

Timer_B

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 from <http://www.ti.com/lit/pdf/slau208>.

Timer_B is a 16-bit timer/counter with multiple capture/compare registers. There can be multiple Timer_B modules on a given device (see the device-specific data sheet). This chapter describes the operation and use of the Timer_B module.

Topic	Page
1.1 Timer_B Introduction.....	2
1.2 Timer_B Operation.....	4
1.3 Timer_B Registers.....	17

1.1 Timer_B Introduction

Timer_B is a 16-bit timer/counter with up to seven capture/compare registers. Timer_B can support multiple capture/compares, PWM outputs, and interval timing. Timer_B also has extensive interrupt capabilities. Interrupts may be generated from the counter on overflow conditions and from each of the capture/compare registers.

Timer_B features include:

- Asynchronous 16-bit timer/counter with four operating modes and four selectable lengths
- Selectable and configurable clock source
- Up to seven configurable capture/compare registers
- Configurable outputs with PWM capability
- Double-buffered compare latches with synchronized loading
- Interrupt vector register for fast decoding of all Timer_B interrupts

The block diagram of Timer_B 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, an associated action does not take place.

NOTE: Nomenclature

There may be multiple instantiations of Timer_B on a given device. The prefix TBx is used, where x is a greater than equal to zero indicating the Timer_B instantiation. For devices with one instantiation, x = 0. The suffix n, where n = 0 to 6, represents the specific capture/compare registers associated with the Timer_B instantiation.

1.1.1 Similarities and Differences From Timer_A

Timer_B is identical to Timer_A with the following exceptions:

- The length of Timer_B is programmable to be 8, 10, 12, or 16 bits.
- Timer_B TBxCCRn registers are double-buffered and can be grouped.
- All Timer_B outputs can be put into a high-impedance state.
- The SCCI bit function is not implemented in Timer_B.

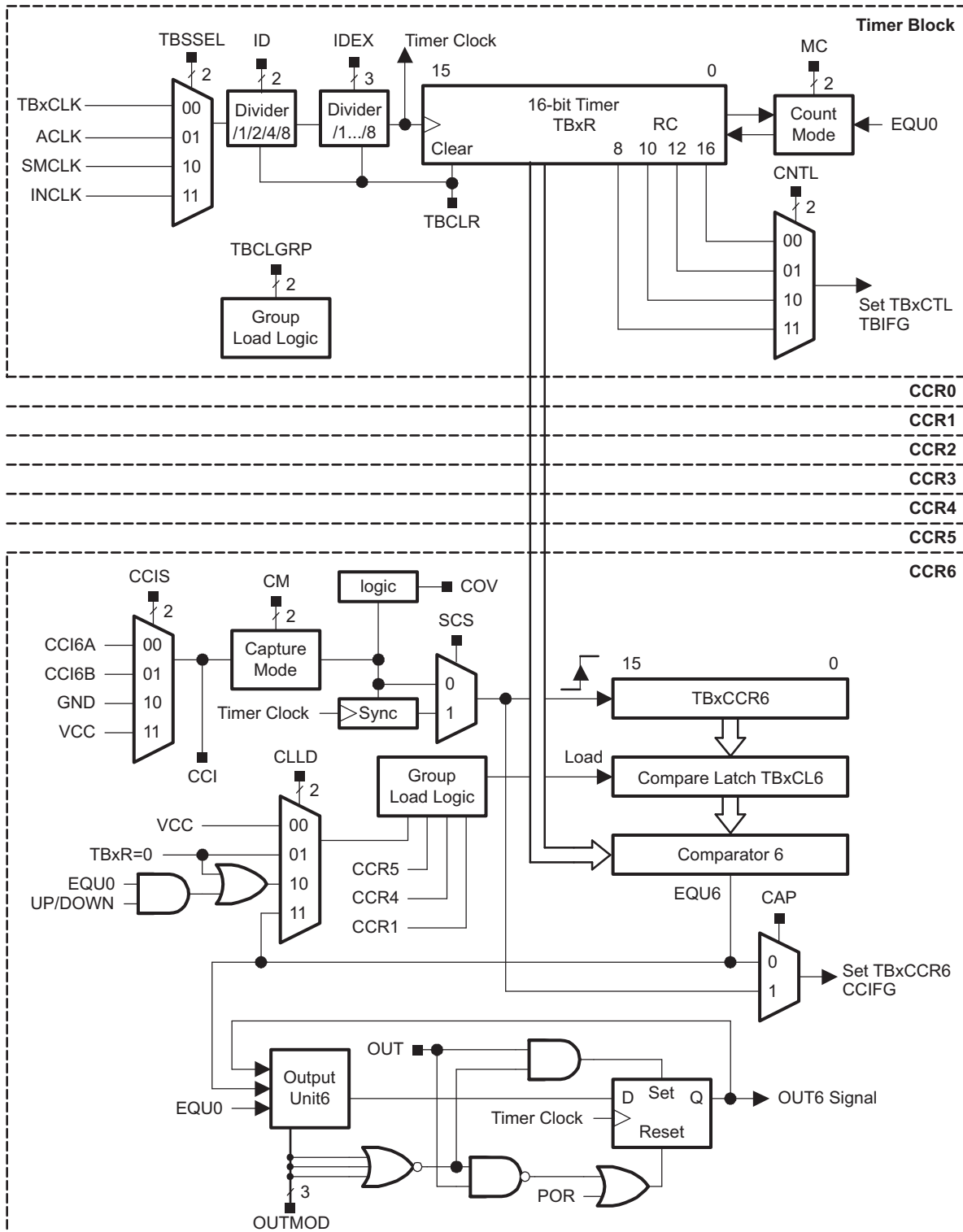


Figure 1-1. Timer_B Block Diagram

1.2 Timer_B Operation

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

1.2.1 16-Bit Timer Counter

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

TBxR may be cleared by setting the TBCLR bit. Setting TBCLR also clears the clock divider counter logic (the divider setting remains unchanged) and count direction for up/down mode.

NOTE: Modifying Timer_B registers

TI recommends stopping the timer before modifying its operation (with exception of the interrupt enable, interrupt flag, and TBCLR) to avoid errant operating conditions.

When the timer clock is asynchronous to the CPU clock, any read from TBxR 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 TBxR takes effect immediately.

1.2.1.1 TBxR Length

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

1.2.1.2 Clock Source Select and Divider

The timer clock can be sourced from ACLK, SMCLK, or externally from TBxCLK or INCLK. The clock source is selected with the TBSEL bits. The selected clock source may be passed directly to the timer or divided by 2, 4, or 8, using the ID bits. The selected clock source can be further divided by 2, 3, 4, 5, 6, 7, or 8 using the TBIDEX bits. The timer clock divider logic is reset when TBCLR is set.

