TMS320C6457 DSP
64-Bit Timer

User's Guide

![Texas Instruments logo]

![Texas Instruments logo]

Contents

List of Figures

List of Tables

Read This First

About This Manual

This document provides an overview of the 64-bit timer in the TMS320C6457 DSP. The timer can be configured as a general-purpose 64-bit timer, dual general-purpose 32-bit timers, or a watchdog timer. When configured as a dual 32-bit timers, each half can operate in conjunction (chain mode) or independently (unchained mode) of each other.

Notational Conventions

This document uses the following conventions.

- Hexadecimal numbers are shown with the suffix h. For example, the following number represents 40 hexadecimal (decimal 64): 40h.
- Registers in this document are shown in figures and described in tables.
  - Each register figure shows a rectangle divided into fields that represent the fields of the register. Each field is labeled with its bit name, its beginning and ending bit numbers above, and its read/write properties below. A legend explains the notation used for the properties.
  - Reserved bits in a register figure designate a bit that is used for future device expansion.

Related Documentation From Texas Instruments

The following documents describe the C6000™ devices and related support tools. Copies of these documents are available on the Internet at www.ti.com. *Tip:* Enter the literature number in the search box provided at www.ti.com.

SPRU189 — *TMS320C6000 DSP CPU and Instruction Set Reference Guide.* Describes the CPU architecture, pipeline, instruction set, and interrupts for the TMS320C6000 digital signal processors (DSPs).

SPRU198 — *TMS320C6000 Programmer's Guide.* Describes ways to optimize C and assembly code for the TMS320C6000™ DSPs and includes application program examples.

SPRU301 — *TMS320C6000 Code Composer Studio Tutorial.* Introduces the Code Composer Studio™ integrated development environment and software tools.

SPRU321 — *Code Composer Studio Application Programming Interface Reference Guide.* Describes the Code Composer Studio™ application programming interface (API), which allows you to program custom plug-ins for Code Composer.

SPRU871 — *TMS320C64x+ Megamodule Reference Guide.* Describes the TMS320C64x+ digital signal processor (DSP) megamodule. Included is a discussion on the internal direct memory access (IDMA) controller, the interrupt controller, the power-down controller, memory protection, bandwidth management, and the memory and cache.

Trademarks

C6000, TMS320C6000, Code Composer Studio are trademarks of Texas Instruments.

All other trademarks are the property of their respective owners.

TMS320C6457 64-Bit Timer

This document provides an overview of the 64-bit timer in the TMS320C6457 (C6457) DSP. The timer can be configured as a general-purpose 64-bit timer, dual general-purpose 32-bit timers, or a watchdog timer. When configured as a dual 32-bit timers, each half can operate in conjunction (chain mode) or independently (unchained mode) of each other.

1 Introduction to the Timer

The timer can be configured in one of three modes using the timer mode (TIMMODE) bits in the timer global control register (TGCR): a 64-bit general-purpose (GP) timer, dual 32-bit timers (TIMLO and TIMHI), or a watchdog timer. When configured as dual 32-bit timers, each half can operate dependently (chain mode) or independently (unchained mode) of each other.

At reset, the timer is configured as a 64-bit GP timer. The watchdog timer function can be enabled if desired, via the TIMMODE bits in timer global control register (TGCR) and WDEN bit in the watchdog timer control register (WDTCR). Once the timer is configured as a watchdog timer, it cannot be re-configured as a regular timer until a device reset occurs.

The timer has one input pin (TINPL) and one output pin (TOUTL). The timer control register (TCR) controls the function of the input and output pin.

Figure 1 shows a high-level block diagram of the timer circuitry.

Figure 1. Timer Block Diagram
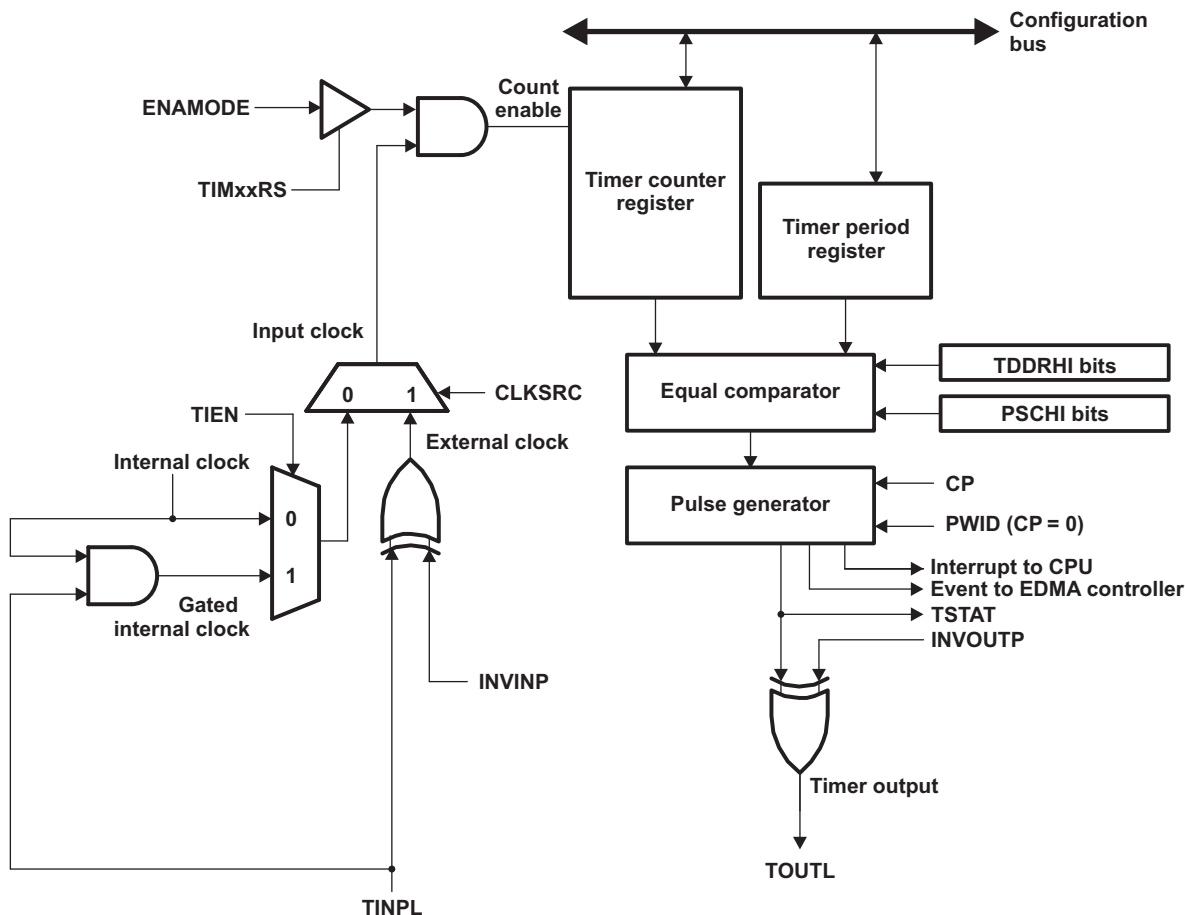
The timer can be driven by an external clock at the timer input pin (TINPL) or by the divide-down clock rate of the internal clock, as shown in Figure 2. The internal clock is generated by the PLL1 controller and is a divided-down version of the CPU clock. For more information, see the *TMS320C6457 Fixed-Point Digital Signal Processor* data manual (SPRS582).

Figure 2. Generation of the Internal Timer Clock
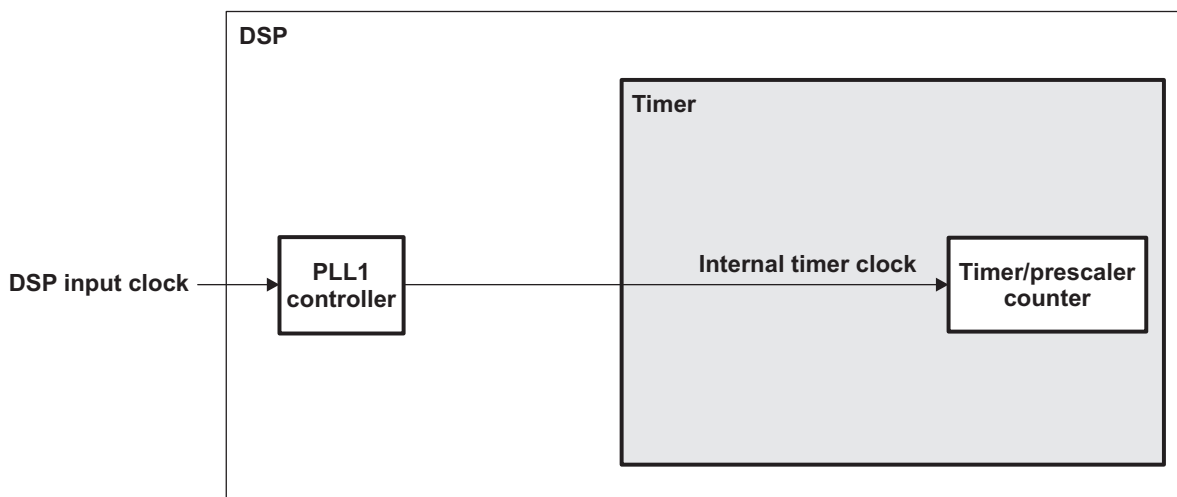
2 Timer Modes

2.1 64-Bit Timer Mode

The timer can be configured as a 64-bit general-purpose (GP) timer, using the TIMMODE bits in TGCR register. At reset, the timer is in 64-bit GP timer mode.

