Section 14. Timers

HIGHLIGHTS

This section of the manual contains the following topics:

14.1 INTRODUCTION

The PIC32MX device family features two different types of timers, depending on the device variant. Timers are useful for generating accurate time-based periodic interrupt events for software applications or real-time operating systems. Other uses include counting external pulses or accurate timing measurement of external events using the timer's gate feature.

With certain exceptions, all of the timers have the same functional circuitry. All timers are classified into two types to account for their functional differences.

- Type A Timer (16-bit synchronous/asynchronous timer/counter with gate)
- Type B Timer (16-bit, 32-bit synchronous timer/counter with gate and Special Event Trigger)

All Timer modules includes the following common features:

- 16-bit timer/counter
- Software-selectable internal or external clock source
- Programmable interrupt generation and priority
- Gated external pulse counter

Beyond the common features, each timer type offers these additional features:

Type A:

- Asynchronous timer/counter with a built-in oscillator
- Operational during CPU SLEEP mode
- Software selectable prescalers 1:1, 1:8, 1:64 and 1:256

Type B:

- Ability to form a 32-bit timer/counter
- Software prescalers 1:1, 1:2, 1:4, 1:8, 1:16, 1:32, 1:64 and 1:256
- Event trigger capability

The following table presents a summary of timer features. For a specific device variant, refer to the PIC32MX device family data sheet for the available type and number of timers.

Table 14-1: Timer Features

Available Timer Types	Secondary Oscillator	Asynchronous External Clock	Synchronous External Clock	16-Bit Synchronous Timer/Counter	32-Bit(1) Synchronous Timer/Counter	Gated Timer	Special Event Trigger
Type A	Yes	Yes	Yes	Yes	No	Yes	No
Type B	No	No	Yes	Yes	Yes	Yes	Yes

Note 1: 32-bit timer/counter configuration requires an even-numbered timer combined with an adjacent odd-numbered timer, e.g., Timer2 and Timer3, or Timer4 and Timer 5.

14.1.1 Type A Timer

Most PIC32MX devices contain at least one Type A timer; usually, Timer1.

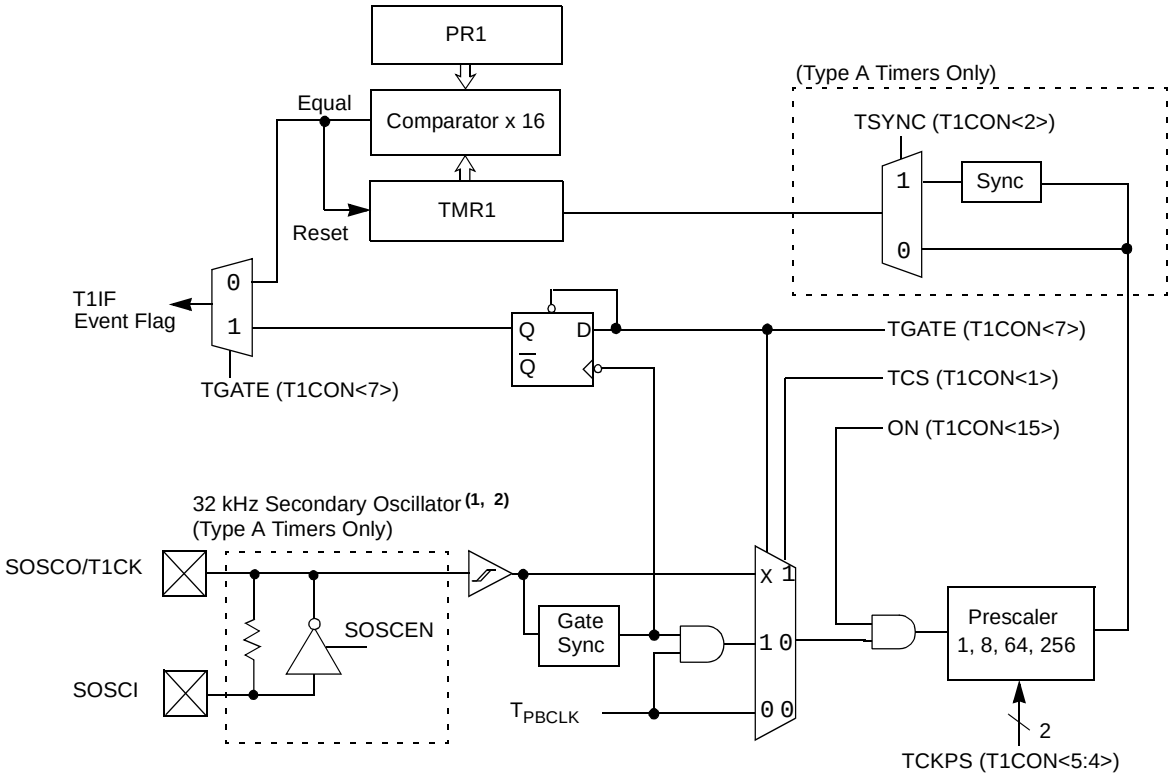
The Type A Timer module is distinct from other types of timers, based on the following features:

- Operable from the external secondary oscillator
- Operable in Asynchronous mode using an external clock source
- Operable during CPU SLEEP mode
- Software selectable prescalers 1:1, 1:8, 1:64 and 1:256

The Type A Timer does not support 32-bit mode.

The unique features of a Type A Timer module allow it to be used for Real-Time Clock (RTC) applications. A block diagram of the Type A Timer module is shown in Figure 14-1.

Figure 14-1: Type A Timer Block Diagram



Note 1: Refer to **Section 6. "Oscillators"** for information on enabling the 32 kHz Secondary Oscillator.

2: The default state of the SOSCEN (OSCCON<1>) during a device Reset is controlled by the FSOSCEN bit in Configuration Word DEVCFG1.

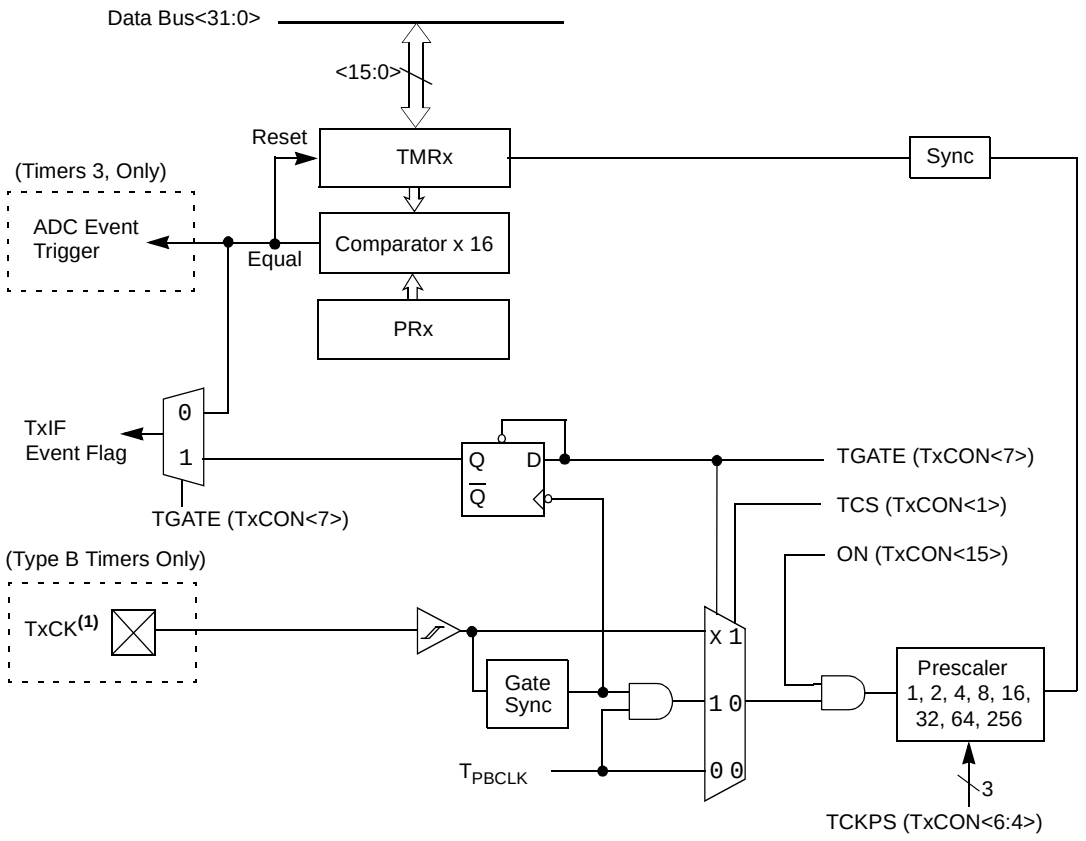
14.1.2 Type B Timer

The Type B timer is distinct from other types of timers, based on the following features:

- Can be combined to form a 32-bit timer
- Software selectable prescalers 1:1, 1:2, 1:4, 1:8, 1:16, 1:32, 1:64 and 1:256
- ADC event trigger capability

A block diagram of Type B timer (16-bit) is shown in Figure 14-2.

Figure 14-2: Type B Timer Block Diagram (16-Bit)

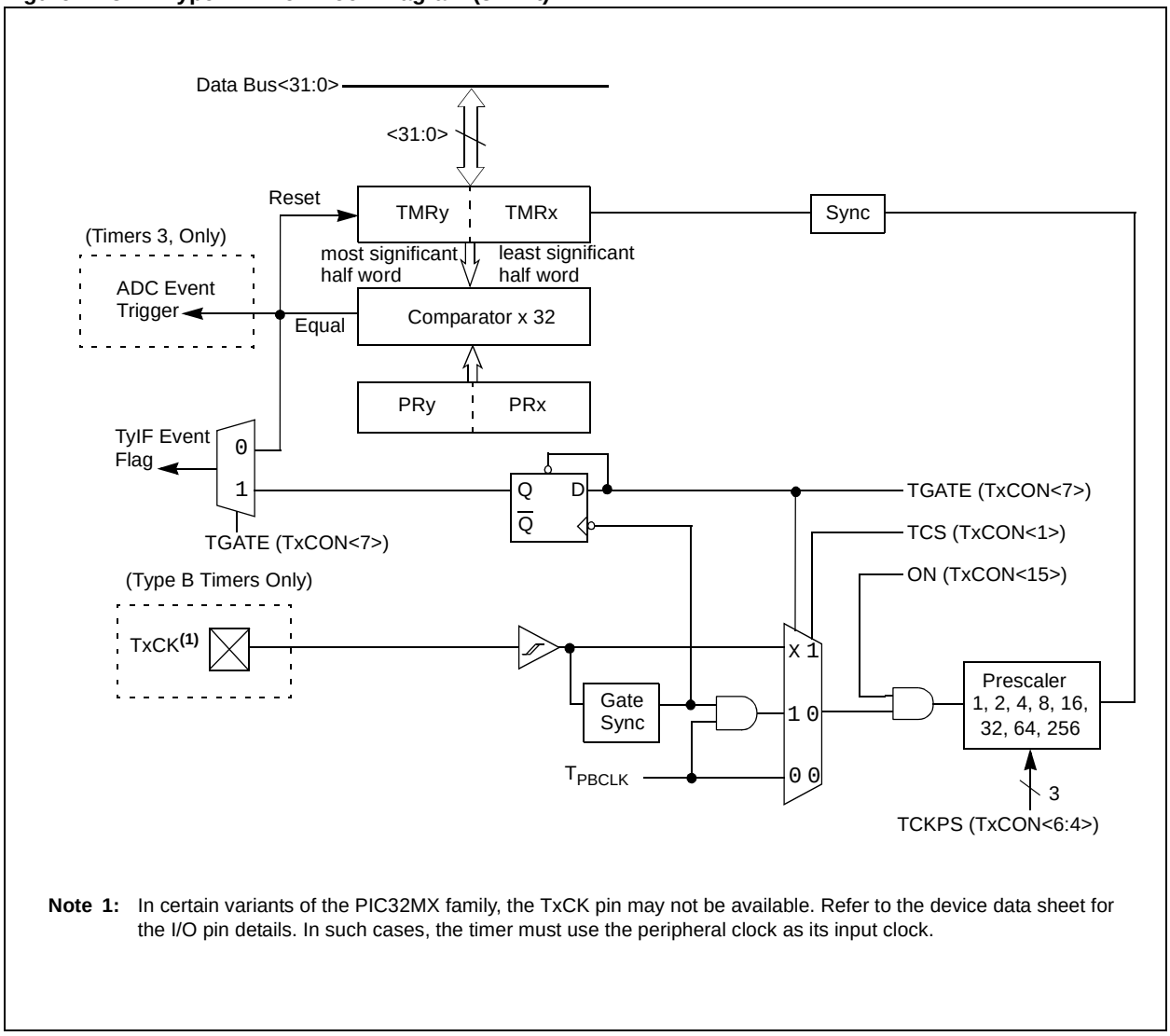


Note 1: In certain variants of the PIC32MX family, the TxCK pin may not be available. Refer to the device data sheet for the I/O pin details. In such cases, the timer must use the peripheral clock as its input clock.

