Timer_D

NOTE: This chapter is an excerpt from the *MSP430x5xx and MSP430x6xx Family User's Guide*. The most recent version of the full user's guide is available here: http://www.ti.com/lit/pdf/slau208.

Timer_D is a 16-bit timer/counter with multiple capture/compare registers. This chapter describes Timer_D.

Topic **Page**

1.1 Timer_D Introduction

Timer_D is a 16-bit timer/counter with multiple capture/compare registers. Timer_D can support multiple capture/compares, interval timing, and PWM outputs both in general and high resolution modes. Timer_D also has extensive interrupt capabilities. Interrupts may be generated from the counter on overflow conditions, from each of the capture/compare registers.

Timer_D features include:

- Asynchronous 16-bit timer/counter with four operating modes and four selectable lengths
- Selectable and configurable clock source
- Configurable capture/compare registers
- Controlling rising and falling PWM edges by combining two neighbor TDCCR registers in one compare channel output
- Configurable outputs with PWM capability
- High-resolution mode with a fine clock frequency up to 16 times the timer input clock frequency
- Double-buffered compare registers with synchronized loading
- Interrupt vector register for fast decoding of all Timer_D interrupts

The block diagram of Timer_D is shown in Figure 1-1.

NOTE: Use of the word *count*

Count is used throughout this chapter. It means the counter must be in the process of counting for the action to take place. If a particular value is directly written to the counter, any associated action does not take place.

1.1.1 Differences From Timer_B

Timer_D is identical to Timer_B with the following exceptions:

- Timer_D supports high-resolution mode.
- Timer_D supports the combination of two adjacent TDCCRx registers in one capture/compare channel.
- Timer_D supports the dual capture event mode.
- Timer_D supports external fault input, external clear input, and signal. See the TEC chapter for detailed information.
- Timer_D can synchronize with a second timer instance when available. See the TEC chapter for detailed information.

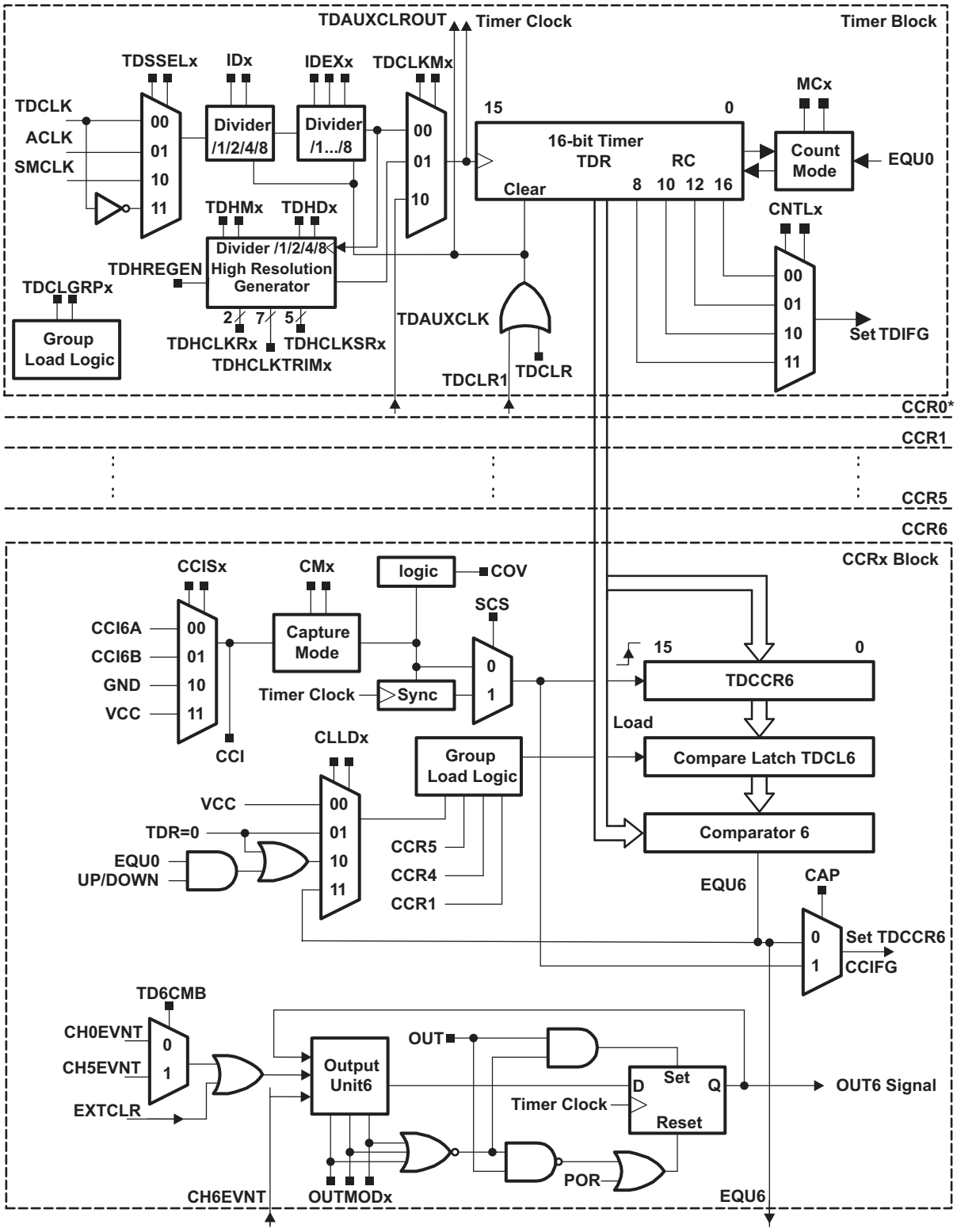

Figure 1-1. Timer_D Block Diagram

NOTE: TDxCCR0 is slightly different from the other capture/compare blocks. The input signals to the output unit are:

- OUT0 signal
- EXTCLR signal
- Timer Counter overflow

This enables channel 0 to generate a PWM output signal in Continuous mode. The events to set and reset the PWM output 0 are the CCR0 match and the Timer Counter overflow (TDIFG).

NOTE: TDCCRx channels with odd channel numbers do not have the TDxCMB bit capabilities. Therefore, there is no MUX, and the CH0EVNT is ORed with EXTCLR.

1.2 Timer_D Operation

The Timer_D module is configured with user software. The setup and operation of Timer_D is discussed in the following sections.

1.2.1 16-Bit Timer Counter

The 16-bit timer/counter register, TDxR, increments or decrements (depending on mode of operation) with each rising edge of the clock signal. TDxR can be read or written with software. Additionally, the timer can generate an interrupt when it overflows.

TDxR may be cleared by any of the following events:

- Writing 1 to TDCLR bit
- A logical high TDCLR1 signal. See the TEC chapter for details.

When the TDxR is cleared, the timer divider is restarted.

NOTE: Modifying Timer_D registers

It is recommended to stop the timer before modifying its operation to avoid errant operating conditions. Stopping the timer is not required for the following cases:

- To enable interrupt
- Setting interrupt flags
- Setting TDCLR bit
- Enabling or disabling calibration in high-resolution mode

When the TDCLK is asynchronous to the CPU clock, any read from TDxR should occur while the timer is not operating, or the results may be unpredictable. Alternatively, the timer may be read multiple times while operating, and a majority vote taken in software to determine the correct reading. Any write to TDxR takes effect immediately.

1.2.1.1 TDxR Length

Timer_D is configurable to operate as an 8-, 10-, 12-, or 16-bit timer with the CNTLx bits. The maximum count value, $TDR_{(max)}$, for the selectable lengths is 0FFh, 03FFh, 0FFFh, and 0FFFFh, respectively. Data written to the TDxR register in 8-, 10-, and 12-bit mode is right justified with leading zeros.

1.2.1.2 Clock Source Select and Divider

The TDCLKMx bits of the control register 1 (TDxCTL1) select the timer clock between the timer input clock source, the high-resolution local clock, or the auxiliary clock of another timer instance.

The timer input clock can be sourced from ACLK, SMCLK, or externally via TDCLK or INCLK. The clock source is selected with the TDSSELx bits. The selected clock source may be passed directly to the timer or to a 2-stage division process by using the ID and IDEX bits. Whenever a timer clear event occurs, the clock divider is restarted. The clock divider is also restarted when passing the selected clock to the high-resolution generator.

1.2.2 High-Resolution Generator

In high-resolution mode, the high-resolution generator is used. Figure 1-2 shows the high resolution clock generator block diagram. It can be operated in free-running mode or in regulated mode.

In free-running mode (TDHREGEN = 0), the high-resolution clock is selected by the high-resolution clock range selection bits TDHCLKRx in the register TDxHCTL1. Each clock range is divided into $2^{TDHCLKSRx}$ sub-ranges. And in each sub-range, a total number of $2^{TDHCLKTRIMx}$ slots can be chosen by configuring the TDHCLKTRIMx bits in the TDxHCTL1 register.

In regulated mode (TDHREGEN = 1), the selected high-resolution generator frequency is adjusted to the timer input clock frequency. The high-resolution clock frequency is 8x (TDHMx = 00) or 16x (TDHMx = 01) higher than the selected timer input clock. The timer input clock frequency can be within the range of 8 MHz to 16 MHz if 16x is selected (TDHMx = 01) or 8 MHz to 25 MHz if 8x (TDHMx = 00) is selected. When the timer input clock frequency is greater than 15 MHz, the TDHCLKCR bit in the TDxHCTL1 register should be set. The TDHCLKRx bits are used to preset the frequency range of the high-resolution generator.

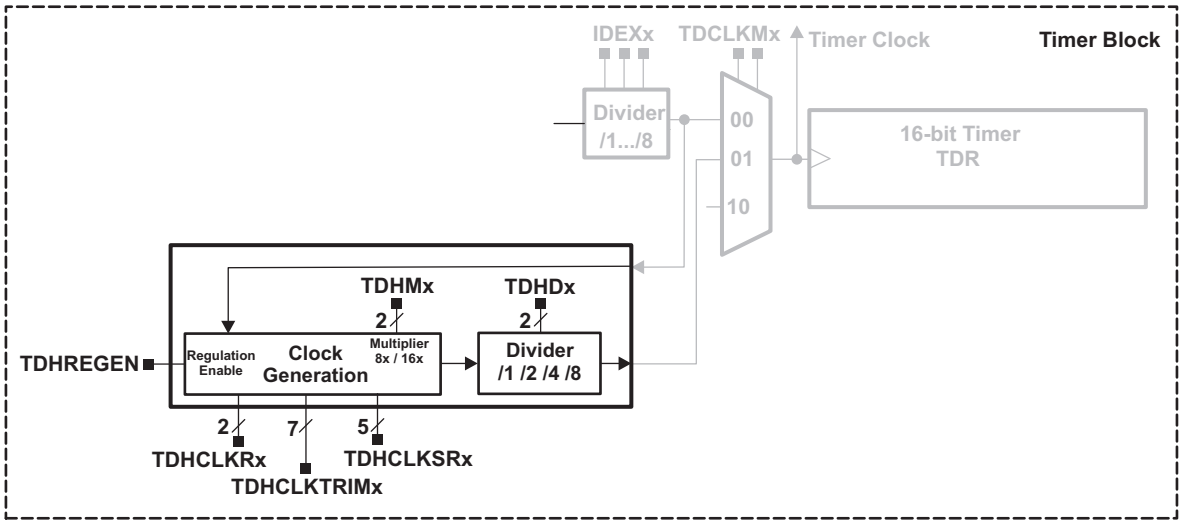

Figure 1-2. High Resolution Clock Generator

1.2.2.1 Free-Running Mode of the High-Resolution Generator

When the high-resolution generator is used in free-running mode, the register TDxHCTL1 is configured by the user software. Before the high-resolution generator is enabled, the bits TDHCLKCR, TDHCLKRx, TDHSRx, and TDHCLKTRIMx must be configured. See the device-specific data sheet for details about the frequency selection. After the high-resolution generator is turned on, only the TDHCLKTRIMx bits should be changed by +1 or -1 at a time. All other settings must not be altered manually. Using TDHCLKTRIMx allows changing the frequency by at least ±20% from TDHCLKTRIM = 64.

There are factory preprogrammed values stored in the TLV structure of the flash memory for the TDHCLKRx, TDHSRx, and TDHCLKTRIMx registers for specific frequencies. To use the calibrated settings, the register setting is copied into the TDHCLKRx, TDHSRx, and TDHCLKTRIMx registers. In addition to these settings, a clock multiplication factor and coarse clock range selection must be applied to the TDHMx and TDHCLKCR bits of the TDHCTL0 register (see Table 1-1).

Table 1-1. Factory Preprogrammed Frequency and TDHMx, TDHCLKCR Bit Settings

Frequency	TLV Label	TDHMx	TDHCLKCR	Description
64	TDHxCTL1_64	00	0	Local clock generator of Timer_Dx generates a frequency of 64 MHz = 8 × 8 MHz.
128	TDHxCTL1_128	00	1	Local clock generator of Timer_Dx generates a frequency of 128 MHz = 8 × 16 MHz.
200	TDHxCTL1_200	00	1	Local clock generator of Timer_Dx generates a frequency of 200 MHz = 8 × 25 MHz.
256	TDHxCTL1_256	01	1	Local clock generator of Timer_Dx generates a frequency of 256 MHz = 16 × 16 MHz.