In this mode, the timer operates as a 64-bit up-counter, as shown in Figure 3. The counter registers (CNTLO, CNTHI) and the period registers (PRDLO, PRDHI) form a 64-bit timer counter register and a 64-bit timer period register, respectively. When the timer is enabled (see Section 3.3), the timer counter starts incrementing by 1 at every timer input clock cycle. When the timer counter matches the timer period, it generates a maskable timer interrupt (TINTLO), a timer event (TEVTLO), and an output signal on the timer output pin, TOUTL. When in pulse mode (CP_LO = 0), the timer output pin (TOUTL) asserts a pulse that is 1, 2, 3, or 4 timer clock cycles wide, depending on the setting of the pulse width (PWID_LO) bits in timer control register TCR. When the timer is configured in continuous mode, the timer counter is reset to 0 on the cycle after the timer counter reaches the timer period. The timer can be stopped, restarted, reset, or disabled using the bits of the timer control register.

Figure 3. 64-Bit Timer Mode Block Diagram
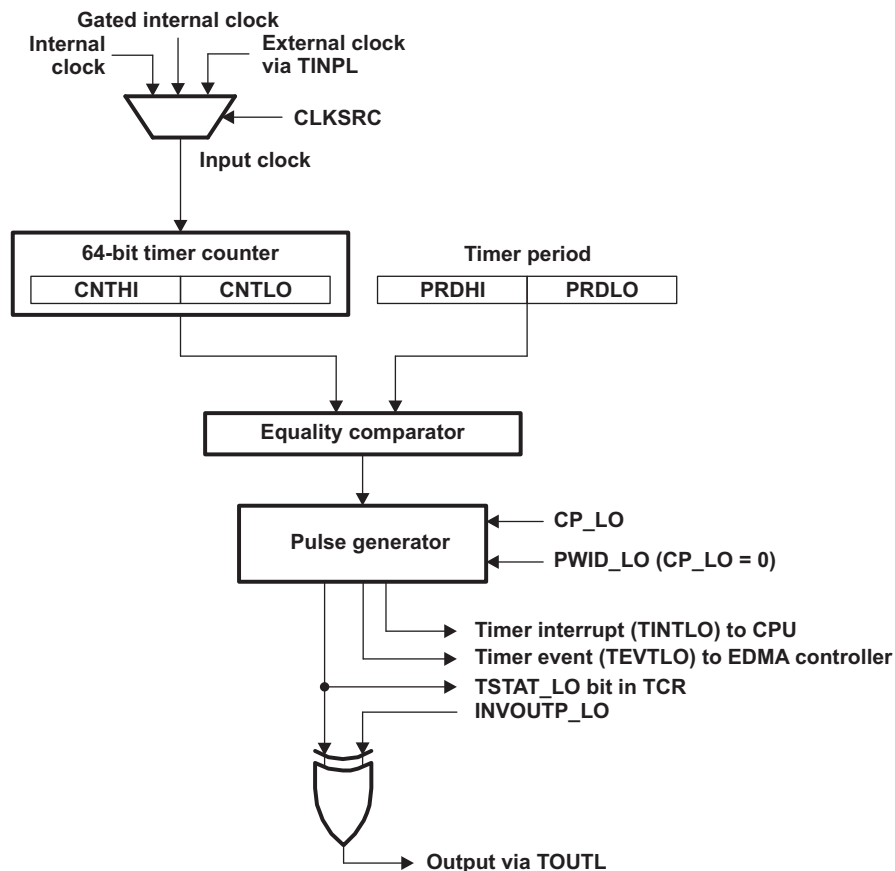
2.2 Dual 32-Bit Timer Modes

The timer can be broken down into two 32-bit timers, using the TIMMODE bits in TGCR. In this mode, the two 32-bit timers can be operated in conjunction with each other (chained mode) or independently (unchained mode).

2.2.1 Chained Mode

In the chained mode, shown in Figure 4, one 32-bit timer (TIMHI) is used as a 32-bit prescaler to a second timer (TIMLO).

The 32-bit prescaler (TIMHI) uses the counter register (CNTHI) and the period register (PRDHI) to form a 32-bit prescale counter register and a 32-bit prescale period register, respectively. When the timer is enabled, the prescale counter starts incrementing by 1 at every timer input clock cycle. One cycle after the prescale counter matches the prescale period, a clock signal is generated and the prescale counter register is reset to 0 (see the example in Figure 5).

The other 32-bit timer (TIMLO) uses the counter register (CNTLO) and the period register (PRDLO) to form a 32-bit timer counter register and a 32-bit timer period register, respectively. This timer is clocked by the output clock from the prescaler (see the example in Figure 5). The timer counter increments by 1 at every prescaler output clock cycle. When the timer counter matches the timer period, a maskable timer interrupt (TINTLO), a timer EDMA event (TEVTLO), and an output signal are generated. When in pulse mode (CP_LO = 0), the timer output (TOUTL) asserts a pulse that is 1, 2, 3, or 4 timer clock cycles wide, depending on the setting of the pulse width (PWID) bits in the timer control register (TCR). When the timer is configured in continuous mode, the timer counter is reset to 0 on the cycle after the timer counter reaches the timer period. The timer can be stopped, restarted, reset, or disabled using the bits of the timer control register. The timer control register (TCR) does not control the TIMHI in this mode.

Figure 4. Dual 32-Bit Timers Chained Mode Block Diagram

Figure 5. Dual 32-Bit Timers Chained Mode Example

32-bit prescaler settings: count = CNTHI = 200; period = PRDHI = 202
32-bit timer settings: count = CNTLO = 3; period = PRDLO= 4
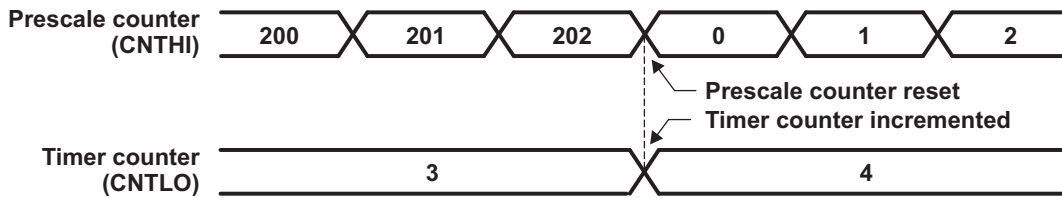
2.2.2   Unchained Mode

In the unchained mode, shown in Figure 6, the timer can operate as two independent 32-bit timers. One 32-bit timer (TIMHI) can be configured as a 32-bit timer being clocked by a 4-bit prescaler. The other (TIMLO) can be used as a 32-bit timer.

Figure 6. Dual 32-Bit Timers Unchained Mode Block Diagram
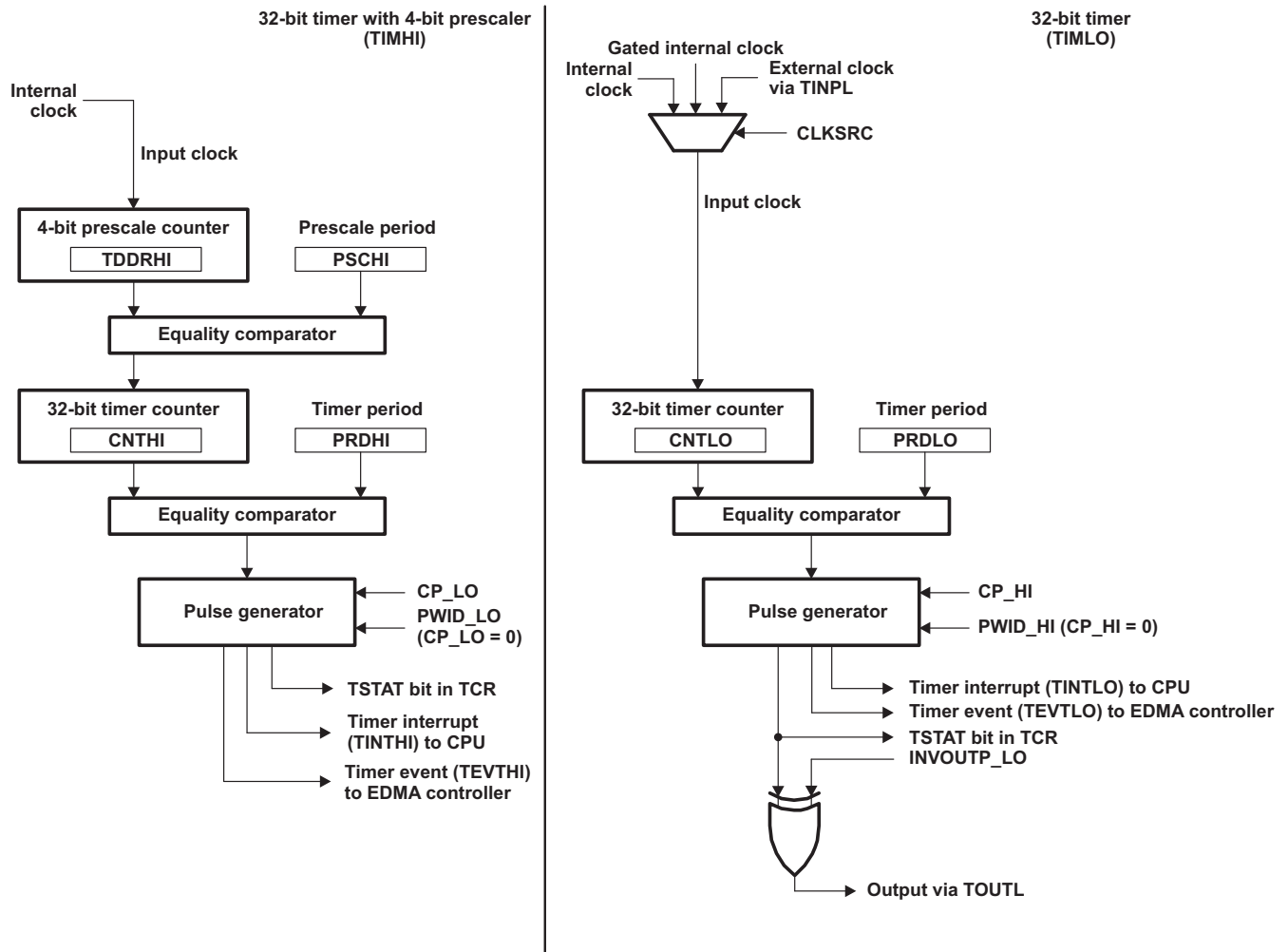
2.2.2.1   32-Bit Timer With a 4-Bit Prescaler (TIMHI)