A block diagram of Type B timer (32-bit) is shown in Figure 14-3.

Note: The Timer Configuration bit, T32 (TxCON<3>), must be set to '1' for a 32-bit timer/counter operation. All control bits are respective to the TxCON register. All interrupt bits are respective to the TyCON register.

Figure 14-3: Type B Timer Block Diagram (32-Bit)



Note 1: In certain variants of the PIC32MX 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 peripheral clock as its input clock.

14.2 CONTROL REGISTERS

Note: Each PIC32MX device variant may have one or more Timer modules. An 'x' used in the names of pins, control/Status bits, and registers denotes the particular module. Refer to the specific device data sheets for more details.

Each timer module is a 16-bit timer/counter that consists of the following Special Function Registers (SFRs):

- TxCON: 16-Bit Control Register Associated with the Timer
- TxCONCLR, TxCONSET, TxCONINV: Atomic Bit Manipulation Write-only Registers for TxCON
- TMRx: 16-Bit Timer Count Register
- TMRxCLR, TMRxSET, TMRxINV: Atomic Bit Manipulation Write-only Registers for TMRx
- PRx: 16-Bit Period Register Associated with the Timer
- PRxCLR, PRxSET, PRxINV: Atomic Bit Manipulation Write-only Registers for PRx

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

- TxIE: Interrupt Enable Control Bit – in IEC0 INT Register
- TxIF: Interrupt Flag Status Bit – in IFS0 INT Register
- TxIP<2:0>: Interrupt Priority Control Bits – in IPC1, IPC2, IPC3, IPC4, IPC5 INT Registers
- TxIS<1:0>: Interrupt Subpriority Control Bits – in IPC1, IPC2, IPC3, IPC4, IPC5 INT Registers

The following table summarizes all Timer-related registers. Corresponding registers appear after the summary, followed by a detailed description of each register.

Table 14-2: Timers SFR Summary

Name		Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
T1CON	31:24	—	—	—	—	—	—	—	—
	23:16	—	—	—	—	—	—	—	—
	7:0	TGATE	—	TCKPS<1:0>		—	TSYNC	TCS	—
T1CONCLR	31:0	Write clears selected bits in T1CON, read yields undefined value							
T1CONSET	31:0	Write sets selected bits in T1CON, read yields undefined value							
T1CONINV	31:0	Write inverts selected bits in T1CON, read yields undefined value							
TxCON	31:24	—	—	—	—	—	—	—	—
	23:16	—	—	—	—	—	—	—	—
	15:8	ON	FRZ	SIDL	—	—	—	—	—
	7:0	TGATE	TCKPS<2:0>(2)			T32(1)	—	TCS	—
TxCONCLR	31:0	Write clears selected bits in TxCON, read yields undefined value							
TxCONSET	31:0	Write sets selected bits in TxCON, read yields undefined value							
TxCONINV	31:0	Write inverts selected bits in TxCON, read yields undefined value							
TMRx	31:24	—	—	—	—	—	—	—	—
	23:16	—	—	—	—	—	—	—	—
	15:8	TMRx<15:8>							
	7:0	TMRx<7:0>							
TMRxCLR	31:0	Write clears selected bits in TMRx, read yields undefined value							
TMRxSET	31:0	Write sets selected bits in TMRx, read yields undefined value							
TMRxINV	31:0	Write inverts selected bits in TMRx, read yields undefined value							

Note 1: Bit T32 is available only on even-numbered Type B timers, e.g., Timer2, Timer4.
2: TCKPS<2:0> is available only on even-numbered Type B timers, e.g., Timer2, Timer4 in 32-bit Timer mode.

Table 14-2: Timers SFR Summary (Continued)

Name		Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
PRx	31:24	—	—	—	—	—	—	—	—
	23:16	—	—	—	—	—	—	—	—
	15:8	PRx<15:8>							
	7:0	PRx<7:0>							
PRxCLR	31:0	Write clears selected bits in PRx, read yields undefined value							
PRxSET	31:0	Write sets selected bits in PRx, read yields undefined value							
PRxINV	31:0	Write inverts selected bits in PRx, read yields undefined value							
IEC0	31:24	I2C1MIE	I2C1SIE	I2C1BIE	U1TXIE	U1RXIE	U1EIE	SPI1RXIE	SPI1TXIE
	23:16	SPI1EIE	OC5IE	IC5IE	T5IE	INT4IE	OC4IE	IC4IE	T4IE
	15:8	INT3IE	OC3IE	IC3IE	T3IE	INT2IE	OC2IE	IC2IE	T2IE
	7:0	INT1IE	OC1IE	IC1IE	T1IE	INT0IE	CS1IE	CS0IE	CTIE
IFS0	31:24	I2C1MIF	I2C1SIF	I2C1BIF	U1TXIF	U1RXIF	U1EIF	SPI1RXIF	SPI1TXIF
	23:16	SPI1EIF	OC5IF	IC5IF	T5IF	INT4IF	OC4IF	IC4IF	T4IF
	15:8	INT3IF	OC3IF	IC3IF	T3IF	INT2IF	OC2IF	IC2IF	T2IF
	7:0	INT1IF	OC1IF	IC1IF	T1IF	INT0IF	CS1IF	CS0IF	CTIF
IPC1	31:24	—	—	—	INT1IP<2:0>			INT1IS<1:0>	
	23:16	—	—	—	OC1IP<2:0>			OC1IS<1:0>	
	15:8	—	—	—	IC1IP<2:0>			IC1IS<1:0>	
	7:0	—	—	—	T1IP<2:0>			T1IS<1:0>	
IPC2	31:24	—	—	—	INT2IP<2:0>			INT2IS<1:0>	
	23:16	—	—	—	OC2IP<2:0>			OC2IS<1:0>	
	15:8	—	—	—	IC2IP<2:0>			IC2IS<1:0>	
	7:0	—	—	—	T2IP<2:0>			T2IS<1:0>	
IPC3	31:24	—	—	—	INT3IP<2:0>			INT3IS<1:0>	
	23:16	—	—	—	OC3IP<2:0>			OC3IS<1:0>	
	15:8	—	—	—	IC3IP<2:0>			IC3IS<1:0>	
	7:0	—	—	—	T3IP<2:0>			T3IS<1:0>	
IPC4	31:24	—	—	—	INT4IP<2:0>			INT4IS<1:0>	
	23:16	—	—	—	OC4IP<2:0>			OC4IS<1:0>	
	15:8	—	—	—	IC4IP<2:0>			IC4IS<1:0>	
	7:0	—	—	—	T4IP<2:0>			T4IS<1:0>	
IPC5	31:24	—	—	—	SPI1IP<2:0>			SPI1IS<1:0>	
	23:16	—	—	—	OC5IP<2:0>			OC5IS<1:0>	
	15:8	—	—	—	IC5IP<2:0>			IC5IS<1:0>	
	7:0	—	—	—	T5IP<2:0>			T5IS<1:0>	

Note 1: Bit T32 is available only on even-numbered Type B timers, e.g., Timer2, Timer4.

2: TCKPS<2:0> is available only on even-numbered Type B timers, e.g., Timer2, Timer4 in 32-bit Timer mode.

Register 14-1: T1CON: Type A Timer Control Register

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 31							bit 24

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 23							bit 16

R/W-0	R/W-0	R/W-0	R/W-0	R-0	r-x	r-x	r-x
ON	FRZ	SIDL	TWDIS	TWIP	—	—	—
bit 15							bit 8

R/W-0	r-x	R/W-0	R/W-0	r-x	R/W-0	R/W-0	r-x
TGATE	—	TCKPS<1:0>		—	TSYNC	TCS	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	P = Programmable bit	r = Reserved bit
U = Unimplemented bit	-n = Bit Value at POR: (‘0’, ‘1’, x = Unknown)		

bit 31-16 **Reserved:** Write ‘0’; ignore read

bit 15 **ON:** Timer On bit

1 = Timer is enabled
0 = Timer is disabled

Note: When using 1:1 PBCLK divisor, the user’s software should not read/write the peripheral’s SFRs in the SYSCLK cycle immediately following the instruction that clears the module’s ON bit.

bit 14 **FRZ:** Freeze in Debug Exception Mode bit

1 = Freeze operation when CPU is in Debug Exception mode
0 = Continue operation even when CPU is in Debug Exception mode

Note: FRZ is writable in Debug Exception mode only, it is forced to ‘0’ in Normal mode.

bit 13 **SIDL:** Stop in IDLE Mode bit

1 = Discontinue operation when device enters IDLE mode
0 = Continue operation even in IDLE mode

bit 12 **TWDIS:** Asynchronous Timer Write Disable bit

1 = Writes to TMR1 are ignored until pending write operation completes
0 = Back to back writes are enabled (Legacy Asynchronous Timer functionality)

bit 11 **TWIP:** Asynchronous Timer Write in Progress bit

In Asynchronous Timer mode:
1 = Asynchronous write to TMR1 register in progress
0 = Asynchronous write to TMR1 register complete

In Synchronous Timer mode:
This bit is read as ‘0’.

bit 10-8 **Reserved:** Write ‘0’; ignore read

Register 14-1: T1CON: Type A Timer Control Register (Continued)

bit 7 **TGATE:** Timer Gated Time Accumulation Enable bit

When TCS = 1:
This bit is ignored and read '0'.

When TCS = 0:
1 = Gated time accumulation is enabled
0 = Gated time accumulation is disabled

bit 6 **Reserved:** Write '0'; ignore read

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 **Reserved:** Write '0'; ignore read

bit 2 **TSYNC:** Timer External Clock Input Synchronization Selection bit

When TCS = 1:
1 = External clock input is synchronized
0 = External clock input is not synchronized

When TCS = 0:
This bit is ignored and read '0'.

bit 1 **TCS:** Timer Clock Source Select bit

1 = External clock from TxCKI pin
0 = Internal peripheral clock

bit 0 **Reserved:** Write '0'; ignore read

Register 14-2: T1CONCLR: Timer Control Clear Register

Write clears selected bits in T1CON, read yields undefined value	
bit 31	bit 0

bit 31-0 **Clears selected bits in T1CON**

A write of '1' in one or more bit positions clears the corresponding bit(s) in T1CON register and does not affect unimplemented or read-only bits. A write of '0' will not affect the register.

