Timer1.

;   When a Timer1 period match interrupt occurs, the interrupt
;   service routine must clear the Timer1 interrupt status flag
;   in software.

    CLR   T1CON               ; Stops the Timer1 and reset control reg.
    CLR   TMR1                ; Clear contents of the timer register
    MOV   #0xFFFF, w0         ; Load the Period register
    MOV   w0, PR1             ; with the value 0xFFFF

    BSET  IPC0, #T1IP0        ; Setup Timer1 interrupt for
    BCLR  IPC0, #T1IP1        ; desired priority level
    BCLR  IPC0, #T1IP2        ; (this example assigns level 1 priority)
    BCLR  IFS0, #T1IF         ; Clear the Timer1 interrupt status flag
    BSET  IEC0, #T1IE         ; Enable Timer1 interrupts
    BSET  T1CON, #TON         ; Start Timer1 with prescaler settings
                              ; at 1:1 and clock source set to
                              ; the internal instruction cycle

; Example code for Timer1 ISR

__T1Interrupt:
    BCLR  IFS0, #T1IF         ; Reset Timer1 interrupt flag
                              ; User code goes here.
    RETFIE                    ; Return from ISR
```

12.4.2 Synchronous Counter Mode Using External Clock Input

When the TCS control bit (TxCON<1>) is set, the clock source for the timer is provided externally and the selected timer increments on every rising edge of clock input on the TxCK pin.

External clock synchronization must be enabled for a Type A time base. This is accomplished by setting the TSYNC control bit (TxCON<2>). For Type B and Type C time bases, the external clock input is always synchronized to the system instruction cycle clock, TCY.

When the timer is operated in the Synchronized Counter mode, there are minimum requirements for the external clock high time and low time. The synchronization of the external clock source with the device instruction clock is accomplished by sampling the external clock signal at two different times within an instruction cycle.

A timer operating from a synchronized external clock source will not operate in Sleep mode, since the synchronization circuit is shut-off during Sleep mode.

Note: The external input clock must meet certain minimum high time and low time requirements when Timerx is used in the Synchronous Counter mode. Refer to the device data sheet “Electrical Specifications” section for further details.

Example 12-2: Initialization Code for 16-bit Synchronous Counter Mode Using an External Clock Input

```
;   The following code example will enable Timer1 interrupts, load the
;   Timer1 Period register and start Timer1 using an external clock
;   and a 1:8 prescaler setting.

;   When a Timer1 period match interrupt occurs, the interrupt service
;   routine must clear the Timer1 interrupt status flag in software.

    CLR     T1CON               ; Stops the Timer1 and reset control reg.
    CLR     TMR1                ; Clear contents of the timer register
    MOV     #0x8CFF, w0         ; Load the Period register
    MOV     w0, PR1             ; with the value 0x8CFF

    BSET    IPC0,   #T1IP0      ; Setup Timer1 interrupt for
    BCLR    IPC0,   #T1IP1      ; desired priority level
    BCLR    IPC0,   #T1IP2      ; (this example assigns level 1 priority)
    BCLR    IFS0,   #T1IF       ; Clear the Timer1 interrupt status flag
    BSET    IEC0,   #T1IE       ; Enable Timer1 interrupts
    MOV     #0x8016, W0         ; Start Timer1 with prescaler settings at
                                ; 1:8 and clock source set to the external
    MOV     w0, T1CON           ; clock in the synchronous mode


;   Example code for Timer1 ISR

__T1Interrupt:
    BCLR    IFS0,   #T1IF       ; Reset Timer1 interrupt flag
                                ; User code goes here.
    RETFIE                      ; Return from ISR
```

12.4.3 Type A Timer Asynchronous Counter Mode Using External Clock Input

A Type A time base has the ability to operate in an Asynchronous Counting mode, using an external clock source connected to the TxCK pin. When the TSYNC control bit (TxCON<2>) is cleared, the external clock input is not synchronized with the device system clock source. The time base continues to increment asynchronously to the internal device clock.

The asynchronous operation time base is beneficial for the following applications:

- The time base can operate during Sleep mode and can generate an interrupt on period register match that will wake-up the processor.
- The time base can be clocked from the low power 32 kHz oscillator for real-time clock applications

Please see **Section 12.12.1 “Timer Operation in Sleep Mode”** for more details..

Note 1: Only Type A time bases support the Asynchronous Counter mode.
2: The external input clock must meet certain minimum high time and low time requirements when Timerx is used in the Asynchronous Counter mode. Refer to the device data sheet “Electrical Specifications” section for further details.
3: Unexpected results may occur when reading Timer1, in asynchronous mode.

Example 12-3: Initialization Code for 16-bit Asynchronous Counter Mode Using an External Clock Input

```
;   The following code example will enable Timer1 interrupts, load the
;   Timer1 Period register and start Timer1 using an asynchronous
;   external clock and a 1:8 prescaler setting.

;   When a Timer1 period match interrupt occurs, the interrupt service
;   routine must clear the Timer1 interrupt status flag in software.

    CLR     T1CON               ; Stops the Timer1 and reset control reg.
    CLR     TMR1                ; Clear contents of the timer register
    MOV     #0x7FFF, w0         ; Load the Period register
    MOV     w0, PR1             ; with the value 0x7FFF

    BSET    IPC0,    #T1IP0     ; Setup Timer1 interrupt for
    BCLR    IPC0,    #T1IP1     ; desired priority level
    BCLR    IPC0,    #T1IP2     ; (this example assigns level 1 priority)
    BCLR    IFS0,    #T1IF      ; Clear the Timer1 interrupt status flag
    BSET    IEC0,    #T1IE      ; Enable Timer1 interrupts
    MOV     #0x8012, w0         ; Start Timer1 with prescaler settings at
                                ; 1:8 and clock source set to the external
    MOV     w0, T1CON           ; clock in the asynchronous mode


; Example code for Timer1 ISR

__T1Interrupt:
    BCLR    IFS0,    #T1IF      ; Reset Timer1 interrupt flag
                                ; User code goes here.
    RETFIE                      ; Return from ISR
```

12.4.4 Timer Operation with Fast External Clock Source

In some applications, it may be desirable to use one of the timers to count clock edges from a relatively high frequency external clock source. In these situations, Type A and Type B time bases are the most suitable choices for counting the external clock source, because the clock synchronization logic for these timers is located after the timer prescaler (see Figure 12-1 and Figure 12-2). This allows a higher external clock frequency to be used that will not violate the minimum high and low times required by the prescaler. When a timer prescaler ratio other than 1:1 is selected for a Type A or Type B time base, the minimum high and low times for the external clock input are reduced by the chosen prescaler ratio.

A Type A time base is unique because it can be operated in an Asynchronous Clock mode, eliminating any prescaler timing requirements.

Note that in all cases, there are minimum high and low times for the external clock signal that cannot be exceeded. These minimum times are required to satisfy the I/O pin timing requirements.

Please refer to the device data sheet for the external clock timing specifications associated with the time bases.

12.4.5 Gated Time Accumulation Mode

The Gated Time Accumulation mode allows the internal timer register to increment based upon the duration of the high time applied to the TxCK pin. In the Gated Time Accumulation mode, the timer clock source is derived from the internal system clock. When the TxCK pin state is high, the timer register will count up until a period match has occurred, or the TxCK pin state is changed to a low state. A pin state transition from high to low will set the TxIF interrupt flag. Depending on when the edge occurs, the interrupt flag is asserted 1 or 2 instruction cycles after the falling edge of the signal on the TxCK pin.

The TGATE control bit (TxCON<6>) must be set to enable the Gated Time Accumulation mode. The timer must be enabled, TON (TxCON<15>) = `1`, and the timer clock source set to the internal clock, TCS (TxCON<1>) = `0`.

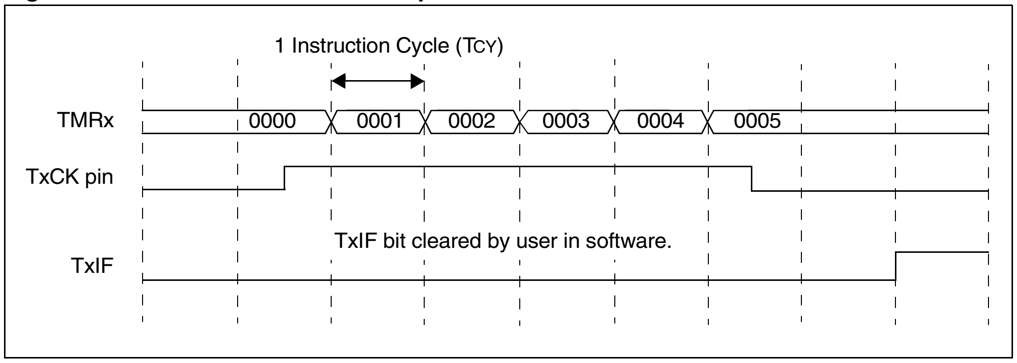
The gate operation starts on a rising edge of the signal applied to the TxCK pin and terminates on the falling edge of the signal applied to the TxCK pin. The respective timer will increment while the external gate signal is high.

The falling edge of the gate signal terminates the count operation, but does not reset the timer. The user must reset the timer if it is desired to start from zero on the next rising edge gate input. The falling edge of the gate signal generates an interrupt.

Note: The timer will not interrupt the CPU when a timer period match occurs in Gate Time Accumulation mode.

The resolution of the timer count is directly related to the timer clock period. For a timer prescaler of 1:1, the timer clock period is one instruction cycle. For a timer prescaler of 1:256, the timer clock period is 256 times the instruction cycle. The timer clock resolution can be associated to the pulse width of the gate signal. Refer to the “Electrical Specifications” section in the device data sheet for further details on the gate width pulse requirements.

Figure 12-4: Gated Timer Mode Operation


1 Instruction Cycle (TCY)
TMRx
0000
0001
0002
0003
0004
0005
TxCK pin
TxIF
TxIF bit cleared by user in software.

Example 12-4: Initialization Code for 16-bit Gated Time Accumulation Mode

```
;   The following code example will enable Timer2 interrupts, load the
;   Timer2 Period register and start Timer2 using an internal clock
;   and an external gate signal. On the falling edge of the gate
;   signal a Timer2 interrupt occurs. The interrupt service
;   routine must clear the Timer2 interrupt status flag in software .

    CLR     T2CON               ; Stops the Timer2 and reset control reg.
    CLR     TMR2                ; Clear contents of the timer register
    MOV     #0xFFFF, w0         ; Load the Period register with
    MOV     w0, PR2             ; the value 0xFFFF

    BSET    IPC1,    #T2IP0     ; Setup Timer2 interrupt for
    BCLR    IPC1,    #T2IP1     ; desired priority level
    BCLR    IPC1,    #T2IP2     ; (this example assigns level 1 priority)
    BCLR    IFS0,    #T2IF      ; Clear the Timer2 interrupt status flag
    BSET    IEC0,    #T2IE      ; Enable Timer2 interrupts
    BSET    T2CON,   #TGATE     ; Set up Timer2 for operation in Gated
                                ; Time Accumulation mode
    BSET    T2CON,   #TON       ; Start Timer2


;   Example code for Timer2 ISR

__T2Interrupt:
    BCLR    IFS0,    #T2IF      ; Reset Timer2 interrupt flag
                                ; User code goes here.
    RETFIE                      ; Return from ISR
```

12.5 Timer Prescalers

The input clock (FOSC/4 or external clock) to all 16-bit timers has prescale options of 1:1, 1:8, 1:64 and 1:256. The clock prescaler is selected using the TCKPS<1:0> control bits (TxCON<5:4>). The prescaler counter is cleared when any of the following occurs:

- A write to the TMRx register
- Clearing TON (TxCON<15>) to '0'
- Any device Reset

Note: The TMRx register is not cleared when TxCON is written.

12.6 Timer Interrupts

A 16-bit timer has the ability to generate an interrupt on a period match or falling edge of the external gate signal, depending on the Operating mode.

The TxIF bit is set when one of the following conditions is true:

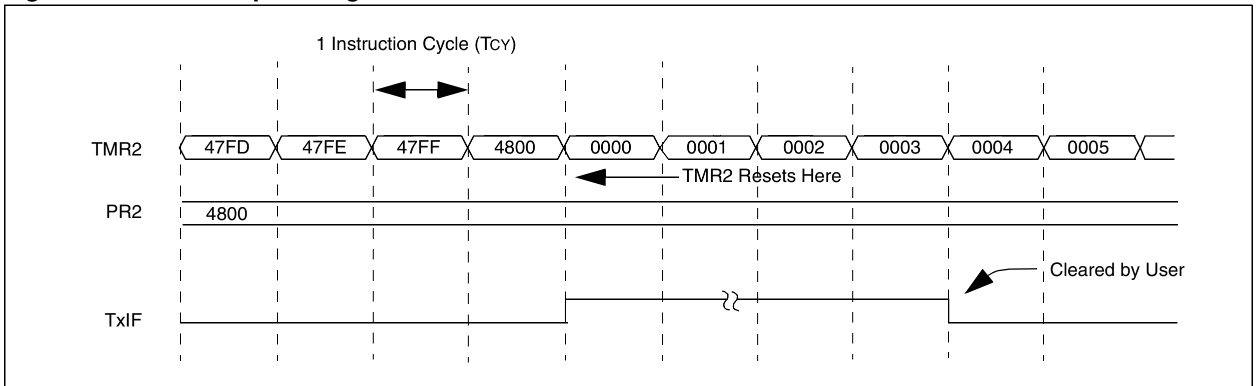
- The timer count matches the respective period register and the timer module is not operating in Gated Time Accumulation mode.
- The falling edge of the "gate" signal is detected when the timer is operating in Gated Time Accumulation mode.

The TxIF bit must be cleared in software.

A timer is enabled as a source of interrupt via the respective timer interrupt enable bit, TxIE. Furthermore, the interrupt priority level bits (TxIP<2:0>) must be written with a non-zero value in order for the timer to be a source of interrupt. Refer to **Section 6. "Reset Interrupts"** for further details.

Note: A special case occurs when the period register is loaded with `0x0000` and the timer is enabled. No timer interrupts will be generated for this configuration.

Figure 12-5: Interrupt Timing for Timer Period Match



12.7 Reading and Writing 16-bit Timer Module Registers

- All timer module SFRs can be written to as a byte (8-bits) or as a word (16-bits).
- All timer module SFRs can only be read as a word (16-bits).

12.7.1 Writing to the 16-bit Timers

The timer and its respective period register can be written to while the module is operating. The user should be aware of the following when byte writes are performed:

- If the timer is incrementing and the low byte of the timer is written to, the upper byte of the timer is not affected. If 0xFF is written into the low byte of the timer, the next timer count clock after this write will cause the low byte to rollover to 0x00 and generate a carry into the high byte of the timer.
- If the timer is incrementing and the high byte of the timer is written to, the low byte of the timer is not affected. If the low byte of the timer contains 0xFF when the write occurs, the next timer count clock will generate a carry from the timer low byte and this carry will cause the upper byte of the timer to increment.

When the TMRx register is written to (word or byte) via an instruction, the TMRx register increment is masked and does not occur during that instruction cycle.

Writes to a timer with an asynchronous clock source should be avoided in a real-timekeeping application. See **Section 12.4.1 “Timer Mode”** for more details.

12.7.2 Reading from the 16-bit Timers

All reads of the timers and their associated SFRs must be word reads (16-bits). A byte read will have no effect (‘0’ will be returned).

The timer and respective period register can be read while the module is operating. A read of the TMRx register does not prevent the timer from incrementing during the same instruction cycle.

12.8 Low Power 32 kHz Crystal Oscillator Input

In each device variant, the Low Power 32 kHz Crystal Oscillator is available to a Type A timer module for Real-Time Clock (RTC) applications.

- The LP Oscillator becomes the clock source for the timer when the LP Oscillator is enabled and the timer is configured to use the external clock source.
- The LP Oscillator is enabled by setting the LPOSCEN control bit in the OSCCON register.
- The 32 kHz crystal is connected to the SOSCO/SOSCI device pins.

Refer to **Section 7. “Oscillator”** for further details.

12.9 32-bit Timer Configuration

A 32-bit timer module can be formed by combining a Type B and a Type C 16-bit timer module. The Type C time base becomes the MSWord of the combined timer and the Type B time base is the LSWord.

When configured for 32-bit operation, the control bits for the Type B time base control the operation of the 32-bit timer. The control bits in the TxCON register for the Type C time base have no effect.

For interrupt control, the combined 32-bit timer uses the interrupt enable, interrupt flag and interrupt priority control bits of the Type C time base. The interrupt control and status bits for the Type B time base are not used during 32-bit timer operation.

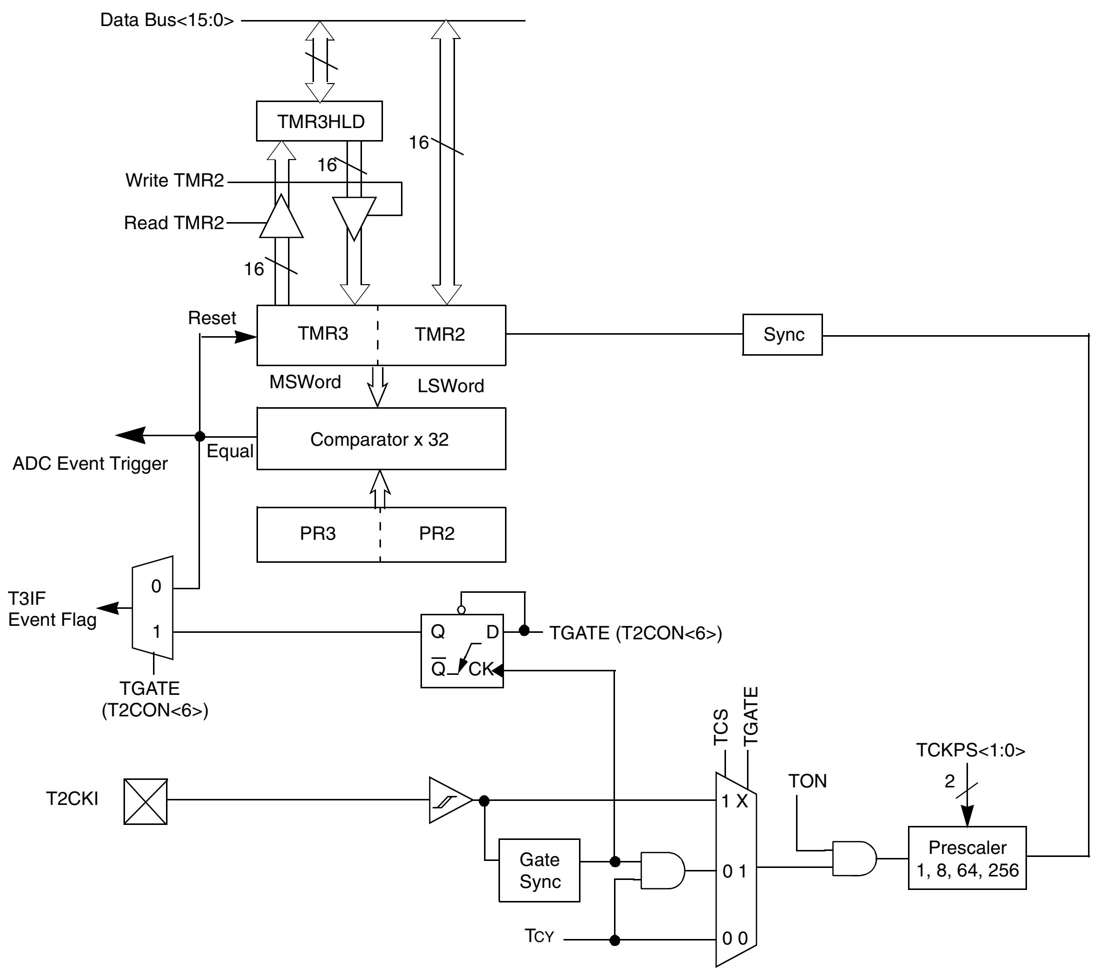
Note:	Refer to the device data sheet for information on the specific Type B and Type C time bases that can be combined.

The following configuration settings assume Timer3 is a Type C time base and Timer2 is a Type B time base:

- TON (T2CON<15>) = `1`.
- T32 (T2CON<3>) = `1`.
- TCKPS<1:0> (T2CON<5:4>) are used to set the Prescaler mode for Timer2 (Type B time base).
- The TMR3:TMR2 register pair contains the 32-bit value of the timer module; the TMR3 (Type C time base) register is the Most Significant Word, while the TMR2 (Type B time base) register is the Least Significant Word of the 32-bit timer value.
- The PR3:PR2 register pair contains the 32-bit period value that is used for comparison with the TMR3:TMR2 timer value.
- T3IE (IEC0<7>) is used to enable the 32-bit timer interrupt for this configuration.
- T3IF (IFS0<7>) is used as a status flag for the timer interrupt.
- T3IP<2:0> (IPC1<14:12>) sets the interrupt priority level for the 32-bit timer.
- T3CON<15:0> are “don’t care” bits.

A block diagram representation of the 32-bit timer module using Timer2 and Timer3 as an example is shown in Figure 12-6.

Figure 12-6: Type B-Type C Timer Pair Block Diagram (32-bit Timer)



Note 1: This block diagram assumes Timer3 is a Type C time base, Timer2 is a Type B time base.

2: Timer configuration bit, T32 (T2CON<3>), must be set to '1' for a 32-bit timer/counter operation. All control bits are respective to the T2CON register.

12.10 32-bit Timer Modes of Operation

12.10.1 Timer Mode

Example 12-5 shows how to configure a 32-bit timer in Timer mode. This example assumes Timer2 is a Type B time base and Timer3 is a Type C time base. For 32-bit timer operation, the T32 control bit must be set in the T2CON register (Type B time base). When Timer2 and Timer3 are configured for a 32-bit timer, the T3CON control bits are ignored. Only the T2CON control bits are required for setup and control. The Timer2 clock and gate input is utilized for the 32-bit timer module, but an interrupt is generated with the T3IF flag. Timer2 is the LSWord and Timer3 is the MSWord of the 32-bit timer. TMR3 is incremented by an overflow (carry-out) from TMR2. The 32-bit timer increments up to a match value preloaded into the combined 32-bit period register formed by PR2 and PR3, then rolls over and continues. For a maximum 32-bit timer count, load PR3:PR2 with a value of `0xFFFFFFFF`. An interrupt is generated on a period match, if enabled.

Example 12-5: Initialization Code for 32-bit Timer Using Instruction Cycle as Input Clock

```
;   The following code example will enable Timer3 interrupts, load the
;   Timer3:Timer2 Period Register and start the 32-bit timer module
;   consisting of Timer3 and Timer2.

;   When a 32-bit period match interrupt occurs, the user must clear
;   the Timer3 interrupt status flag in software.

    CLR     T2CON               ; Stops any 16/32-bit Timer2 operation
    CLR     T3CON               ; Stops any 16-bit Timer3 operation
    CLR     TMR3                ; Clear contents of the Timer3 timer register
    CLR     TMR2                ; Clear contents of the Timer2 timer register
    MOV     #0xFFFF, w0         ; Load the Period Register 3
    MOV     w0, PR3             ; with the value 0xFFFF
    MOV     w0, PR2             ; Load the Period Register2 with value 0xFFFF

    BSET    IPC1,   #T3IP0      ; Setup Timer3 interrupt for
    BCLR    IPC1,   #T3IP1      ; desired priority level
    BCLR    IPC1,   #T3IP2      ; (this example assigns level 1 priority)
    BCLR    IFS0,   #T3IF       ; Clear the Timer3 interrupt status flag
    BSET    IEC0,   #T3IE       ; Enable Timer3 interrupts
    BSET    T2CON,  #T32        ; Enable 32-bit Timer operation
    BSET    T2CON,  #TON        ; Start 32-bit timer with prescaler
                                ; settings at 1:1 and clock source set to
                                ; the internal instruction cycle


;   Example code for Timer3 ISR

__T3Interrupt:
    BCLR    IFS0,   #T3IF       ; Reset Timer3 interrupt flag
                                ; User code goes here.
    RETFIE                      ; Return from ISR
```

12.10.2 Synchronous Counter Mode

The 32-bit timer operates similarly to a 16-bit timer in Synchronous Counter mode. Example 12-6 shows how to configure a 32-bit timer in Synchronous Counter mode. This example assumes Timer2 is a Type B time base and Timer3 is a Type C time base.

Example 12-6: Initialization Code for 32-bit Synchronous Counter Mode Using an External Clock Input

```
;   The following code example will enable Timer2 interrupts, load
;   the Timer3:Timer2 Period register and start the 32-bit timer
;   module consisting of Timer3 and Timer2.

;   When a 32-bit period match interrupt occurs, the user must clear
;   the Timer3 interrupt status flag in the software.

    CLR     T2CON           ; Stops any 16/32-bit Timer2 operation
    CLR     T3CON           ; Stops any 16-bit Timer3 operation
    CLR     TMR3            ; Clear contents of the Timer3 timer register
    CLR     TMR2            ; Clear contents of the Timer2 timer register
    MOV     #0xFFFF, w0     ; Load the Period Register3
    MOV     w0, PR3         ; with the value 0xFFFF
    MOV     w0, PR2         ; Load the Period Register2 with value 0xFFFF

    BSET    IPC1,   #T3IP0  ; Setup Timer3 interrupt for
    BCLR    IPC1,   #T3IP1  ; desired priority level
    BCLR    IPC1,   #T3IP2  ; (this example assigns level 1 priority)
    BCLR    IFS0,   #T3IF   ; Clear the Timer3 interrupt status flag
    BSET    IEC0,   #T3IE   ; Enable Timer3 interrupts
    MOV     #0x801A, w0     ; Enable 32-bit Timer operation and start
    MOV     w0, T2CON       ; 32-bit timer with prescaler settings at
                            ; 1:8 and clock source set to external clock


;   Example code for Timer3 ISR

__T3Interrupt:
    BCLR    IFS0,   #T3IF   ; Reset Timer3 interrupt flag
                            ; User code goes here.
    RETFIE                  ; Return from ISR
```

12.10.3 Asynchronous Counter Mode

Type B and Type C time bases do not support the Asynchronous External Clock mode. Therefore, no 32-bit Asynchronous Counter mode is supported.

12.10.4 Gated Time Accumulation Mode

The 32-bit timer operates similarly to a 16-bit timer in Gated Time Accumulation mode. Example 12-7 shows how to configure a 32-bit timer in Gated Time Accumulation mode. This example assumes Timer2 is a Type B time base and Timer3 is a Type C time base.

Example 12-7: Initialization Code for 32-bit Gated Time Accumulation Mode

```
;   The following code example will enable Timer2 interrupts, load the
;   Timer3:Timer2 Period register and start the 32-bit timer module
;   consisting of Timer3 and Timer2. When a 32-bit period match occurs
;   the timer will simply roll over and continue counting.

;   However, when at the falling edge of the Gate signal on T2CK
;   an interrupt is generated, if enabled. The user must clear the
;   Timer3 interrupt status flag in the software.

    CLR     T2CON               ; Stops any 16/32-bit Timer2 operation
    CLR     T3CON               ; Stops any 16-bit Timer3 operation
    CLR     TMR3                ; Clear contents of the Timer3 register
    CLR     TMR2                ; Clear contents of the Timer2 register
    MOV     #0xFFFF, w0         ; Load the Period Register3
    MOV     w0, PR3             ; with the value 0xFFFF
    MOV     w0, PR2             ; Load the Period Register2 with value 0xFFFF

    BSET    IPC1,    #T3IP0     ; Setup Timer3 interrupt for
    BCLR    IPC1,    #T3IP1     ; desired priority level
    BCLR    IPC1,    #T3IP2     ; (this example assigns level 1 priority)
    BCLR    IFS0,    #T3IF      ; Clear the Timer3 interrupt status flag
    BSET    IEC0,    #T3IE      ; Enable Timer3 interrupts
    MOV     #0x804C, w0         ; Enable 32-bit Timer operation and
    MOV     w0, T2CON           ; Start 32-bit timer in gated time
                                ; accumulation mode.


;   Example code for Timer3 ISR

__T3Interrupt:
    BCLR    IFS0,    #T3IF      ; Reset Timer3 interrupt flag
                                ; User code goes here.
    RETFIE                      ; Return from ISR
```

12.11 Reading and Writing into 32-bit Timers

In order for 32-bit read/write operations to be synchronized between the LSWord and MSWord of the 32-bit timer, additional control logic and holding registers are utilized (see Figure 12-6). Each Type C time base has a register called TMRxHLD, that is used when reading or writing the timer register pair. The TMRxHLD registers are only used when their respective timers are configured for 32-bit operation.

Assuming TMR3:TMR2 form a 32-bit timer pair; the user should first read the LSWord of the timer value from the TMR2 register. The read of the LSWord will automatically transfer the contents of TMR3 into the TMR3HLD register. The user can then read TMR3HLD to get the MSWord of the timer value. This is shown in the example below:

Example 12-8: Reading from a 32-bit Timer

```
;   The following code segment reads the 32-bit timer formed by the
;   Timer3-Timer2 pair into the registers W1(MS Word) and W0(LS Word).

    MOV TMR2, W0       ;Transfer the LSW into W1
    MOV TMR3HLD, W1    ;Transfer the MSW from the holding register to W0
```

To write a value to the TMR3:TMR2 register pair, the user should first write the MSWord to the TMR3HLD register. When the LSWord of the timer value is written to TMR2, the contents of TMR3HLD will automatically be transferred to the TMR3 register.

12.12 Timer Operation in Power Saving States

12.12.1 Timer Operation in Sleep Mode

When the device enters Sleep mode, the system clock is disabled. If the timer module is running from the internal clock source (FOSC/4), it will also be disabled.

A Type A timer is different from the other timer modules because it can operate asynchronously from an external clock source. Because of this distinction, the Type A time base module can continue to operate during Sleep mode. To operate in Sleep mode, Type A time base must be configured as follows:

- The Timer1 module is enabled, TON = 1 (T1CON<15>) and
- The Timer1 clock source is selected as external, TCS = 1 (T1CON<1> = 1) and
- The TSYNC bit (T1CON<2>) is set to logic '0' (Asynchronous Counter mode enabled).

Note: Asynchronous counter operation is only supported for the Timer1 module.

When all of the above conditions are met, Timer1 will continue to count and detect period matches when the device is in Sleep mode. When a match between the timer and the period register occurs, the TxIF bit will be set and an interrupt can be generated to optionally wake the device from Sleep. Refer to **Section 10. "Watchdog Timer and Power Saving Modes"** for further details.

When executing the SLEEP instruction in asynchronous mode using a 32.768 kHz real time oscillator to keep track of real time in seconds, it is important to insure the crystal connected to the inputs is operating. This can be done by checking for a non-zero value in TMR1 and then executing the SLEEP instruction. Failure to do so may result in a normal sleep execution with no wake-up from sleep due to timer time out.

Example Code:

```
mov.b   #0x46,w1           ; follow write sequence …
mov.b   #0x57,w2           ; for OSCCONL writes.
mov     #OSCCONL,w3
mov.b   w1,[w3]
mov.b   w2,[w3]
bset    OSCCONL,#LPOSCEN   ;enable 32Khz external xtal
clr     TMR1               ; set up TMR1 for …
mov     #0x7FFF,W0         ;interrupts every 1.0 Sec
mov     W0,PR1
```

```
    bclr    IFS0,#T1IF          ; clr interrupt flag
    mov     #0x8002,W0          ; External clock, 1:1; start TMR1
    mov     W0,T1CON
    bset    IEC0,#T1IE          ; enable interrupt
TestLPOperation:
    mov     TMR1,W0             ; check for non-zero value in TMR1
    cp      W0,#0
    bra     Z,TestLPOperation   ; if zero then recheck
    pwrsav  #0                  ; non-zero then execute sleep
```

12.12.2 Timer Operation in Idle Mode

When the device enters Idle mode, the system clock sources remain functional and the CPU stops executing code. The timer modules can optionally continue to operate in Idle mode.

The TSIDL bit (TxCON<13>) selects if the timer module will stop in Idle mode, or continue to operate normally. If TSIDL = `0`, the module will continue operation in Idle mode. If TSIDL = `1`, the module will stop in Idle mode.

12.13 Peripherals Using Timer Modules

12.13.1 Time Base for Input Capture/Output Compare

The Input Capture and Output Compare peripherals can select one of two timer modules as their time base. Refer to **Section 13. "Input Capture"**, **Section 14. "Output Compare"**, and the device data sheet for further details.

12.13.2 A/D Special Event Trigger

On each device variant, one Type C time base has the capability to generate a special A/D conversion trigger signal on a period match, in both 16 and 32-bit modes. The timer module provides a conversion start signal to the A/D sampling logic.

- If T32 = `0`, when a match occurs between the 16-bit timer register (TMRx) and the respective 16-bit period register (PRx), the A/D special event trigger signal is generated.
- If T32 = `1`, when a match occurs between the 32-bit timer (TMRx:TMRy) and the 32-bit respective combined period register (PRx:PRy), the A/D special event trigger signal is generated.

The special event trigger signal is always generated by the timer. The trigger source must be selected in the A/D converter control registers. Refer to **Section 17. "10-bit A/D Converter"**, **Section 18. "12-bit A/D Converter"**, and the device data sheet for additional information.

12.13.3 Timer as an External Interrupt Pin

The external clock input pin for each timer can be used as an additional interrupt pin. To provide the interrupt, the timer period register, PRx, is written with a non-zero value and the TMRx register is initialized to a value of 1 less than the value written to the period register. The timer must be configured for a 1:1 clock prescaler. An interrupt will be generated when the next rising edge of the external clock signal is detected.

12.13.4 I/O Pin Control

When a timer module is enabled and configured for external clock or gate operation, the user must ensure the I/O pin direction is configured for an input. Enabling the timer module does not configure the pin direction.

Table 12-1: Special Function Registers Associated with Timer Modules

Name SFR	Address	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on All Resets
TMR1	0100	Timer1 Register																0000 0000 0000 0000
PR1	0102	Timer1 Period Register																1111 1111 1111 1111
T1CON	0104	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	TSYNC	TCS	—	0000 0000 0000 0000
TMR2	0106	Timer2 Register																0000 0000 0000 0000
TMR3HLD	0108	Timer3 Holding Register (used in 32-bit mode only)																0000 0000 0000 0000
TMR3	010A	Timer3 Register																0000 0000 0000 0000
PR2	010C	Timer2 Period Register																1111 1111 1111 1111
PR3	010E	Timer3 Period Register																1111 1111 1111 1111
T2CON	0110	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	T32	—	TCS	—	0000 0000 0000 0000
T3CON	0112	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	—	TCS	—	0000 0000 0000 0000
TMR4	0114	Timer4 Register																0000 0000 0000 0000
TMR5HLD	0116	Timer5 Holding Register (used in 32-bit mode only)																0000 0000 0000 0000
TMR5	0118	Timer5 Register																0000 0000 0000 0000
PR4	011A	Timer4 Period Register																1111 1111 1111 1111
PR5	011C	Timer5 Period Register																1111 1111 1111 1111
T4CON	011E	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	T32	—	TCS	—	0000 0000 0000 0000
T5CON	0120	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	—	TCS	—	0000 0000 0000 0000
IFS0	0084	CNIF	MI2CIF	SI2CIF	NVMIF	ADIF	U1TXIF	U1RXIF	SPI1IF	T3IF	T2IF	OC2IF	IC2IF	T1IF	OC1IF	IC1IF	INT01F	0000 0000 0000 0000
IFS1	0086	IC61F	IC5IF	IC4IF	IC3IF	C1IF	SPI2IF	U2TXIF	U2RXIF	INT2IF	T5IF	T4IF	OC4IF	OC3IF	IC8IF	IC7IF	INT1IF	0000 0000 0000 0000
IEC0	008C	CNIE	MI2CIE	IC2IE	NVMIE	ADIE	U1TXIE	U1RXIE	SPI1IE	T3IE	T2IE	OC2IE	IC2IE	T1IE	OC1IE	IC1IE	INT0IE	0000 0000 0000 0000
IEC1	008E	IC6IE	IC5IE	IC4IE	IC3IE	C1IE	SPI2IE	U2TXIE	U2RXIE	INT2IE	T5IE	T4IE	OC4IE	OC3IE	IC8IE	IC7IE	INT1IE	0000 0000 0000 0000
IPC0	0094	—	T1IP<2:0>			—	OC1IP<2:0>			—	IC1IP<2:0>			INT0IP<2:0>				0100 0100 0100 0100
IPC1	0096	—	T3IP<2:0>			—	T2IP<2:0>			—	OC2IP<2:0>			IC2IP<2:0>				0100 0100 0100 0100
IPC5	009E	—	INT2IP<2:0>			—	T5IP<2:0>			—	T4IP<2:0>			OC4IP<2:0>				0100 0100 0100 0100

Note: Please refer to the device data sheet for specific memory map details.

12.14 Design Tips

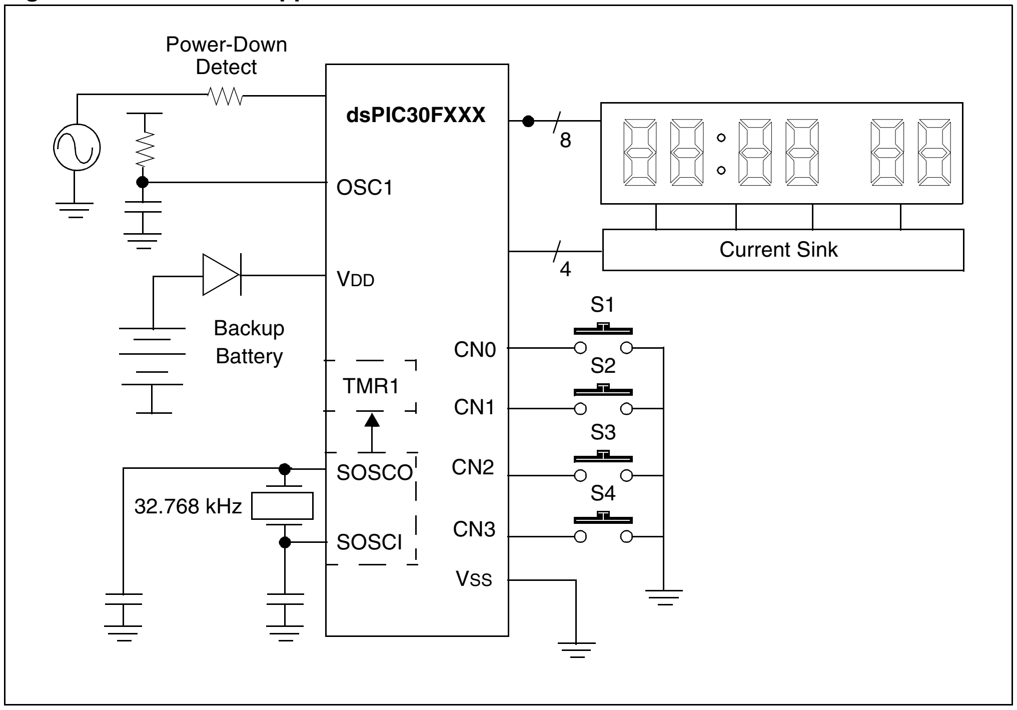
Question 1: ***Can a timer module be used to wake the device from Sleep mode?***

Answer: Yes, but only Timer1 has the ability to wake the device from Sleep mode. This is because Timer1 allows the TMR1 register to increment from an external, unsynchronized clock source. When the TMR1 register is equal to the PR1 register, the device will wake from Sleep mode, if Timer1 interrupts have been enabled using the T1IE control bit. Refer to **Section 12.12.1 "Timer Operation in Sleep Mode"** for further details.

12.14.1 Example Application

An example application is shown in Figure 12-7, where Timer1 (Type A time base) is driven from an external 32.768 kHz oscillator. The external 32.768 kHz oscillator is typically used in applications where real-time needs to be kept, but it is also desirable to have the lowest possible power consumption. The Timer1 oscillator allows the device to be placed in Sleep while the timer continues to increment. When Timer1 overflows, the interrupt wakes up the device so that the appropriate registers can be updated.

Figure 12-7: Timer1 Application



In this example, a 32.768 kHz crystal is used as the time base for the Real-Time Clock. If the clock needs to be updated at 1 second intervals, then the period register, PR1, must be loaded with a value to allow the Timer1 to PR1 match at the desired rate. In the case of a 1 second Timer1 match event, the PR1 register should be loaded with a value of `0x8000`.

Note: The TMR1 register should never be written for correct real-time clock functionality, since the Timer1 clock source is asynchronous to the system clock. Writes to the TMR1 register may corrupt the real-time counter value, resulting in inaccurate timekeeping.

12.15 Related Application Notes

This section lists application notes that are related to this section of the manual. These application notes may not be written specifically for the dsPIC30F Product Family, but the concepts are pertinent and could be used with modification and possible limitations. The current application notes related to the Timers module are:

Title	Application Note #
Using Timer1 in Asynchronous Clock Mode	AN580
Yet Another Clock Featuring the PIC16C924	AN649

Note: Please visit the Microchip web site (www.microchip.com) for additional Application Notes and code examples for the dsPIC30F Family of devices.

12.16 Revision History

Revision A

This is the initial released revision of this document.

Revision B

This revision incorporates technical content changes for the dsPIC30F Timers module.

Revision C

There were no technical content revisions to this section of the manual, however, this section was updated to reflect Revision C throughout the manual.

Revision D

Added example code in **Section 12.12.1 “Timer Operation in Sleep Mode”**.