In the unchained mode, the 4-bit prescaler must be clocked by the internal clock; an external clock source cannot be used for TIMHI. The 4-bit prescaler uses the timer divide-down ratio bits (TDDRHI) and the prescale counter bits (PSCHI) in TCR to form a 4-bit prescale counter register and a 4-bit prescale period register, respectively. When the timer is enabled, the prescale counter starts incrementing by 1 at every timer input clock cycle. One cycle after the prescale counter matches the prescale period, a clock signal is generated for the 32-bit timer.

The 32-bit timer uses the counter register (CNTHI) and the period register (PRDHI) to form a 32-bit timer counter register and a 32-bit timer period register, respectively. The 32-bit timer is clocked by the output clock from the 4-bit prescaler (see the example in Figure 7). When the timer is enabled, the timer counter increments by 1 at every prescaler output clock cycle. When the timer counter matches the timer period, a maskable timer interrupt (TINTHI) and a timer EDMA event (TEVTHI) are generated. The state of the output signal is read in the timer status (TSTAT_HI) bit of the timer control register (TCR). When in pulse mode (CP_HI = 0), TSTAT_HI stays high or low for 1, 2, 3, or 4 timer clock cycles. The pulse width depends on the setting of the pulse width (PWID_HI) bits in TCR. When in clock mode (CP_HI = 1), the TSTAT bit changes state (high-to-low or low-to-high) every time the timer counter matches the timer period. When the timer is configured in continuous mode, the timer counter is reset to 0 on the cycle after the timer counter reaches the timer period (see the example in Figure 7). The timer can be stopped, restarted, reset, or disabled using TCR and the timer global control register (TGCR).
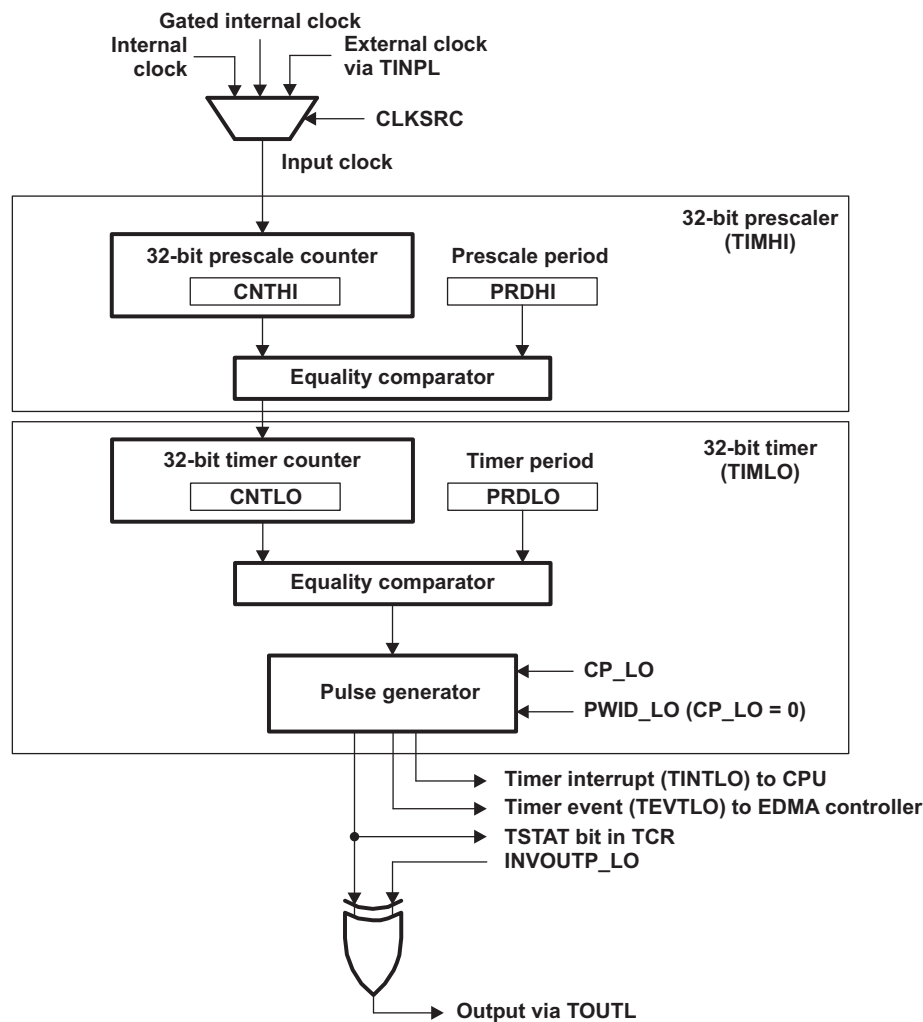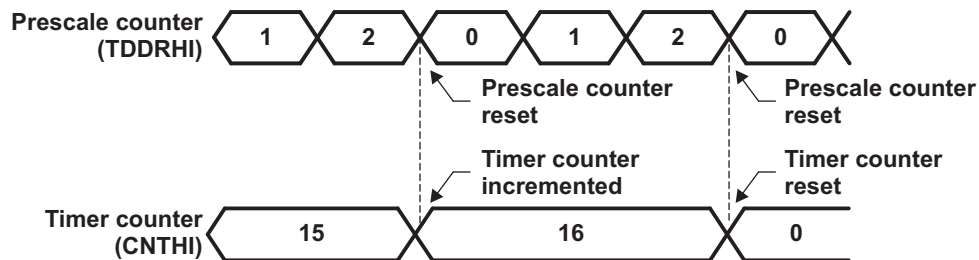
Figure 7. Dual 32-Bit Timers Unchained Mode Example

4-bit prescaler settings: count = TDDRHI = 1; period = PSCHI = 2

32-bit timer settings: count = CNTHI= 15; period = PRDHI = 16

2.2.2.2   32-Bit Timer (TIMLO)

The other 32-bit timer (TIMLO) uses the counter register (CNTLO) and the period register (PRDLO) to form a 32-bit timer counter register and a 32-bit timer period register, respectively. When the timer is enabled, the timer counter increments by 1 at every timer input clock cycle. When the timer counter matches the timer period, a maskable timer interrupt (TINTLO), a timer EDMA event (TEVTLO), and an output signal (TOUTL) are generated; the state of the output signal is also read in the timer status (TSTAT) bit of the timer control register (TCR). When in pulse mode (CP_LO = 0) and depending on the timer output inverter control (INVOUTP) bit in TCR, the timer output pin (TOUTL) stays high or low for 1, 2, 3, or 4 timer clock cycles. The pulse width depends on the setting of the pulse width (PWID_LO) bits in TCR. When in clock mode (CP_LO = 1), the timer output and the TSTAT_LO bit change state (high-to-low or low-to-high) every time the timer counter matches the timer period. When the timer is configured in continuous mode, the timer counter is reset to 0 on the cycle after the timer counter reaches the timer period. The timer can be stopped, restarted, reset, or disabled using TCR and the timer global control register (TGCR).

2.3 *Counter and Period Registers Used in GP Timer Modes*

Table 1 summarizes the counter registers (CNTLO and CNTHI) and period registers (PRDLO and PRDHI) used in each GP timer mode.

Table 1. Counter and Period Registers Used in GP Timer Modes

Timer Mode	Counter Registers	Period Registers
64-bit general-purpose	CNTHI:CNTLO	PRDHI:PRDLO
Dual 32-bit chained		
Prescaler (TIMHI)	CNTHI	PRDHI
Timer (TIMLO)	CNTLO	PRDLO
Dual 32-bit unchained		
Timer (TIMLO)	CNTLO	PRDLO
Timer with prescaler (TIMHI)	PSCHI bits and CNTHI	TDDRHI bits and PRDHI

3 Timer Operation

The following sections describe the overall timer operation. For specific details on the watchdog timer operation, see Section 4.

3.1 Timer Mode Selection

The timer can be configured as a 64-bit general-purpose timer or dual 32-bit timers (chained or unchained), or a watchdog timer using the timer mode (TIMMODE) bits in timer global control register (TGCR) (see Table 2). At reset, the timer is configured as a 64-bit GP timer as default. These bits can be written to select dual 32-bit timers (chained or unchained) or a watchdog timer function as shown in Table 2.