Example: `T1CONCLR = 0x00008001` will clear bits 15 and 0 in T1CON register.

Register 14-3: T1CONSET: Timer Control Set Register

Write sets selected bits in T1CON, read yields undefined value	
bit 31	bit 0

bit 31-0 **Sets selected bits in T1CON**

A write of '1' in one or more bit positions sets the corresponding bit(s) in T1CON register and does not affect unimplemented or read-only bits. A write of '0' will not affect the register.

Example: `T1CONSET = 0x00008001` will set bits 15 and 0 in T1CON register.

Register 14-4: T1CONINV: Timer Control Invert Register

Write inverts selected bits in T1CON, read yields undefined value	
bit 31	bit 0

bit 31-0 **Inverts selected bits in T1CON**

A write of '1' in one or more bit positions inverts the corresponding bit(s) in T1CON register and does not affect unimplemented or read-only bits. A write of '0' will not affect the register.

Example: `T1CONINV = 0x00008001` will invert bits 15 and 0 in T1CON register.

Register 14-5: TxCON: Type B Timer Control Register

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 31							bit 24

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 23							bit 16

R/W-0	R/W-0	R/W-0	r-x	r-x	r-x	r-x	r-x
ON	FRZ	SIDL	—	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	r-x	R/W-0	r-x
TGATE	TCKPS<2:0>			T32	—	TCS	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	P = Programmable bit	r = Reserved bit
U = Unimplemented bit	-n = Bit Value at POR: (‘0’, ‘1’, x = Unknown)		

bit 31-16 **Reserved:** Write ‘0’; ignore read

bit 15 **ON:** Timer On bit

1 = Module is enabled
0 = Module is disabled

Note: When using 1:1 PBCLK divisor, the user’s software should not read/write the peripheral’s SFRs in the SYSCLK cycle immediately following the instruction that clears the module’s ON bit.

bit 14 **FRZ:** Freeze in Debug Exception Mode bit

1 = Freeze operation when CPU is in Debug Exception mode
0 = Continue operation even when CPU is in Debug Exception mode
Note: FRZ is writable in Debug Exception mode only, it is forced to ‘0’ in Normal mode.

bit 13 **SIDL:** Stop in IDLE Mode bit

1 = Discontinue operation when device enters IDLE mode
0 = Continue operation even in IDLE mode

bit 12-8 **Reserved:** Write ‘0’; ignore read

bit 7 **TGATE:** Timer Gated Time Accumulation Enable bit

When TCS = 1:
This bit is ignored and read ‘0’.

When TCS = 0:
1 = Gated time accumulation is enabled
0 = Gated time accumulation is disabled

Register 14-5: TxCON: Type B Timer Control Register (Continued)

bit 6-4 **TCKPS<2:0>:** Timer Input Clock Prescale Select bits

111 = 1:256 prescale value
110 = 1:64 prescale value
101 = 1:32 prescale value
100 = 1:16 prescale value
011 = 1:8 prescale value
010 = 1:4 prescale value
001 = 1:2 prescale value
000 = 1:1 prescale value

bit 3 **T32:** 32-Bit Timer Mode Select bit

1 = TMRx and TMRy form a 32-bit timer
0 = TMRx and TMRy form separate 16-bit timer

Note: Bit T32 is available only on even-numbered Type B timers: Timer 2, Timer 4, etc.

bit 2 **Reserved:** Write '0'; ignore read

bit 1 **TCS:** Timer Clock Source Select bit

1 = External clock from TxCKI pin
0 = Internal peripheral clock

bit 0 **Reserved:** Write '0'; ignore read

Register 14-6: TxCONCLR: Type B Timer Control Clear Register

Write clears selected bits in TxCON, read yields undefined value	
bit 31	bit 0

bit 31-0 **Clears selected bits in TxCON**

A write of '1' in one or more bit positions clears the corresponding bit(s) in TxCON register and does not affect unimplemented or read-only bits. A write of '0' will not affect the register.

Example: `TxCONCLR = 0x00008001` will clear bits 15 and 0 in TxCON register.

Register 14-7: TxCONSET: Type B Timer Control Set Register

Write sets selected bits in TxCON, read yields undefined value	
bit 31	bit 0

bit 31-0 **Sets selected bits in TxCON**

A write of '1' in one or more bit positions sets the corresponding bit(s) in TxCON register and does not affect unimplemented or read-only bits. A write of '0' will not affect the register.

Example: `TxCONSET = 0x00008001` will set bits 15 and 0 in TxCON register.

Register 14-8: TxCONINV: Type B Timer Control Invert Register

Write inverts selected bits in TxCON, read yields undefined value	
bit 31	bit 0

bit 31-0 **Inverts selected bits in TxCON**

A write of '1' in one or more bit positions inverts the corresponding bit(s) in TxCON register and does not affect unimplemented or read-only bits. A write of '0' will not affect the register.

Example: `TxCONINV = 0x00008001` will invert bits 15 and 0 in TxCON register.

Register 14-9: TMRx: Timer 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
TMR<15:8>							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TMR<7:0>							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	P = Programmable bit	r = Reserved bit
U = Unimplemented bit	-n = Bit Value at POR: (‘0’, ‘1’, x = Unknown)		

bit 15-0 **TMRx<15:0>:** Timer Count Register

16-bit mode:
These bits represent the complete 16-bit timer count.

32-bit mode (Timer Type B only):
Timer2 and Timer4
These bits represent the least significant half word (16 bits) of the 32-bit timer count.
Timer3 and Timer5
These bits represent the most significant half word (16 bits) of the 32-bit timer count.

Register 14-10: TMRxCLR: Timer Clear Register

Write clears selected bits in TMRx, read yields undefined value	
bit 31	bit 0

bit 31-0 **Clears selected bits in TMRx**

A write of '1' in one or more bit positions clears the corresponding bit(s) in TMRx register and does not affect unimplemented or read-only bits. A write of '0' will not affect the register.

Example: `TMRxCLR = 0x00008001` will clear bits 15 and 0 in TMRx register.

Register 14-11: TMRxSET: Timer Set Register

Write sets selected bits in TMRx, read yields undefined value	
bit 31	bit 0

bit 31-0 **Sets selected bits in TMRx**

A write of '1' in one or more bit positions sets the corresponding bit(s) in TMRx register and does not affect unimplemented or read-only bits. A write of '0' will not affect the register.

Example: `TMRxSET = 0x00008001` will set bits 15 and 0 in TMRx register.

Register 14-12: TMRxINV: Timer Invert Register

Write inverts selected bits in TMRx, read yields undefined value	
bit 31	bit 0

bit 31-0 **Inverts selected bits in TMRx**

A write of '1' in one or more bit positions inverts the corresponding bit(s) in TMRx register and does not affect unimplemented or read-only bits. A write of '0' will not affect the register.

Example: `TMRxINV = 0x00008001` will invert bits 15 and 0 in TMRx register.

Register 14-13: PRx: Period 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
PR<15:8>							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PR<7:0>							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	P = Programmable bit	r = Reserved bit
U = Unimplemented bit	-n = Bit Value at POR: (‘0’, ‘1’, x = Unknown)		

bit 15-0 **PRx<15:0>:** Period Register

16-bit mode:
These bits represent the complete 16-bit period match

32-bit mode (Timer Type B only):
Timer2 and Timer4
These bits represent the least significant half word (16 bits) of the 32-bit period match.
Timer3 and Timer5
These bits represent the most significant half word (16 bits) of the 32-bit period match.

Register 14-14: PRxCLR: Period Clear Register

Write clears selected bits in PRx, read yields undefined value	
bit 31	bit 0

bit 31-0 **Clears selected bits in PRx**

A write of '1' in one or more bit positions clears the corresponding bit(s) in PRx register and does not affect unimplemented or read-only bits. A write of '0' will not affect the register.

Example: `PRxCLR = 0x00008001` **will clear bits 15 and 0 in PRx register.**

Register 14-15: PRxSET: Period Set Register

Write sets selected bits in PRx, read yields undefined value	
bit 31	bit 0

bit 31-0 **Sets selected bits in PRx**

A write of '1' in one or more bit positions sets the corresponding bit(s) in PRx register and does not affect unimplemented or read-only bits. A write of '0' will not affect the register.

Example: `PRxSET = 0x00008001` **will set bits 15 and 0 in PRx register.**

Register 14-16: PRxINV: Period Invert Register

Write inverts selected bits in PRx, read yields undefined value	
bit 31	bit 0

bit 31-0 **Inverts selected bits in PRx**

A write of '1' in one or more bit positions inverts the corresponding bit(s) in PRx register and does not affect unimplemented or read-only bits. A write of '0' will not affect the register.

Example: `PRxINV = 0x00008001` **will invert bits 15 and 0 in PRx register.**

Register 14-17: IEC0: Interrupt Enable Control Register(1)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
I2C1MIE	I2C1SIE	I2C1BIE	U1TXIE	U1RXIE	U1EIE	SPI1RXIE	SPI1TXIE
bit 31							bit 24

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SPI1EIE	OC5IE	IC5IE	T5IE	INT4IE	OC4IE	IC4IE	T4IE
bit 23							bit 16

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INT3IE	OC3IE	IC3IE	T3IE	INT2IE	OC2IE	IC2IE	T2IE
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INT1IE	OC1IE	IC1IE	T1IE	INT0IE	CS1IE	CS0IE	CTIE
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	P = Programmable bit	r = Reserved bit
U = Unimplemented bit	-n = Bit Value at POR: (‘0’, ‘1’, x = Unknown)		

bit 20 **T5IE:** Timer5 Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

bit 16 **T4IE:** Timer4 Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

bit 12 **T3IE:** Timer3 Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

bit 8 **T2IE:** Timer2 Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

bit 4 **T1IE:** Timer1 Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

Note 1: Shaded bit names in this Interrupt register control other PIC32MX peripherals and are not related to the Timers.

Register 14-18: IFS0: Interrupt Flag Status Register 0(1)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
I2C1MIF	I2C1SIF	I2C1BIF	U1TXIF	U1RXIF	U1EIF	SPI1RXIF	SPI1TXIF
bit 31							bit 24

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SPI1EIF	OC5IF	IC5IF	T5IF	INT4IF	OC4IF	IC4IF	T4IF
bit 23							bit 16

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INT3IF	OC3IF	IC3IF	T3IF	INT2IF	OC2IF	IC2IF	T2IF
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INT1IF	OC1IF	IC1IF	T1IF	INT0IF	CS1IF	CS0IF	CTIF
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	P = Programmable bit	r = Reserved bit
U = Unimplemented bit	-n = Bit Value at POR: (‘0’, ‘1’, x = Unknown)		

bit 20 **T5IF:** Timer5 Interrupt Request Flag bit
1 = Interrupt request has occurred
0 = No interrupt request has occurred

bit 16 **T4IF:** Timer4 Interrupt Request Flag bit
1 = Interrupt request has occurred
0 = No interrupt request has occurred

bit 12 **T3IF:** Timer3 Interrupt Request Flag bit
1 = Interrupt request has occurred
0 = No interrupt request has occurred

bit 8 **T2IF:** Timer2 Interrupt Request Flag bit
1 = Interrupt request has occurred
0 = No interrupt request has occurred

bit 4 **T1IF:** Timer1 Interrupt Request Flag bit
1 = Interrupt request has occurred
0 = No interrupt request has occurred

Note 1: Shaded bit names in this Interrupt register control other PIC32MX peripherals and are not related to the Timers.