1.2.2.2 Change Frequency of High-Resolution Generator (Free-Running Mode)

To change the frequency of the high-resolution generator in free-running mode:

1. Increment or decrement TDHCLKTRIM by 1 until TDHCLKTRIM = 64 is reached. If the clock range must be changed, proceed to Step 1a for a higher clock range or to Step 1b for a lower clock range.
    1. If the clock range, TDHCLKRx, must be changed to a higher clock range, then the TDHCLKSR must be brought to TDHCLKSR = 31 by incrementing by 1.
    2. If the clock range, TDHCLKRx, must be changed to a lower clock range, then the TDHCLKSR must be brought to TDHCLKSR = 0 by decrementing by 1.
2. Change the clock range, TDHCLKR, after step 1a or 1b. Increment or decrement by 1 at a time.
3. Increment or decrement TDHCLKSR by 1 until the desired clock frequency is reached to an accuracy of approximately ±3%.
4. Increment or decrement TDHCLKTRIM by 1 until the desired clock frequency is reached to an accuracy of approximately ±1.5%.

Further changes to track the frequency because of changes in the environment can be done by changing only the TDHCLTRIM values.

1.2.2.3 Regulated Mode of the High-Resolution Generator

In regulated mode, the high-resolution generator produces 8 or 16 equidistant events per timer input clock cycle. Regulation is enabled by setting the high-resolution calibration enable bit TDHREGEN. The high-resolution generator tracks changes of the timer input clock after locking to the timer input clock frequency. Locking is indicated by setting the lock interrupt flag TDHLKIFG. As long as the high-resolution generator is not locked, the interrupt flag TDHUNLKIFG is set.

If the timer input clock is out of the frequency range of the high-resolution generator, then the fail-high interrupt flag (TDHFHIFG) or the fail-low interrupt flag (TDHFLIFG) is set.

If the TDHREGEN bit is cleared, the continuous regulation is stopped and the high-resolution frequency enters free-running mode. The latest settings are kept.

Example 1-1 shows how to set the timer to high-resolution mode.

Example 1-1. Set the Timer to High-Resolution Mode

```
; Example: TDCLK=12MHz ; Generate the high-resolution frequency 12x16MHz
;
 MOV #TDHCLKCR_0+TDHCLKR_x+TDHSR_x+TDHCLKTRIM_x, &TDxHCTL1 ; pre-selected Clock Range
 MOV #TDSSEL_0, &TDxCTL0                                  ; TDCLK, High-Res input
                                                          ; clock
 MOV #TDCLKM_1, &TDxCTL1                                  ; Select High-Res Clock
 MOV #TDHM_1+TDHREGEN+TDHEN+TDHRON,&TDxHCTL0              ; 16x, Regulation
                                                          ; Waiting for TDHLKIFG
TD_HANDLER ...                                            ; Int handler Timer_D
 ADD &TDxIV, PC                                           ; Add offset to Jump table
 RETI                                                     ; Vector  0: No Interrupt
 ...                                                      ; Other vectors
 JMP TDHLK_HANDLER                                        ; Vector 24: Locked handler
 ...
TDHLK_HANDLER                                             ; Begin of Clock Locked handler
 MOV #MC_1, &TDxCTL0                                      ; Up Mode. Start Timer_D
 ...
 RETI
```

1.2.3 Starting the Timer

The timer may be started or restarted by any of the following methods:

- The timer counts when MCx > 0 and the clock source is active.
- TDHEN = 0: When the timer mode is either up or up/down, the timer may be stopped by loading 0 to TDxCL0. The timer may then be restarted by loading a nonzero value to TDxCL0. In this case, the timer starts incrementing in the up direction from zero.
- TDHEN = 1: When the timer mode is in either up mode or up/down mode, the timer may be stopped by loading 0 to TDxCL0. The timer may then be restarted by loading a nonzero value to TDxCL0. In this case, the timer starts incrementing in the up direction from zero.

1.2.4 Timer Mode Control

The timer has four modes of operation: stop, up, continuous, and up/down. The operating mode is selected with the MSCx bits (see Table 1-2).

Table 1-2. Timer Modes

MCx	Mode	Description
00	Stop	The timer is halted.
01	Up	The timer repeatedly counts from zero to the value of compare register TDxCL0.
10	Continuous	The timer repeatedly counts from zero to the value selected by the TDCNTLx bits.
11	Up/down	The timer repeatedly counts from zero up to the value of TDxCL0 and then back down to zero.

1.2.4.1 Up Mode

The up mode is used if the timer period must be different from $TDR_{(max)}$ counts. The timer repeatedly counts up to the value of compare latch TDxCL0, which defines the period (see Figure 1-3). The number of timer counts in the period is TDCL0 + 1. When the timer value equals TDCL0, the timer restarts counting from zero. If up mode is selected when the timer value is greater than TDCL0, the timer immediately restarts counting from zero.

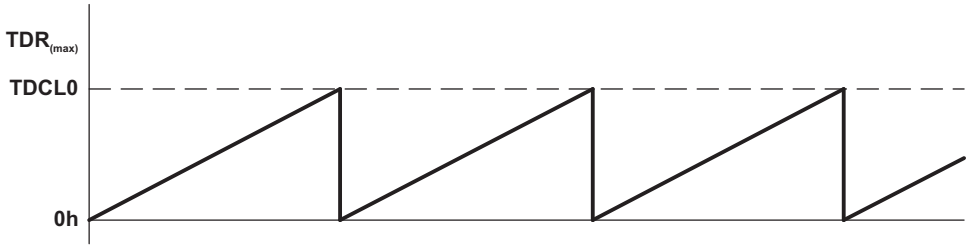

Figure 1-3. Up Mode

The TDxCCR0 CCIFG interrupt flag is set when the timer *counts* to the TDCL0 value. The TDIFG interrupt flag is set when the timer *counts* from TDCL0 to zero. Figure 1-4 shows the flag set cycle.

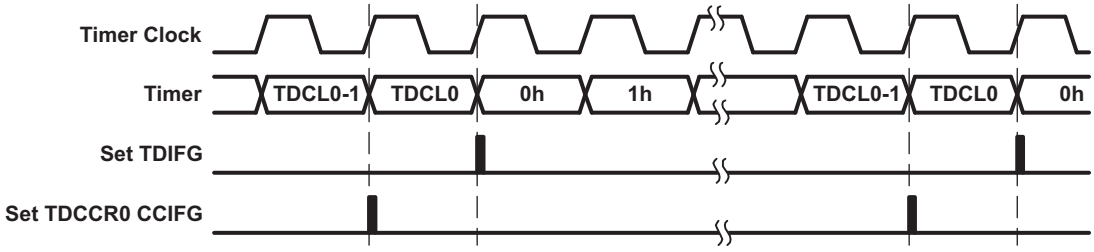

Figure 1-4. Up Mode Flag Setting

1.2.4.1.1 Changing Period Register TDxCL0

When changing TDxCL0 while the timer is running and when the TDxCL0 load mode is *immediate*, if the new period is greater than or equal to the old period or greater than the current count value, the timer counts up to the new period. If the new period is less than the current count value, the timer rolls to zero. However, one additional count may occur before the counter rolls to zero.

NOTE: When TDxCL0 is used as period register under high-resolution mode, the four LSB of the TDxCL0 in the 16x case or the three LSB in the 8x case are ignored.

1.2.4.2 Continuous Mode

In continuous mode, the timer repeatedly counts up to $TDR_{(max)}$ and restarts from zero (see Figure 1-5). The compare latch TDxCL0 works the same way as the other capture/compare registers.

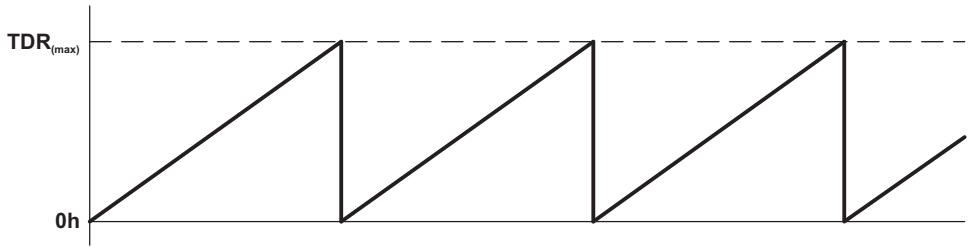

Figure 1-5. Continuous Mode

The TDIFG interrupt flag is set when the timer *counts* from $TDR_{(max)}$ to zero. Figure 1-6 shows the flag set cycle.

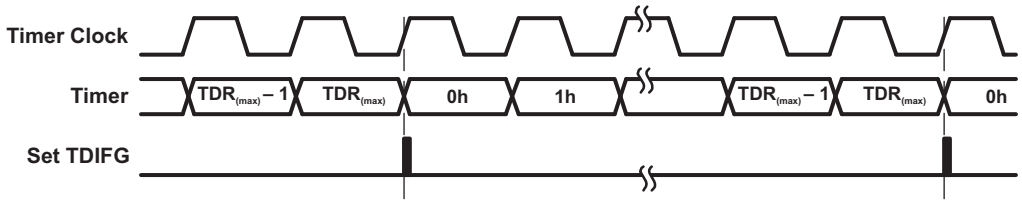

Figure 1-6. Continuous Mode Flag Setting

1.2.4.3 Use of Continuous Mode

The continuous mode can be used to generate independent time intervals and output frequencies. Each time an interval is completed, an interrupt is generated. The next time interval is added to the TDCLx latch in the interrupt service routine. Figure 1-7 shows two separate time intervals, t_0 and t_1, being added to the capture/compare registers. The time interval is controlled by hardware, not software, without impact from interrupt latency. Up to three (for Timer_D3) or 7 (for Timer_D7) independent time intervals or output frequencies can be generated using capture/compare registers.

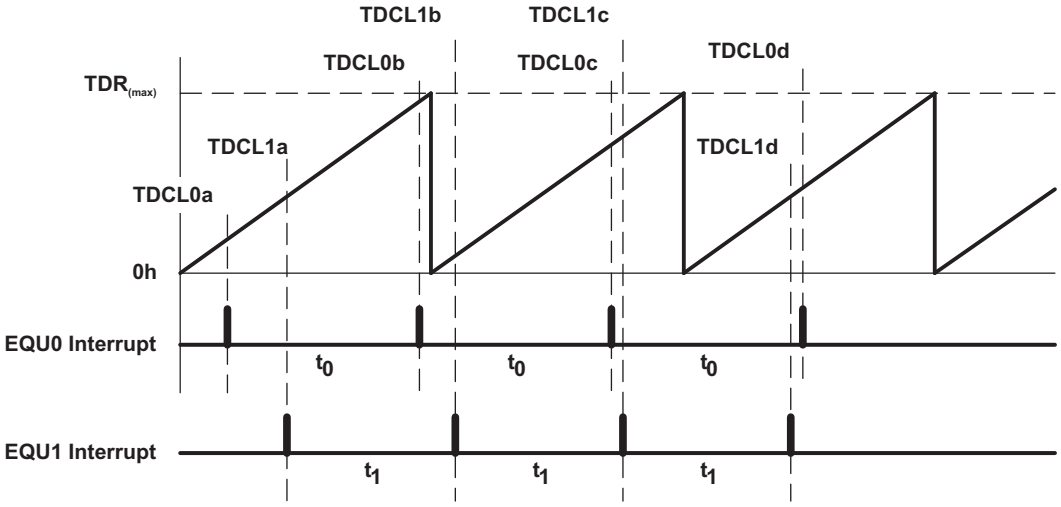

Figure 1-7. Continuous Mode Time Intervals

Time intervals can be produced with other modes as well, using TDxCL0 as the period register. Their handling is more complex, because the sum of the old TDCLx data and the new period can be higher than the TDxCL0 value. When the sum of the previous TDCLx value plus t_x is greater than the TDCL0 data, the old TDCL0 value must be subtracted to obtain the correct time interval.

PWM output can be generated in capture/compare channel 0 by using the timer overflow signal together with TDCL0. Figure 1-8 shows an example.

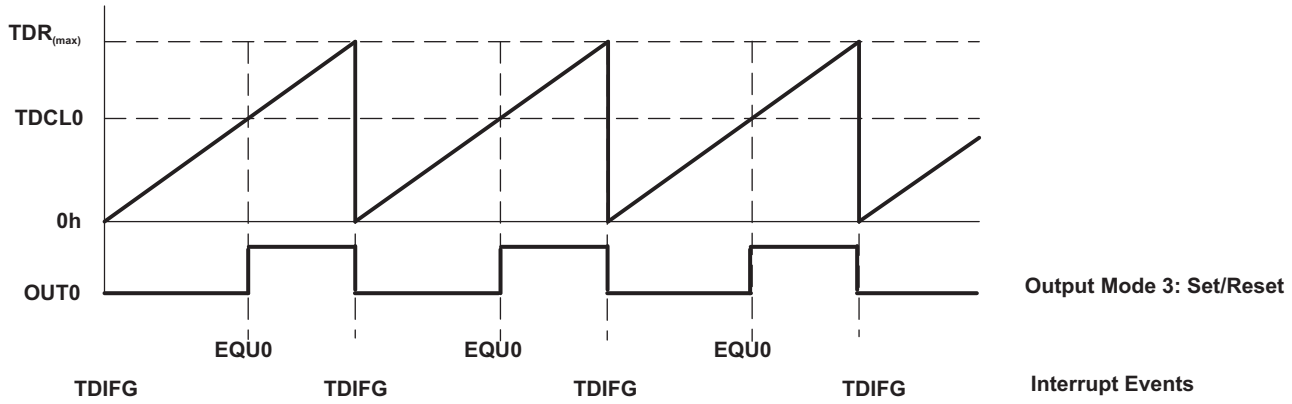

Figure 1-8. TDxCCR0 PWM Generation Under Continuous Mode

1.2.4.4 Up/Down Mode

The up/down mode is used if the timer period must be different from $TDR_{(max)}$ counts and if symmetrical pulse generation is needed. The timer repeatedly counts up to the value of compare latch TDxCL0, and back down to zero (see Figure 1-9). The period is twice the value in TDCL0.

NOTE: TDCL0 > $TDR_{(max)}$

If TDCL0 > $TDR_{(max)}$, the counter operates as if it were configured for continuous mode. It does not count down from $TDR_{(max)}$ to zero.

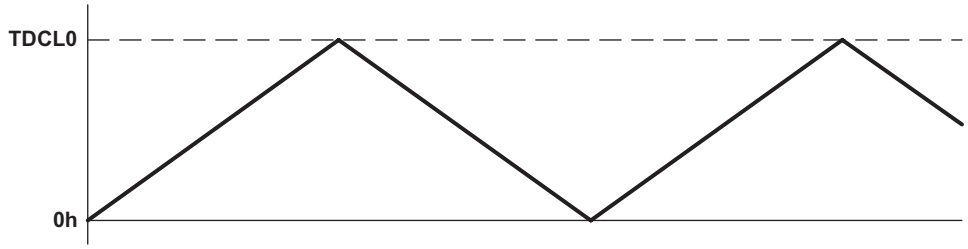

Figure 1-9. Up/Down Mode

The count direction is latched. This allows the timer to be stopped and then restarted in the same direction it was counting before it was stopped. If this is not desired, the TDCLR bit must be used to clear the direction. The TDCLR bit also clears the TDR value and the TDCLK divider.

In up/down mode, the TDxCCR0 CCIFG interrupt flag and the TDIFG interrupt flag are set only once during the period, separated by one-half the timer period. The TDxCCR0 CCIFG interrupt flag is set when the timer *counts* from TDCL0 – 1 to TDCL0, and TDIFG is set when the timer completes *counting* down from 0001h to 0000h. Figure 1-10 shows the flag set cycle.

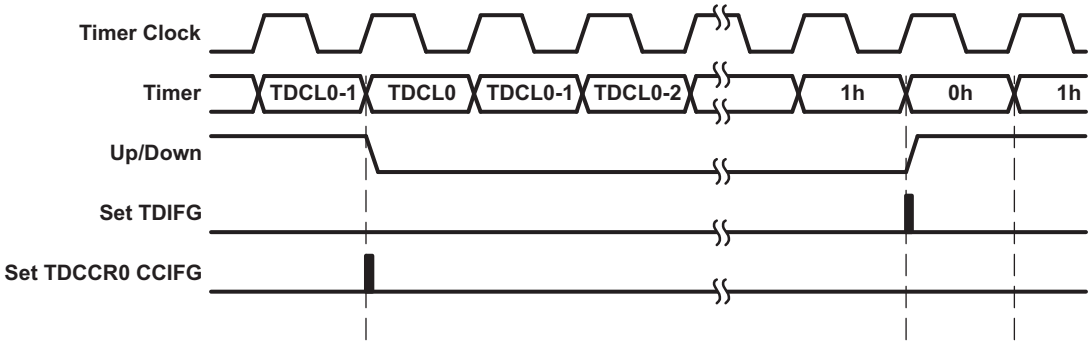

Figure 1-10. Up/Down Mode Flag Setting

1.2.4.4.1 Changing the Value of Period Register TDxCL0

When changing TDxCL0 while the timer is running and counting in the down direction, and when the TDxCL0 load mode is *immediate*, the timer continues its descent until it reaches zero. The new period takes effect after the counter counts down to zero.

If the timer is counting in the up direction when the new period is latched into TDxCL0, and the new period is greater than or equal to the old period or greater than the current count value, the timer counts up to the new period before counting down. When the timer is counting in the up direction and the new period is less than the current count value when TDxCL0 is loaded, the timer begins counting down. However, one additional count may occur before the counter begins counting down.

1.2.4.5 Use of Up/Down Mode

The up/down mode supports applications that require dead times between output signals (see Section 1.2.9). For example, to avoid overload conditions, two outputs driving an H-bridge must never be in a high state simultaneously. In the example shown in Figure 1-11, the t_{dead} is:

$t_{dead} = t_{timer} \times (TDCL1 – TDCL3)$

Where:

t_{dead} = Time during which both outputs need to be inactive
t_{timer} = Cycle time of the timer clock
TDCLx = Content of compare latch x

The ability to simultaneously load grouped compare latches ensures the dead times.

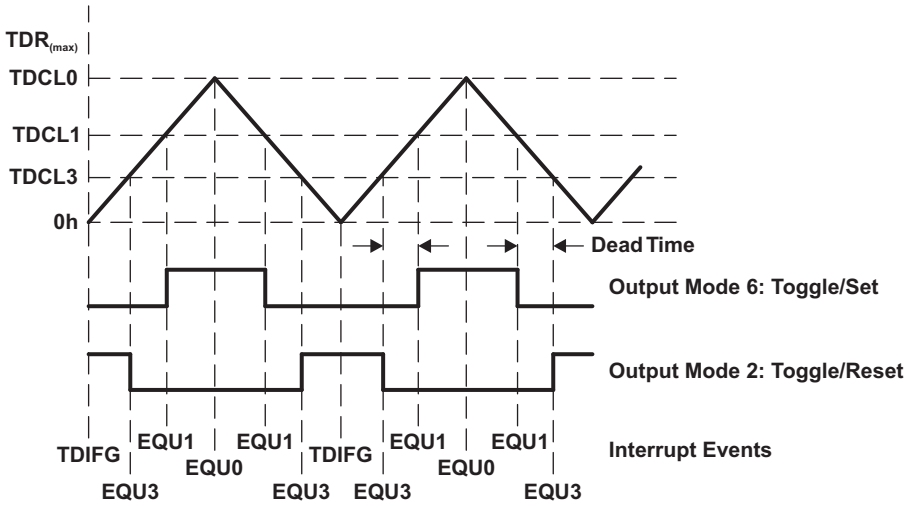

Figure 1-11. Output Unit in Up/Down Mode

1.2.5 PWM Generation

The previous examples have shown that PWM output can be produced by using a single TDCCRx. In TDxCCR0, for example, either the falling or the rising edge of the PWM signal is controlled by the TDxCCR0 or the period overflow (see Figure 1-8). In the other TDCCRx (x ≠ 0), PWM can be generated by setting the output to up/down mode and using the TDCCRx register to control both the rising and the falling edges of the signal (see Figure 1-11). However, this cannot always meet the application requirements; for example, in motor control applications.

The control bits, TDxCMB of the TDxCTL1 register, are implemented so that two neighbor TDCCR registers can be used for one channel to control both edges of one PWM output signal. If TDxCMB is set, $TDCCR_{x-1}$ and $TDCCR_x$ are combined to channel TD_x; for example, TDxCCR1 and TDxCCR2 for TD2 output, TDxCCR3 and TDxCCR4 for TD4 output, and TDxCCR5 and TDxCCR6 for TD6. By configuring TDxCMB bit, a Timer_D5 can then generate either two fully controlled PWM signals or four independent channels.

The PWM duty cycle can be zero. The smallest pulse width depends on the TDHEN and TDHMx settings.

- In normal timer mode, TDHEN = 0, TDHMx = don't care: CCRx = 1
- In high-resolution timer mode, 8x reference frequency, TDHEN = 1, TDHMx = 00: CCRx = 0x0010, 8 high-resolution cycles
- In high-resolution timer mode, 16x reference frequency, TDHEN = 1, TDHMx = 01: CCRx = 0x0020, 16 high-resolution cycles

Figure 1-12 shows an example.

Table 1-3 and Table 1-4 summarize the duty cycle limitations of the Timer_D. Duty cycles close to 0% and 100% should be avoided. Table 1-3 and Table 1-4 show how the output signal behaves in case the TDCCRx register is programmed to values beyond the recommended range of values dependent on the OUTMODx setting.

Table 1-3. High-Resolution Mode Limitation (TDHEN = 1) - Minimum Duty Cycle

OUTMODx	Mode	Condition	Description
000	Output	n/a	n/a
001	Set	TDHMx = 0: TDCCRx < 0x0007 TDHMx = 1: TDCCRx < 0x000F	Output is not set.
010	Toggle/Reset	-	Not recommended.
011	Set/Reset	TDHMx = 0: TDCCRx < 0x0007 TDHMx = 1: TDCCRx < 0x000F	Result is 100% duty cycle.
100	Toggle	-	Not recommended.
101	Reset	TDHMx = 0: TDCCRx < 0x0007 TDHMx = 1: TDCCRx < 0x000F	Output is not reset.
110	Toggle/Set	-	Not recommended.
111	Reset/Set	TDHMx = 0: TDCCRx < 0x0007 TDHMx = 1: TDCCRx < 0x000F	Result is 0% duty cycle.

Table 1-4. High-Resolution Mode Limitation (TDHEN = 1) - Maximum Duty Cycle

OUTMODx	Mode	Condition	Description
000	Output	n/a	n/a
001	Set	TDCCRx > TDCCR0 - 0x0008	Output is not set.
010	Toggle/Reset	-	Not recommended.
011	Set/Reset	TDCCRx > TDCCR0 - 0x0008	Result is 100% duty cycle.
100	Toggle	-	Not recommended.
101	Reset	TDCCRx > TDCCR0 - 0x0008	Output is not reset.
110	Toggle/Set	-	Not recommended.
111	Reset/Set	TDCCRx > TDCCR0 - 0x0008	Result is 0% duty cycle.

In high-resolution compare mode there are limitations of the minimum and maximum duty cycles.

Minimum duty cycle:

TDHMx = 0: TDCCRx < 8 are set to TDCCRx = 0; that is, return 0 duty cycle.

TDHMX = 1: TDCCRx < 16 are set to TDCCRx = 0; that is, return 0 duty cycle.

Maximum duty cycle:

TDHMx = 0 or 1: The maximum duty cycle is limited to TDCCRx < TDCCR0 - 0x000F. Values of TDCCRx greater than TDCCR0 - 0x000F lead to 0 duty cycle.

NOTE: In high-resolution mode, when the two CCRx channels are combined by setting TDxCMB = 1, the minimum difference between the output compare registers (CCRx and CCRx + 1 must be:

if TDHMx = 00: $| TDCCR_x - TDCCR_{x-1} | > 7$

if TDHMx = 01: $| TDCCR_x - TDCCR_{x-1} | > 15$

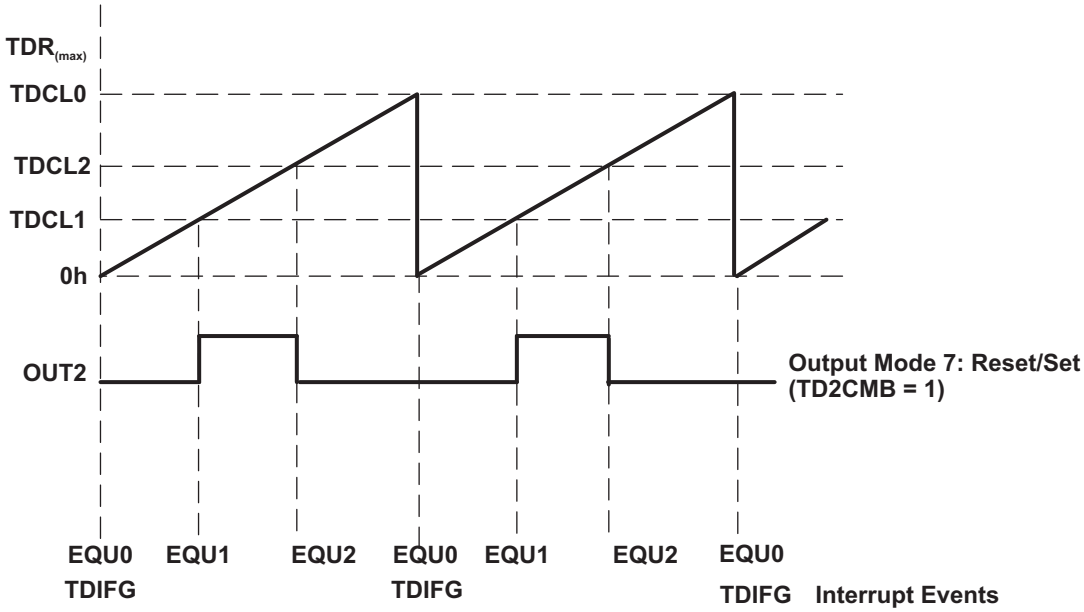

Figure 1-12. Controlling Rising and Falling Edge of PWM Output in Up Mode

Example 1-2 shows how to set up the timer for the case shown in Figure 1-12.

Example 1-2.

```
MOV #TD2CMB+TDCLKM_0, &TDxCTL1              ; Combine TDxCCR1&TDxCCR2,
                                            ; External Clock
MOV #OUTMOD_7, &TDxCCTL2                    ; TDxCCR2 Reset/Set
MOV 0x80, &TDxCCR0                          ; PWM Period TDxCCR0 128
MOV 0x28, &TDxCCR1                          ; TDxCCR1 is 40, Set
MOV 0x50, &TDxCCR2                          ; TDxCCR2 is 80, Reset
MOV #TDSSEL_1+MC_1, &TDxCTL0                ; ACLK, Up Mode
BIS #CPUOFF,SR                              ; Enter LPM0
```

NOTE: Channels TD1, TD3, and TD5 are still controlled by TDxCCR1, TDxCCR3, and TDxCCR5, respectively, even when the TDxCCRx registers are paired.

Combining two pairs of CCRx channels to two PWM outputs can be used to generate non-overlapping PWM channels.

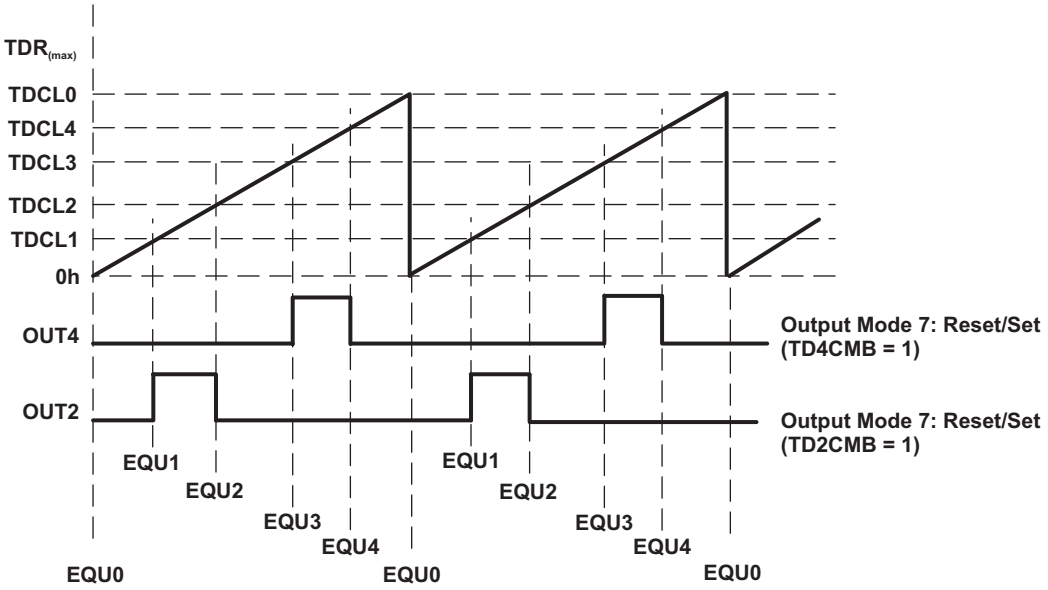

Figure 1-13. Deadband Generation (TDxCMB = 1)

1.2.6 Capture/Compare Blocks

Several identical capture/compare blocks, TDCCRx, are present in Timer_D. Any of the blocks may be used to capture the timer data or to generate time intervals. See the device-specific data sheet for the number of capture/compare blocks available.

1.2.6.1 Capture Mode

The capture mode is used to record time events. It can be used for speed computations or time measurements. The capture inputs CCIxA and CCIxB are connected to external pins or internal signals and are selected with the CCISx bits. The CMx bits select the capture edge of the input signal as rising, falling, or both. A capture occurs on the selected edge of the input signal. If a capture is performed:

- The timer value is copied into the TDCCRx register.
- The TDCCRx register is copied into the TDCLx register.
- The interrupt flag CCIFG is set.
- The Capture Overflow bit is set at a capture overflow condition.

The input signal level can be read at any time via the CCI bit. Devices may have different signals connected to CCIxA and CCIxB. See the device-specific data sheet for the connections of these signals.

The capture signal can be asynchronous to the timer clock and cause a race condition. Setting the SCS bit synchronizes the capture with the next timer clock. Setting the SCS bit to synchronize the capture signal with the timer clock is recommended (see Figure 1-14). In high-resolution mode, the SCS bit is ignored.

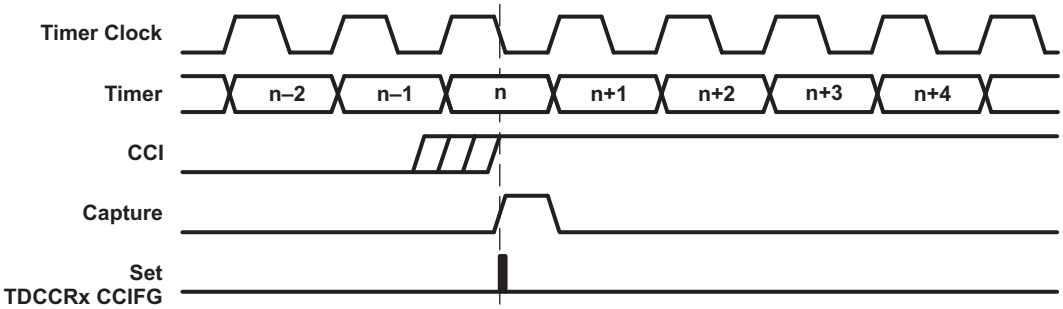

Figure 1-14. Capture Signal (SCS = 1)

Overflow logic is provided in each capture/compare register to indicate a capture overflow condition.

1.2.6.1.1 Single Capture Mode (TDCAPMx = 0)

Each input capture channel can be operated in single capture (compatibility) mode or in dual capture mode. The single capture mode is selected by TDCAPMx = 0. In single capture mode, the content of register TDCCRx is shifted to TDCLx register at each capture event. In single capture mode, the capture overflow bit (COV) is set if a capture events occurs before the TDCCRx register has been read (see Figure 1-15). COV must be reset by software.

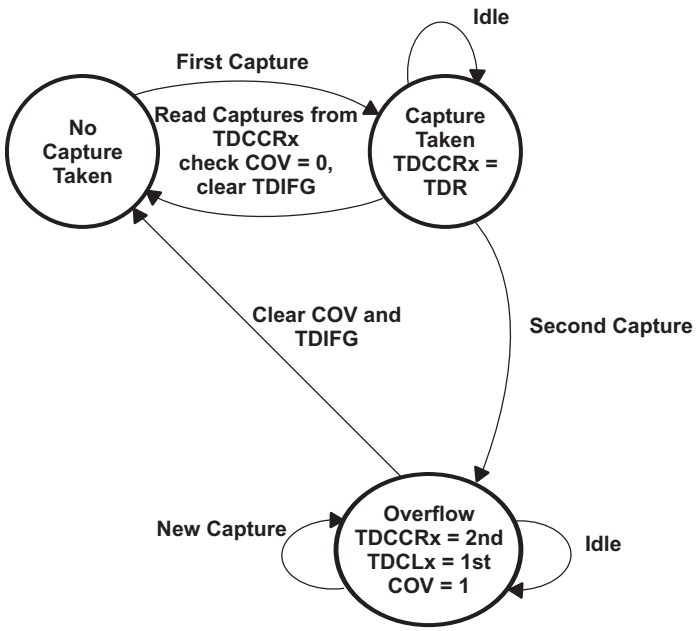

Figure 1-15. Single Capture Cycle

1.2.6.1.2 Dual Capture Mode (TDCAPMx = 1)

Each input capture channel can be operated in single capture (compatibility) mode or in dual capture mode. The dual capture mode is selected by TDCAPMx = 1. In dual capture mode, the register TDCCRx content is shifted to the TDCLx register per each capture event; that is, the second capture event shifts the first capture value from TDCCRx into the TDCLx register. Then TDCCRx holds the second capture value while TDCLx holds the first, and the CCIFG interrupt flag is set. To make sure there is no earlier captured value in the TDCCRx and TDCLx registers these registers should be read prior to using the dual capture mode. The COV bit is set at the third capture event if both TDCCRx and TDCLx have not been read prior to the third capture event (see Figure 1-16). The third capture event again shifts the TDCCRx register content to the TDCLx register. The first capture value is no longer available.

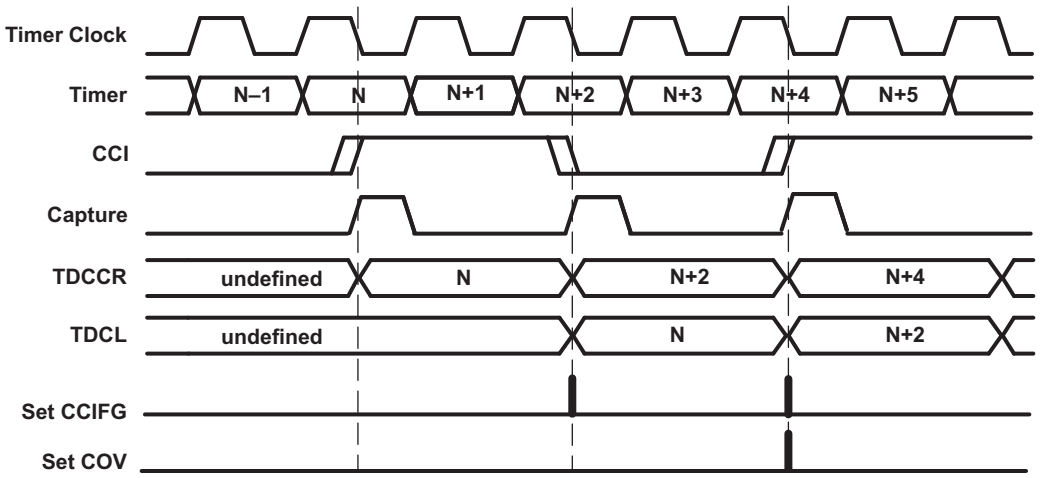

Figure 1-16. Sequential Capture Events in Dual Capture Mode

The COV bit is set at the third capture if TDCCRx and TDCLx are not read prior to the third capture event (see Figure 1-17). Checking for COV = 0 after reading the capture registers TDCCRx and TDCLx validates the register content.

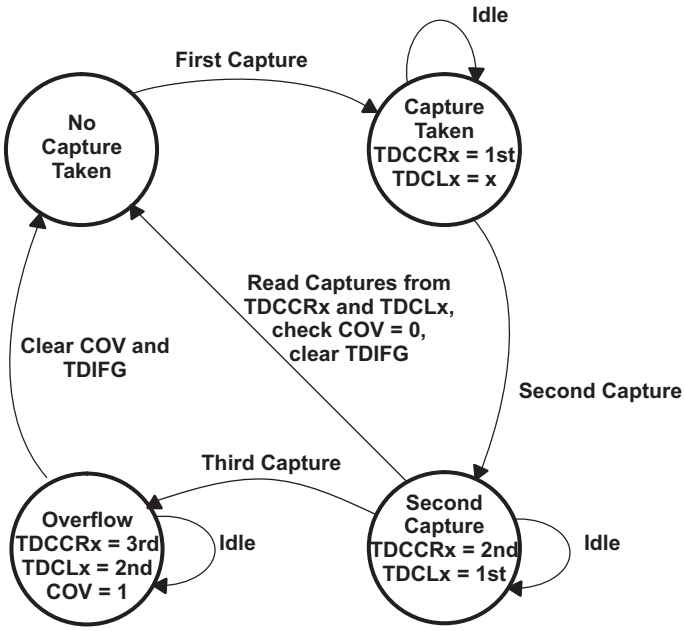

Figure 1-17. COV in Dual Capture Mode

1.2.6.1.3 Capture Initiated by Software

Captures can be initiated by software. The CMx bits can be set for capture on both edges. Software then sets bit CCIS1 = 1 and toggles bit CCIS0 to switch the capture signal between V_{CC} and GND, initiating a capture each time CCIS0 changes state:

```
MOV #CAP+SCS+CCIS1+CM_3,&TDxCCTLx          ; Set up TDCCTLx
XOR #CCIS0,&TDCCTLx                        ; TDCCRx = TDR
```

1.2.7 Compare Mode

The compare mode is selected when CAP = 0. Compare mode is used to generate PWM output signals or interrupts at specific time intervals. When TDxR *counts* to the value in a TDCLx:

- Interrupt flag CCIFG is set.
- Internal signal EQUx = 1.
- EQUx affects the output according to the output mode.

1.2.7.1 Compare Latch TDCLx

The TDCCRx compare latch, TDCLx, holds the data for the comparison to the timer value in compare mode. TDCLx is buffered by TDCCRx. The buffered compare latch gives the user control over when a compare period updates. Compare data is written to each TDCCRx and automatically transferred to TDCLx. The timing of the transfer from TDCCRx to TDCLx is user selectable by setting the CLLDx bits as described in Table 1-5.

Table 1-5. TDCLx Load Events

CLLDx	Description
00	New data is transferred from TDCCRx to TDCLx immediately when TDCCRx is written to.
01	New data is transferred from TDCCRx to TDCLx when TDxR *counts* to 0.
10	New data is transferred from TDCCRx to TDCLx when TDxR *counts* to 0 for up and continuous modes. New data is transferred to from TDCCRx to TDCLx when TDxR *counts* to the old TDCL0 value or to 0 for up/down mode.
11	New data is transferred from TDCCRx to TDCLx when TDxR *counts* to the old TDCLx value.

1.2.7.2 Grouping Compare Latches

Multiple compare latches may be grouped together for simultaneous updates with the TDCLGRPx bits. When using groups, the CLLDx bits of the lowest numbered TDCCRx in the group determine the load event for each compare latch of the group, except when TDCLGRP = 3 (see Table 1-6). The CLLDx bits of the controlling TDCCRx must not be set to zero. When the CLLDx bits of the controlling TDCCRx are set to zero, all compare latches update immediately when their corresponding TDCCRx is written; no compare latches are grouped.

Two conditions must exist for the compare latches to be loaded when grouped. First, all TDCCRx registers of the group must be updated, even when new TDCCRx data equals the old TDCCRx data. Second, the load event must occur.

Table 1-6. Compare Latch Operating Modes

TDCLGRPx	Grouping	Update Control
00	None	Individual
01	TDxCL1+TDxCL2 TDxCL3+TDxCL4 TDxCL5+TDxCL6	TDxCCR1 TDxCCR3 TDxCCR5
10	TDxCL1+TDxCL2+TDxCL3 TDxCL4+TDxCL5+TDxCL6	TDxCCR1 TDxCCR4
11	TDxCL0+TDxCL1+TDxCL2+TDxCL3+TDxCL4+TDxCL5+TDxCL6	TDxCCR1

1.2.8 Switching From Capture to Compare Mode

As discussed in Section 1.2.7, the TDCLx register holds the first captured value when a second capture is taking place. Therefore, when the capture/compare block switches from capture mode to compare mode, TDCLx may have a non-zero initial value. It is important to clear the TDCLx before use or to update the TDCLx immediately when TDCCRx is written; that is, set CLLDx = 0x0.

1.2.9 Output Unit

Each capture/compare block contains an output unit. The output unit is used to generate output signals, such as PWM signals. Each output unit has eight operating modes that generate signals based on the EQU0 and EQUx signals. The TDOUTH pin function can be used to put all Timer_D outputs into a high-impedance state. When the TDOUTH pin function is selected for the pin (corresponding PSEL bit is set, and port configured as input) and when the pin is pulled high, all Timer_D outputs are in a high-impedance state.

1.2.9.1 Output Modes

Table 1-7 lists the output modes defined by the OUTMODx bits. The OUTx signal is changed with the rising edge of the timer clock for all modes except mode 0. In , the term TDyR period match is used. The period match in continuous mode is defined by the counter length (TDyCNTLx), and in Up mode and Up/Down mode by TDyCL0 register.

Table 1-7. Output Modes

OUTMODx	Mode	Description
000	Output	The output signal OUTx is defined by the OUTx bit. The OUTx signal updates immediately when OUTx is updated.
001	Set	Set events: Timer counts to TDyCLx, external fault (TECyFLTx). It remains set until a reset of the timer, or until another output mode is selected and affects the output.
010	Toggle/Reset	TDxCMB = 0: Toggle events: Timer counts to TDyCLx, external fault (TECyFLTx). Reset events: Timer counts to TDyCLx, TDyR period match, external fault (TECyFLT0), external clear (TECyCLR). TD2CMB = 1: Toggle events: Timer counts to TDyCL2, external fault (TECyFLT2). Reset events: Timer counts to TDyCL1, external fault (TECyFLT1), external clear (TECyCLR). TD4CMB = 1: Toggle events: Timer counts to TDyCL4, external fault (TECyFLT4). Reset events: Timer counts to TDyCL3, external fault (TECyFLT3), external clear (TECyCLR). TD6CMB = 1: Toggle events: Timer counts to TDyCL6, external fault (TECyFLT6). Reset events: Timer counts to TDyCL5, external fault (TECyFLT5), external clear (TECyCLR).
011	Set/Reset	TDxCMB = 0: Set events: Timer counts to TDyCLx, external fault (TECyFLTx). Reset events: Timer counts to TDyCLx, TDyR period match, external fault (TECyFLT0), external clear (TECyCLR). TD2CMB = 1: Set events: Timer counts to TDyCL2, external fault (TECyFLT2). Reset events: Timer counts to TDyCL1, external fault (TECyFLT1),, external clear (TECyCLR). TD4CMB = 1: Set events: Timer counts to TDyCL4, external fault (TECyFLT4). Reset events: Timer counts to TDyCL3, external fault (TECyFLT3), external clear (TECyCLR). TD6CMB = 1: Set events: Timer counts to TDyCL6, external fault (TECyFLT6). Reset events: Timer counts to TDyCL5, external fault (TECyFLT5), external clear (TECyCLR).
100	Toggle	Toggle events: Timer counts to TDyCLx, external fault (TECyFLTx). The output period is twice the timer period.
101	Reset	Reset events: Timer counts to TDyCLx, external fault (TECyFLTx). It remains reset until another output mode is selected and affects the output.

Table 1-7. Output Modes (continued)

OUTMODx	Mode	Description
110	Toggle/Set	TDxCMB = 0: Toggle events: Timer counts to TDyCLx, external fault (TECyFLTx). Set events: Timer counts to TDyCLx, TDyR period match, external fault (TECyFLT0), external clear (TECyCLR). TD2CMB = 1: Toggle events: Timer counts to TDyCL2, external fault (TECyFLT2). Set events: Timer counts to TDyCL1, external fault (TECyFLT1), external clear (TECyCLR). TD4CMB = 1: Toggle events: Timer counts to TDyCL4, external fault (TECyFLT4). Set events: Timer counts to TDyCL3, external fault (TECyFLT3), external clear (TECyCLR). TD6CMB = 1: Toggle events: Timer counts to TDyCL6, external fault (TECyFLT6). Set events: Timer counts to TDyCL5, external fault (TECyFLT5), external clear (TECyCLR).TDxCMB=0:
111	Reset/Set	TDxCMB = 0: Reset events: Timer counts to TDyCLx, external fault (TECyFLTx). Set events: Timer counts to TDyCLx, TDyR period match, external fault (TECyFLT0), external clear (TECyCLR). TD2CMB = 1: Reset events: Timer counts to TDyCL2, external fault (TECyFLT2). Set events: Timer counts to TDyCL1, external fault (TECyFLT1), external clear (TECyCLR). TD4CMB = 1: Reset events: Timer counts to TDyCL4, external fault (TECyFLT4). Set events: Timer counts to TDyCL3, external fault (TECyFLT3), external clear (TECyCLR). TD6CMB = 1: Reset events: Timer counts to TDyCL6, external fault (TECyFLT6). Set events: Timer counts to TDyCL5, external fault (TECyFLT5), external clear (TECyCLR).TDxCMB=0:

1.2.9.1.1 Output Example – Timer in Up Mode

The OUTx signal is changed when the timer *counts* up to the TDCLx value, and rolls from TDCL0 to zero, depending on the output mode. An example is shown in Figure 1-18 using TDxCL0 and TDxCL1.

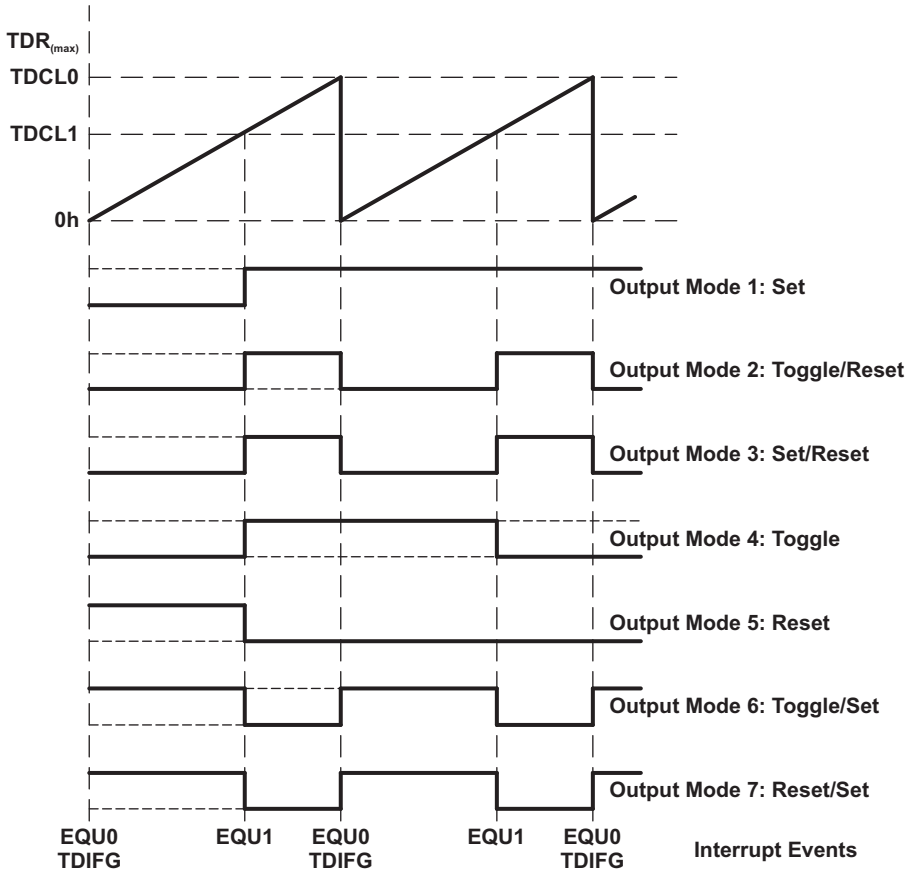

Figure 1-18. Output Example, Channel 1 – Timer in Up Mode

The OUTx signal is changed when the timer *counts* up to the TDCLx value, and rolls from TDCL0 to zero, depending on the output mode. The activity selected for the **TDCCRx match** event (TDR = TDCCLx) occurs at the point in time where the **external fault** event happens. An example is shown in Figure 1-19 using TDxCL0 and TDxCL1.

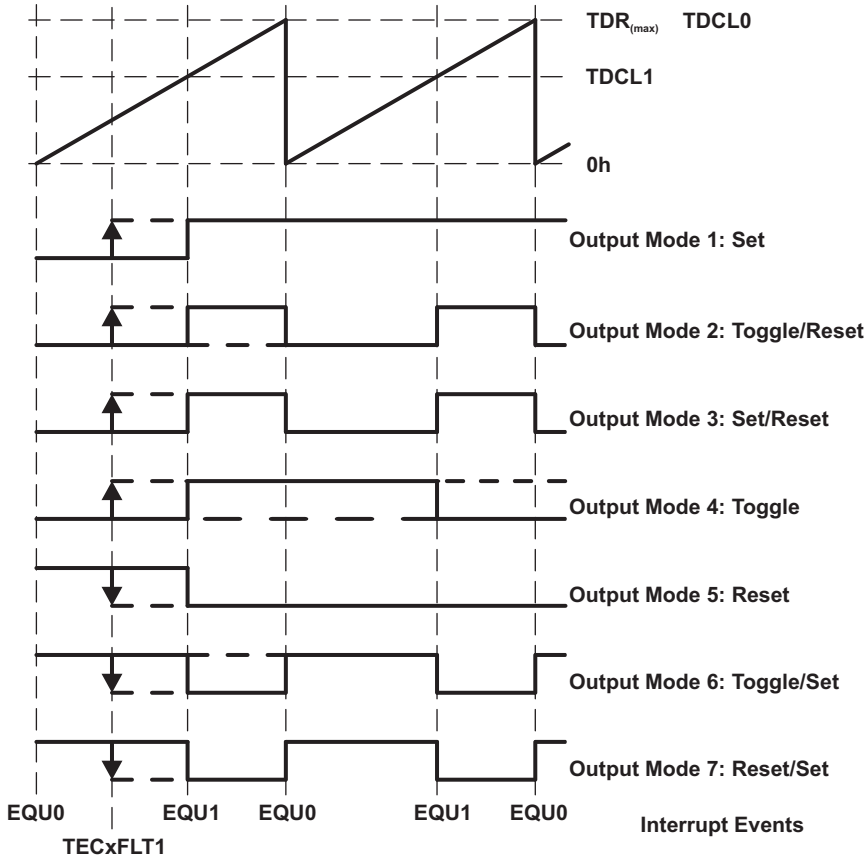

Figure 1-19. Output Example, Channel 1 - Timer in Up Mode With External Fault Signal

The OUTx signal is changed when the timer *counts* up to the TDCLx value, and rolls from TDCL0 to zero, depending on the output mode. The activity selected for the **period match** event (TDR = TDCCL0) occurs at the point in time when the external **clear event** happens. The external clear event restarts the timer counter. As a consequence, the next period starts earlier, and all following events happen earlier as well. An example is shown in Figure 1-20 using TDxCL0 and TDxCL1.

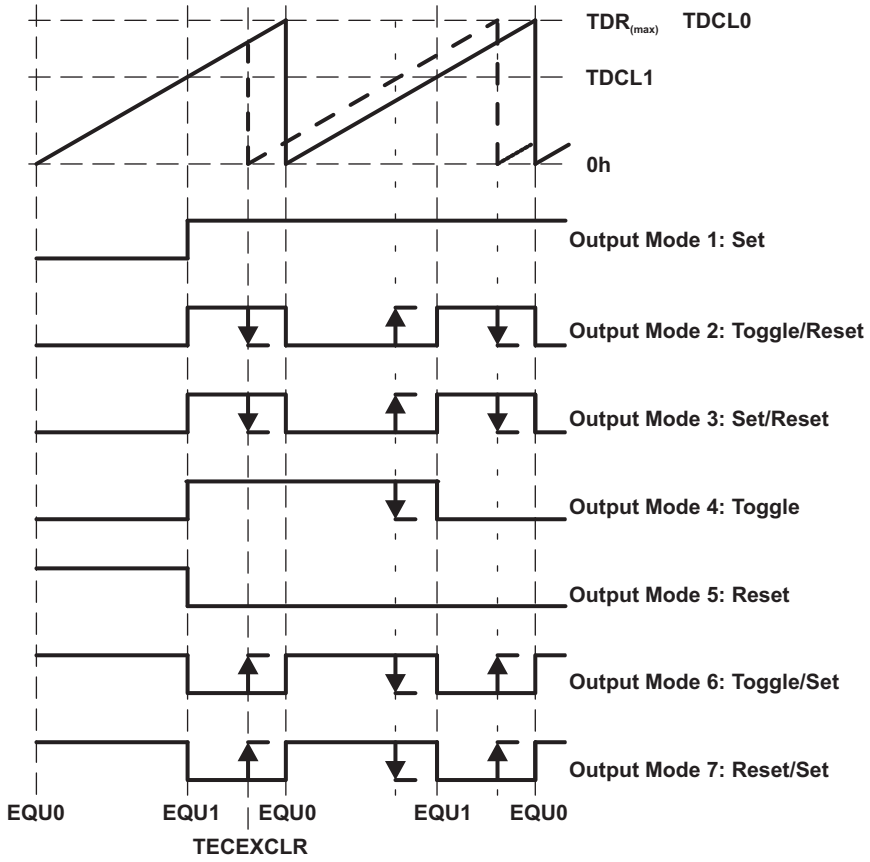

Figure 1-20. Output Example - Timer in Up Mode with External Timer Clear Signal

1.2.9.1.2 Output Example – Timer in Continuous Mode

The OUTx signal is changed when the timer reaches the TDCLx and TDCL0 values, depending on the output mode. An example is shown in Figure 1-21 using TDxCL0 and TDxCL1. The external fault and external clear signals have the same impact to the output signals as in Up mode as shown in and .

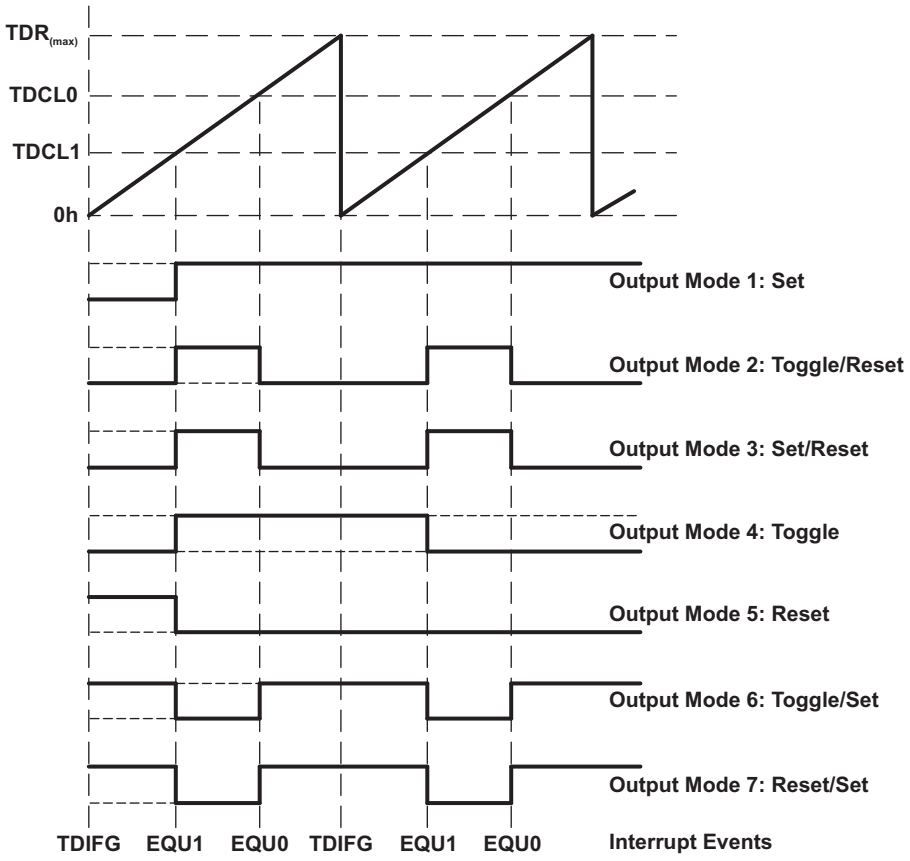

Figure 1-21. Output Example – Timer in Continuous Mode

1.2.9.1.3 Output Example – Timer in Up/Down Mode

The OUTx signal changes when the timer equals TDCLx in either count direction and when the timer equals TDCL0, depending on the output mode. An example is shown in Figure 1-22 using TDxCL0 and TDxCL3.

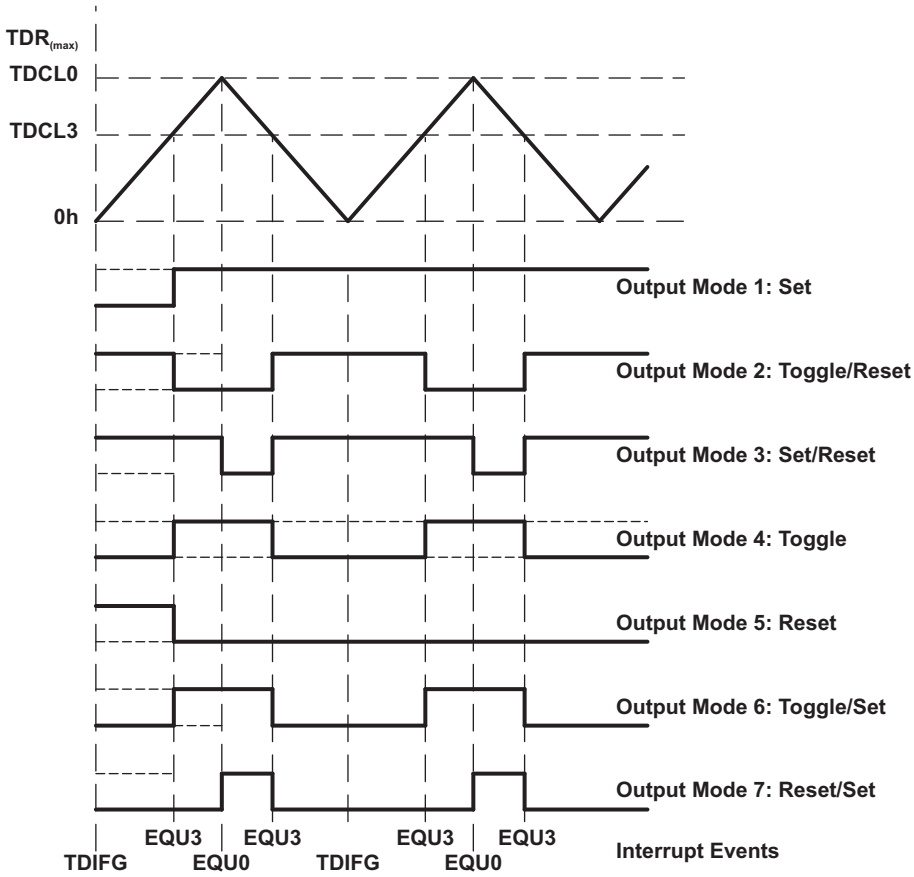

Figure 1-22. Output Example – Timer in Up/Down Mode

NOTE: Switching between output modes

When switching between output modes, one of the OUTMODx bits should remain set during the transition, unless switching to mode 0. Otherwise, output glitching can occur because a NOR gate decodes output mode 0. A safe method for switching between output modes is to use output mode 7 as a transition state:

```
BIS #OUTMOD_7,&TDCCTLx   ; Set output mode=7
BIC #OUTMODx,&TDCCTLx    ; Clear unwanted bits
```

1.2.10 Synchronization Between Timer_D Instances

For some devices that contain more than one Timer_D instance, internal signals assist the synchronization between different timers. Those signals include Timer Clock, TDAUXCLK, TDAUXCLROUT, and some other signals that connect the Timer_D module and the Timer Event Control module. See the TEC chapter for details.

1.2.11 Timer_D Interrupts

Two interrupt vectors are associated with the 16-bit Timer_D module:

- TDxCCR0 interrupt vector for TDxCCR0 CCIFG
- TDIV interrupt vector for all other interrupt flags

In capture mode, any CCIFG flag is set when a timer value is captured in the associated TDCCRx register. In compare mode, any CCIFG flag is set when TDxR *counts* to the associated TDCLx value. Software may also set or clear any CCIFG flag. All CCIFG flags request an interrupt when their corresponding CCIE bit and the GIE bit are set.

1.2.11.1 TDxCCR0 Interrupt Vector

The TDxCCR0 CCIFG flag has the highest Timer_D interrupt priority and has a dedicated interrupt vector (see Figure 1-23). The TDxCCR0 CCIFG flag is automatically reset when the TDxCCR0 interrupt request is serviced.

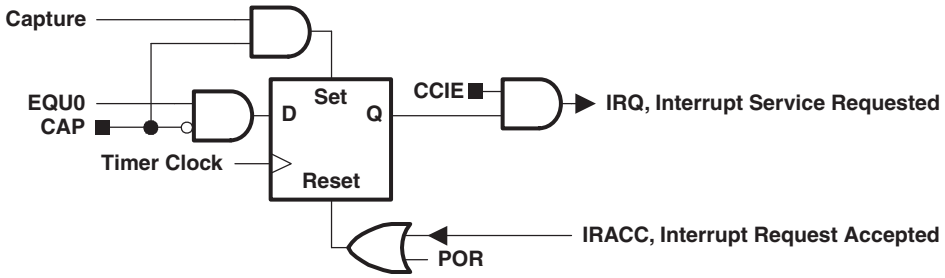

Figure 1-23. Capture/Compare TDxCCR0 Interrupt Flag

1.2.11.2 TDIV, Interrupt Vector Generator

The TDIFG flag, TDCCRx CCIFG flags (excluding TDxCCR0 CCIFG), and all of the high-resolution related interrupts are prioritized and combined to source a single interrupt vector. The interrupt vector register TDIV is used to determine which flag requested an interrupt.

The highest-priority enabled interrupt (excluding TDxCCR0 CCIFG) generates a number in the TDxIV register (see Section 1.3.11). This number can be evaluated or added to the program counter to automatically enter the appropriate software routine. Disabled Timer_D interrupts do not affect the TDIV value.

Read access of the TDIV register automatically resets the highest-pending interrupt flag. If another interrupt flag is set, another interrupt is immediately generated after servicing the initial interrupt. For example, if the TDxCCR1 and TDxCCR2 CCIFG flags are set when the interrupt service routine accesses the TDIV register, TDxCCR1 CCIFG is reset automatically. After the RETI instruction of the interrupt service routine is executed, the TDxCCR2 CCIFG flag generates another interrupt.

Write access of the TDIV register clears all pending interrupt conditions and flags.

1.2.11.3 TDIV, Interrupt Handler Examples

The following software example shows the recommended use of TDIV. The TDIV value is added to the PC to automatically jump to the appropriate routine.

Example 1-3 shows the recommended use of TDIV for Timer_D3.

Example 1-3.

```
; Interrupt handler for TDxCCR0 CCIFG.
CCIFG_0_HANDLER
            ...                         ; Start of handler Interrupt latency
            RETI
; Interrupt handler for TDxIFG, TDxCCR1 and TDxCCR2 CCIFG.
TD_HANDLER  ...                         ; Interrupt latency
            ADD &TDxIV,PC               ; Add offset to Jump table
            RETI                        ; Vector 0: No interrupt
            JMP CCIFG_1_HANDLER         ; Vector 2: Module 1
            JMP CCIFG_2_HANDLER         ; Vector 4: Module 2
            RETI                        ; Vector 6
            RETI                        ; Vector 8
            RETI                        ; Vector 10
            RETI                        ; Vector 12
            JMP TDxIFG_HANDLER          ; Vector 14
            RETI                        ; Vector 16
            RETI                        ; Vector 18
            RETI                        ; Vector 20
            JMP TDHxUNLKIFG_HANDLER     ; Vector 22
            ...

TDHxUNLKIFG_HANDLER                     ; Vector 22: TDHUNLKIFG Flag
            ...                         ; Task starts here
            RETI

TDxIFG_HANDLER                          ; Vector 14: TIMOV Flag
            ...                         ; Task starts here
            RETI
            ...

CCIFG_2_HANDLER                         ; Vector 4: Module 2
            ...                         ; Task starts here
            RETI                        ; Back to main program

; The Module 1 handler shows a way to look if any other
; interrupt is pending: 5 cycles have to be spent, but
; 9 cycles may be saved if another interrupt is pending
 CCIFG_1_HANDLER                        ; Vector 2: Module 1
            ...                         ; Task starts here
            JMP TD_HANDLER              ; Look for pending ints
```

1.3 Timer_D Registers

The Timer_D registers and address offsets are listed in Table 1-8. The base address for each instance of Timer_D can be found in the device-specific data sheet.

Table 1-8. Timer_D Registers

Offset	Acronym	Register Name	Type	Reset	Section
0000h	TDxCTL0	Timer_D Control 0	Read/write	On POR	Section 1.3.1
0002h	TDxCTL1	Timer_D Control 1	Read/write	On POR	Section 1.3.2
0004h	TDxCTL2	Timer_D Control 2	Read/write	On POR	Section 1.3.3
0006h	TDxR	Timer_D Counter	Read/write	On POR	Section 1.3.4
0008h	TDxCCTL0	Timer_D Capture/Compare Control 0	Read/write	On POR	Section 1.3.5
000Ah	TDxCCR0	Timer_D Capture/Compare 0	Read/write	On POR	Section 1.3.6
000Ch	TDxCL0	Timer_D Capture/Compare Latch 0	Read only	On POR	Section 1.3.7
000Eh	TDxCCTL1	Timer_D Capture/Compare Control 1	Read/write	On POR	Section 1.3.5
0010h	TDxCCR1	Timer_D Capture/Compare 1	Read/write	On POR	Section 1.3.6
0012h	TDxCL1	Timer_D Capture/Compare Latch 1	Read only	On POR	Section 1.3.7
0014h	TDxCCTL2	Timer_D Capture/Compare Control 2	Read/write	On POR	Section 1.3.5
0016h	TDxCCR2	Timer_D Capture/Compare 2	Read/write	On POR	Section 1.3.6
0018h	TDxCL2	Timer_D Capture/Compare Latch 2	Read only	On POR	Section 1.3.7
001Ah	TDxCCTL3	Timer_D Capture/Compare Control 3	Read/write	On POR	Section 1.3.5
001Ch	TDxCCR3	Timer_D Capture/Compare 3	Read/write	On POR	Section 1.3.6
001Eh	TDxCL3	Timer_D Capture/Compare Latch 3	Read only	On POR	Section 1.3.7
0020h	TDxCCTL4	Timer_D Capture/Compare Control 4	Read/write	On POR	Section 1.3.5
0022h	TDxCCR4	Timer_D Capture/Compare 4	Read/write	On POR	Section 1.3.6
0024h	TDxCL4	Timer_D Capture/Compare Latch 4	Read only	On POR	Section 1.3.7
0026h	TDxCCTL5	Timer_D Capture/Compare Control 5	Read/write	On POR	Section 1.3.5
0028h	TDxCCR5	Timer_D Capture/Compare 5	Read/write	On POR	Section 1.3.6
002Ah	TDxCL5	Timer_D Capture/Compare Latch 5	Read only	On POR	Section 1.3.7
002Ch	TDxCCTL6	Timer_D Capture/Compare Control 6	Read/write	On POR	Section 1.3.5
002Eh	TDxCCR6	Timer_D Capture/Compare 6	Read/write	On POR	Section 1.3.6
0030h	TDxCL6	Timer_D Capture/Compare Latch 6	Read only	On POR	Section 1.3.7
0032h		Reserved			
0034h		Reserved			
0036h		Reserved			
0038h	TDxHCTL0	Timer_D High-Resolution Control 0	Read/write	On POR	Section 1.3.8
003Ah	TDxHCTL1	Timer_D High-Resolution Control 1	Read/write	On POR	Section 1.3.9
003Ch	TDxHINT	Timer_D High-Resolution Interrupt	Read/write	On POR	Section 1.3.10
003Eh	TDxIV	Timer_D Interrupt Vector	Read only	On POR	Section 1.3.11

1.3.1 TDxCTL0 Register

Timer_D x Control Register 0

NOTE: To synchronize TD0 and TD1, TD0 must be cleared after it starts to count. Set CCRx to 1 to cause an interrupt when TD0 starts counting. After the interrupt occurs, set the CCRx to the desired value and then clear the counter.

```
TD0CCRx = 1;                // Set CCRx to cause interrupt when
                            // TD0 starts counting
TD0CTL0 |= MC_1 + TDCLR;    // Start timer
while(!(TD0CCTL1 & CCIFG)); // wait for CCRx interrupt flag
                            // signaling TD0 has started counting
TD0CCRx = x;                // User resets CCRx to value required
                            // for application
TD0CTL0 |= MC_1 + TDCLR;    // Clear the counters
```

Figure 1-24. TDxCTL0 Register

15	14	13	12	11	10	9	8
Reserved	TDCLGRPx		CNTLx		Reserved	TDSSELx	
r0	rw-(0)	rw-(0)	rw-(0)	rw-(0)	r0	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
ID		MCx		Reserved	TDCLR	TDIE	TDIFG
rw-(0)	rw-(0)	rw-(0)	rw-(0)	r0	w-(0)	rw-(0)	rw-(0)

Table 1-9. TDxCTL0 Register Description

Bit	Field	Type	Reset	Description
15	Reserved	R	0h	Reserved. Always reads as 0.
14-13	TDCLGRPx	RW	0h	TDCLx group 00b = Each TDCLx latch loads independently. 01b = TDxCL1+TDxCL2 (TDxCCR1 CLLDx bits control the update) TDxCL3+TDxCL4 (TDxCCR3 CLLDx bits control the update) TDxCL5+TDxCL6 (TDxCCR5 CLLDx bits control the update) TDxCL0 independent 10b = TDxCL1+TDxCL2+TDxCL3 (TDxCCR1 CLLDx bits control the update) TDxCL4+TDxCL5+TDxCL6 (TDxCCR4 CLLDx bits control the update) TDxCL0 independent 11b = TDxCL0+TDxCL1+TDxCL2+TDxCL3+TDxCL4+TDxCL5+TDxCL6 (TDxCCR1 CLLDx bits control the update)
12-11	CNTLx	RW	0h	Counter length 00b = 16-bit, TDR(max) = 0FFFFh 01b = 12-bit, TDR(max) = 0FFFh 10b = 10-bit, TDR(max) = 03FFh 11b = 8-bit, TDR(max) = 0FFh
10	Reserved	R	0h	Reserved. Always reads as 0.
9-8	TDSSELx	RW	0h	Timer_D clock source select 00b = TDCLK 01b = ACLK 10b = SMCLK 11b = Inverted TDCLK

Table 1-9. TDxCTL0 Register Description (continued)

Bit	Field	Type	Reset	Description
7-6	ID	RW	0h	Input divider. These bits, along with the IDEX bits in TDxCTL1, select the divider for the input clock. 00b = Divide by 1 01b = Divide by 2 10b = Divide by 4 11b = Divide by 8
5-4	MCx	RW	0h	Mode control. Setting MCx = 00h when Timer_D is not in use saves power. 00b = Stop mode: Timer is halted 01b = Up mode: Timer counts up to TDCL0 10b = Continuous mode: Timer counts up to the value set by CNTLx (counter length) 11b = Up/down mode: Timer counts up to TDCL0 and down to 0000h
3	Reserved	R	0h	Reserved. Always reads as 0.
2	TDCLR	W	0h	Timer_D clear. Setting this bit resets TDR, the TDCLK divider, and the count direction. The TDCLR bit always read as zero.
1	TDIE	RW	0h	Timer_D interrupt enable. This bit enables the TDIFG interrupt request. 0b = Interrupt disabled 1b = Interrupt enabled
0	TDIFG	RW	0h	Timer_D interrupt flag 0b = No interrupt pending 1b = Interrupt pending

1.3.2 TDxCTL1 Register

Timer_D x Control Register 1

Figure 1-25. TDxCTL1 Register

15	14	13	12	11	10	9	8
Reserved					IDEX		
r0	r0	r0	r0	r0	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
Reserved	TD6CMB	TD4CMB	TD2CMB	Reserved		TDCLKMx	
r0	rw-(0)	rw-(0)	rw-(0)	r0	r0	rw-(0)	rw-(0)

Table 1-10. TDxCTL1 Register Description

Bit	Field	Type	Reset	Description
15-11	Reserved	R	0h	Reserved. Always reads as 0.
10-8	IDEX	RW	0h	Input divider extension. These bits, along with the ID bits in TDxCTL0, select the divider for the input clock. 000b = Divide by 1 001b = Divide by 2 010b = Divide by 3 011b = Divide by 4 100b = Divide by 5 101b = Divide by 6 110b = Divide by 7 111b = Divide by 8
7	Reserved	R	0h	Reserved. Always reads as 0.
6	TD6CMB	RW	0h	Control bit for TDCCR registers combination in TD6 (only available on Timer_D7) 0b = TDxCCR5 and TDxCCR6 are not combined 1b = TDxCCR5 and TDxCCR6 are combined
5	TD4CMB	RW	0h	Control bit for TDCCR registers combination in TD4 (available on Timer_D5, Timer_D7) 0b = TDxCCR3 and TDxCCR4 are not combined 1b = TDxCCR3 and TDxCCR4 are combined
4	TD2CMB	RW	0h	Control bit for TDCCR registers combination in TD2 0b = TDxCCR1 and TDxCCR2 are not combined 1b = TDxCCR1 and TDxCCR2 are combined
3-2	Reserved	R	0h	Reserved. Always reads as 0.
1-0	TDCLKMx	RW	0h	Timer_D clocking mode register 00b = External clock source is used 01b = High-resolution local clock is used 10b = Auxiliary clock source from another timer instance is used 11b = Reserved

1.3.3 TDxCTL2 Register

Timer_D x Control Register 2

Figure 1-26. TDxCTL2 Register

15	14	13	12	11	10	9	8
Reserved							
r0	r0	r0	r0	r0	r0	r0	r0
7	6	5	4	3	2	1	0
Reserved	TDCAPM6	TDCAPM5	TDCAPM4	TDCAPM3	TDCAPM2	TDCAPM1	TDCAPM0
r0							

Table 1-11. TDxCTL2 Register Description

Bit	Field	Type	Reset	Description
15-7	Reserved	R	0h	Reserved. Always reads as 0.
6	TDCAPM6	RW	0h	Capture mode of channel 6 0b = Single capture mode 1b = Dual capture mode
5	TDCAPM5	RW	0h	Capture mode of channel 5 0b = Single capture mode 1b = Dual capture mode
4	TDCAPM4	RW	0h	Capture mode of channel 4 0b = Single capture mode 1b = Dual capture mode
3	TDCAPM3	RW	0h	Capture mode of channel 3 0b = Single capture mode 1b = Dual capture mode
2	TDCAPM2	RW	0h	Capture mode of channel 2 0b = Single capture mode 1b = Dual capture mode
1	TDCAPM1	RW	0h	Capture mode of channel 1 0b = Single capture mode 1b = Dual capture mode
0	TDCAPM0	RW	0h	Capture mode of channel 0 0b = Single capture mode 1b = Dual capture mode

1.3.4 TDxR Register

Timer_D x Counter Register

Figure 1-27. TDxR Register

15	14	13	12	11	10	9	8
TDRx							
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
TDRx							
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

Table 1-12. TDxR Register Description

Bit	Field	Type	Reset	Description
15-0	TDRx	RW	0h	Timer_D register. The TDxR register is the count of Timer_D. In high-resolution mode, the bits 0 to 3 return zero when the TDxR register is read.

1.3.5 TDxCCTLn Register

Timer_D x Capture/Compare Control Register

Figure 1-28. TDxCCTLn Register

15	14	13	12	11	10	9	8
CMx		CCISx		SCS	CLLDx		CAP
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
OUTMODx			CCIE	CCI	OUT	COV	CCIFG
rw-(0)	rw-(0)	rw-(0)	rw-(0)	r	rw-(0)	rw-(0)	rw-(0)

Table 1-13. TDxCCTLn Register Description

Bit	Field	Type	Reset	Description
15-14	CMx	RW	0h	Capture mode 00b = No capture 01b = Capture on rising edge 10b = Capture on falling edge 11b = Capture on both rising and falling edges
13-12	CCISx	RW	0h	Capture/compare input select. These bits select the TDCCRx input signal. See the device-specific data sheet for specific signal connections. 00b = CCIxA 01b = CCIxB 10b = GND 11b = VCC
11	SCS	RW	0h	Synchronize capture source. This bit is used to synchronize the capture input signal with the timer clock. In high-resolution mode, the capture is always synchronous to the high-resolution clock, and this setting is ignored. 0b = Asynchronous capture 1b = Synchronous capture
10-9	CLLDx	RW	0h	Compare latch load. These bits select the compare latch load event. 00b = TDCLx loads on write to TDCCRx 01b = TDCLx loads when TDR counts to 0 10b = TDCLx loads when TDR counts to 0 (up or continuous mode). TDCLx loads when TDR counts to TDxCL0 or to 0 (up/down mode). 11b = TDCLx loads when TDR counts to TDCLx
8	CAP	RW	0h	Capture mode 0b = Compare mode 1b = Capture mode
7-5	OUTMODx	RW	0h	Output mode 000b = OUT bit value 001b = Set 010b = Toggle/reset 011b = Set/reset 100b = Toggle 101b = Reset 110b = Toggle/set 111b = Reset/set
4	CCIE	RW	0h	Capture/compare interrupt enable. Enables the interrupt request of the corresponding CCIFG flag. 0b = Interrupt disabled 1b = Interrupt enabled
3	CCI	R	0h	Capture/compare input. The selected input signal can be read by this bit.

Table 1-13. TDxCCTLn Register Description (continued)

Bit	Field	Type	Reset	Description
2	OUT	RW	0h	Output. For output mode 0, this bit directly controls the state of the output. 0b = Output low 1b = Output high
1	COV	RW	0h	Capture overflow. Indicates a capture overflow occurred. COV must be reset with software. 0b = No capture overflow occurred 1b = Capture overflow occurred
0	CCIFG	RW	0h	Capture/compare interrupt flag 0b = No interrupt pending 1b = Interrupt pending

1.3.6 TDxCCRn Register

Timer_D x Capture/Compare Register

Figure 1-29. TDxCCRn Register

15	14	13	12	11	10	9	8
TDCCRx							
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
TDCCRx							
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

Table 1-14. TDxCCRn Register Description

Bit	Field	Type	Reset	Description
15-0	TDCCRx	RW	0h	Timer_D capture/compare register. The TDCCRx register is the count of capture/compare block x. Note: In high-resolution compare mode, there are limitations of the minimum and maximum duty cycles. See Table 1-3 and Table 1-4 for details.

1.3.7 TDxCLn Register

Timer_D x Capture/Compare Latch Register

Figure 1-30. TDxCLn Register

15	14	13	12	11	10	9	8
TDCLx							
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
TDCLx							
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

Table 1-15. TDxCLn Register Description

Bit	Field	Type	Reset	Description
15-0	TDCLx	RW	0h	Timer_D capture/compare latch register. TDCLx register holds the comparison value in compare mode. In capture mode the register content of TDxCCRx is copied into TDxCLx at each capture event.

1.3.8 TDxHCTL0 Register

Timer_D x High-Resolution Control Register 0

NOTE: The TDHMx bits of master and slave must be the same for Timer_D synchronization.

Figure 1-31. TDxHCTL0 Register

15	14	13	12	11	10	9	8
Reserved							TDHFW
r0	r0	r0	r0	r0	r0	r0	rw-(0)
7	6	5	4	3	2	1	0
TDHDx		TDHMx		TDHRON	TDHEAEN	TDHREGEN	TDHEN
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

Table 1-16. TDxHCTL0 Register Description

Bit	Field	Type	Reset	Description
15-9	Reserved	R	0h	Reserved. Always reads as 0.
8	TDHFW	RW	0h	High-resolution generator fast wakeup enable 0b = High-resolution generator fast wakeup disabled 1b = High-resolution generator fast wakeup enable
7-6	TDHDx	RW	0h	High-resolution clock divider. These bits select the divider for the high-resolution clock. 00b = Divide by 1 01b = Divide by 2 10b = Divide by 4 11b = Divide by 8
5-4	TDHMx	RW	0h	Timer_D high-resolution clock multiplication factor 00b = High-resolution clock 8x Timer_D clock 01b = High-resolution clock 16x Timer_D clock 10b = Reserved 11b = Reserved
3	TDHRON	RW	0h	Timer_D high-resolution generator forced on. 0b = High-resolution generator is on if the Timer_D counter MCx bits are 01, 10 or 11. 1b = High-resolution generator is on in all Timer_D MCx modes. The PMM remains in high-current mode.
2	TDHEAEN	RW	0h	Timer_D high-resolution clock enhanced accuracy enable bit. Setting this bit reduces the accumulated frequency offset of the high-resolution clock generator and the reference clock. 0b = Normal accuracy 1b = Enhanced accuracy enable
1	TDHREGEN	RW	0h	Timer_D regulation enable. Set this bit to synchronize the high-resolution clock to the Timer_D input clock defined by TDSSELx. 0b = Regulation disabled 1b = Regulation enabled
0	TDHEN	RW	0h	Timer_D high-resolution enable bit. This bit must be set to enable high-resolution operation mode. Whenever a high-resolution TDAUXCLK from another Timer_D instance is used, this bit must also be set. 0b = High-resolution mode disable 1b = High-resolution mode enable

1.3.9 TDxHCTL1 Register

Timer_D x High-Resolution Control Register 1

NOTE: The TDHMx bit of the slave timer must be equal to the TDHMx bit of the master timer even if high resolution is not enabled for the slave timer.

Figure 1-32. TDxHCTL1 Register

15	14	13	12	11	10	9	8
TDHCLKCR	TDHCLKRx		TDHCLKSRx				
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
TDHCLKTRIMx							Reserved
rw-(1)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	r0

Table 1-17. TDxHCTL1 Register Description

Bit	Field	Type	Reset	Description
15	TDHCLKCR	RW	0h	Timer_D high-resolution coarse clock range selection bits. For detailed frequency numbers, see the device-specific data sheet. 0b = Timer_D input clock to the high-resolution generator is <15 MHz. 1b = Timer_D input clock to the high-resolution generator is >15 MHz.
14-13	TDHCLKRx	RW	0h	Timer_D high-resolution clock range selection bits. These bits are used to define the coarse clock range of the high-resolution clock generator. If TDHREGEN = 1 these register bits are modified by hardware. 00b = Clock range 0. See data sheet for frequency details. 01b = Clock range 1. See data sheet for frequency details. 10b = Clock range 2. See data sheet for frequency details. 11b = Reserved
12-8	TDHCLKSRx	RW	0h	Timer_D high-resolution clock sub-range selection bits. These bits are used to define the subclock range of the high-resolution clock generator. The frequency change due to a change of the TDHCLKTRIMx bits is approximately half of the TDHCLKSRx bits in each clock range. If TDHREGEN = 1, these register bits are modified by hardware.
7-1	TDHCLKTRIMx	RW	40h	Timer_D high-resolution clock trim selection bits. These bits are used to change the clock frequency slightly. The frequency change due to a change of the TDHCLKTRIMx bits is approximately half of the TDHCLKSRx bits in each clock range. If TDHREGEN = 1, these register bits are modified by hardware.
0	Reserved	R	0h	Reserved. Always reads as 0.

1.3.10 TDxHINT Register

Timer_D x High-Resolution Interrupt Register

Figure 1-33. TDxHINT Register

15	14	13	12	11	10	9	8
Reserved				TDHUNLKIE	TDHLKIE	TDHFHIE	TDHFLIE
r0	r0	r0	r0	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
Reserved				TDHUNLKIFG	TDHLKIFG	TDHFHIFG	TDHFLIFG
r0	r0	r0	r0	rw-(0)	rw-(0)	rw-(0)	rw-(0)

Table 1-18. TDxHINT Register Description

Bit	Field	Type	Reset	Description
15-12	Reserved	R	0h	Reserved. Always reads as 0.
11	TDHUNLKIE	RW	0h	Timer_D interrupt enable. This bit enables the TDHUNLKIFG interrupt request. 0b = Interrupt disabled 1b = Interrupt enabled
10	TDHLKIE	RW	0h	Timer_D interrupt enable. This bit enables the TDHLKIFG interrupt request. 0b = Interrupt disabled 1b = Interrupt enabled
9	TDHFHIE	RW	0h	Timer_D interrupt enable. This bit enables the TDHFHIFG interrupt request. 0b = Interrupt disabled 1b = Interrupt enabled
8	TDHFLIE	RW	0h	Timer_D interrupt enable. This bit enables the TDHFLIFG interrupt request. 0b = Interrupt disabled 1b = Interrupt enabled
7-4	Reserved	R	0h	Reserved. Always reads as 0.
3	TDHUNLKIFG	RW	0h	Timer_D high-resolution frequency unlock interrupt flag if TDHREGEN is set to 1. This bit is set if the frequency is still unlocked and the calibration is not yet completed. If the bit is set until cleared by writing to it. 0b = No interrupt pending 1b = Interrupt pending
2	TDHLKIFG	RW	0h	Timer_D high-resolution frequency lock interrupt flag if TDHREGEN is set to 1. This bit is set if the frequency is locked and the calibration is completed. If the bit is set it remains set until cleared by writing to it. 0b = No interrupt pending 1b = Interrupt pending
1	TDHFHIFG	RW	0h	Timer_D high-resolution fail high interrupt flag. This bit is set if the high-resolution generator cannot generate the high-resolution frequency because the input frequency is too high. If the bit is set until cleared by writing to it. 0b = No interrupt pending 1b = Interrupt pending
0	TDHFLIFG	RW	0h	Timer_D high-resolution fail low interrupt flag. This bit is set if the high-resolution generator cannot generate the high-resolution frequency because the input frequency is too low. If the bit is set until cleared by writing o it. 0b = No interrupt pending 1b = Interrupt pending

1.3.11 TDxIV Register

Timer_D x Interrupt Vector Register

Figure 1-34. TDxIV Register

15	14	13	12	11	10	9	8
TDIVx							
r0	r0	r0	r0	r0	r0	r0	r0
7	6	5	4	3	2	1	0
TDIVx							
r0	r0	r0	r-(0)	r-(0)	r-(0)	r-(0)	r0

Table 1-19. TDxIV Register Description

Bit	Field	Type	Reset	Description
15-0	TDIVx	R	0h	Timer_D external interrupt vector value 00h = No interrupt pending 02h = Interrupt Source: Capture/compare 1; Interrupt Flag: TDxCCR1 CCIFG; Interrupt Priority: Highest 04h = Interrupt Source: Capture/compare 2; Interrupt Flag: TDxCCR2 CCIFG 06h = Interrupt Source: Capture/compare 3; Interrupt Flag: TDxCCR3 CCIFG 08h = Interrupt Source: Capture/compare 4; Interrupt Flag: TDxCCR4 CCIFG 0Ah = Interrupt Source: Capture/compare 5; Interrupt Flag: TDxCCR5 CCIFG 0Ch = Interrupt Source: Capture/compare 6; Interrupt Flag: TDxCCR6 CCIFG 0Eh = Reserved 10h = Interrupt Source: Timer overflow; Interrupt Flag: TDxIFG 12h = Interrupt Source: Clock fail low; Interrupt Flag: TDxHINT TDHFLIFG 14h = Interrupt Source: Clock fail high; Interrupt Flag: TDxHINT TDHFHIFG 16h = Interrupt Source: High-resolution frequency locked; Interrupt Flag: TDxHINT TDHLKIFG 18h = Interrupt Source: High-resolution frequency unlocked; Interrupt Flag: TDxHINT TDHUNLKIFG 1Ah = Reserved 1Ch = Reserved 1Eh = Reserved ; Interrupt Priority: Lowest

IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

Texas Instruments Incorporated ('TI") technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, "TI Resources") are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI's provision of TI Resources does not expand or otherwise alter TI's applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

TI RESOURCES ARE PROVIDED "AS IS" AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your non-compliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include; without limitation, TI's standard terms for semiconductor products http://www.ti.com/sc/docs/stdterms.htm), evaluation modules, and samples (http://www.ti.com/sc/docs/sampterms.htm).

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265