Table 2. Timer Mode Selection

TIMMODE Bits		
Bit 3	Bit 2	Timer Mode
0	0	64-bit general-purpose timer (default)
0	1	Dual 32-bit timers (unchained)
1	0	64-bit watchdog timer
1	1	Dual 32-bit timers (chained)

3.2 Timer Enabling

In the 64-bit timer mode or the dual 32-bit timers chained mode, the timer can be enabled by setting the TIMLORS and TIMHIRS bits in the timer global control register (TGCR) to 1 and setting the ENAMODE_LO bits in the timer control register (TCR) to 01b or 10b.

In the dual 32-bit timers unchained mode, the 32-bit timer (TIMLO) can be enabled by setting the TIMLORS bit in TGCR to 1 and the ENAMODE_LO bits in TCR to 01b or 10b. The 32-bit timer with prescaler (TIMHI) can be enabled by setting the TIMHIRS bit in TGCR to 1 and the ENAMODE_HI bits in the timer control register (TCR) to 01b or 10b.

Table 3 is a summary of timer enabling.

Table 3. Timer Enabling

Timer Mode	TCR ENAMODE bits				TGCR		Timer Status
	Bit = 23	Bit = 22	Bit = 7	Bit = 6	TIMHIRS	TIMLORS	
64-bit general-purpose	X	X	0	0	X	X	Disabled (default)
	X	X	0	1	1	1	Enabled one time
	X	X	1	0	1	1	Enabled continuously
Dual 32-bit chained	X	X	0	0	X	X	Disabled (default)
	X	X	0	1	1	1	Enabled one time
	X	X	1	0	1	1	Enabled continuously
Dual 32-bit unchained	0	0	0	0	X	X	Both timers disabled (default)
32-bit timer	X	X	0	1	X	1	32-bit timer = enabled one time
	X	X	1	0	X	1	32-bit timer enabled continuously

Table 3. Timer Enabling (continued)

Timer Mode	TCR ENAMODE bits				TGCR		Timer Status
	Bit = 23	Bit = 22	Bit = 7	Bit = 6	TIMHIRS	TIMLORS	
32-bit timer with prescaler	0	1	X	X	1	X	32-bit timer = enabled one time
	1	0	X	X	1	X	32-bit timer = enabled continuously

3.3 Timer Clock Source Selection

As shown in Table 4 and Figure 8, the timer clock source for TIMLO is selected using the clock source (CLKSRC) bit and timer input enable (TIEN_LO) bit in timer control register (TCR). The input clock source for TIMHI is always the internal clock.

Three clock sources are available to drive the timer clock:

- The internal clock, by setting CLKSRC_LO = 0 and TIEN_LO = 0.
- The internal clock gated by the timer input signal, by setting CLKSRC_LO = 0 and TIEN_LO = 1.
- The external clock on the timer input pin (TINPL), by setting CLKSRC_LO = 1. This input signal is synchronized internally and can be inverted by setting the timer inverter control (INVINP_LO) bit in TCR to 1.

At reset, the clock source is the internal clock. The internal clock is derived from the DSP clock generator, as shown in Figure 2.

When the clock source is the gated internal clock, the timer starts counting when the timer input transitions from low to high and the timer stops counting when the timer input transitions from high to low.

Table 4. Timer Clock Source Selection

CLKSRC_LO	TIEN_LO	Input Clock
0	0	Internal clock (default)
0	1	Gated internal clock
1	X	External clock on timer input (TINPL)

Figure 8. Timer Clock Source Block Diagram
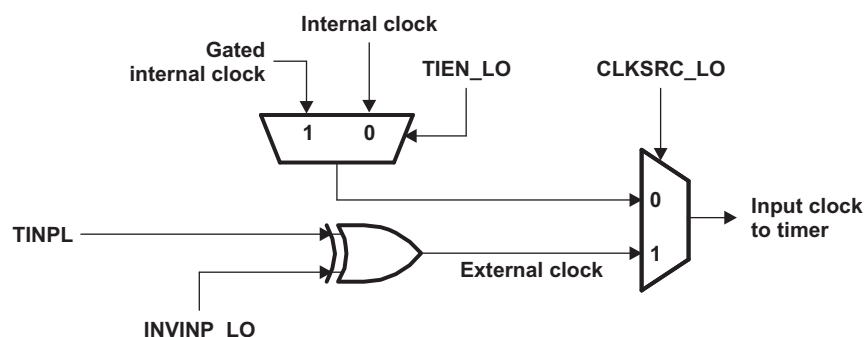
3.4 Timer Output Mode Selection

The two basic timer output modes are pulse mode and clock mode. The timer output mode is selected using the clock/pulse mode bits (CP_LO and CP_HI) in the timer control register (TCR).

When in the pulse mode (CP_LO or CP_HI = 0), the pulse width bits (PWID_LO or PWID_HI) set the pulse width to 1, 2, 3, or 4 timer clock cycles. This pulse can be inverted by setting the timer output inverter control bits (INVOUTP_LO or INVOUTP_HI) to 1.

When in the clock mode (CP_LO or CP_HI = 1), the timer output signal has a 50% duty cycle. The signal toggles (from high-to-low or from low-to-high) each time the timer counter reaches the timer period.

The output signal of TIMLO is driven on both TSTAT_LO and the timer output pin TOUTL. The output signal of TIMHI is driven on TSTAT_HI only.

3.5 Timer Counting

The timer counter runs at the timer clock rate specified by the clock source bit (CLKSRC) in the timer control register (TCR). Counting is enabled by setting the enabling mode (ENAMODE) bits in TCR to 01b or 10b. When enabled, the timer counter starts incrementing until the counter reaches a value equal to the value in the timer period register. Once the timer counter matches the timer period:

- If the timer is set to enable one time (ENAMODE = 01b), the timer counter is reset to 0, then stops.
- If the timer is set to enable continuously (ENAMODE = 10b), the timer counter is reset to 0, then continues counting.

Once the timer stops, if an external clock is used as the timer clock, the disable period must last at least one external clock period or the timer will not start counting again. When using the external clock, the count value is synchronized to the internal clock.

Note that when both the timer counter and timer period are cleared to 0, the timer can be enabled but the timer counter does not increment because the timer period is 0.

3.6 Timer Reset Sources

The timer has two reset sources: hardware reset and the timer reset bits (TIMLORS and TIMHIRS) in the timer global control register (TGCR).

- When a hardware reset is asserted, all the registers are set to their default values.
- When TIMLORS is cleared to 0, TSTAT_LO in TCR is reset to 0 and TOUTL is in the high-impedance state.
- When TIMHIRS is cleared to 0, TSTAT_HI in TCR is reset to 0.

3.7 Timer Interrupt Rate

To receive periodic interrupts, configure the timer to run in the continuous mode (ENAMODE = 10b). Each time the timer finishes counting, it can generate a timer interrupt for the CPU and a timer event for the EDMA controller. The rate at which this occurs (the timer interrupt rate) depends on whether the timer has a prescaler.

If the timer does not have a prescaler, there is only one counter. When the timer counter reaches the programmed timer period, the timer generates an interrupt and an EDMA event. Because the timer is in the continuous mode, one cycle after the timer counter matches the timer period, the timer counter is reset to 0 and starts counting again. The timer interrupt rate is:

$$TINT\ rate = \frac{Timer\ input\ clock\ rate}{Programmed\ timer\ period + 1}$$

If a timer has prescaler, there are two counters. One cycle after the prescale counter reaches the programmed prescale period, the timer counter is incremented by 1, and the prescale counter is reset to start counting again. If the prescaler continues long enough, it increments the timer counter to the programmed timer period. At that time, the timer generates an interrupt and an EDMA event. One cycle later (assuming the continuous mode), the timer counter is reset to 0 and starts counting again. The timer interrupt rate in this case is:

$$TINT\ rate = \frac{Timer\ input\ clock\ rate}{(Programmed\ prescale\ period + 1) \cdot (Programmed\ timer\ period + 1)}$$

3.8 Timer Emulation Modes

The timer has an emulation management and clock speed register (EMUMGT_CLKSPD). As shown in Table 5, the FREE and SOFT bits of EMUMGT_CLKSPD determine how the timer responds to an emulation suspend event. An emulation suspend event corresponds to any type of emulator access to the DSP, such as a hardware or software breakpoint, a probe point, or a printf instruction.

Table 5. Timer Emulation Modes Selection

FREE	SOFT	Emulation Mode
0	0	Default: The timer stops immediately.
0	1	The timer stops when the timer counter value increments and reaches the value in the timer period register.
1	X	The timer runs free regardless of SOFT bit status.

When using an internal clock as the timer clock source, the timer counter increments properly when single stepping. For example, the timer increments by one for each single step if the timer clock is equal to the CPU clock; or increments by one for every six single steps if the timer clock is equal to one-sixth of the CPU clock.

3.9 Timer Operation Boundary Conditions

The following boundary conditions affect the timer operation.

3.9.1 Writing to and Reading From the Reserved Registers

Write the reset value to the reserved registers. Reading from the reserved registers returns zeros.

3.9.2 Timer Count = 0 and Timer Period = 0 (No Prescaler)

Consider a timer that has no prescaler:
* The 64-bit GP timer.
* TIMLO in the 32-bit dual timers configuration (unchained mode).