Register 14-19: IPC1: Interrupt Priority Control Register 1(1)

r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	INT1IP<2:0>			INT1IS<1:0>	
bit 31							bit 24

r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	OC1IP<2:0>			OC1IS<1:0>	
bit 23							bit 16

r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	IC1IP<2:0>			IC1IS<1:0>	
bit 15							bit 8

r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	T1IP<2:0>			T1IS<1:0>	
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	P = Programmable bit	r = Reserved bit
U = Unimplemented bit	-n = Bit Value at POR: (‘0’, ‘1’, x = Unknown)		

bit 4-2 **T1IP<2:0>:** Timer1 Interrupt Priority bits
- 111 = Interrupt Priority is 7
- 110 = Interrupt Priority is 6
- 101 = Interrupt Priority is 5
- 100 = Interrupt Priority is 4
- 011 = Interrupt Priority is 3
- 010 = Interrupt Priority is 2
- 001 = Interrupt Priority is 1
- 000 = Interrupt is disabled

bit 1-0 **T1IS<1:0>:** Timer1 Interrupt Subpriority bits
- 11 = Interrupt Subpriority is 3
- 10 = Interrupt Subpriority is 2
- 01 = Interrupt Subpriority is 1
- 00 = Interrupt Subpriority is 0

Note 1: Shaded bit names in this Interrupt register control other PIC32MX peripherals and are not related to the Timer1.

Register 14-20: IPC2: Interrupt Priority Control Register 2(1)

r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	INT2IP<2:0>			INT2IS<1:0>	
bit 31							bit 24

r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	OC2IP<2:0>			OC2IS<1:0>	
bit 23							bit 16

r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	IC2IP<2:0>			IC2IS<1:0>	
bit 15							bit 8

r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	T2IP<2:0>			T2IS<1:0>	
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	P = Programmable bit	r = Reserved bit
U = Unimplemented bit	-n = Bit Value at POR: (‘0’, ‘1’, x = Unknown)		

bit 4-2 **T2IP<2:0>:** Timer2 Interrupt Priority bits

111 = Interrupt Priority is 7
110 = Interrupt Priority is 6
101 = Interrupt Priority is 5
100 = Interrupt Priority is 4
011 = Interrupt Priority is 3
010 = Interrupt Priority is 2
001 = Interrupt Priority is 1
000 = Interrupt is disabled

bit 1-0 **T2IS<1:0>:** Timer2 Interrupt Subpriority bits

11 = Interrupt Subpriority is 3
10 = Interrupt Subpriority is 2
01 = Interrupt Subpriority is 1
00 = Interrupt Subpriority is 0

Note 1: Shaded bit names in this Interrupt register control other PIC32MX peripherals and are not related to the Timer2.

Register 14-21: IPC3: Interrupt Priority Control Register 3(1)

r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	INT3IP<2:0>			INT3IS<1:0>	
bit 31							bit 24

r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	OC3IP<2:0>			OC3IS<1:0>	
bit 23							bit 16

r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	IC3IP<2:0>			IC3IS<1:0>	
bit 15							bit 8

r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	T3IP<2:0>			T3IS<1:0>	
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	P = Programmable bit	r = Reserved bit
U = Unimplemented bit	-n = Bit Value at POR: (‘0’, ‘1’, x = Unknown)		

bit 4-2 **T3IP<2:0>:** Timer3 Interrupt Priority bits

111 = Interrupt Priority is 7
110 = Interrupt Priority is 6
101 = Interrupt Priority is 5
100 = Interrupt Priority is 4
011 = Interrupt Priority is 3
010 = Interrupt Priority is 2
001 = Interrupt Priority is 1
000 = Interrupt is disabled

bit 1-0 **T3IS<1:0>:** Timer3 Interrupt Subpriority bits

11 = Interrupt Subpriority is 3
10 = Interrupt Subpriority is 2
01 = Interrupt Subpriority is 1
00 = Interrupt Subpriority is 0

Note 1: Shaded bit names in this Interrupt register control other PIC32MX peripherals and are not related to the Timer3.

Register 14-22: IPC4: Interrupt Priority Control Register 4(1)

r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	INT4IP<2:0>			INT4IS<1:0>	
bit 31							bit 24

r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	OC4IP<2:0>			OC4IS<1:0>	
bit 23							bit 16

r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	IC4IP<2:0>			IC4IS<1:0>	
bit 15							bit 8

r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	T4IP<2:0>			T4IS<1:0>	
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	P = Programmable bit	r = Reserved bit
U = Unimplemented bit	-n = Bit Value at POR: (‘0’, ‘1’, x = Unknown)		

bit 4-2 **T4IP<2:0>:** Timer4 Interrupt Priority bits
- 111 = Interrupt Priority is 7
- 110 = Interrupt Priority is 6
- 101 = Interrupt Priority is 5
- 100 = Interrupt Priority is 4
- 011 = Interrupt Priority is 3
- 010 = Interrupt Priority is 2
- 001 = Interrupt Priority is 1
- 000 = Interrupt is disabled

bit 1-0 **T4IS<1:0>:** Timer4 Interrupt Subpriority bits
- 11 = Interrupt Subpriority is 3
- 10 = Interrupt Subpriority is 2
- 01 = Interrupt Subpriority is 1
- 00 = Interrupt Subpriority is 0

Note 1: Shaded bit names in this Interrupt register control other PIC32MX peripherals and are not related to the Timer4.

Register 14-23: IPC5: Interrupt Priority Control Register 5(1)

r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	SPI1IP<2:0>			SPI1IS<1:0>	
bit 31							bit 24

r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	OC5IP<2:0>			OC5IS<1:0>	
bit 23							bit 16

r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	IC5IP<2:0>			IC5IS<1:0>	
bit 15							bit 8

r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	T5IP<2:0>			T5IS<1:0>	
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	P = Programmable bit	r = Reserved bit
U = Unimplemented bit	-n = Bit Value at POR: (‘0’, ‘1’, x = Unknown)		

bit 4-2 **T5IP<2:0>:** Timer5 Interrupt Priority bits
- 111 = Interrupt Priority is 7
- 110 = Interrupt Priority is 6
- 101 = Interrupt Priority is 5
- 100 = Interrupt Priority is 4
- 011 = Interrupt Priority is 3
- 010 = Interrupt Priority is 2
- 001 = Interrupt Priority is 1
- 000 = Interrupt is disabled

bit 1-0 **T5IS<1:0>:** Timer5 Interrupt Subpriority bits
- 11 = Interrupt Subpriority is 3
- 10 = Interrupt Subpriority is 2
- 01 = Interrupt Subpriority is 1
- 00 = Interrupt Subpriority is 0

Note 1: Shaded bit names in this Interrupt register control other PIC32MX peripherals and are not related to the Timer5.

14.3 MODES OF OPERATION

14.3.1 16-Bit Modes

Both Type A and Type B timer modules support the following 16-bit modes:

- 16-bit Synchronous Clock Counter
- 16-bit Synchronous External Clock Counter
- 16-bit Asynchronous External Counter (Type A Timer module only)
- 16-bit Gated Timer

The 16-bit 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 Timer module only)
- TGATE (TxCON<7>): Timer Gate Control bit

14.3.1.1 16-Bit Timer Considerations

The following should be considered when using a 16-bit timer:

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

14.3.2 32-Bit Modes (Type B Timer)

Only Type B timer modules support 32-bit modes of operation. A 32-Bit Timer module is formed by combining an even numbered Type B timer (referred to as TimerX) with a consecutive odd numbered Type B timer (referred to as TimerY). For example, 32-bit timer combinations are Timer2 and Timer3, Timer4 and Timer5, etc. The number of timer pairs depends on the device family variant.

The 32-Bit Timer pairs can operate in the following modes:

- 32-Bit Synchronous Clock Counter
- 32-Bit Synchronous External Clock Counter
- 32-Bit Gated Timer

The 32-Bit Timer modes are determined by the following bits:

- T32 (TxCON<3>): 32-Bit mode Control Bit (TimerX only)
- TCS (TxCON<1>): Timer Clock Source Control Bit
- TGATE (TxCON<7>): Timer Gate Control Bit

Specific behavior in 32-bit Timer mode:

- TimerX is the master timer; TimerY is the slave timer
- TMRx count register is least significant half word (lshw) of the 32-bit timer value
- TMRy count register is most significant half word (mshw) of the 32-bit timer value
- PRx period register is least significant half word of the 32-bit period value
- PRy period register is most significant half word of the 32-bit period value
- TimerX control bits (TxCON) configure the operation for the 32-bit timer pair
- TimerY control bits (TyCON) have no effect
- TimerX interrupt and Status bits are ignored
- TimerY provides the interrupt enable, interrupt flag and interrupt priority control bits

14.3.2.1 32-Bit Timer Considerations

The following should be considered when using a 32-bit timer:

- Ensure that the timer pair is configured for 32-bit mode by setting T32 (TxCON<3>) = 1, before writing any 32-bit value to the TMRxy count registers or PRxy period registers.
- All timer module SFRs can be written to as a byte (8 bits), a half word (16 bits) or a word (32 bits).
- All timer module SFRs can be read from as a byte, a half word or a word.
- TMRx and TMRy count register pairs can be read as well as written as a single 32-bit value.
- PRx and PRy period register pairs can be read as well as written as a single 32-bit value.

14.3.3 16-Bit Synchronous Clock Counter Mode

The Synchronous Clock Counter operation provides the following capabilities:

- Elapsed time measurements
- Time delays
- Periodic timer interrupts

Type A and B timers have the ability to operate in Synchronous Clock Counter mode. In this mode, the input clock source for the timer is the internal peripheral bus clock, PBCLK, and is selected by clearing the clock source control bit TCS, (TxCON<1>) = 0. Type A and B Timers automatically provide synchronization to the peripheral bus clock; therefore, the Type A Timer Synchronous mode control bit TSYNC (T1CON<2>) is ignored in this mode.

Type A and B timers that use a 1:1 clock prescale operate at a timer clock rate which is the same as the PBCLK, and which increments the TMR count register on every rising timer clock edge. The timer continues to increment until the TMR count register matches the PR period register value. The TMR count register resets to 0000h on the next timer clock cycle, then continues to increment and repeat the period match until the timer is disabled. If the PR period register value = 0000h, the TMR count register resets to 0000h on the next timer clock cycle, but does not continue to increment.

Type A and B timers using a clock prescale = N (other than 1:1) operate at a timer clock rate (PBCLK/N) and the TMR count register increments on every Nth timer clock rising edge. For example, if the clock prescale is 1:8, then the timer increments on every 8th timer clock cycle. The timer continues to increment until the TMR count register matches the PR period register value. The TMR count register then resets to 0000h after N more timer clock cycles, then continues to increment and repeat the period match until the timer is disabled. If the PR period register value = 0000h, the TMR count register resets to 0000h on the next Nth timer clock cycle, but will not continue to increment.

Type A timers generate a timer event one-half timer clock cycle (on the falling edge) after the TMR count register matches the PR period register value. Type B timers generate a timer event within 1 PBCLK + 2 SYSCLK system clock cycles after the TMR count register matches the PR period register value. Both Type A and B timer interrupt flag bits, TxIF, are set within 1 PBCLK + 2 SYSCLK cycles of this event and if the timer interrupt enable bit TxIE is set, an interrupt is generated.

14.3.3.1 16-Bit Synchronous Clock Counter Considerations

This section describes items that should be considered when using a 16-bit Synchronous Clock Counter.