NOTE: Timer_B dividers

After programming ID or TBIDEX bits, set the TBCLR bit. This clears the contents of TBxR and resets the clock divider logic to a defined state. The clock dividers are implemented as down counters. Therefore, when the TBCLR bit is cleared, the timer clock immediately begins clocking at the first rising edge of the Timer_B clock source selected with the TBSEL bits and continues clocking at the divider settings set by the ID and TBIDEX bits.

1.2.2 Starting the Timer

The timer may be started or restarted in the following ways:

- The timer counts when $MC > \{ 0 \}$ and the clock source is active.
- When the timer mode is either up or up/down, the timer may be stopped by loading 0 to TBxCL0. The timer may then be restarted by loading a nonzero value to TBxCL0. In this scenario, the timer starts incrementing in the up direction from zero.

1.2.3 Timer Mode Control

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

Table 1-1. Timer Modes

MC	Mode	Description
00	Stop	The timer is halted.
01	Up	The timer repeatedly counts from zero to the value of compare register TBxCL0.
10	Continuous	The timer repeatedly counts from zero to the value selected by the CNTL bits.
11	Up/down	The timer repeatedly counts from zero up to the value of TBxCL0 and then back down to zero.

1.2.3.1 Up Mode

The up mode is used if the timer period must be different from $TBxR_{(max)}$ counts. The timer repeatedly counts up to the value of compare latch TBxCL0, which defines the period (see Figure 1-2). The number of timer counts in the period is $TBxCL0 + 1$. When the timer value equals TBxCL0, the timer restarts counting from zero. If up mode is selected when the timer value is greater than TBxCL0, the timer immediately restarts counting from zero.

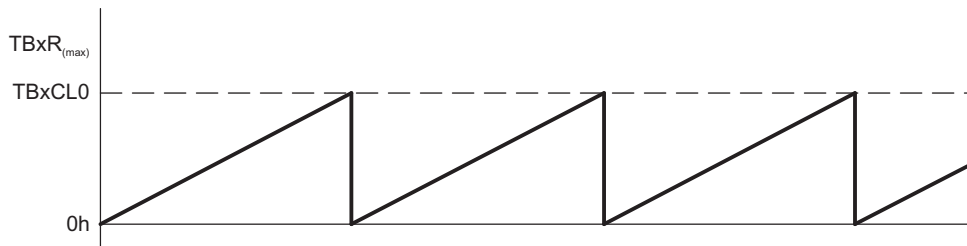


Figure 1-2. Up Mode

The TBxCCR0 CCIFG interrupt flag is set when the timer counts to the TBxCL0 value. The TBIFG interrupt flag is set when the timer counts from TBxCL0 to zero. Figure 1-3 shows the flag set cycle.

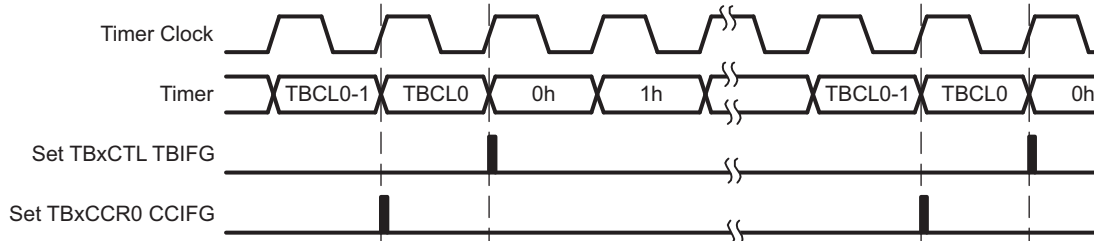


Figure 1-3. Up Mode Flag Setting

1.2.3.1.1 Changing Period Register TBxCL0

When changing TBxCL0 while the timer is running and when the TBxCL0 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.

1.2.3.2 Continuous Mode

In continuous mode, the timer repeatedly counts up to $TBxR_{(max)}$ and restarts from zero (see Figure 1-4). The compare latch $TBxCL0$ works the same way as the other capture/compare registers.

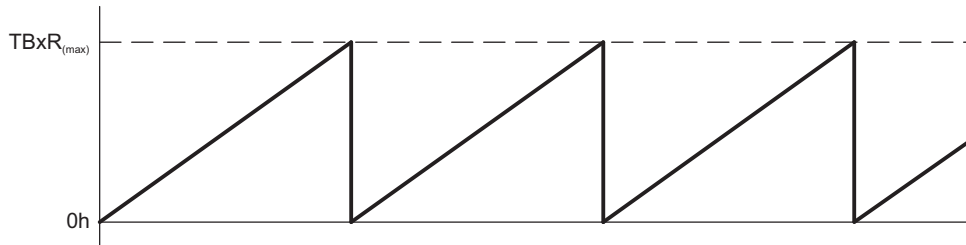


Figure 1-4. Continuous Mode

The TBIFG interrupt flag is set when the timer *counts* from $TBxR_{(max)}$ to zero. Figure 1-5 shows the flag set cycle.

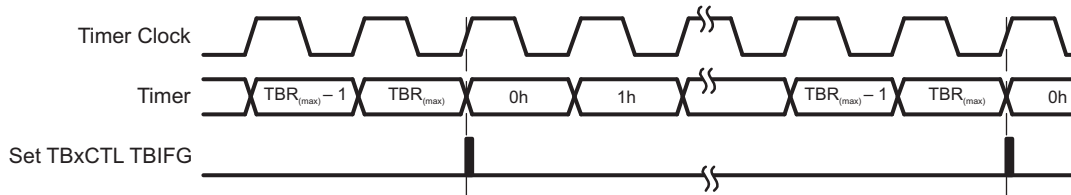


Figure 1-5. Continuous Mode Flag Setting

1.2.3.3 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 $TBxCLn$ latch in the interrupt service routine. Figure 1-6 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 n (where $n = 0$ to 7), independent time intervals or output frequencies can be generated using capture/compare registers.

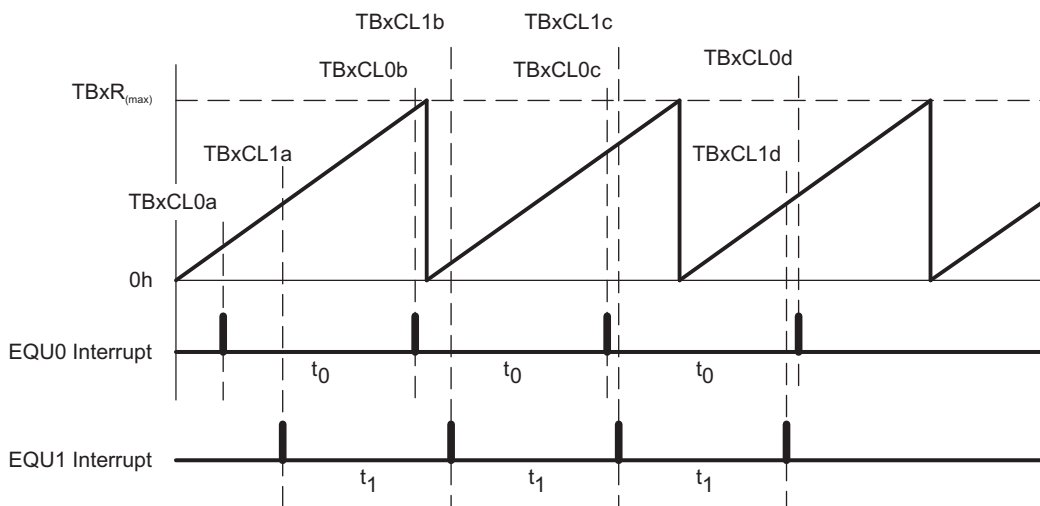


Figure 1-6. Continuous Mode Time Intervals

Time intervals can be produced with other modes as well, where TBxCL0 is used as the period register. Their handling is more complex, because the sum of the old TBxCLn data and the new period can be higher than the TBxCL0 value. When the sum of the previous TBxCLn value plus t_x is greater than the TBxCL0 data, the old TBxCL0 value must be subtracted to obtain the correct time interval.

1.2.3.4 Up/Down Mode

The up/down mode is used if the timer period must be different from $TBxR_{(max)}$ counts and if symmetrical pulse generation is needed. The timer repeatedly counts up to the value of compare latch TBxCL0, and back down to zero (see Figure 1-7). The period is twice the value in TBxCL0.

NOTE: $TBxCL0 > TBxR_{(max)}$

If $TBxCL0 > TBxR_{(max)}$, the counter operates as if it were configured for continuous mode. It does not count down from $TBxR_{(max)}$ to zero.

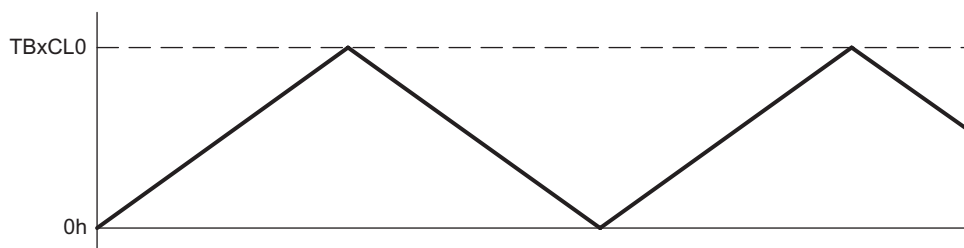


Figure 1-7. Up/Down Mode

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 TBCLR bit must be used to clear the direction. Setting TBCLR also clears the TBxR value and the clock divider counter logic (the divider setting remains unchanged).

In up/down mode, the TBxCCR0 CCIFG interrupt flag and the TBIFG interrupt flag are set only once during the period, separated by one-half the timer period. The TBxCCR0 CCIFG interrupt flag is set when the timer counts from $TBxCL0-1$ to $TBxCL0$, and TBIFG is set when the timer completes counting down from $0001h$ to $0000h$. Figure 1-8 shows the flag set cycle.

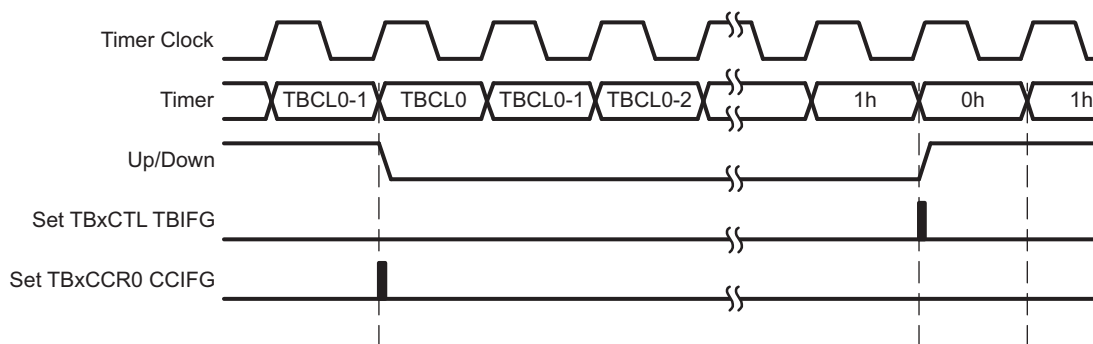


Figure 1-8. Up/Down Mode Flag Setting

1.2.3.4.1 Changing the Value of Period Register TBxCL0

When changing TBxCL0 while the timer is running and counting in the down direction, and when the TBxCL0 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 TBxCL0, 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 TBxCL0 is loaded, the timer begins counting down. However, one additional count may occur before the counter begins counting down.

1.2.3.5 Use of Up/Down Mode

The up/down mode supports applications that require dead times between output signals (see [Section 1.2.5](#)). 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 1-9](#), the t_{dead} is:

$$t_{\text{dead}} = t_{\text{timer}} \times (\text{TBxCL1} - \text{TBxCL3})$$

Where:

t_{dead} = Time during which both outputs need to be inactive

t_{timer} = Cycle time of the timer clock

TBxCLn = Content of compare latch n

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

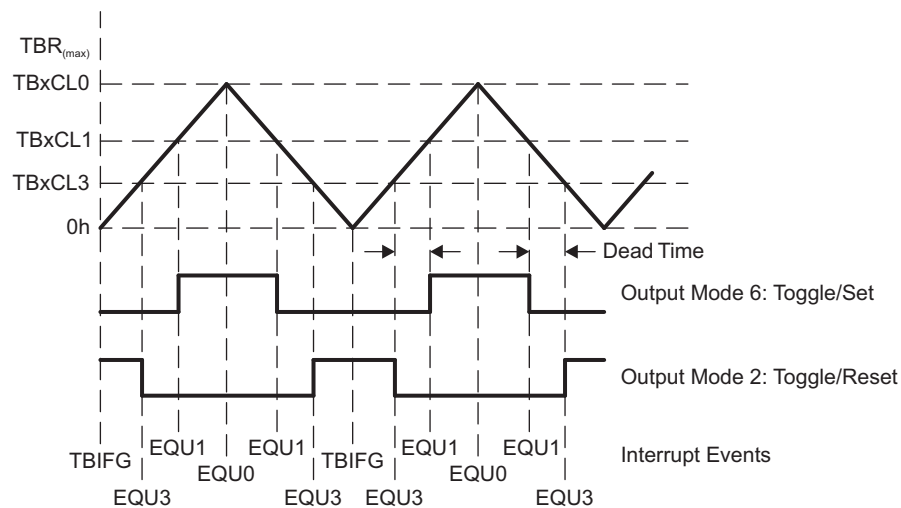


Figure 1-9. Output Unit in Up/Down Mode

1.2.4 Capture/Compare Blocks

Up to seven identical capture/compare blocks, TBxCCRn (where n = 0 to 6), are present in Timer_B. Any of the blocks may be used to capture the timer data or to generate time intervals.

1.2.4.1 Capture Mode

The capture mode is selected when CAP = 1. 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 CCIS bits. The CM 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 TBxCCRn register.
- The interrupt flag CCIFG is set.

The input signal level can be read at any time from 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. TI recommends setting the SCS bit to synchronize the capture signal with the timer clock (see [Figure 1-10](#)).

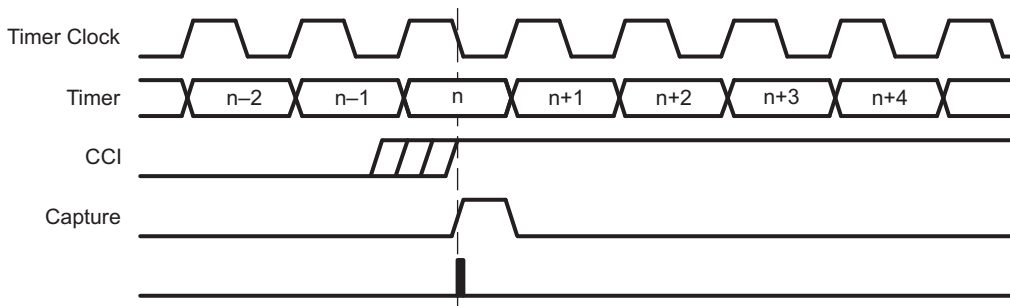


Figure 1-10. Capture Signal (SCS = 1)

NOTE: Changing Capture Inputs

Changing capture inputs while in capture mode may cause unintended capture events. To avoid this scenario, capture inputs should only be changed when capture mode is disabled (CM = {0} or CAP = 0).

Overflow logic is provided in each capture/compare register to indicate if a second capture was performed before the value from the first capture was read. Bit COV is set when this occurs (see [Figure 1-11](#)). COV must be reset with software.

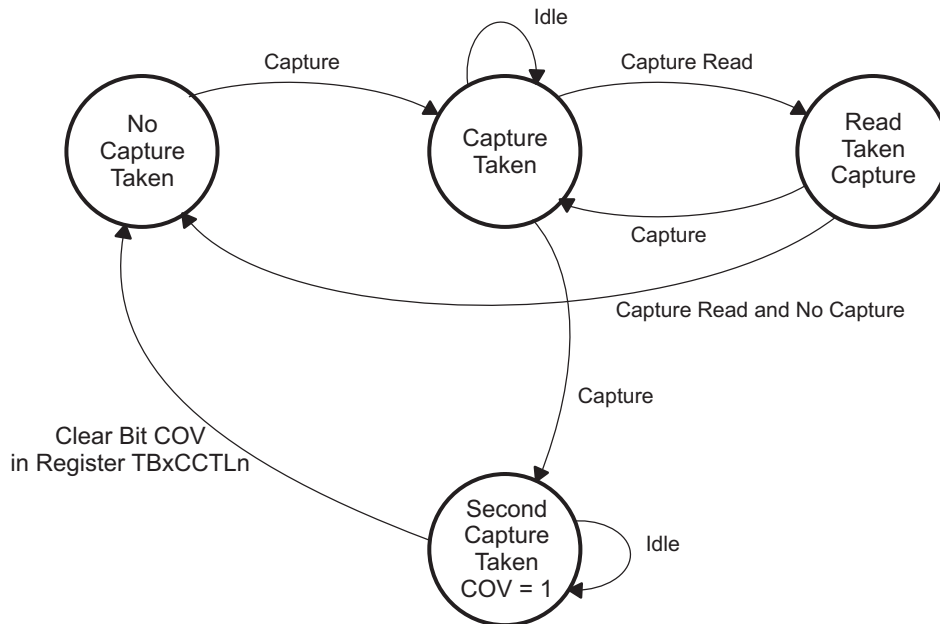


Figure 1-11. Capture Cycle

1.2.4.1.1 Capture Initiated by Software

Captures can be initiated by software. The CM 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,&TB0CCTL1 ; Setup TB0CCTL1
XOR      #CCIS0,&TB0CCTL1             ; TB0CCR1 = TB0R
```

NOTE: Capture Initiated by Software

In general, changing capture inputs while in capture mode may cause unintended capture events. For this scenario, switching the capture input between VCC and GND, disabling the capture mode is not required.

1.2.4.2 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 TBxR *counts* to the value in a TBxCLn, where n represents the specific capture/compare latch:

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

1.2.4.2.1 Compare Latch TBxCLn

The TBxCCRn compare latch, TBxCLn, holds the data for the comparison to the timer value in compare mode. TBxCLn is buffered by TBxCCRn. The buffered compare latch gives the user control over when a compare period updates. The user cannot directly access TBxCLn. Compare data is written to each TBxCCRn and automatically transferred to TBxCLn. The timing of the transfer from TBxCCRn to TBxCLn is user selectable, with the CLLD bits as described in [Table 1-2](#).

Table 1-2. TBxCLn Load Events

CLLD	Description
00	New data is transferred from TBxCCRn to TBxCLn immediately when TBxCCRn is written to.
01	New data is transferred from TBxCCRn to TBxCLn when TBxR <i>counts</i> to 0.
10	New data is transferred from TBxCCRn to TBxCLn when TBxR <i>counts</i> to 0 for up and continuous modes. New data is transferred to from TBxCCRn to TBxCLn when TBxR <i>counts</i> to the old TBxCL0 value or to 0 for up/down mode.
11	New data is transferred from TBxCCRn to TBxCLn when TBxR <i>counts</i> to the old TBxCLn value.

1.2.4.2.2 Grouping Compare Latches

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

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

Table 1-3. Compare Latch Operating Modes

TBCLGRP _x	Grouping	Update Control
00	None	Individual
01	TBxCL1+TBxCL2 TBxCL3+TBxCL4 TBxCL5+TBxCL6	TBxCCR1 TBxCCR3 TBxCCR5
10	TBxCL1+TBxCL2+TBxCL3 TBxCL4+TBxCL5+TBxCL6	TBxCCR1 TBxCCR4
11	TBxCL0+TBxCL1+TBxCL2+TBxCL3+TBxCL4+TBxCL5+TBxCL6	TBxCCR1

1.2.5 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 EQU_n signals. The TBOUTH pin function can be used to put all Timer_B outputs into a high-impedance state. When the TBOUTH 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_B outputs are in a high-impedance state.

1.2.5.1 Output Modes

The output modes are defined by the OUTMOD bits and are described in Table 1-4. The OUT_n signal is changed with the rising edge of the timer clock for all modes except mode 0. Output modes 2, 3, 6, and 7 are not useful for output unit 0 because EQU_n = EQU₀.

Table 1-4. Output Modes

OUTMOD	Mode	Description
000	Output	The output signal OUT _n is defined by the OUT bit. The OUT _n signal updates immediately when OUT is updated.
001	Set	The output is set when the timer <i>counts</i> to the TBxCL _n value. It remains set until a reset of the timer, or until another output mode is selected and affects the output.
010	Toggle/Reset	The output is toggled when the timer <i>counts</i> to the TBxCL _n value. It is reset when the timer <i>counts</i> to the TBxCL ₀ value.
011	Set/Reset	The output is set when the timer <i>counts</i> to the TBxCL _n value. It is reset when the timer <i>counts</i> to the TBxCL ₀ value.
100	Toggle	The output is toggled when the timer <i>counts</i> to the TBxCL _n value. The output period is double the timer period.
101	Reset	The output is reset when the timer <i>counts</i> to the TBxCL _n value. It remains reset until another output mode is selected and affects the output.
110	Toggle/Set	The output is toggled when the timer <i>counts</i> to the TBxCL _n value. It is set when the timer <i>counts</i> to the TBxCL ₀ value.
111	Reset/Set	The output is reset when the timer <i>counts</i> to the TBxCL _n value. It is set when the timer <i>counts</i> to the TBxCL ₀ value.

1.2.5.1.1 Output Example – Timer in Up Mode

The OUTn signal is changed when the timer *counts* up to the TBxCLn value, and rolls from TBxCL0 to zero, depending on the output mode. An example is shown in [Figure 1-12](#) using TBxCL0 and TBxCL1.

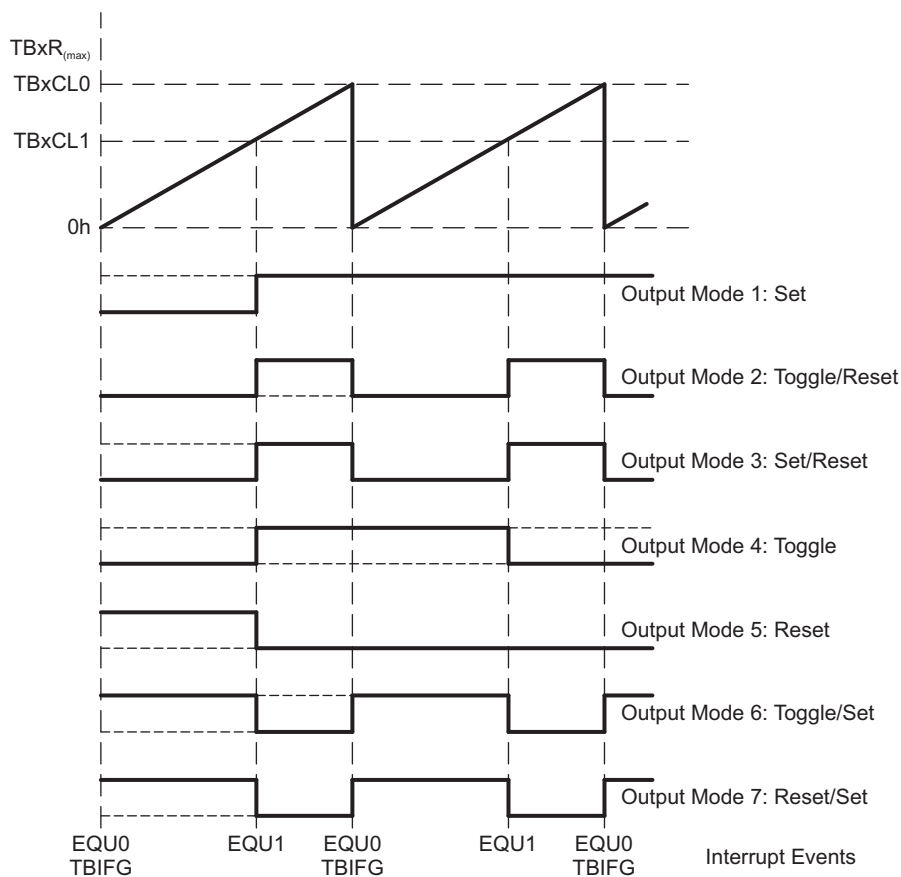


Figure 1-12. Output Example – Timer in Up Mode

1.2.5.1.2 Output Example – Timer in Continuous Mode

The OUTn signal is changed when the timer reaches the TBxCLn and TBxCL0 values, depending on the output mode. An example is shown in Figure 1-13 using TBxCL0 and TBxCL1.

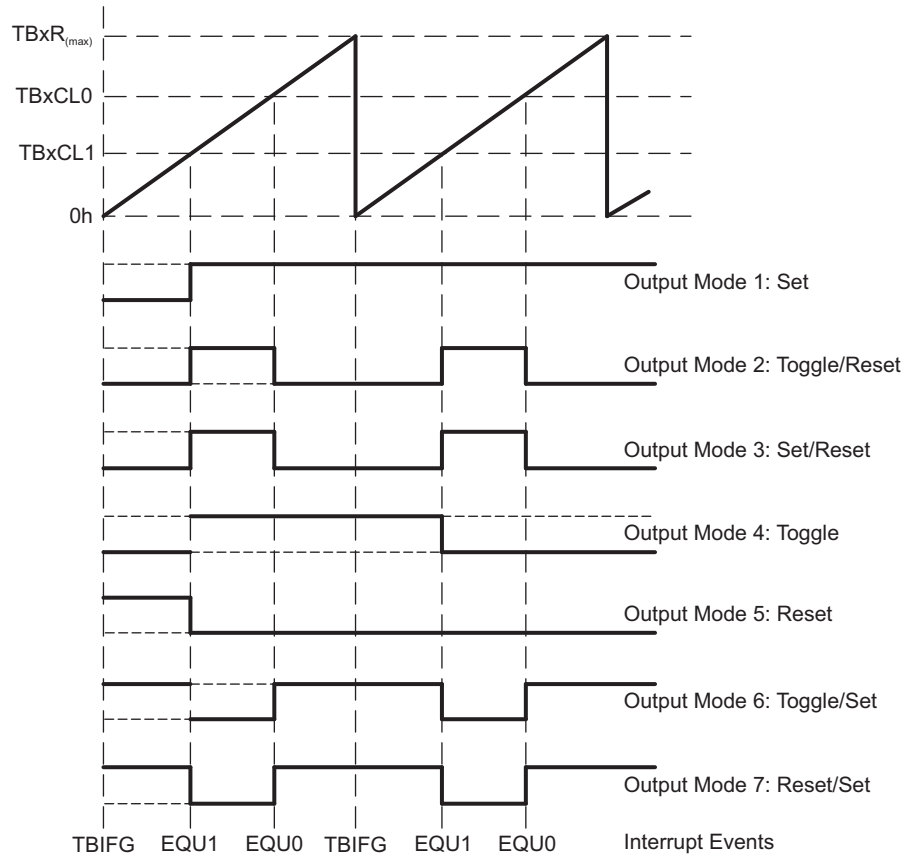


Figure 1-13. Output Example – Timer in Continuous Mode

1.2.5.1.3 Output Example – Timer in Up/Down Mode

The OUTn signal changes when the timer equals TBxCLn in either count direction and when the timer equals TBxCL0, depending on the output mode. An example is shown in Figure 1-14 using TBxCL0 and TBxCL3.

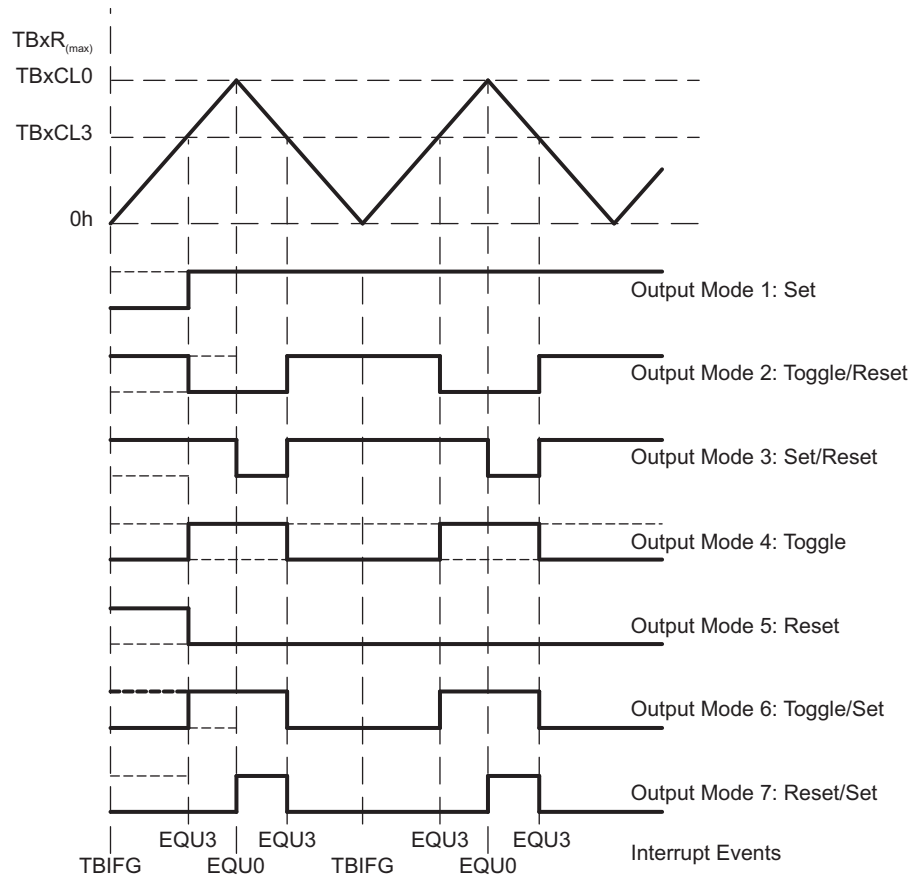


Figure 1-14. Output Example – Timer in Up/Down Mode

NOTE: Switching between output modes

When switching between output modes, one of the OUTMOD 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,&TBCCTLx ; Set output mode=7
BIC #OUTMOD,&TBCCTLx ; Clear unwanted bits
```

1.2.6 Timer_B Interrupts

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

- TBxCCR0 interrupt vector for TBxCCR0 CCIFG
- TBIV interrupt vector for all other CCIFG flags and TBIFG

In capture mode, any CCIFG flag is set when a timer value is captured in the associated TBxCCRn register. In compare mode, any CCIFG flag is set when TBxR *counts* to the associated TBxCLn 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.

1.2.6.1 TBxCCR0 Interrupt Vector

The TBxCCR0 CCIFG flag has the highest Timer_B interrupt priority and has a dedicated interrupt vector (see Figure 1-15). The TBxCCR0 CCIFG flag is automatically reset when the TBxCCR0 interrupt request is serviced.

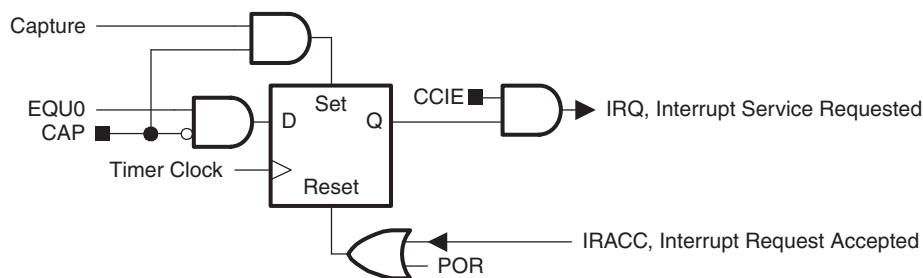


Figure 1-15. Capture/Compare TBxCCR0 Interrupt Flag

1.2.6.2 TBxIV, Interrupt Vector Generator

The TBIFG flag and TBxCCRn CCIFG flags (excluding TBxCCR0 CCIFG) are prioritized and combined to source a single interrupt vector. The interrupt vector register TBxIV is used to determine which flag requested an interrupt.

The highest-priority enabled interrupt (excluding TBxCCR0 CCIFG) generates a number in the TBxIV register (see register description). This number can be evaluated or added to the program counter to automatically enter the appropriate software routine. Disabled Timer_B interrupts do not affect the TBxIV value.

Any access, read or write, of the TBxIV 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 TBxCCR1 and TBxCCR2 CCIFG flags are set when the interrupt service routine accesses the TBxIV register, TBxCCR1 CCIFG is reset automatically. After the RETI instruction of the interrupt service routine is executed, the TBxCCR2 CCIFG flag generates another interrupt.

1.2.6.3 TBxIV, Interrupt Handler Examples

The following software example shows the recommended use of TBxIV and the handling overhead. The TBxIV value is added to the PC to automatically jump to the appropriate routine. The example assumes a single instantiation of the largest timer configuration available.

The numbers at the right margin show the necessary CPU clock cycles for each instruction. The software overhead for different interrupt sources includes interrupt latency and return-from-interrupt cycles, but not the task handling itself. The latencies are:

- Capture/compare block CCR0: 11 cycles
- Capture/compare blocks CCR1 to CCR6: 16 cycles
- Timer overflow TBIFG: 14 cycles

The following software example shows the recommended use of TBxIV for Timer_B3.

```

; Interrupt handler for TB0CCR0 CCIFG.
CCIFG_0_HND
;      ...      ; Start of handler Interrupt latency    6
      RETI      ;                                       5

; Interrupt handler for TB0IFG, TB0CCR1 through TB0CCR6 CCIFG.

TB0_HND      ...      ; Interrupt latency                6
      ADD      &TB0IV,PC ; Add offset to Jump table    3
      RETI      ; Vector 0: No interrupt                5
      JMP      CCIFG_1_HND ; Vector 2: TB0CCR1          2
      JMP      CCIFG_2_HND ; Vector 4: TB0CCR2          2
      JMP      CCIFG_3_HND ; Vector 6: TB0CCR3          2
      JMP      CCIFG_4_HND ; Vector 8: TB0CCR4          2
      JMP      CCIFG_5_HND ; Vector 10: TB0CCR5         2
      JMP      CCIFG_6_HND ; Vector 12: TB0CCR6         2

TB0IFG_HND      ; Vector 14: TB0IFG Flag
      ...      ; Task starts here
      RETI      ;                                       5

CCIFG_6_HND      ; Vector 12: TB0CCR6
      ...      ; Task starts here
      RETI      ; Back to main program                5

CCIFG_5_HND      ; Vector 10: TB0CCR5
      ...      ; Task starts here
      RETI      ; Back to main program                5

CCIFG_4_HND      ; Vector 8: TB0CCR4
      ...      ; Task starts here
      RETI      ; Back to main program                5

CCIFG_3_HND      ; Vector 6: TB0CCR3
      ...      ; Task starts here
      RETI      ; Back to main program                5

CCIFG_2_HND      ; Vector 4: TB0CCR2
      ...      ; Task starts here
      RETI      ; Back to main program                5

CCIFG_1_HND      ; Vector 2: TB0CCR1
      ...      ; Task starts here
      RETI      ; Back to main program                5

```


1.3 Timer_B Registers

The Timer_B registers are listed in [Table 1-5](#). The base address can be found in the device-specific data sheet. The address offset is listed in [Table 1-5](#).

Table 1-5. Timer_B Registers

Offset	Acronym	Register Name	Type	Access	Reset	Section
00h	TBxCTL	Timer_B Control	Read/write	Word	0000h	Section 1.3.1
02h	TBxCCTL0	Timer_B Capture/Compare Control 0	Read/write	Word	0000h	Section 1.3.3
04h	TBxCCTL1	Timer_B Capture/Compare Control 1	Read/write	Word	0000h	Section 1.3.3
06h	TBxCCTL2	Timer_B Capture/Compare Control 2	Read/write	Word	0000h	Section 1.3.3
08h	TBxCCTL3	Timer_B Capture/Compare Control 3	Read/write	Word	0000h	Section 1.3.3
0Ah	TBxCCTL4	Timer_B Capture/Compare Control 4	Read/write	Word	0000h	Section 1.3.3
0Ch	TBxCCTL5	Timer_B Capture/Compare Control 5	Read/write	Word	0000h	Section 1.3.3
0Eh	TBxCCTL6	Timer_B Capture/Compare Control 6	Read/write	Word	0000h	Section 1.3.3
10h	TBxR	Timer_B Counter	Read/write	Word	0000h	Section 1.3.2
12h	TBxCCR0	Timer_B Capture/Compare 0	Read/write	Word	0000h	Section 1.3.4
14h	TBxCCR1	Timer_B Capture/Compare 1	Read/write	Word	0000h	Section 1.3.4
16h	TBxCCR2	Timer_B Capture/Compare 2	Read/write	Word	0000h	Section 1.3.4
18h	TBxCCR3	Timer_B Capture/Compare 3	Read/write	Word	0000h	Section 1.3.4
1Ah	TBxCCR4	Timer_B Capture/Compare 4	Read/write	Word	0000h	Section 1.3.4
1Ch	TBxCCR5	Timer_B Capture/Compare 5	Read/write	Word	0000h	Section 1.3.4
1Eh	TBxCCR6	Timer_B Capture/Compare 6	Read/write	Word	0000h	Section 1.3.4
2Eh	TBxIV	Timer_B Interrupt Vector	Read only	Word	0000h	Section 1.3.5
20h	TBxEX0	Timer_B Expansion 0	Read/write	Word	0000h	Section 1.3.6

1.3.1 TBxCTL Register

Timer_B x Control Register

Figure 1-16. TBxCTL Register

15	14	13	12	11	10	9	8
Reserved	TBCLGRP _x		CNTL		Reserved	TBSSEL	
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
ID		MC		Reserved	TBCLR	TBIE	TBIFG
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	w-(0)	rw-(0)	rw-(0)

Table 1-6. TBxCTL Register Description

Bit	Field	Type	Reset	Description
15	Reserved	R	0h	Reserved. Always reads as 0.
14-13	TBCLGRP	RW	0h	TBxCL _n group 00b = Each TBxCL _n latch loads independently. 01b = TBxCL1+TBxCL2 (TBxCCR1 CLLD bits control the update); TBxCL3+TBxCL4 (TBxCCR3 CLLD bits control the update); TBxCL5+TBxCL6 (TBxCCR5 CLLD bits control the update); TBxCL0 independent 10b = TBxCL1+TBxCL2+TBxCL3 (TBxCCR1 CLLD bits control the update); TBxCL4+TBxCL5+TBxCL6 (TBxCCR4 CLLD bits control the update); TBxCL0 independent 11b = TBxCL0+TBxCL1+TBxCL2+TBxCL3+TBxCL4+TBxCL5+TBxCL6 (TBxCCR1 CLLD bits control the update)
12-11	CNTL	RW	0h	Counter length 00b = 16-bit, TBxR(max) = 0FFFFh 01b = 12-bit, TBxR(max) = 0FFFh 10b = 10-bit, TBxR(max) = 03FFh 11b = 8-bit, TBxR(max) = 0FFh
10	Reserved	R	0h	Reserved. Always reads as 0.
9-8	TBSSEL	RW	0h	Timer_B clock source select 00b = TBxCLK 01b = ACLK 10b = SMCLK 11b = INCLK
7-6	ID	RW	0h	Input divider. These bits, along with the TBIDEX bits, select the divider for the input clock. 00b = /1 01b = /2 10b = /4 11b = /8
5-4	MC	RW	0h	Mode control. Setting MC = 00h when Timer_B is not in use conserves power. 00b = Stop mode: Timer is halted 01b = Up mode: Timer counts up to TBxCL0 10b = Continuous mode: Timer counts up to the value set by CNTL 11b = Up/down mode: Timer counts up to TBxCL0 and down to 0000h
3	Reserved	R	0h	Reserved. Always reads as 0.
2	TBCLR	RW	0h	Timer_B clear. Setting this bit clears TBR, the clock divider logic (the divider setting remains unchanged), and the count direction. The TBCLR bit is automatically reset and is always read as zero.
1	TBIE	RW	0h	Timer_B interrupt enable. This bit enables the TBIFG interrupt request. 0b = Interrupt disabled 1b = Interrupt enabled

Table 1-6. TBxCTL Register Description (continued)

Bit	Field	Type	Reset	Description
0	TBIFG	RW	0h	Timer_B interrupt flag 0b = No interrupt pending 1b = Interrupt pending

1.3.2 TBxR Register

Timer_B x Counter Register

Figure 1-17. TBxR Register

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

Table 1-7. TBxR Register Description

Bit	Field	Type	Reset	Description
15-0	TBxR	RW	0h	Timer_B register. The TBxR register is the count of Timer_B.

1.3.3 TBxCCTLn Register

Timer_B x Capture/Compare Control Register n

Figure 1-18. TBxCCTLn Register

15	14	13	12	11	10	9	8
CM		CCIS		SCS	CLLD		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
OUTMOD			CCIE	CCI	OUT	COV	CCIFG
rw-(0)	rw-(0)	rw-(0)	rw-(0)	r	rw-(0)	rw-(0)	rw-(0)

Table 1-8. TBxCCTLn Register Description

Bit	Field	Type	Reset	Description
15-14	CM	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	CCIS	RW	0h	Capture/compare input select. These bits select the TBxCCRn 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. 0b = Asynchronous capture 1b = Synchronous capture
10-9	CLLD	RW	0h	Compare latch load. These bits select the compare latch load event. 00b = TBxCLn loads on write to TBxCCRn 01b = TBxCLn loads when TBxR counts to 0 10b = TBxCLn loads when TBxR counts to 0 (up or continuous mode). TBxCLn loads when TBxR counts to TBxCL0 or to 0 (up/down mode). 11b = TBxCLn loads when TBxR counts to TBxCLn
8	CAP	RW	0h	Capture mode 0b = Compare mode 1b = Capture mode
7-5	OUTMOD	RW	0h	Output mode. Modes 2, 3, 6, and 7 are not useful for TBxCL0 because EQU _n = EQU ₀ . 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. This bit enables the interrupt request of the corresponding CCIFG flag. 0b = Interrupt disabled 1b = Interrupt enabled
3	CCI	R	Undef	Capture/compare input. The selected input signal can be read by this bit.

Table 1-8. TBxCCTLn 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. This bit 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

1.3.4 TBxCCRn Register

Timer_B x Capture/Compare Register n

Figure 1-19. TBxCCRn Register

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

Table 1-9. TBxCCRn Register Description

Bit	Field	Type	Reset	Description
15-0	TBxCCRn	RW	0h	Timer_B capture/compare register. Compare mode: TBxCCRn holds the data for the comparison to the timer value in the Timer_B Register, TBR. Capture mode: The Timer_B Register, TBR, is copied into the TBxCCRn register when a capture is performed.

1.3.5 TBxIV Register

Timer_B x Interrupt Vector Register

Figure 1-20. TBxIV Register

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

Table 1-10. TBxIV Register Description

Bit	Field	Type	Reset	Description
15-0	TBIV	R	0h	Timer_B interrupt vector value 00h = No interrupt pending 02h = Interrupt Source: Capture/compare 1; Interrupt Flag: TBxCCR1 CCIFG; Interrupt Priority: Highest 04h = Interrupt Source: Capture/compare 2; Interrupt Flag: TBxCCR2 CCIFG 06h = Interrupt Source: Capture/compare 3; Interrupt Flag: TBxCCR3 CCIFG 08h = Interrupt Source: Capture/compare 4; Interrupt Flag: TBxCCR4 CCIFG 0Ah = Interrupt Source: Capture/compare 5; Interrupt Flag: TBxCCR5 CCIFG 0Ch = Interrupt Source: Capture/compare 6; Interrupt Flag: TBxCCR6 CCIFG 0Eh = Interrupt Source: Timer overflow; Interrupt Flag: TBxCTL TBIFG; Interrupt Priority: Lowest

1.3.6 TBxEX0 Register

Timer_B x Expansion Register 0

Figure 1-21. TBxEX0 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				TBIDEX ⁽¹⁾			
r0	r0	r0	r0	r0	rw-(0)	rw-(0)	rw-(0)

⁽¹⁾ After programming TBIDEX bits and configuration of the timer, set TBCLR bit to ensure proper reset of the timer divider logic.

Table 1-11. TBxEX0 Register Description

Bit	Field	Type	Reset	Description
15-3	Reserved	R	0h	Reserved. Always reads as 0.
2-0	TBIDEX	RW	0h	Input divider expansion. These bits along with the ID bits 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

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/sampterm.htm) (<http://www.ti.com/sc/docs/sampterm.htm>).

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2018, Texas Instruments Incorporated