In the special case when timer count = 0 and timer period = 0:
* After a hardware reset and before the timer starts counting (ENAMODE bits = 00b), the timer output signal is held low.
* Once the timer is enabled, its behavior depends on the selected enabling mode (ENAMODE bits = 01b or 10b in the timer control register) and the selected timer output mode (CP bits = 0 or 1 in the timer control register). The options are summarized in Table 6.
* The timer interrupt is not generated.

Table 6. Timer Operation When Timer Count = 0 and Timer Period = 0

Timer Enabling Mode	Timer Output Mode	Timer Operation When Timer Count = 0 and Timer Period = 0 (No Prescaler)
One-time mode (ENAMODE bits = 01b)	Pulse mode (CP bits = 0)	The timer output pulses once at the first timer clock cycle, and the timer stops counting at the next timer clock cycle. The pulse width is defined by the PWID bits of the timer control register.
	Clock mode (CP bits = 1)	The timer output toggles once at the first timer clock cycle. The timer stops counting at the next timer clock cycle.
Continuous mode (ENAMODE bits = 10b)	Pulse mode (CP bits = 0)	The timer output pulses once at the first timer clock cycle, and the timer continues to count up. Whenever the timer counter reaches its maximum value, it rolls around to 0 (see Section 3.9.4), generating another pulse. The pulse width is defined by the PWID bits of the timer control register.
	Clock mode (CP bits = 1)	The timer output toggles once at the first timer clock cycle and then toggles with a frequency of half the timer clock frequency as the timer continues to count.

3.9.3 Timer Count = 0, Timer Period = 0, Prescale Count = 0, and Prescale Period = 0

Consider a timer that has a prescaler:

- The combination timer in the 32-bit dual timers chained mode.
- TIMHI in the 32-bit dual timer configuration (unchained mode).

In the special case when timer count = 0, timer period = 0, prescale count = 0, and prescale period = 0, the timer operates in the same manner as a non-prescaled timer with timer count = 0 and timer period = 0 (see Section 3.9.2).

3.9.4 Timer Counter Overflow

Timer counter overflow can happen when the timer counter register is set to a value greater than the value in the timer period register. The counter reaches its maximum value (FFFF FFFFh or FFFF FFFF FFFF FFFFh), rolls over to 0, and continues counting until it reaches the timer period. An example is shown in Figure 9.

Figure 9. 32-Bit Timer Counter Overflow Example

Timer interrupt and
timer event generated
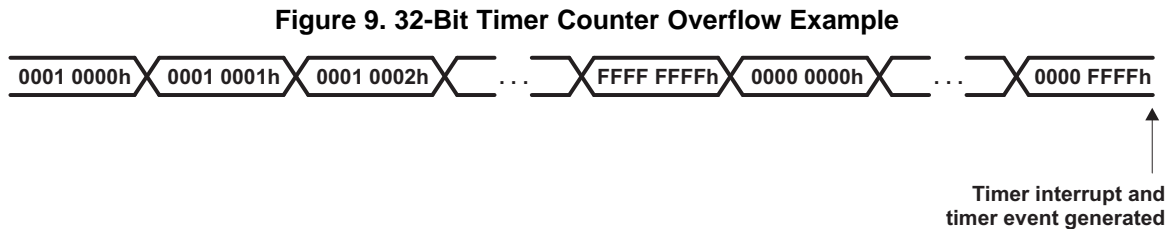
3.9.5 Writing to Registers of an Active Timer

Writes from the configuration bus to the timer registers are not allowed when the timer is active, except for stopping or resetting the timers. In the 64-bit and dual 32-bit timer modes, registers that are protected by hardware include CNTLO, CNTHI, PRDLO, PRDHI, TGCR (except the TIMLORS and TIMHIRS bits), and TCR (except the ENAMODE bits).

3.9.6 Small Timer Period Value in Pulse Mode

Small timer periods in pulse mode (CP = 0) can cause TSTAT to remain high when ENAMODE is not 0. This condition can occur when PRD . PWID + 1.

3.9.7 Reading the Counter Registers

Table 7 summarizes how to read the counter registers. When reading the timer counter in 64-bit GP timer mode, the CPU must read the first 32-bit word from the CNTLO registers. When this occurs, the timer takes a snapshot of the CNTHI register and copies it into a shadow register CNTHIS. Note that reading CNTHI instead of CNTLO will not cause the timer to take a snap shot of the timer counters and copy them into the shadow registers.

Table 7. Reading Counter Registers

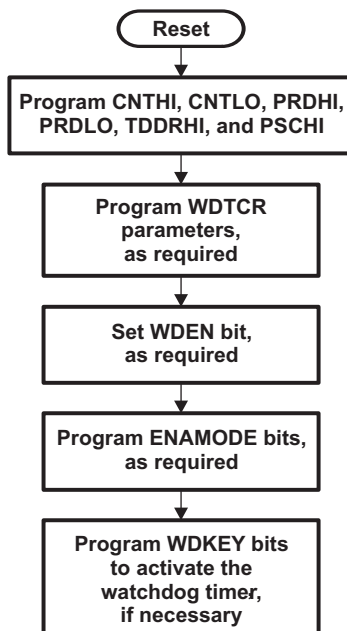| Timer Mode	CPU	
64-bit timer	Read CNTLO →	Read CNTLO
		Copy CNTHI to CNTHIS
	Read CNTHI →	Read from CNTHIS
32-bit timer	Read CNTLO →	Read CNTLO
	Read CNTHI →	Read CNTHI

3.10 *Initializing the Timer*

After a hardware reset, the enabling mode (ENAMODE) bits in the timer control register (TCR) are cleared to 0 and the timer is disabled. The timer counter and period registers are cleared to 0. The timer can be configured to the desired mode by programming the control registers, TCR and (in the case of the watchdog timer mode) WDTCR.

Figure 10 shows a typical timer initialization:

1. Write the timer counter and period values to CNTHI and/or CNTLO, and PRDHI and/or PRDLO registers.
2. If the 4-bit prescaler is used, write the values to the TDDRHI and PSCHI bits.
3. Set the remaining control bits to the required state.
4. Set WDEN = 1 to use the timer as watchdog timer, if necessary.
5. To start the timer, set the ENAMODE bits to use the timer as a continuous interrupt generator (ENAMODE bits = 10b) or as a one-time counter (ENAMODE bits = 01b).
6. Program the WDKEY bits, if the watchdog timer mode is selected.

Figure 10. Timer Initialization

4   Watchdog Timer Mode

The timer can also be configured as a 64-bit watchdog timer. As a watchdog timer, it can be used to prevent system lockup when the software becomes trapped in loops with no controlled exit. After a hardware reset, the timer is configured as a 64-bit GP timer and the watchdog mode is disabled. The timer then can be reconfigured as a watchdog timer using the timer mode (TIMMODE) bits in the timer global control register (TGCR) and the watchdog timer enable (WDEN) bit in the watchdog timer control register (WDTCR). In the watchdog timer mode, the timer requires a special service sequence to be executed periodically. Without this periodic servicing, the timer counter increments until it matches the timer period and causes a watchdog timeout event.

Once the timer is configured as a watchdog timer, it cannot be reconfigured as a GP timer until a device reset occurs. When the timer counter matches the timer period, the timer generates two signals: an output signal and an interrupt signal (described in Section 4.1). Typically, one or the other is used, depending on whether an external or internal trigger is desired.

4.1   Timer Output Signal and Timer Interrupt Signal in Watchdog Mode

When the periodic service sequence is not met, the timer counter increments until it matches the period and times out. During a time out, a pulse will be asserted on the timer output pin, and an internal maskable interrupt (TINTLO) will be triggered. The timer output pin can be externally connected to the non-maskable interrupt (NMI) pin of the device. Note that the timer pulse width must be configured to generate an active low pulse long enough for the CPU to recognize it as a NMI pulse. The pulse width is configured using the PWID bits of the timer control register (TCR).

4.2   Watchdog Timer Mode Restrictions

The watchdog timer mode is selected and enabled when TIMMODE = 10b in TGCR and WDEN = 1 in WDTCR. This mode has the following restrictions:

- No dual 32-bit timers mode
- No gated clock
- No external clock
- No one-time enabling
- No clock mode (only pulse mode).

4.3   Watchdog Timer Mode Operation

Figure 11 shows the timer when it is used in the watchdog timer mode. Note that in this mode, the timer clock must be set to the internal clock (CLKSRC_LO = 0). The CP_LO bit is forced to 0 because the pulse mode is required for the watchdog timer operation. The counter registers (CNTLO and CNTHI) form a 64-bit timer counter register and the period registers (PRDLO and PRDHI) form a 64-bit period register. When the timer counter matches the timer period, the timer generates a watchdog timeout event. This event:

- Drives the timer output signal (TOUTL) and/or the timer interrupt signal (TINTLO).
- Resets the timer counter to 0.
- Sets the TSTAT_LO bit, which is copied to the WDFLAG bit of WDTCR.
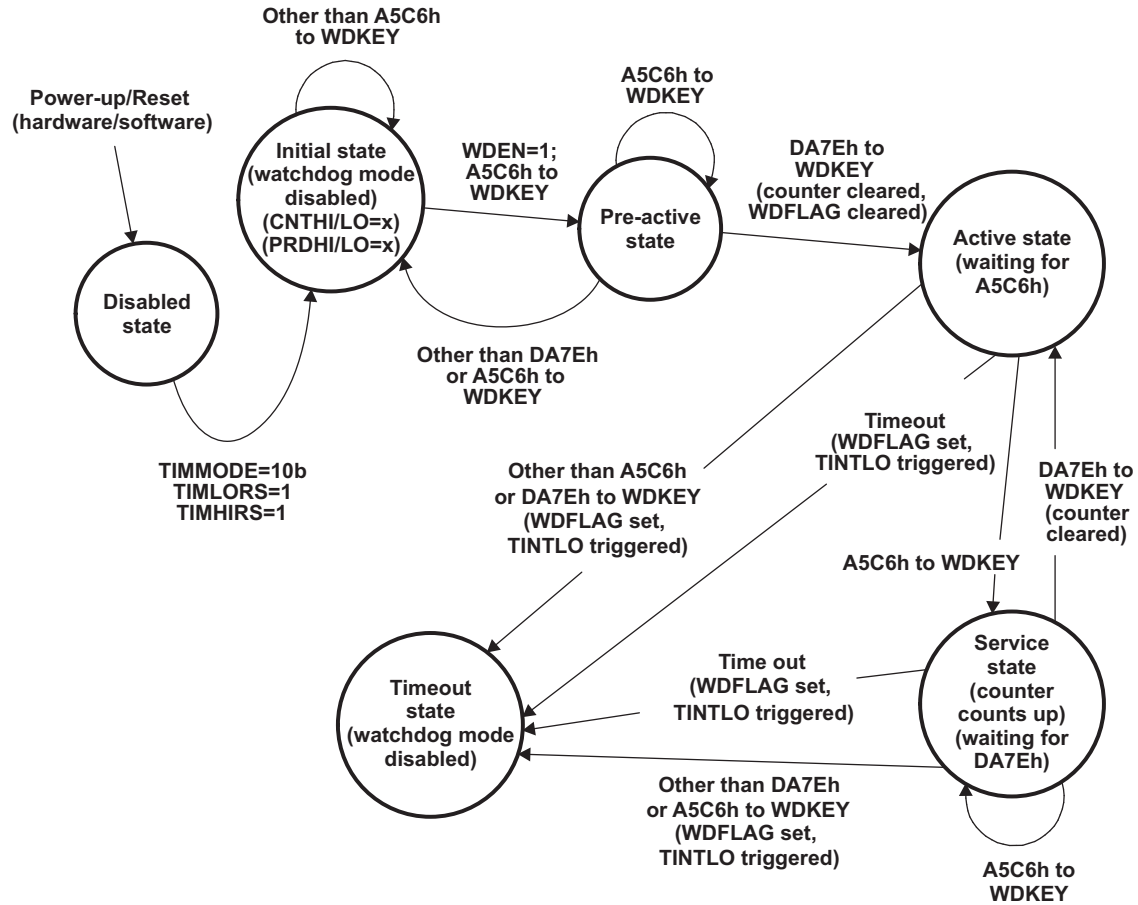
The timer output signal can be connected externally to the NMI pin to generate a non-maskable interrupt, if so desired.

Figure 11. Timer in Watchdog Timer Mode
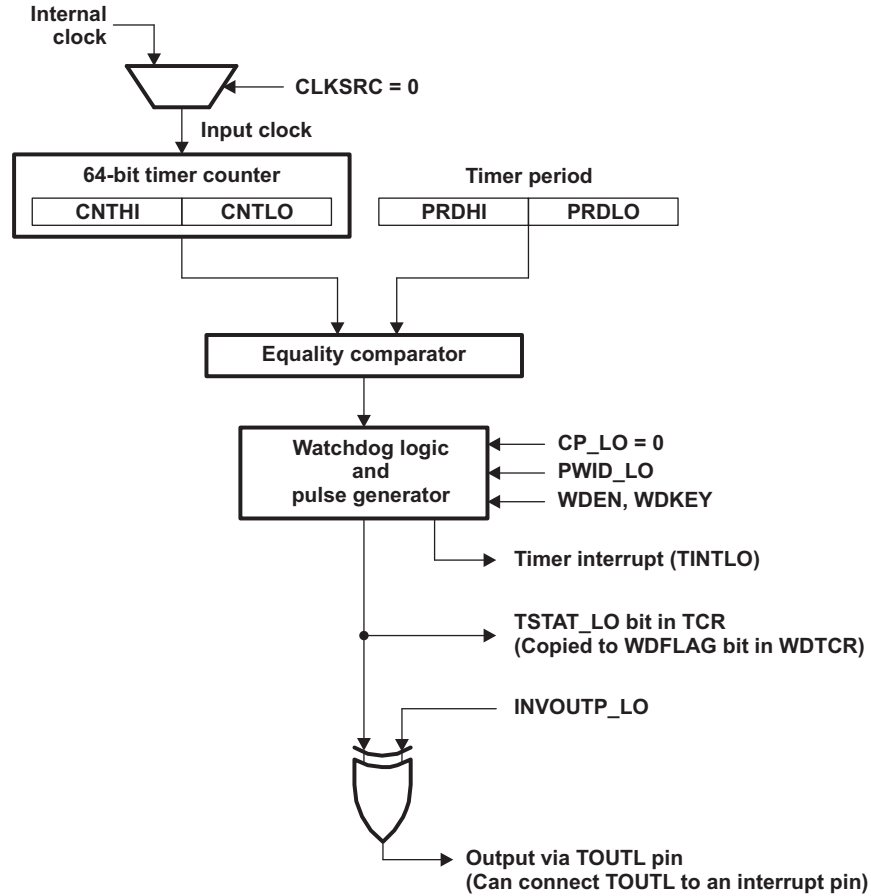
To activate the watchdog timer, a certain sequence of events must be followed, as shown in the state diagram, Figure 12.

Figure 12. Watchdog Timer Operation State Diagram

Once the watchdog timer is activated, it can be disabled only by a watchdog timeout event or by a hardware reset. A special key sequence is required to prevent the watchdog timer from being accidentally serviced while the software is trapped in a dead loop or by some other software failure.

To prevent a watchdog timeout event, the timer has to be serviced periodically (you could use another on-chip timer or an off-chip timer) by writing A5C6h followed by DA7Eh to the watchdog timer service key (WDKEY) bits of WDTCR before the timer finishes counting up. Both A5C6h and DA7Eh are allowed to be written to the WDKEY bits, but only the correct sequence of A5C6h followed by DA7Eh to the WDKEY bits services the watchdog timer. Any other writes to the WDKEY bits triggers the watchdog timeout event immediately. Writes to other bits in the WDTCR are ignored when the watchdog timer is active (see Section 4.4).

When the watchdog timer is in the timeout state, the watchdog timer is disabled, the WDEN bit is cleared to 0, and the timer is reset. After entering the timeout state, the watchdog timer cannot be enabled again until a hardware reset occurs.

After a hardware reset, the watchdog timer is disabled; however, reads or writes to the watchdog timer registers are allowed. Once the WDEN bit is set and A5C6h is written to the WDKEY bits, the watchdog timer enters the pre-active state. In the pre-active state:

- A write to WDTCR is allowed only when the write comes with the correct key (A5C6h or DA7Eh) to the WDKEY bits.
- A write of DA7Eh to the WDKEY bits when the WDEN bit is set to 1 resets the counters and activates the watchdog timer.

The PRDHI, PRDLO, TCR, and WDTCR registers must be configured before the watchdog timer enters the active state. The WDEN bit must be set to 1 before writing DA7Eh to the WDKEY bits in the pre-active state. Every time the watchdog timer is serviced by the correct WDKEY sequence, the watchdog timer counter is automatically reset.

> **Note:** Before the watchdog timer enters the active state, the timer output signal is never asserted. Only the timer interrupt is asserted when the timer finishes counting up. In this case, the timer interrupt can be used to:
>
> - Indicate that the watchdog timer is counting but is not in the active state.
> - Generate a periodic interrupt, without having to service the watchdog timer.
>
> The watchdog timer can always be disabled before entering the active state.

4.4 Watchdog Timer Register Write Protection

Once the watchdog timer enters the pre-active state, writes to registers CNTHI, CNTLO, PRDHI, PRDLO, TCR, and WDTCR (except for the WDKEY bits) will have no effect. Writes to WDEN when the watchdog timer is in the timeout state have no effect.

The value 0xA5C6 or 0xDA7E must be written to the WDKEY bits depending on the current state (see Figure 12). After the watchdog timer has entered the initial state, clearing the TIMLORS and TIMHIRS bits is prohibited.

5 Timer Registers

The timer contains a set of registers as indicated in Table 8. All timer register bits are read-write unless otherwise specified. For specific address locations, see the *TMS320C6457 Fixed-Point Digital Signal Processor* data manual (SPRS582).

Table 8. Timer Registers

Offset	Acronym	Name	See
0004	EMUMGT_CLKSPD	Emulation management and clock speed register	Section 5.1
0010	CNTLO	Counter register low	Section 5.2
0014	CNTHI	Counter register high	Section 5.2
0018	PRDLO	Period register low	Section 5.3
001C	PRDHI	Period register high	Section 5.3
0020	TCR	Timer control register	Section 5.4
0024	TGCR	Timer global control register	Section 5.5
0028	WDTCR	Watchdog timer control register	Section 5.6

5.1 Emulation Management and Clock Speed Register (EMUMGT_CLKSPD)

The EMUMGT_CLKSPD register contains the FREE and SOFT bits that determine how the timer responds to an emulation suspend event (see Figure 13 and Table 9). An emulation suspend event corresponds to any type of emulator access to the DSP, such as a hardware or software breakpoint, a probepoint, or a printf instruction. For additional emulation information, see Section 3.8.

The CLKDIV field of this register can also be read to identify the ratio of the CPU clock to the timer input clock. For example, in devices where the internal timer clock frequency is equal to the CPU frequency divided by 6, the CLKDIV field will read as 6 on those devices.

Figure 13. Emulation Management and Clock Speed Register (EMUMGT_CLKSPD)

31	20	19	16
Reserved		CLKDIV	
R/W-0		R-n[A]	

15	2	1	0
Reserved		SOFT	FREE
R/W-0		R-0	R-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

A. The reset value of this field is based on the ratio of the CPU clock to the timer internal clock. To determine what this ratio is for your device, see the *TMS320C6457 Fixed-Point Digital Signal Processor* data manual (SPRS582).

Table 9. Emulation Management and Clock Speed Register (EMUMGT_CLKSPD) Field Descriptions

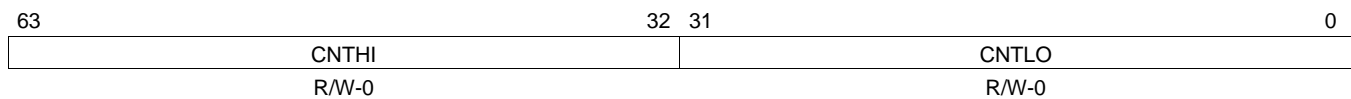| Bit	Field	Value	Description
31-20	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
19-16	CLKDIV		Clock divide-down ratio bits. Defines the ratio of the CPU clock to the timer input clock. The CLKDIV bits are read-only bits.
		1h	Internal clock source for the timer is the CPU clock divided by 1.
		2h	Internal clock source for the timer is the CPU clock divided by 2.
		3h	Reserved
		4h	Internal clock source for the timer is the CPU clock divided by 4.
		5h	Reserved
		6h	Internal clock source for the timer is the CPU clock divided by 6.
		7h	Reserved
		8h	Internal clock source for the timer is the CPU clock divided by 8.
		9-15h	Reserved
15-2	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
1	SOFT		Used in conjunction with FREE bit to determine how the timer responds to an emulation suspend event. When the FREE bit is 0, the SOFT bit selects the timer response.
		0	The timer stops immediately.
		1	The timer stops when the timer counter register increments and reaches the value in the timer period register.
0	FREE		Used in conjunction with SOFT bit to determine how the timer responds to an emulation suspend event. When the FREE bit is 0, the SOFT bit selects the timer response.
		0	The SOFT bit selects the timer response.
		1	The timer runs free, regardless of the value of the SOFT bit.

5.2 Timer Counter Registers (CNTHI and CNTLO)

The timer counter registers (CNTLO and CNTHI) are 32-bit wide registers that can be used in conjunction to form a 64-bit counter, or separately as 32-bit counters. The use of these registers depends on the configuration of the timer. In the 64-bit general-purpose timer mode and watchdog mode, the two registers work as a single 64-bit counter (see Figure 14). The 64-bit counter increments when the timer is enabled to count. The timer counter registers are cleared to 0 at reset. These two registers are shown in Figure 15 and described in Table 10 and are accessed via a separate address.

In a dual 32-bit timer mode, the counter registers work as separate 32-bit registers. These two register pairs can be configured as chained or unchained.

A hardware reset clears both counter registers, but software resets do not affect them. When the TIMLORS bit is cleared, CNTLO keeps its current value, and when the TIMHIRS bit is cleared, CNTHI keeps its current value.

Figure 14. 64-Bit Timer Counter Register

63 32	31 0
CNTHI	CNTLO
R/W-0	R/W-0

LEGEND: R/W = Read/Write; -*n* = value after reset

Figure 15. Timer Counter Registers (CNTHI and CNTLO)

31   0
CNT
R/W-0
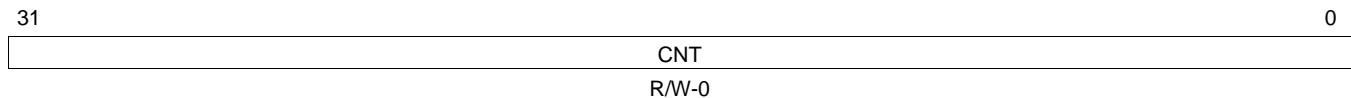
LEGEND: R/W = Read/Write; -*n* = value after reset

Table 10. Timer Counter Registers (CNTHI and CNTLO) Field Descriptions

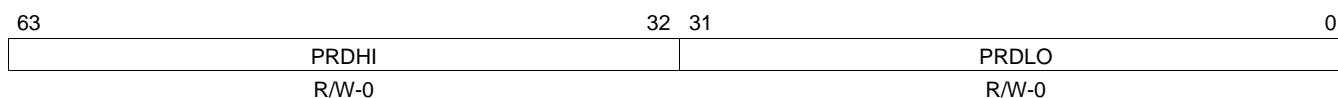| Bit	Field	Value	Description
31-0	CNT	00000000h - FFFFFFFFh	Counter register. This register is a 32-bit prescale counter or 32-bit timer counter, or one half of a 64-bit timer counter.

5.3 Timer Period Registers (PRDHI and PRDLO)

The timer period registers (PRDLO and PRDHI) are 32-bit wide registers which can be used in conjunction to form a single 64-bit period register (Figure 16) or separately as 32-bit period registers. These two registers have the field shown in Figure 17 and described in Table 11 and are accessed via a separate address.

In a 64-bit general-purpose timer mode and watchdog mode, all 64 period bits contain the number of timer input clock cycles to count. This number controls the frequency of the timer output. In a 32-bit dual timer mode, the period registers work as separate 32-bit registers.

These two registers are used in conjunction with the two counter-registers, CNTHI and CNTLO.

Figure 16. 64-Bit Timer Period Register

63	32	31	0
PRDHI		PRDLO	
R/W-0		R/W-0	

LEGEND: R/W = Read/Write; -*n* = value after reset

Figure 17. Timer Period Registers (PRDHI and PRDLO)

31	0
PRD	
R/W-0	
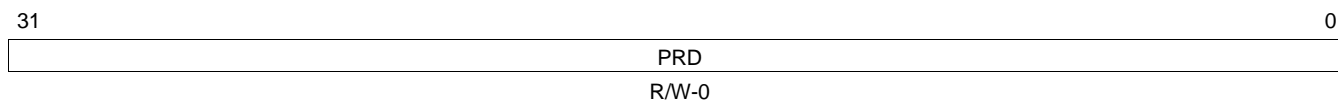
LEGEND: R/W = Read/Write; -*n* = value after reset

Table 11. Timer Period Registers (PRDHI and PRDLO) Field Descriptions

Bit	Field	Value	Description
31-0	PRD	00000000h - FFFFFFFFh	Period register. This register contains the full timer period for a 32-bit timer configuration or half the timer period for a 64-bit timer configuration.

5.4   Timer Control Register (TCR)

The timer control register (TCR) is shown in Figure 18 and described in Table 12. The lower 16 bits of TCR determine the operating mode and monitor the status of TIMLO, as well as control the function of TINPL and TOUTL. The upper 16 bits determine the operating mode and monitor the status of TIMHI. The upper 16 bits of TCR are used only when the timer is configured in dual 32-bit timers unchained mode (TIMMODE = 01b in TGCR).

Figure 18. Timer Control Register (TCR)

31							24
Reserved							
R/W-0							

23	22	21	20	19	18	17	16
ENAMODE_HI		PWID_HI		CP_HI	Reserved	INVOUTP_HI	TSTAT_HI
R/W-0		R/W-0		R/W-0	R/W-0	R/W-0	R-0

15					10	9	8
Reserved						TIEN_LO	CLKSRC_LO
R/W-0						R/W-0	R/W-0

7	6	5	4	3	2	1	0
ENAMODE_LO		PWID_LO		CP_LO	INVINP_LO	INVOUTP_LO	TSTAT_LO
R/W-0		R/W-0		R/W-0	R/W-0	R/W-0	R-0

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 12. Timer Control Register (TCR) Field Descriptions

Bit	Field	Value	Description
31-24	Reserved		Reserved. The reserved bit location is always read as 0. A value written to this
23-22	ENAMODE_ HI		Enabling mode bits determine the timer mode for TIMHI.
		00b	The timer is disabled (not counting) and maintains the current value.
		01b	The timer is enabled one time. The timer stops after the timer counter reaches the timer period.
		10b	The timer is enabled continuously. The timer counter increments until it reaches the timer period. One timer clock cycle later, the timer counter is reset to 0 and continues counting.
		11b	Reserved
21-20	PWID_HI		Pulse width bits for TIMHI. PWID_HI is only used in pulse mode (CP_HI = 0). PWID_HI controls the width of the timer output signal. The polarity of the pulse is controlled by the INVOUTP_HI bit. The timer output signal is recorded in the TSTAT_HI bit and can be made visible on the timer output pin.
		00b	The pulse width is 1 timer clock cycle.
		01b	The pulse width is 2 timer clock cycles.
		10b	The pulse width is 3 timer clock cycles.
		11b	The pulse width is 4 timer clock cycles.
19	CP_HI		Clock/pulse mode bit for TIMHI. In the watchdog timer mode (TIMMODE = 10b), the pulse mode is selected automatically and the CP_HI bit is a don't care.
		0	Pulse mode. When the timer counter reaches the timer period, the timer output appears as a pulse with the width defined by the PWID_HI bits and the polarity defined by the INVOUTP_HI bits.
		1	Clock mode. The timer output signal has a 50% duty cycle signal. When the timer counter reaches the timer period, the level of the timer output signal is toggled (from high to low or from low to high).
18	Reserved		Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
17	INVOUTP_HI		Timer output inverter control bit for TIMHI.
		0	The timer output is not inverted.
		1	The timer output is inverted.

Table 12. Timer Control Register (TCR) Field Descriptions (continued)

Bit	Field	Value	Description
16	TSTAT_HI		Timer status bit for TIMHI. This is a read-only bit that shows the value of the timer output.
		0	Timer output is low.
		1	Timer output is high.
15-10	Reserved		Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
9	TIEN_LO		Timer input enable bit determines if the timer clock is gated by the timer input. Applicable only when CLKSRC_LO = 0.
		0	Timer clock is not gated by the timer input.
		1	Timer clock is gated by a high state of the timer input synchronized with the internal clock. Timer starts counting when timer input transitions from low to high. Timer stops counting when timer input transitions from high to low.
8	CLKSRC_LO		Clock source bit determines the clock source for the timer.
		0	The clock source is the internal clock.
		1	The clock source is the signal on the timer pin.
7-6	ENAMODE_ LO		Enabling mode bits determine the timer mode.
		00b	The timer is disabled (not counting) and maintains the current value.
		01b	The timer is enabled one time. The timer stops after the timer counter reaches the timer period.
		10b	The timer is enabled continuously. The timer counter increments until it reaches the timer period. One timer clock cycle later, the timer counter is reset to 0 and continues counting.
		11b	Reserved
5-4	PWID_LO		Pulse width bits. PWID_LO is only used in pulse mode (CP_LO = 0). PWID_LO controls the width of the timer output signal. The polarity of the pulse is controlled by the INVOUTP_LO bit. The timer output signal is recorded in the TSTAT_LO bit and can be made visible on the timer output pin.
		00b	The pulse width is 1 timer clock cycle.
		01b	The pulse width is 2 timer clock cycles.
		10b	The pulse width is 3 timer clock cycles.
		11b	The pulse width is 4 timer clock cycles.
3	CP_LO		Clock/pulse mode bit for timer output. In the watchdog timer mode (TIMMODE = 10b), the pulse mode is selected automatically and the CP_LO bit is a don't care.
		0	Pulse mode. When the timer counter reaches the timer period, the timer output appears as a pulse with the width defined by the PWID_LO bits and the polarity defined by the INVOUTP_LO bits.
		1	Clock mode. The timer output signal has a 50% duty cycle signal. When the timer counter reaches the timer period, the level of the timer output signal is toggled (from high to low or from low to high).
2	INVINP_LO		Timer input inverter control bit. Only affects operation if CLKSRC_LO = 1.
		0	A non-inverted timer input drives the timer.
		1	An inverted timer input drives the timer.
1	INVOUTP_LO		Timer output inverter control bit.
		0	The timer output is not inverted.
		1	The timer output is inverted.
0	TSTAT_LO		Timer status bit. This is a read-only bit that shows the value of the timer output. TSTAT_LO drives the timer pin (TOUTL) when the pin is used as a timer output pin and may be inverted by setting INVOUTP_LO = 1.
		0	Timer output is low.
		1	Timer output is high.

5.5 Timer Global Control Register (TGCR)

The timer global control register (TGCR) is shown in Figure 19 and described in Table 13. This register contains a field for selecting the operating mode of the timer (TIMMODE), timer reset bits (TIMHIRS and TIMLORS), and counters for TIMHI in the dual 32-bit timers unchained mode (TDDRHI and PSCHI).

Figure 19. Timer Global Control Register (TGCR)

31	16
Reserved	
R/W-0	

15	12	11	8
TDDRHI		PSCHI	
R/W-0		R/W-0	

7		4	3	2	1	0
Reserved			TIMMODE		TIMHIRS	TIMLORS
R-0			R/W-0		R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 13. Timer Global Control Register (TGCR) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved		Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
15-12	TDDRHI		Timer divide-down ratio bits. This field is the prescale counter for TIMHI in the dual 32-bit timers unchained mode (TIMMODE = 01b). When the timer is enabled, TDDRHI increments every timer clock cycle. The timer counter (CNTHI) increments on the cycle after the TDDRHI matches the value of PSCHI. TDDRHI resets to 0 and continues. If enabled one time, when CNTHI matches the PRDHI, the timer stops; if the timer is enabled continuously, CNTHI resets to 0 on the cycle after matching the PRDHI and the timer continues counting. The default value is 0000b.
11-8	PSCHI		Prescale period bits. This field specifies the prescale period for TIMHI in the dual 32-bit timers unchained mode (TIMMODE = 01b). The default value is 0000b.
7-4	Reserved		Reserved. The reserved bit location is always read as 0. Read-only.
3-2	TIMMODE		Timer mode bits determine the timer operating mode.
		00b	The timer is in the 64-bit general-purpose timer mode.
		01b	The timer is in the dual 32-bit timers unchained mode.
		10b	The timer is in the 64-bit watchdog timer mode.
		11b	The timer is in the dual 32-bit timers chained mode.
1	TIMHIRS		TIMHI reset bit. Note that in order for the timer to function properly in 64-bit general-purpose timer mode both the TIMHIRS and TIMLORS bits must be set to 1. If the timer is in the watchdog timer active state, changing this bit does not affect the timer.
		0	TIMHI is in reset. The TSTAT_HI bit of TCR is reset to 0. However, the counter register CNTHI keeps its current value.
		1	TIMHI is not in reset. TIMHI can be used as a 32-bit timer.
0	TIMLORS		TIMLO reset bit. Note that in order for the timer to function properly in 64-bit general-purpose timer mode both the TIMHIRS and TIMLORS bits must be set to 1. If the timer is in the watchdog timer active state, changing this bit does not affect the timer.
		0	TIMLO is in reset. The TSTAT_LO bit of TCR is reset to 0, and the timer output signal is in the high-impedance state. However, the counter register CNTLO keeps its current value.
		1	TIMLO is not in reset. TIMLO can be used as a 32-bit timer.

5.6 *Watchdog Timer Control Register (WDTCR)*

The watchdog timer control register (WDTCR) is shown in Figure 20 and described in Table 14. This register determines the state of the watchdog timer and monitors the watchdog timer status.

Figure 20. Watchdog Timer Control Register (WDTCR)

31	16
WDKEY	
R/W-0	

15	14	13	12	11				0
WDFLAG	WDEN	WDKEY		Reserved				
R/W-0	R/W-0	R/W-0			R/W-0			
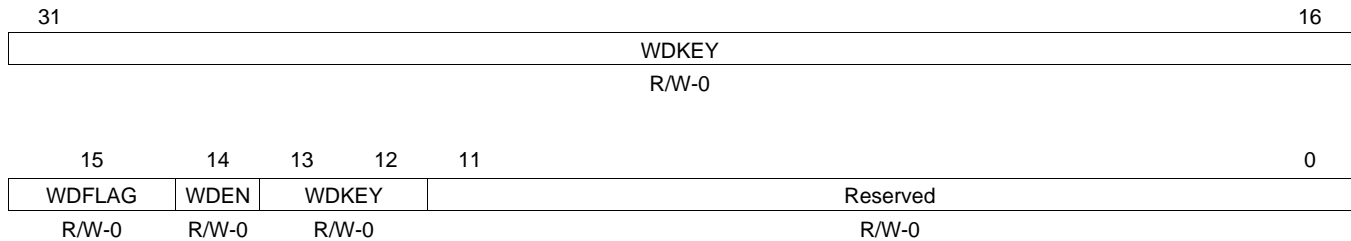
LEGEND: R/W = Read/Write; -*n* = value after reset

Table 14. Watchdog Timer Control Register (WDTCR) Field Descriptions

Bit	Field	Value	Description
31-16	WDKEY	0000h - FFFFh	Watchdog timer service key bits. Before a watchdog timeout event occurs, only a write sequence of a A5C6h followed by a DA7Eh services the watchdog timer. Any other write triggers a watchdog timeout event immediately. The default value of WDKEY is 0000h. WDKEY is not applicable in the general-purpose timer mode.
15	WDFLAG		Watchdog timer flag bit. WDFLAG is cleared when the watchdog timer is moved to the active state, when a hardware reset occurs, and when 1 is written to WDFLAG.
		0	No watchdog timer timeout event occurred.
		1	Watchdog timer timeout event occurred.
14	WDEN		Watchdog timer enable bit. WDEN must be set to move the watchdog timer to the pre-active state (see Figure 12).
		0	Watchdog timer is disabled. Watchdog timer output pin is disconnected from the watchdog timer timeout event, and the timer counter starts to count. The timer is in the general-purpose timer mode.
		1	Watchdog timer is enabled. The watchdog timer output pin is connected to the watchdog timeout event. The watchdog timer can be disabled by a watchdog timeout event or by a hardware reset.
13-12	WDKEY	00b - 11b	Watchdog idle enable key bits. A write sequence of 01b followed by 10b is required before the watchdog timer can enter the idle mode. Writing 00b to WDKEY prevents the watchdog timer from entering the idle mode. WDKEY is not applicable in the general-purpose timer mode.
11-0	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.

IMPORTANT NOTICE