The timer period is determined by the value in the PR period register. To initialize the timer period, a user may write to the PR period register directly at any time while the timer is disabled, ON bit = 0, or during a timer match Interrupt Service Routine (ISR) while the timer is enabled, ON bit = 1. In all other cases, writing to the period register while the timer is enabled is not recommended and may allow unintended period matches to occur.

The maximum period that can be loaded is FFFFh.

Writing 0000h to PRx period register allows a TMRx match to occur; however, no interrupt will be generated.

14.3.4 32-Bit Synchronous Clock Counter Mode (Type B Timer)

Only Type B timers have the ability to operate in 32-bit Synchronous Counter mode. To enable 32-bit Synchronous Clock Counter operation, Type B (TimerX) T32 control bit (TxCON<3>) must be set (= 1). In this mode, the input clock source for the timer is the internal peripheral bus clock, PBCLK, and is selected by clearing the clock source control bit TCS, (TxCON<1>) = 0. Type B timers automatically provide synchronization to the peripheral bus clock.

Type B Timers that use a 1:1 clock prescale operate at a timer clock rate which is the same as the PBCLK, and increments the TMRxy count register on every rising timer clock edge. The timer continues to increment until the TMRxy count register matches the PRxy period register value. The TMRxy count register resets to 00000000h on the next timer clock cycle, then continues to increment and repeat the period match until the timer is disabled. If the PR period register value = 00000000h, the TMR count register resets to 00000000h on the next timer clock cycle, but does not continue to increment.

Type B timers using a clock prescale = N (other than 1:1) operate at a timer clock rate (PBCLK/N) and the TMRxy count register increments on every Nth timer clock rising edge. For example, if the clock prescale is 1:8, then the timer increments on every 8th timer clock cycle. The timer continues to increment until the TMRxy count register matches the PRxy period register value. The TMRxy count register resets to 00000000h after N more timer clock cycles, then continues to increment and repeat the period match until the timer is disabled.

Type B timers generate a timer event within 1 PBCLK + 2 SYSCLK system clock cycles after the TMRxy count register matches the PRxy period register value. The Type B timer interrupt flag bit, TyIF, is set within 1 PBCLK + 2 SYSCLK cycles of this event and if the timer interrupt enable bit TyIE is set, an interrupt is generated.

14.3.4.1 32-Bit Synchronous Clock Counter Considerations

This section describes items that should be considered when using the 32-bit Synchronous Clock Counter.

The timer period is determined by the value in the PRxy period register. To initialize the timer period, a user may write to the PRxy period register directly at any time while the timer is disabled, ON bit = 0, or during a timer match Interrupt Service Routine while the timer is enabled, ON bit = 1. In all other cases, writing to the period register while the timer is enabled is not recommended, and may allow unintended period matches to occur.

The maximum period that can be loaded is FFFFFFFFh.

Writing 00000000h to the PRxy period register will allow a TMRxy match to occur; however, no interrupt is generated.

14.3.4.2 16-Bit Synchronous Counter Initialization Steps

Performed the following steps to configure the timer for 16-bit Synchronous Timer mode.

1. Clear control bit ON (TxCON<15> = 0) to disable timer.
2. Clear control bit TCS (TxCON<1> = 0) to select internal PBCLK source.
3. Select desired clock prescale.
4. Load/Clear timer register TMRx.
5. Load period register PRx with desired 16-bit match value.
6. If interrupts are used:
   i. Clear interrupt flag bit TxIF in IFS0 register.
   ii. Configure interrupt priority and subpriority levels in IPCn register.
   iii. Set interrupt enable bit TxIE in IEC0 registers.
7. Set control bit ON (TxCON<15> = 1) to enable the timer.

Example 14-1: 16-Bit Synchronous Clock Counter Example Code

```
T2CON = 0x0;              // Stop Timer and clear control register,
                          // set prescaler at 1:1, internal clock source
TMR2 = 0x0;               // Clear timer register
PR2 = 0xFFFF;             // Load period register
T2CONSET = 0x8000;        // Start Timer
```

14.3.4.3 32-Bit Synchronous Clock Counter Initialization Steps

Performed the following steps to configure the timer for 32-bit Synchronous Clock Counter mode.

1. Clear control bit ON (TxCON<15> = 0) to disable timer.
2. Clear control bit TCS (TxCON<1> = 0) to select internal PBCLK source.
3. Set control bit T32 (TxCON<3> = 1) to select 32-bit operations.
4. Select desired clock prescale.
5. Load/Clear timer register TMRxy.
6. Load period register PRxy with desired 32-bit match value.
7. If interrupts are used:
   i. Clear interrupt flag bit TyIF in IFSn register.
   ii. Configure interrupt priority and subpriority levels in IPCn register.
   iii. Set interrupt enable bit TyIE in IECn registers.
8. Set control bit ON (TxCON<15> = 1) to enable the timer.

Example 14-2: 32-Bit Synchronous Clock Counter Example Code

```
T4CON = 0x0;              // Stop any 16/32-bit Timer4 operation
T5CON = 0x0;              // Stop any 16-bit Timer5 operation
T4CONSET = 0x0038;        // Enable 32-bit mode, prescaler 1:8,
                          // internal peripheral clock source

TMR4= 0x0;                // Clear contents of the TMR4 and TMR5
PR4 = 0xFFFFFFFF;         // Load PR4 and PR5 registers with 32-bit value

T4CONSET = 0x8000;        // Start Timer45
```


14.3.5 16-Bit Synchronous External Clock Counter Mode

The Synchronous External Clock Counter operation provides the following capabilities:

- Counting periodic or non-periodic pulses
- Use external clock as time base for timers

Type A and B timers have the ability to operate in Synchronous External Clock Counter mode. In this mode, the input clock source for the timer is an external clock applied to the TxCK pin and is selected by setting the clock source control bit TCS (TxCON<1>) = 1. Type B timers automatically provide synchronization for the external clock source; however, the Type A timer does not, and requires the external clock synchronization bit TSYNC (T1CON<2>) be set = 1.

Type A and B timers that use a 1:1 clock prescale increment the TMR count register on every rising external clock edge after synchronization. The timer continues to increment until the TMR count register matches the PR period register value. The TMR count register resets to 0000h on the next timer clock cycle, then continues to increment and repeat the period match until the timer is disabled. If the PR period register value = 0000h, the TMR count register resets to 0000h on the next timer clock cycle, but will not continue to increment.

Type A and B timers using a clock prescale = N (other than 1:1) operate at a timer clock rate (external clock/N) and the TMR count register increments on every Nth external clock rising edge after synchronization. For example, if the clock prescale is 1:8, then the timer increments on every 8th external clock cycle. The timer continues to increment until the TMR count register matches the PR period register value. The TMR count register then resets to 0000h after N more external clock cycles, then continues to increment and repeat the period match until the timer is disabled. If the PR period register value = 0000h, the TMR count register resets to 0000h on the next external clock cycle, but does not continue to increment.

Type A timers generate a timer event one-half timer clock cycle (on the falling edge) after the TMR count register matches the PR period register value. Type B timers generate a timer event within 1 PBCLK + 2 SYSCLK system clock cycles after the TMR count register matches the PR period register value. Both Type A and B timer interrupt flag bits, TxIF, are set within 1 PBCLK + 2 SYSCLK cycles of this event and if the timer interrupt enable bit TxIE is set, an interrupt is generated.

14.3.5.1 16-Bit Synchronous External Clock Counter Considerations

This section describes items that should be considered when using the 16-bit Synchronous External Clock Counter.

Type A or Type B timers operating from a synchronized external clock source will not operate in SLEEP mode, since the synchronization circuit is disabled during SLEEP mode.

Type A and B Timers using a clock prescale = N (other than 1:1) require 2 to 3 external clock cycles, after the ON bit = 1, before the TMR count register increments. Refer to **Section 14.3.12 “Timer Latency Considerations”** for more information.

When operating the timer in Synchronous Counter mode, the external input clock must meet certain minimum high time and low time requirements. Refer to the device data sheet **“Electrical Specifications”** section for further details.

14.3.6 32-Bit Synchronous External Clock Counter Mode

The 32-bit Synchronous External Clock counter operation provides the following capabilities:

- Counting large number of periodic or non-periodic pulses
- Use external clock as large time base for timers

Only Type B timers have the ability to operate in 32-bit Synchronous External Clock Counter mode. To enable 32-bit Synchronous External Clock Counter operation, a Type B (TimerX) T32 control bit (TxCON<3>) must be set = 1. In this mode, the input clock source for the timer is an external clock applied to the TxCK pin and is selected by setting the clock source control bit TCS (TxCON<1>) = 1. Type B timers automatically provide synchronization for the external clock source.

Type B timers that use a 1:1 clock prescale increment the TMRxy count register on every rising external clock edge after synchronization. The timer continues to increment until the TMRxy count register matches the PRxy period register value. The TMRxy count register resets to 00000000h on the next timer clock cycle, then continues to increment and repeat the period match until the timer is disabled. If the PRxy period register value = 0000h, the TMR count register resets to 00000000h on the next timer clock cycle, but does not continue to increment.

Type B timers that use a clock prescale = N (other than 1:1) operate at a timer clock rate (external clock/N) and the TMRxy count register increments on every Nth external clock rising edge after sychronization. For example, if the clock prescale is 1:8, then the timer increments on every 8th external clock cycle. The timer continues to increment until the TMRxy count register matches the PRxy period register value. The TMRxy count register resets to 0000h after N more external clock cycles, then continues to increment and repeat the period match until the timer is disabled. If the PRxy period register value = 00000000h, the TMRxy count register resets to 00000000h on the next external clock cycle, but does not continue to increment.

Type B timers generate a timer event within 1 PBCLK + 2 SYSCLK system clock cycles after the TMRxy count register matches the PRxy period register value. The Type B timer interrupt flag bit, TyIF, is set within 1 PBCLK + 2 SYSCLK cycles of this event and if the timer interrupt enable bit TyIE is set, an interrupt is generated.

14.3.6.1 32-Bit Synchronous External Clock Counter Considerations

This section describes items that should be considered when using the 32-bit Synchronous External Clock Counter.

Type B timers operating from a synchronized external clock source will not operate in SLEEP mode, since the synchronization circuit is disabled during SLEEP mode.

Type B timers using a clock prescale = N (other than 1:1) require 2 to 3 external clock cycles, after the ON bit = 1, before the TMR count register increments. Refer to **Section 14.3.12 “Timer Latency Considerations”**.

When operating the timer in Synchronous Counter mode, the external input clock must meet certain minimum high time and low time requirements. Refer to the device data sheet **“Electrical Specifications”** section for further details.

14.3.6.2 16-Bit Synchronous External Counter Initialization Steps

Perform the following steps to configure the timer for 16-bit Synchronous Counter mode:

1. Clear control bit ON (TxCON<15> = 0) to disable timer.
2. Set control bit TCS (TxCON<1> = 1) to select external clock source.
3. If Type A Timer, set control bit TSYNC (T1CON<2> = 1) to enable clock synchronization.
4. Select desired clock prescale.
5. Load/Clear timer register TMRx.
6. If using period match:
   a. Load period register PRx with desired 16-bit match value.
7. If interrupts are used:
   i. Clear interrupt flag bit TxIF in IFS0 register.
   ii. Configure interrupt priority and subpriority levels in IPCn register.
   iii. Set interrupt enable bit TxIE in IEC0 registers.
8. Set control bit ON (TxCON<15> = 1) to enable the timer.

Example 14-3: 16-Bit Synchronous External Counter Example Code

```
T3CON = 0x0;              // Stop Timer and clear control register
T3CONSET = 0x0072;        // Set prescaler at 1:256, external clock source
TMR3 = 0x0;               // Clear timer register
PR3 = 0x3FFF;             // Load period register
T3CONSET = 0x8000;        // Start Timer
```

14.3.6.3 32-Bit Synchronous External Clock Counter Initialization Steps

Perform the following steps to configure the timer for 32-bit Synchronous External Clock Counter mode:

1. Clear control bit ON (TxCON<15> = 0) to disable timer.
2. Set control bits TCS (TxCON<1> = 1) to select external clock source.
3. Set T32 (TxCON<3> = 1) to enable 32-bit operations.
4. Select desired clock prescale.
5. Load/Clear timer register TMRxy.
6. Load period register PRxy with desired 32-bit match value.
7. If interrupts are used:
   i. Clear interrupt flag bit TyIF in IFSn register.
   ii. Configure interrupt priority and subpriority levels in IPCn register.
   iii. Set interrupt enable bit TyIE in IECn registers.
8. Set control bit ON (TxCON<15> = 1) to enable the timer.

Example 14-4: 32-Bit Synchronous External Clock Counter Example Code

```
T4CON = 0x0;              // Stop any 16/32-bit Timer4 operation
T5CON = 0x0;              // Stop any 16-bit Timer5 operation
T4CONSET = 0x006A;        // 32-bit mode, external clock, 1:64 prescale
TMR4 = 0x0;               // Clear contents of the TMR4 and TMR5

PR4 = 0xFFFFFFFF;         // Load PR4 and PR5 registers with 32-bit value

T4CONSET = 0x8000;        // Start 32-bit timer
```

14.3.7 16-Bit Gated Timer Mode

The gate operation starts on a rising edge of the signal applied to the TxCK pin. The TMRx count register increments while the external gate signal remains high. The gate operation terminates on the falling edge of the signal applied to the TxCK pin. The timer interrupt flag, TxIF, is set.

Both Type A and B timers can operate in Gated Timer mode. The timer clock source is the internal peripheral bus clock, PBCLK, and is selected by clearing the TCS control bit = 0, (TxCON<1>). Type A and B Timers automatically provide synchronization to the peripheral bus clock, therefore the Type A Timer Synchronous mode control bit TSYNC (T1CON<2>) is ignored in this mode. In Gated Timer mode, the input clock is gated by the signal applied to the TxCK pin. The Gated Timer mode is enabled by setting the TGATE control bit = 1, (TxCON<7>).

Type A and B timers using a 1:1 clock prescale operate at a timer clock rate the same as the PBCLK and increment the TMR count register on every rising timer clock edge. The timer continues to increment until the TMR count register matches the PR period register value. The TMR count register then resets to 0000h on the next timer clock cycle, then continues to increment and repeat the period match until the falling edge of the gate signal or the timer is disabled. The timer does not generate an interrupt when a timer period match occurs.

Type A and B timers using a clock prescale = N (other than 1:1) operate at a timer clock rate (PBCLK/N) and the TMR count register increments on every Nth timer clock rising edge. For example, if the clock prescale is 1:8, then the timer increments on every 8th timer clock cycle. The timer continues to increment until the TMR count register matches the PR period register value. The TMR count register then resets to 0000h after N more timer clock cycles, and continues to increment and repeat the period match until the falling edge of the gate signal or the timer is disabled. The timer does not generate an interrupt when a timer period match occurs.

On the falling edge of the gate signal, the count operations terminates, a Timer event is generated and the interrupt flag bit TxIF is set 1 PBCLK + 2 SYSCLK system clock cycles after the falling edge of the signal on the gate pin. The TMR count register is not reset to 0000h. The user must reset the TMR count register if it is desired to start from zero on the next rising edge gate input.

The resolution of the timer count is directly related to the timer clock period. When the timer prescaler is 1:1, the timer clock period is one peripheral bus clock cycle T_{PBCLK}. For a timer prescaler of 1:8, the timer clock period is 8 times the peripheral bus clock cycle.

14.3.7.1 Special Gated Timer Mode Considerations

This section describes items that should be considered when using the special Gated Timer mode.

Gated Timer mode is overridden if the clock source bit TCS is set to external clock source, TCS = 1. For Gated Timer operation, the internal clock source must be selected, TCS = 0.

Type A and B timers using a clock prescale = N (other than 1:1) require 2 to 3 timer clock cycles, after the ON bit = 1, before the TMR count register increments. Refer to **Section 14.3.12 “Timer Latency Considerations”** for more information.

Refer to the **“Electrical Specifications”** section in the device data sheet for details on the gate width pulse requirements.

14.3.8 32-Bit Gated Timer Mode

The gate operation starts on a rising edge of the signal applied to the TxCK pin. The TMRx count register increments while the external gate signal remains high. The gate operation terminates on the falling edge of the signal applied to the TxCK pin. The timer interrupt flag, TyIF, is set.

Only Type B timers can operate in 32-bit Gated Timer mode. The timer clock source is the internal peripheral bus clock, PBCLK, and is selected by clearing the TCS control bit = 0, (TxCON<1>). Type B timers automatically provide synchronization to the peripheral bus clock. In 32-bit Gated Timer mode, the input clock is gated by the signal applied to the TxCK pin. The Gated Timer mode is enabled by setting the TGATE control bit (TxCON<7>) = 1.

The gate operation starts on a rising edge of the signal applied to the TxCK pin and the TMRxy count register increments while the external gate signal remains high.

Type B timers using a 1:1 clock prescale operate at a timer clock rate the same as the PBCLK and increment the TMRxy count register on every rising timer clock edge. The timer continues to increment until the TMRxy count register matches the PRxy period register value. The TMRxy count register then resets to 00000000h on the next timer clock cycle, then continues to increment and repeat the period match until the falling edge of the gate signal or the timer is disabled. The timer does not generate an interrupt when a timer period match occurs.

Type B timers using a clock prescale = N (other than 1:1) operate at a timer clock rate (PBCLK/N) and the TMRxy count register increments on every Nth timer clock rising edge. For example, if the clock prescale is 1:8, then the timer increments on every 8th timer clock cycle. The timer continues to increment until the TMRxy count register matches the PRxy period register value. The TMRxy count register then resets to 00000000h after N more timer clock cycles, then continues to increment and repeat the period match until the falling edge of the gate signal or the timer is disabled. The timer does not generate an interrupt when a timer period match occurs.

On the falling edge of the gate signal, the count operations terminate, a timer event is generated, and the interrupt flag bit TyIF is set 1 PBCLK + 2 SYSCLK system clock cycles after the falling edge of the signal on the gate pin. The TMR count register is not reset to 00000000h. The user must reset the TMRxy count register if it is desired to start from zero on the next rising edge gate input.

The resolution of the timer count is directly related to the timer clock period. When the timer prescaler is 1:1, the timer clock period is 1 PBCLK peripheral bus clock cycle. For a timer prescaler of 1:8, the timer clock period is 8 times the peripheral bus clock cycle.

14.3.8.1 32-Bit Gated Timer Mode Considerations

This section describes items that should be considered when using the 32-bit Gated Timer mode.

Gated Timer mode is overridden if the clock source bit TCS is set to external clock source, TCS = 1. For Gated Timer operation, the internal clock source must be selected, TCS = 0.

Refer to the **"Electrical Specifications"** section in the device data sheet for details on the gate width pulse requirements.

14.3.8.2 16-Bit Gated Timer Initialization Steps

Perform the following steps to configure the timer for 16-bit Gated Timer mode:

1. Clear control bit ON (TxCON<15> = 0) to disable timer.
2. Set control bits TCS (TxCON<1> = 0) to select internal PBCLK source.
3. Set control bit TGATE (T1CON<7> = 1) to enable gated Timer mode.
4. Select desired prescaler.
5. Clear timer register TMRx.
6. Load period register PRx with desired 16-bit match value.
7. If interrupts are used:
   i. Clear interrupt flag bit TxIF in IFS0 register.
   ii. Configure interrupt priority and subpriority levels in IPCn register.
   iii. Set Interrupt enable bit TxIE in IEC0 registers.
8. Set control bit ON (TxCON<15> = 1) to enable the timer.

Example 14-5: 16-Bit Gated Timer Example Code

```
T4CON = 0x0;              // Stop Timer and clear control register
T4CON = 0x00E0;           // Gated timer mode, prescaler at 1:64, internal clock source
TMR4 = 0;                 // Clear timer register
PR4 = 0xFFFF;             // Load period register with 16-bit match value
T4CONSET = 0x8000;        // Start Timer
```

14.3.8.3 32-Bit Gated Timer Initialization Steps

Perform the following steps to configure the timer for 32-bit Gated Timer Accumulation mode:

1. Clear control bit ON (TxCON<15> = 0) to disable Timer.
2. Clear control bit TCS (TxCON<1>) = 0 to select internal PBCLK source.
3. Set control bit T32 (TxCON<3>= 1) = 1 to enable 32-bit operations.
4. Set control bit TGATE (TxCON<7> = 1) to enable gated Timer mode.
5. Select desired clock prescale.
6. Load/Clear timer register TMRx.
7. Load period register PRx with desired 32-bit match value.
8. If interrupts are used:
   i. Clear interrupt flag bit TyIF in IFSn register.
   ii. Configure interrupt priority and subpriority levels in IPCn register.
   iii. Set interrupt enable bit TyIE in IECn registers.

Set control bit ON (TxCON<15> = 1) to enable the timer.

Example 14-6: 32-Bit Gated Timer Example Code

```
T2CON = 0x0;              // Stops any 16/32-bit Timer2 operation
T3CON = 0x0;              // Stops any 16-bit Timer3 operation
T2CONSET = 0x00C8;        // 32-bit mode, gate enable, internal clock,
                          // 1:16 prescale
TMR2 = 0x0;               // Clear contents of the TMR2 and TMR3

PR2 = 0xFFFFFFFF;         // Load PR2 and PR3 registers with 32-bit match value

T2CONSET = 0x8000;        // Start 32-bit timer
```


14.3.9 Asynchronous Clock Counter Mode (Type A Timer Only)

The Asynchronous Timer operation provides the following capabilities:

- The timer can operate during SLEEP mode and can generate an interrupt on period register match that will wake-up the processor from SLEEP or IDLE mode.
- The timer can be clocked from the Secondary Oscillator for real-time clock applications.

The Type A timer has the ability to operate in an Asynchronous Counting mode, using an external clock source connected to the T1CK pin, and is selected by setting the clock source control bit TCS (TxCON<1>) = 1. This requires the external clock synchronization be disabled, bit TSYNC (T1CON<2>) = 0. It is also possible to utilize the Secondary Oscillator with a 32 kHz crystal connected to SOSCI/SOSCO pins as an asynchronous clock source. Refer to **Section 14.3.13 "Secondary Oscillator"** for more information.

Type A timer using a 1:1 clock prescale operates at the same clock rate as the applied external clock rate, and increments the TMR count register on every rising timer clock edge. The timer continues to increment until the TMR count register matches the PR period register value. The TMR count register resets to 0000h on the next timer clock cycle, then continues to increment and repeat the period match until the timer is disabled. If the PR period register value = 0000h, the TMR count register resets to 0000h on the next timer clock cycle, but will not continue to increment.

Type A timers generate a timer event when the TMR count register matches the PR period register value. The timer interrupt flag bit, TxIF is set within 1 PBCLK + 2 SYSCLK system clock cycles of this event. If the timer interrupt enable bit is set, TxIE = 1, an interrupt is generated.

14.3.9.1 Asynchronous Mode TMR1 Read and Write Operations

Due to the asynchronous nature of Timer1 operating in this mode, reading and writing to the TMR1 count register requires synchronization between the asynchronous clock source and the internal PBCLK peripheral bus clock. Timer1 features a TWDIS (Timer Write Disable) control bit (T1CON<12>) and a TWIP (TImer Write in Progress) Status bit (T1CON<11>) to provide the user with 2 options for safely writing to the TMR1 count register while Timer1 is enabled. These bits have no affect in Synchronous Clock Counter modes.

Option 1 is the legacy Timer1 Write mode, TWDIS bit = 0. To determine when it is safe to write to the TMR1 count register, it is recommended to poll the TWIP bit. When TWIP = 0, it is safe to perform the next write operation to the TMR1 count register. When TWIP = 1, the previous Write operation to the TMR1 count register is still being synchronized and any additional write operations should wait until TWIP = 0.

Option 2 is the new synchronized Timer1 Write mode, TWDIS bit = 1. A write to the TMR1 count register can be performed at any time. However, if the previous write operation to the TMR1 count register is still being synchronized, any additional write operations are ignored.

When performing a write to the TMR1 count register, 2 to 3 asynchronous external clock cycles are required for the value to be synchronized into the register.

When performing a read from the TMR1 count register, synchronization requires 2 PBCLK cycle delays between the current unsynchronized value in the TMR1 count register and the synchronized value returned by the read operation. In other words, the value read is always 2 PBCLK cycles behind the actual value in the TMR1 count register.

14.3.9.2 Asynchronous Clock Counter Considerations

This section describes items that should be considered when using the Asynchronous Clock Counter.

Regardless of the clock prescale, Type A timers require 2 to 3 timer clock cycles, after the ON bit = 1 before the TMR count register increments. Refer to **Section 14.3.12 "Timer Latency Considerations"** for more information.

The external input clock must meet certain minimum high time and low time requirements when used in the Asynchronous Counter mode. Refer to the device data sheet **"Electrical Specifications"** section for further details.

14.3.9.3 Asynchronous External Clock Counter Initialization Steps

Perform the following steps to configure the Timer for 16-bit Asynchronous Counter mode.

1. Clear control bit ON (T1CON<15> = 0) to disable timer.
2. Set control bit TCS (T1CON<1> = 1) to enable external clock source.
3. Clear control bit TSYNC (T1CON<2> = 0) to disable clock synchronization.
4. Select desired prescaler.
5. Load/Clear timer register TMR1.
6. If using period match:
   i. Load period register PR1 with desired 16-bit match value.
7. If interrupts are used:
   i. Clear interrupt flag bit T1IF in IFS0 register.
   ii. Configure interrupt priority and subpriority levels in IPCn register.
   iii. Set interrupt enable bit T1IE in IEC0 registers.
8. Set control bit ON (T1CON<15> = 1) to enable the timer.

Example 14-7: Example Code: 16-Bit Asynchronous Counter Mode

```
/*
    16-bit asynchronous counter mode example
*/
    T1CON = 0x0;           // Stops the Timer1 and reset control reg.
    T1CON = 0x0042;        // Set prescaler 1:16, external clock, asynch mode

    TMR1 = 0x0;            // Clear timer register
    PR1 = 0x7FFF;          // Load period register
    T1CONSET = 0x8000;     // Start Timer
```


14.3.10 Timer Prescalers

Type A timers provide input clock (peripheral bus clock or external clock) prescale options of 1:1, 1:8, 1:64 and 1:256 selected using TCKPS<1:0> (TxCON<5:4>).

Type B timers provide input clock (peripheral bus clock or external clock) prescale options of 1:1, 1:2, 1:4, 1:8, 1:16, 1:32, 1:64 and 1:256 selected using TCKPS<2:0> (TxCON<6:4>).

The prescaler counter is cleared when any of the following occurs:

- A write to the TMRx register
- Disabling the timer, ON (TxCON<15>) = 0
- Any device Reset, except Power-on Reset

14.3.11 Writing to TxCON, TMR, and PR Registers

A timer module is disabled and powered off when the ON bit (TxCON<15>) = 0, thus providing maximum power savings.

To prevent unpredictable timer behavior, it is recommended that the timer be disabled, ON bit = 0, before writing to any of the TxCON register bits or timer prescaler. Attempting to set ON bit = 1 and write to any TxCON register bits in the same instruction may cause erroneous timer operation.

The PRx period register can be written to while the module is operating. However, to prevent unintended period matches, writing to the PRx period register while the timer is enabled, (ON bit = 1)is not recommended.

The TMRx count 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.

Additionally, TMR1 count register can be written to while the module is operating. However, see **Section 14.3.9.1 "Reading and Writing TMR1 register"** regarding asynchronous clock operations.

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

A TMR count register is not reset to zero when the module is disabled.

14.3.12 Timer Latency Considerations

This section describes items that should be considered regarding timer latency.

Since both Type A and Type B timers can use the Internal Peripheral Bus Clock (PBCLK) or an external clock (Type A also supports asynchronous clock), there are considerations regarding latencies of operations performed on the timer. These latencies represent the time delay between the moment an operation is executed (read or write) and the moment its first effect begins, as shown in Table 14-3 and Table 14-4.

For Type A and Type B timers, reading and writing the TxCON, TMRx, and PRx registers in any Synchronized Clock mode does not require synchronization of data between the main SYSCLK clock domain and the timer module clock domain. Therefore, the operation is immediate. However, when operating Timer1 in Asynchronous Clock mode, reading the TMR1 count register requires 2 PBCLK cycles for synchronization, while writing to theTMR1 count register requires 2 to 3 timer clock cycles for synchronization.

For example, Timer1 is using an asynchronous clock source and a read operation of TMR1 register is executed. There are 2 PBCLK peripheral bus clocks required to synchronize this data to the TMR1 count register. The effect is a value which is always 2 PBCLK cycles behind the actual TMR1 count.

Additionally, any timer using an external clock source requires 2-3 external clock cycles, after the ON bit (TxCON<15>) has been set (= 1), before the timer starts incrementing.

The interrupt flag latency represents the time delay between the timer event and the moment the timer interrupt flag is active.

Table 14-3: Type A Timer Latencies

Operation	PBCLK Internal clock	Synchronous External clock	Asynchronous External clock
Set ON = 1 (enable timer)	0 PBCLK	2-3 $TMRCLK_{CY}$	2-3 $TMRCLK_{CY}$
Set ON = 0 (disable timer)	0 PBCLK	2-3 $TMRCLK_{CY}$	2-3 $TMRCLK_{CY}$
Read PRx	0 PBCLK	0 PBCLK	0 PBCLK
Write PRx	0 PBCLK	0 PBCLK	0 PBCLK
Read TMRx	0 PBCLK	0 PBCLK	2 PBCLK
Write TMRx	0 PBCLK	0 PBCLK	2-3 $TMRCLK_{CY}$
Interrupt Flag INTF = 1	1 PBCLK + 2 to 3 SYSCLK	1 PBCLK + 2 to 3 SYSCLK	($TMRCLK_{CY}$ / 2) + 2 to 3 SYSCLK

Note: $TMRCLK_{CY}$ = External synchronous or asynchronous timer clock cycles.

Table 14-4: Type B Timer Latencies

Operation	PBCLK Internal clock	Synchronous External clock
Set ON = 1 (enable timer)	0 PBCLK	0 PBCLK
Set ON = 0 (disable timer)	0 PBCLK	0 PBCLK
Read PRx	0 PBCLK	0 PBCLK
Write PRx	0 PBCLK	0 PBCLK
Read TMRx	0 PBCLK	0 PBCLK
Write TMRx	0 PBCLK	0 PBCLK
Interrupt Flag INTF = 1	1 PBCLK + 2 to 3 SYSCLKs	1 PBCLK + 2 to 3 SYSCLKs

14.3.13 Secondary Oscillator

In each device variant, the secondary oscillator is available to the Type A timer module for Real-Time Clock (RTC) applications.

- The secondary oscillator becomes the clock source for the timer when the secondary oscillator is enabled and the timer is configured to use the external clock source.
- The secondary oscillator is enabled when the Configuration Fuse bit FSOSCEN (DEVCFG1<5>) = 0 and by setting the SOSCEN control bit (OSCCON<1>).

Refer to **Section 6. "Oscillators"** for further details.

14.4 INTERRUPTS

A 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 (TyIF bit in 32-bit mode) is set when one of the following conditions is true:

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

The TxIF bit (TyIF bit in 32-bit mode) must be cleared in software.

A timer is enabled as a source of interrupt via the respective timer interrupt enable bit, TxIE (TyIE for 32-bit mode). The interrupt priority level bits TxIP<2:0> (TyIP<2:0> for 32-bit mode) and interrupt subpriority level bits TxIS<1:0> (TyIS<1:0> for 32-bit mode) also must be configured. Refer to **Section 8. "Interrupts"** in this manual for further details.

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

14.4.1 Interrupt Configuration

Each Time Base module has a dedicated interrupt flag bit TxIF and a corresponding interrupt enable/mask bit TxIE. These bits determine the source of an interrupt, and enable or disable an individual interrupt source. Each Timer module can have its own priority level independent of other Timer modules.

The TxIF is set when the timer count matches the respective period register and the timer module is not operating in Gated Time Accumulation mode, or when the falling edge of the gate signal is detected when the timer is operating in Gated Time Accumulation mode. The TxIF bit is set without regard to the state of the corresponding TxIE bit. The TxIF bit can be polled by software if desired.

The TxIE bit is used to define the behavior of the Interrupt Controller when a corresponding TxIF is set. When the TxIE bit is clear, the Interrupt Controller does not generate a CPU interrupt for the event. If the TxIE bit is set, the Interrupt Controller will generate an interrupt to the CPU when the corresponding TxIF bit is set (subject to the priority and subpriority as outlined below).

It is the responsibility of the user's software routine that services a particular interrupt to clear the appropriate Interrupt Flag bit before the service routine is complete.

The priority of each timer module can be set independently with the TxIP<2:0> bits. This priority defines the priority group to which the interrupt source will be assigned. The priority groups range from a value of 7 (the highest priority) to a value of 0 (which does not generate an interrupt). An interrupt being serviced will be preempted by an interrupt in a higher priority group.

The subpriority bits allow setting the priority of a interrupt source within a priority group. The values of the subpriority, TxIS<1:0>, range from 3 (the highest priority), to 0 (the lowest priority). An interrupt with the same priority group, but having a higher subpriority value, will preempt a lower subpriority interrupt that is in progress.

The priority group and subpriority bits allow more than one interrupt source to share the same priority and subpriority. If simultaneous interrupts occur in this configuration, the natural order of the interrupt sources within a priority/subgroup pair determines the interrupt generated. The natural priority is based on the vector numbers of the interrupt sources. The lower the vector number, the higher the natural priority of the interrupt. Any interrupts that were overridden by natural order will then generate their respective interrupts based on priority, subpriority, and natural order after the interrupt flag for the current interrupt is cleared.

After an enabled interrupt is generated, the CPU will jump to the vector assigned to that interrupt. The vector number for the interrupt is the same as the natural order number. The CPU will then begin executing code at the vector address. The user's code at this vector address should perform any application specific operations and clear the TxIF interrupt flag, and then exit. Refer to **Section 8. "Interrupts"** for the vector address table details for more information on interrupts.

Table 14-5: Timer Interrupt Vectors for Various Offsets with EBASE = 0x8000:0000

Interrupt	Vector/Natural Order	IRQ Number	Vector Address IntCtl.VS = 0x01	Vector Address IntCtl.VS = 0x02	Vector Address IntCtl.VS = 0x04	Vector Address IntCtl.VS = 0x08	Vector Address IntCtl.VS = 0x10
Timer1	4	4	8000_0280	8000 0300	8000_0400	8000_0600	8000 0A00
Timer2	8	8	8000_0300	8000_0400	8000_0600	8000_0A00	8000 1200
Timer3	12	12	8000_0380	8000_0500	8000_0800	8000_0E00	8000 1A00
Timer4	16	16	8000_0400	8000_0600	8000_0A00	8000 1200	8000 2200
Timer5	20	20	8000_0480	8000_0700	8000_0C00	8000 1600	8000 2A00

Table 14-6: Example of Priority and Subpriority Assignment

Interrupt	Priority Group	Subpriority	Vector/Natural Order
Timer1	7	3	4
Timer2	7	3	8
Timer3	7	2	12
Timer4	6	1	16
Timer5	0	3	20

Example 14-8: 16-Bit Timer Interrupt Initialization Code Example

```
/*
    The following code example will enable Timer2 interrupts, load the Timer2 Period
    register and start the Timer.

    When a Timer2 period match interrupt occurs, the interrupt service routine must clear
    the Timer2 interrupt status flag in software.
*/
    T2CON = 0x0;           // Stop Timer and clear control register,
                           // prescaler at 1:1,internal clock source

    TMR2 = 0x0;            // Clear timer register
    PR2 = 0xFFFF;          // Load period register

    IPC2SET = 0x0000000C;  // Set priority level=3
    IPC2SET = 0x00000001;  // Set sub-priority level=1
                           // Could have also done this in single
                           // operation by assigning IPC2SET = 0x0000000D

    IFS0CLR = 0x00000100;  // Clear Timer interrupt status flag
    IEC0SET = 0x00000100;  // Enable Timer interrupts

    T2CONSET = 0x8000;     // Start Timer
```

Example 14-9: Timer ISR Code Example

```
/*
    The following code example demonstrates a simple interrupt service routine for Timer
    interrupts. The user's code at this ISR handler should perform any application
    specific operations and must clear the corresponding Timer interrupt status flag
    before exiting.
*/
void __ISR(_Timer_1_Vector,ipl3)Timer1Handler(void)
{
    ... perform application specific operations in response to the interrupt

    IFS0CLR = 0x00000010;  // Be sure to clear the Timer 2 interrupt status
}
```

Note: The Timer ISR code example shows MPLAB® C32 C-compiler specific syntax. Refer to your compiler manual regarding support for ISRs.

Example 14-10: 32-bit Timer Interrupt Initialization Code Example

```
/*
    The following code example will enable Timer5 interrupts, load the Timer4:Timer5 Period
    Register pair and start the 32-bit timer module.

    When a 32-bit period match interrupt occurs, the user must clear the Timer5 interrupt
    status flag in software.
 */

    T4CON = 0x0;           // Stop 16-bit Timer4 and clear control register
    T5CON = 0x0;           // Stop 16-bit Timer5 and clear control register
    T4CONSET = 0x0038;     // Enable 32-bit mode, prescaler at 1:8,
                           // internal clock source

    TMR4 = 0x0;            // Clear contents of the TMR4 and TMR5

    PR4 = 0xFFFFFFFF;      // Load PR4 and PR5 registers with 32-bit value

    IPC5SET = 0x00000004;  // Set priority level=1 and
    IPC5SET = 0x00000001;  // Set sub-priority level=1
                           // Could have also done this in single
                           // operation by assigning IPC5SET = 0x00000005

    IFS0CLR = 0x00100000;  // Clear the Timer5 interrupt status flag
    IEC0SET = 0x00100000;  // Enable Timer5 interrupts

    T4CONSET = 0x8000;     // Start Timer
```


14.5 OPERATION IN POWER-SAVING AND DEBUG MODES

Note: In this manual, a distinction is made between a power mode as it is used in a specific module, and a power mode as it is used by the device, e.g., Sleep mode of the Comparator and SLEEP mode of the CPU. To indicate which type of power mode is intended, uppercase and lowercase letters (Sleep, Idle, Debug) signify a module power mode, and all uppercase letters (SLEEP, IDLE, DEBUG) signify a device power mode.

14.5.1 Timer Operation in SLEEP Mode

As the device enters SLEEP mode, the system clock SYSCLK and peripheral bus clock PBCLK are disabled. For both timer types (A and B) operating in Synchronous mode, the timer module stops operating.

Type A timer module is different from the Type B timer module because it can operate asynchronously from an external clock source. Because of this distinction, the Type A timer module can continue to operate during SLEEP mode.

To operate in SLEEP mode, Type A timer module must be configured as follows:

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

When all of the preceding conditions are met, Timer1 continues 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 T1IF Status bit is set. If the T1IE bit is set, and its priority is greater than current CPU priority, the device wakes from SLEEP or IDLE mode and executes the Timer1 Interrupt Service Routine.

If the assigned priority level of the Timer1 interrupt is less than, or equal to, the current CPU priority level, the CPU is not awakened and the device enters IDLE mode.

14.5.2 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 SIDL bit (TxCON<13>) selects whether the timer module stops in IDLE mode, or continues to operate normally. If TSIDL = 0, the module continues operation in IDLE mode. If SIDL = 1, the module stops in IDLE mode.

14.5.3 Timer Operation in DEBUG Mode

The FRZ bit (TxCON<14>) determines whether the timer module will run or stop while the CPU is executing debug exception code (i.e., the application is halted) in DEBUG mode. When FRZ = 0, the timer module continues to run, even when application is halted in DEBUG mode. When FRZ = 1 and the application is halted in DEBUG mode, the module freezes its operations and makes no changes to the state of the timer module. The module will resume its operation after the CPU resumes execution.

Note: The FRZ bit is readable and writable only when the CPU is executing in DEBUG mode. In all other modes, FRZ reads as '0'. If the FRZ bit is changed during DEBUG mode, the new value does not take effect until the current DEBUG mode is exited and reentered. During DEBUG mode, FRZ reads the last written value, which may or may not be in effect (depending on when the last value was written).

14.6 EFFECTS OF VARIOUS RESETS

14.6.1 Device Reset

All timer registers are forced to their reset states upon a device Reset.

14.6.2 Power-on Reset

All timer registers are forced to their reset states upon a Power-on Reset.

14.6.3 Watchdog Reset

All timer registers are forced to their reset states on a Watchdog Reset.

14.7 PERIPHERALS USING TIMER MODULES

14.7.1 Time Base for Input Capture/Output Compare

The Input Capture and Output Compare peripherals can select one of two timer modules or a combined 32-bit timer as their timer source. Refer to the device data sheet, and to **Section 15. "Input Capture"** and **Section 16. "Output Compare"** in this manual for details.

14.7.2 A/D Special Event Trigger

On each device variant, a Type B Timer3 or Timer5 has the capability to generate a special A/D conversion trigger signal on a period match in both 16-bit 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), a 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 the device data sheet, and to **Section 17. "10-Bit A/D Converter"** in this manual for additional information.

14.8 I/O PIN CONTROL

Enabling the timer module does not configure the I/O pin direction. 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 as an input by setting the corresponding TRIS control register bit = 1.

On PIC32MX devices, the TxCK pins become the gate inputs when Gated Timer mode is selected, TGATE bit (TxCON<7>) = 1, and internal peripheral bus clock source PBCLK, TCS bit (TxCON<1>) = 0, are selected. The TxCK pins can be external clock inputs for other modes when the external clock source TCS (TxCON<1>) = 1 is selected. If not used as a gate or external clock input, these pins can be general purpose I/O pins.

14.8.1 I/O Pin Resources

A summary of timer/counter modes, and the specific I/O pins required for each mode is provided in Table 14-7. The table illustrates which I/O pin is required for a certain mode of operation.

Refer to Table 14-8 to configure the I/O pins.

Table 14-7: Required I/O Pin Resources

	16/32-Bit Timer Modes			16/32-Bit Counter Modes
I/O Pin Name	**Internal Clock Source(1)**	**External Clock Source**	**Gate For Internal Clock Source**	**External Clock Source**
T1CK	No	Yes	Yes	Yes
T2CK	No	Yes	Yes	Yes
T3CK	No	Yes	Yes	Yes
T4CK	No	Yes	Yes	Yes
T5CK	No	Yes	Yes	Yes

Note 1: “No” indicates the pin is not required and can be used as a general purpose I/O pin.

14.8.2 I/O Pin Configuration

Table 14-8 provides a summary of I/O pin resources associated with the timer modules. The table also shows the settings required to make each I/O pin work with a specific timer module.

Table 14-8: I/O Pin Configuration for Use with Timer Modules

		Required Settings for Module Pin Control					
I/O Pin Name	**Required(1)**	**Module Control**	**Bit Field**	**TRIS**	**Pin Type**	**Buffer Type**	**Description**
T1CK	No	ON	TCS,TGATE	Input	I	ST	Timer 1 External Clock/Gate Input
T2CK	No	ON	TCS,TGATE	Input	I	ST	Timer 2 External Clock/Gate Input
T3CK	No	ON	TCS,TGATE	Input	I	ST	Timer 3 External Clock/Gate Input
T4CK	No	ON	TCS,TGATE	Input	I	ST	Timer 4 External Clock/Gate Input
T5CK	No	ON	TCS,TGATE	Input	I	ST	Timer 5 External Clock/Gate Input

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

Note 1: These pins are only required for modes that use gated timer or external clock inputs. Otherwise, these pins can be used for general purpose I/O and require the user to set the corresponding TRIS control register bits.

14.9 FREQUENTLY ASKED QUESTIONS

Question 1: ***Can the lower half of the 32-bit timer generate an interrupt?***

Answer: No. When two 16-bit timers are combined in 32-bit mode (TxCON<TGATE> = 1), the interrupt enable bit TxIE, interrupt flag bit TxIF, interrupt priority bit TxIP, and interrupt subpriority bit TxIS associated with the upper timer module are used. The interrupt functions of the lower timer module are disabled.

Question 2: ***If I do not use the TxCK input for my timer mode, is this I/O pin available as a general purpose I/O pin?***

Answer: Yes. If the timer module is configured to use an internal clock source (TxCON<TCS = 0>) and not use the Gated Timer mode (TxCON<TGATE> = 0), then the associated I/O pin is available for general purpose I/O. Note, though, that when the I/O pin is used as a general purpose I/O pin, the user is responsible for configuring the respective TRIS register to input or output.

14.10 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 PIC32MX device 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 #
No related application notes at this time.	N/A

Note: Please visit the Microchip web site (www.microchip.com) for additional application notes and code examples for the PIC32MX family of devices.

14.11 REVISION HISTORY

Revision A (August 2007)

This is the initial released version of this document.

Revision B (October 2007)

Updated document to remove Confidential status.

Revision C (April 2008)

Revised status to Preliminary; Revised U-0 to r-x; Revised Table 14-2; Revised Register 14-1; Revised Section 14.3.9.1

Revision D (May 2008)

Added note to Registers 14-17, 14-18, 14-19, 14-20, 14-21, 14-22, 14-23; Revised Tables 14-1, 14-5; Revised Examples 14-9, 14-10; Revised Section 14.3.9.1 Title; Revised Section 14.3.11; Change Reserved bits from “Maintain as” to “Write”; Added Note to ON bit (T1CON, TxCON Registers).

NOTES:

