Lecture 7
Timers

Embedded Systems

DCSE, UET Peshawar

Bilal Habib

Basic Timer Structure

- A Binary Counter Driven by a Periodic Signal
  - Mux: Clock source selector
  - Prescaler: Clock frequency divider
  - Counter: *n*-bit binary counter
  - Comparator: compares counter output Vs. compare register

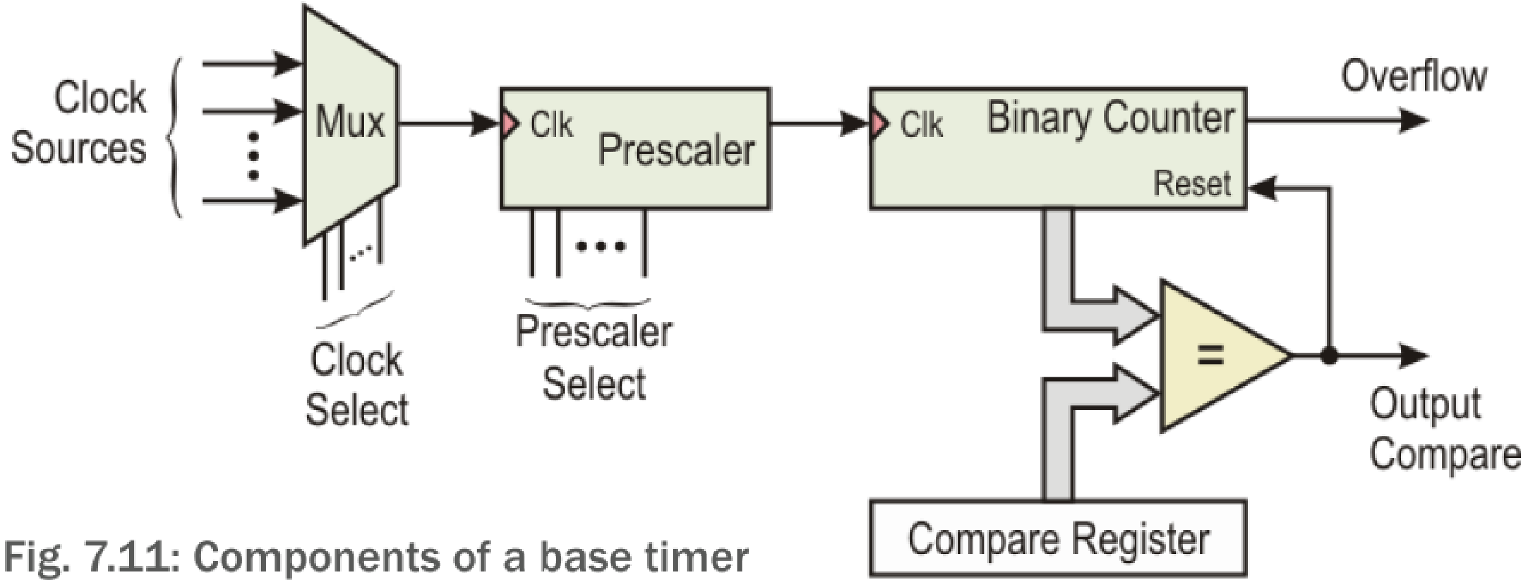


Fig. 7.11: Components of a base timer

Overflow VS Output Compare

Overflow Output Operation

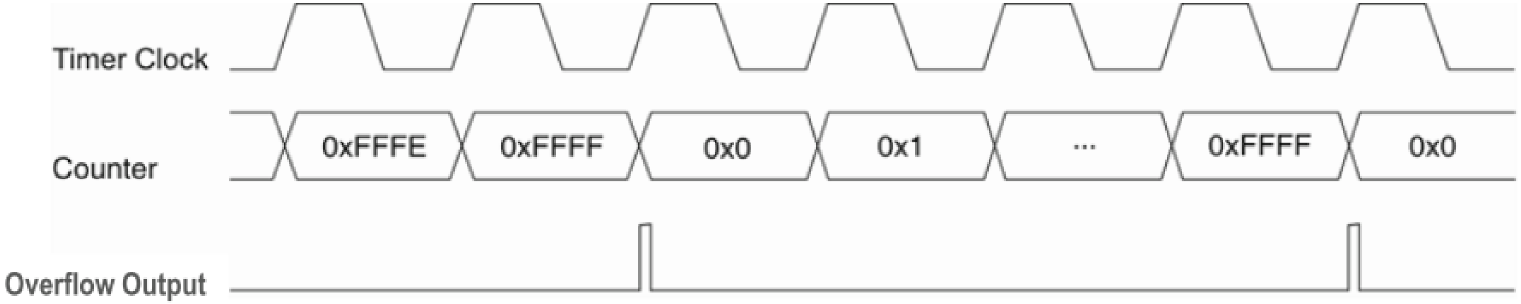


Fig. 7.12 Overflow signal obtained from a 16-bit timer.

Output Compare Operation

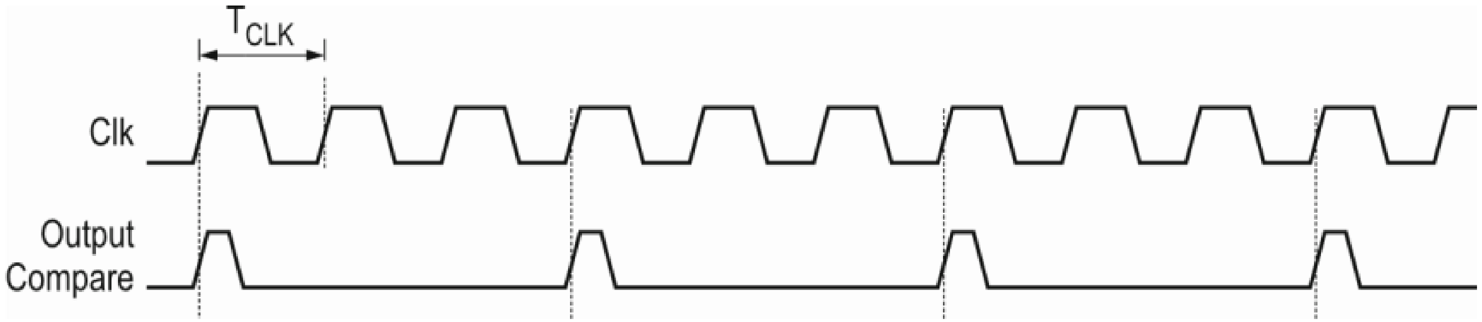
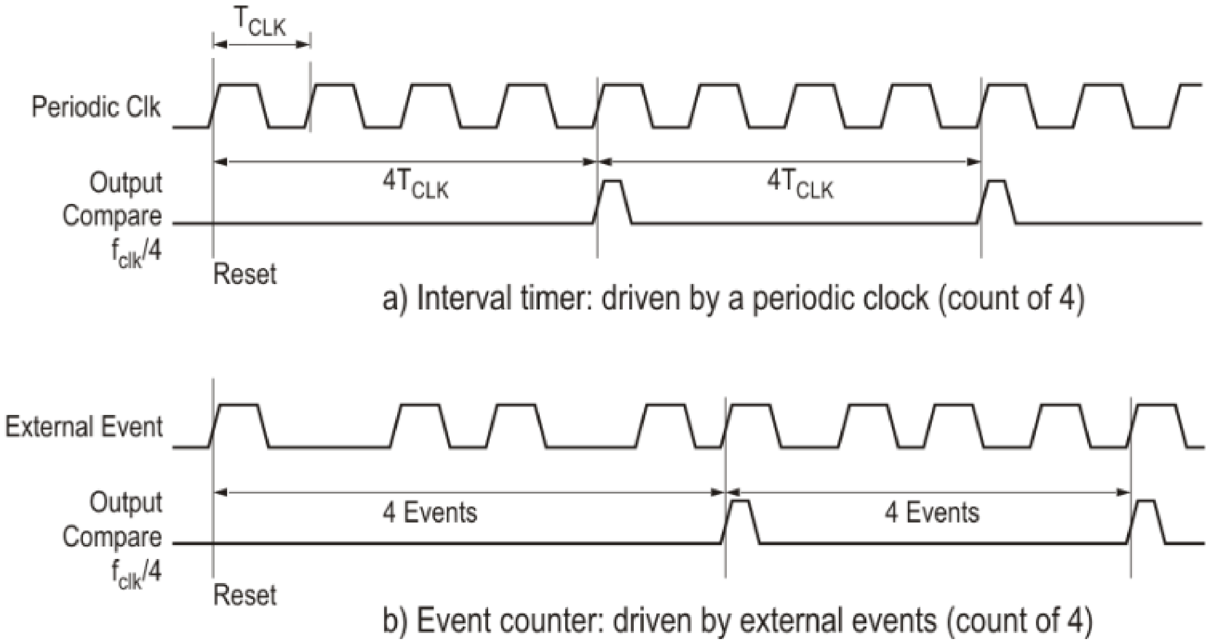


Fig. 7.13 Output compare signal obtained when loading the compare register with a value of three (3)

Interval Timer VS Event Counter

- Interval Timer
  - Measures the time elapsed after *k* clock cycles
  - As the clock period *T* is known, the time interval is *kT*
- Event Counter
  - Counts the occurrence of *k* external events
  - The clock is driven by the signal marking the external event



T_CLK
Periodic Clk
4T_CLK
4T_CLK
Output Compare f_clk/4
Reset
a) Interval timer: driven by a periodic clock (count of 4)
External Event
4 Events
4 Events
Output Compare f_clk/4
Reset
b) Event counter: driven by external events (count of 4)

Fig. 7.14: Periodic interval of 4 clock Vs. counting four aperiodic events

Extending the Timer Count

- Cascading Multiple Timers

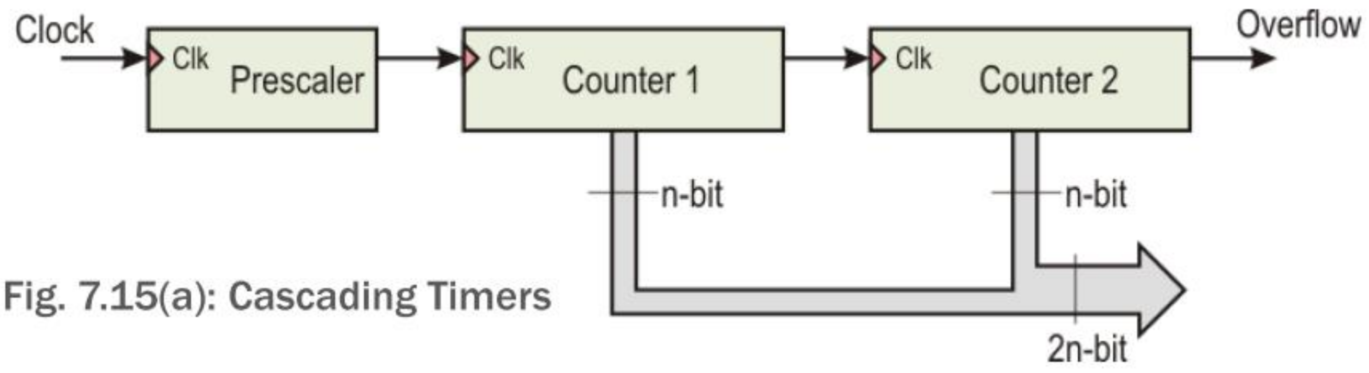


Fig. 7.15(a): Cascading Timers

- Using Software Variable

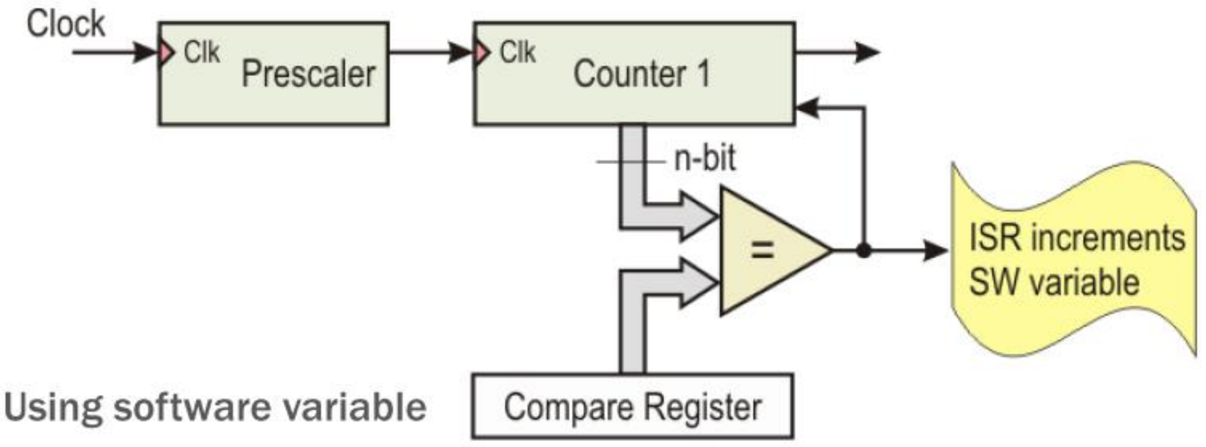
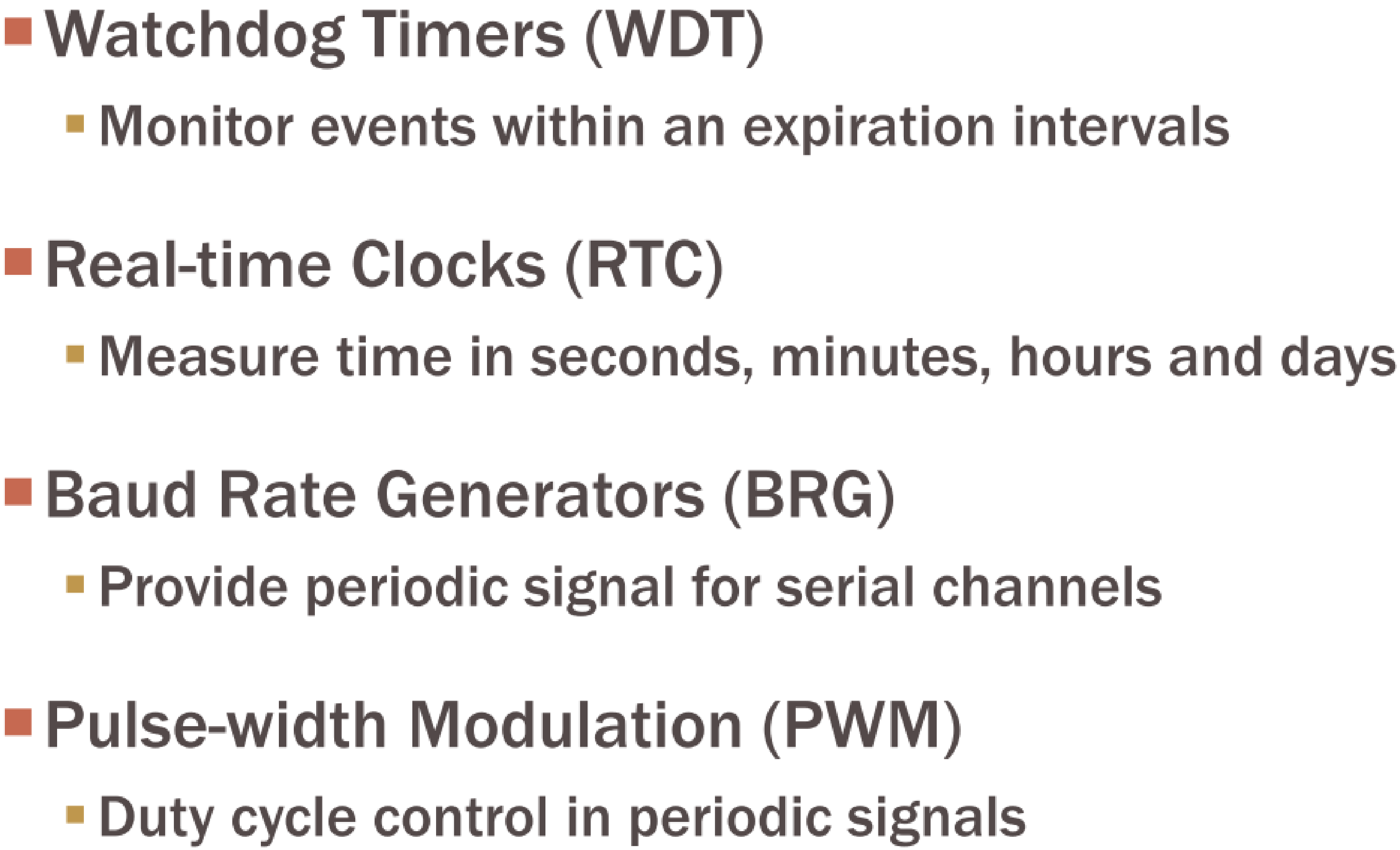


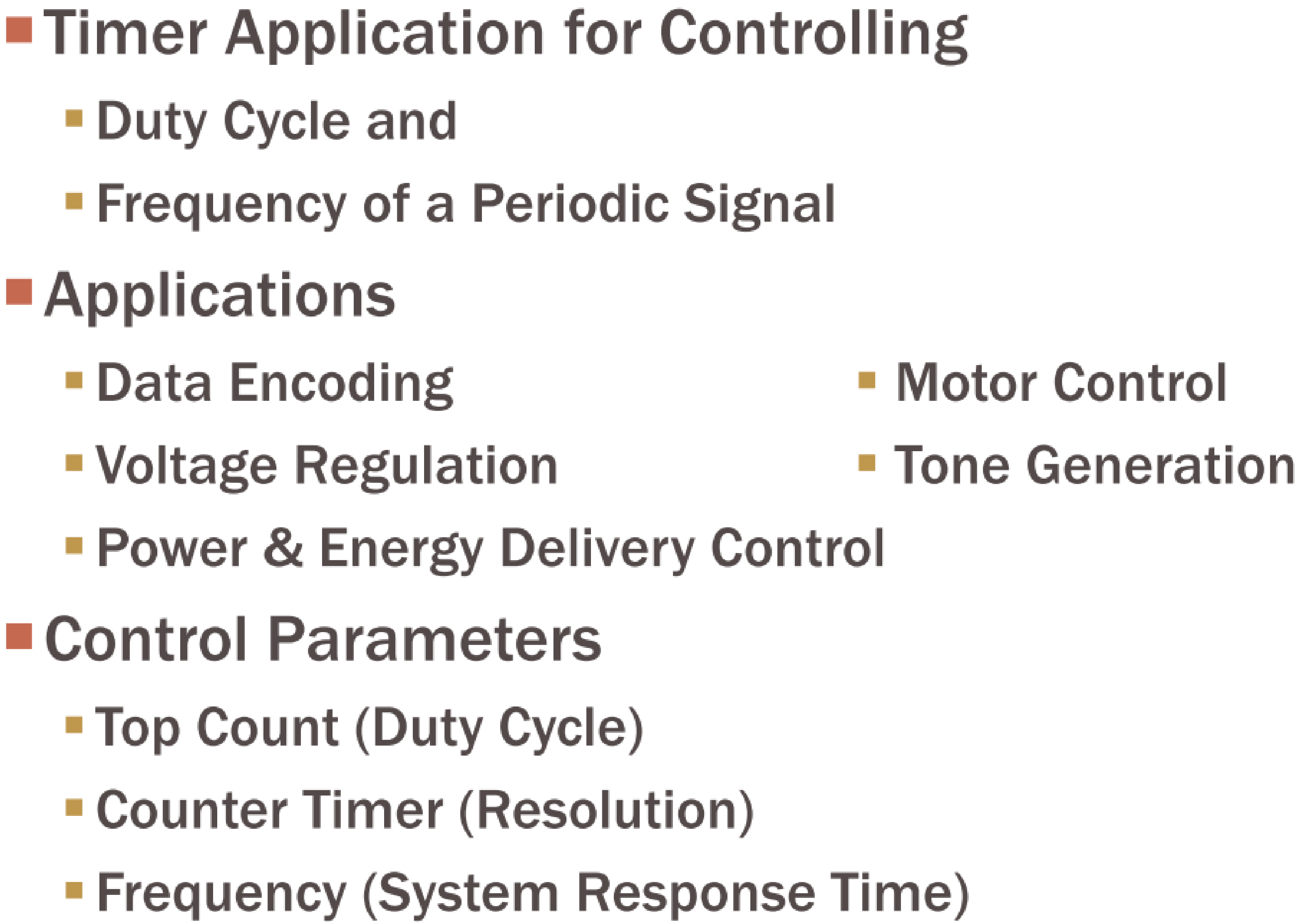
Fig. 7.15(b): Using software variable

Signature Timer Applications (STA)



- Watchdog Timers (WDT)
  - Monitor events within an expiration intervals
- Real-time Clocks (RTC)
  - Measure time in seconds, minutes, hours and days
- Baud Rate Generators (BRG)
  - Provide periodic signal for serial channels
- Pulse-width Modulation (PWM)
  - Duty cycle control in periodic signals

STA: Pulse Width Modulation (PWM)



- Timer Application for Controlling
  - Duty Cycle and
  - Frequency of a Periodic Signal
- Applications
  - Data Encoding
  - Voltage Regulation
  - Motor Control
  - Tone Generation
  - Power & Energy Delivery Control
- Control Parameters
  - Top Count (Duty Cycle)
  - Counter Timer (Resolution)
  - Frequency (System Response Time)

MSP430 Timer_A

- 16-bit Timer/Counter
- 3-bit Prescaler
- 3 Capture/Compare Registers
- Four Modes
  - Stop, Up, Continuous, Up/Down
- Selectable Clock Source
- Configurable Outputs
- PWM Capable

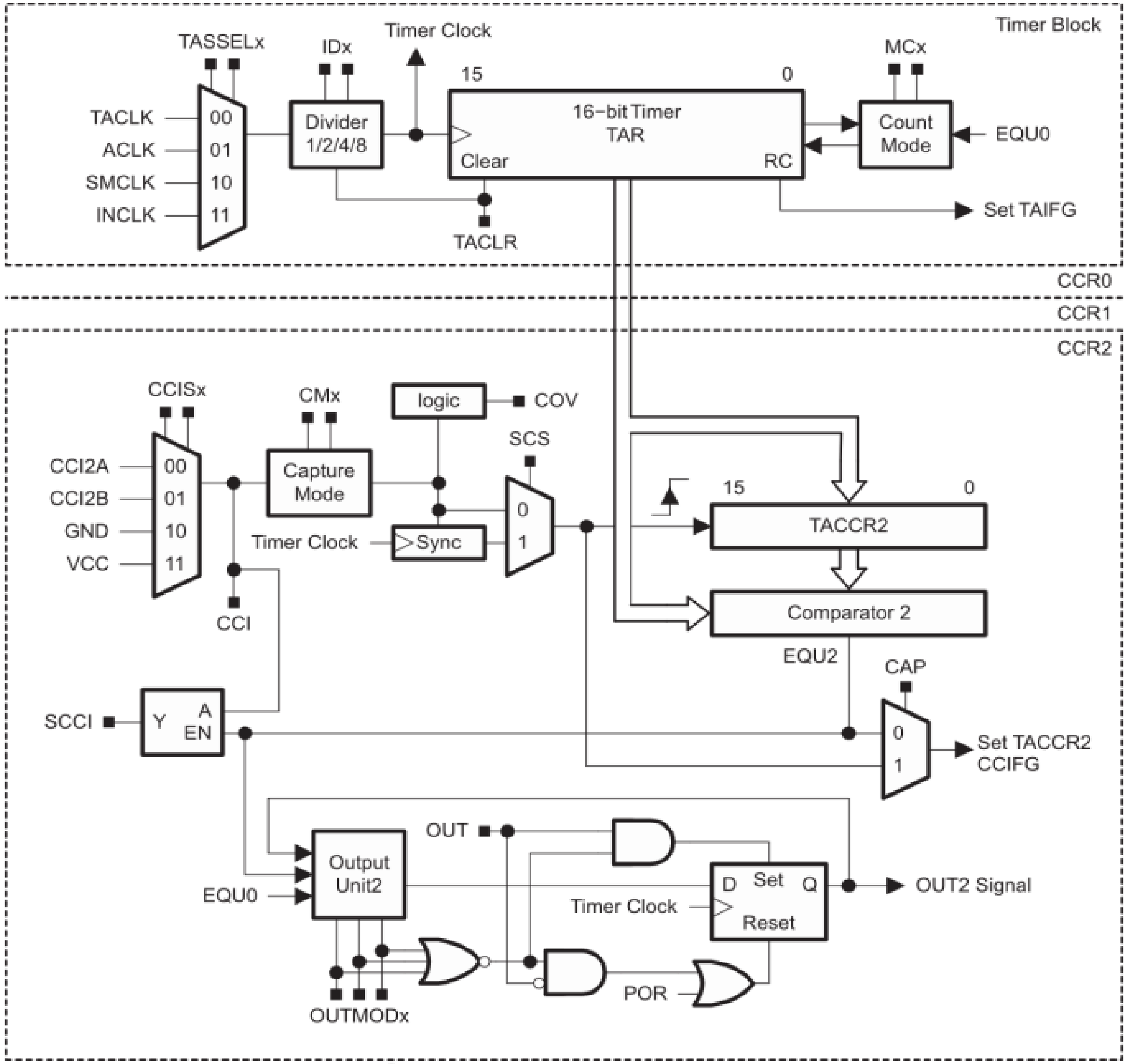


Fig. 7.22: MSP430 Timer_A block diagram

(Courtesy of Texas Instruments, Inc.)

Timer_A Operating Modes

- Stop: Timer Halted
- Up: Repeatedly Counts from 0 to TACCR0
- Continuous: Repeatedly Counts from 0 to 0FFFFh
- Up/Down: Repeatedly Counts from 0 to TACCRO and back to 0

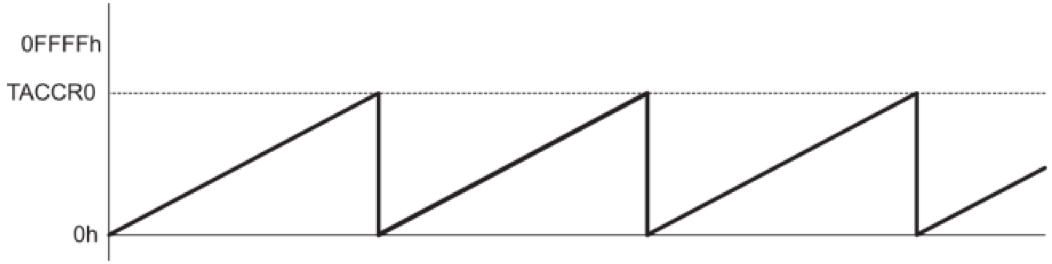


Fig. 7.24 Timer_A operating in up mode

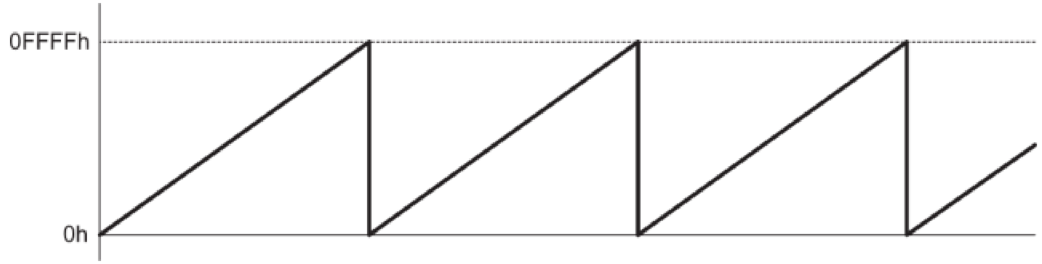


Fig. 7.25 Timer_A operating in continuous mode

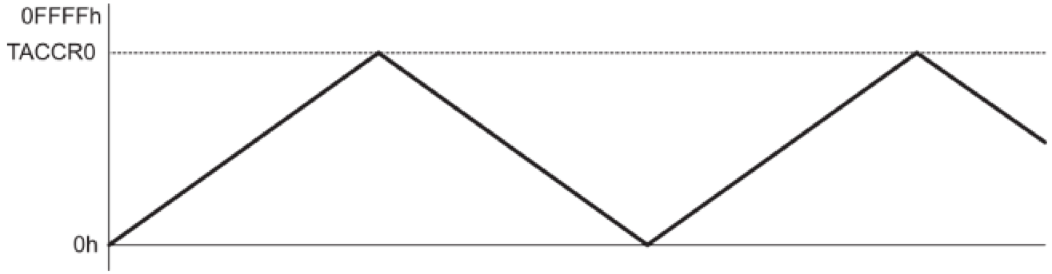
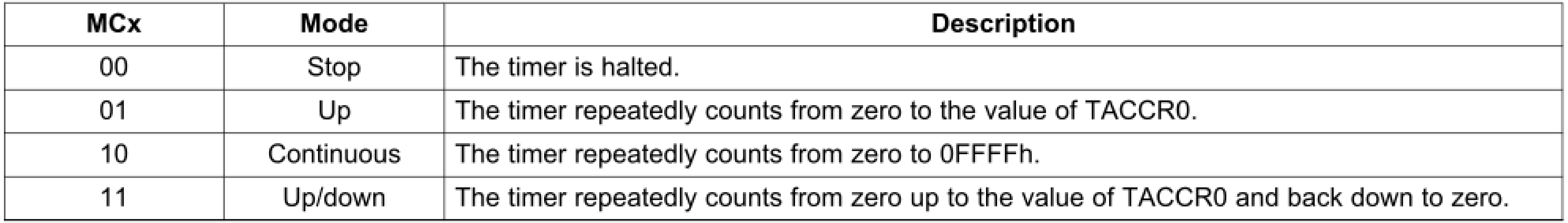


Fig. 7.26 Timer_A operating in up/down mode

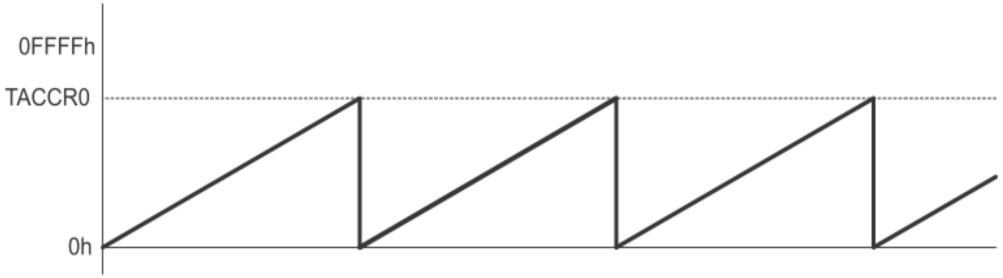
Timer Mode Control

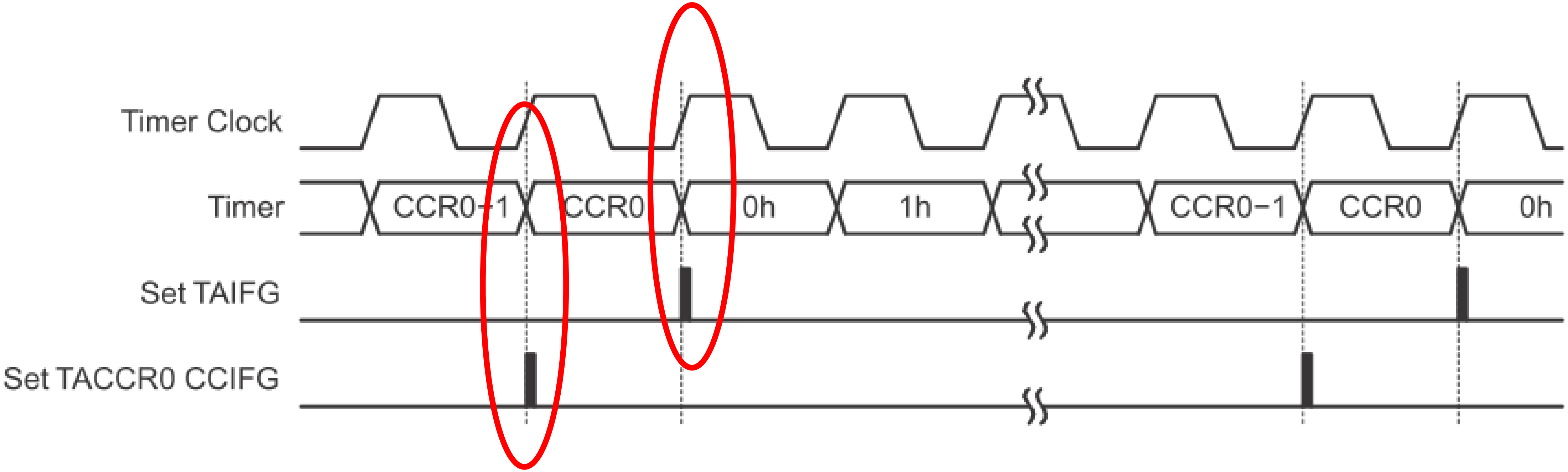


MCx	Mode	Description
00	Stop	The timer is halted.
01	Up	The timer repeatedly counts from zero to the value of TACCR0.
10	Continuous	The timer repeatedly counts from zero to 0FFFFh.
11	Up/down	The timer repeatedly counts from zero up to the value of TACCR0 and back down to zero.

Up-Mode Flag setting

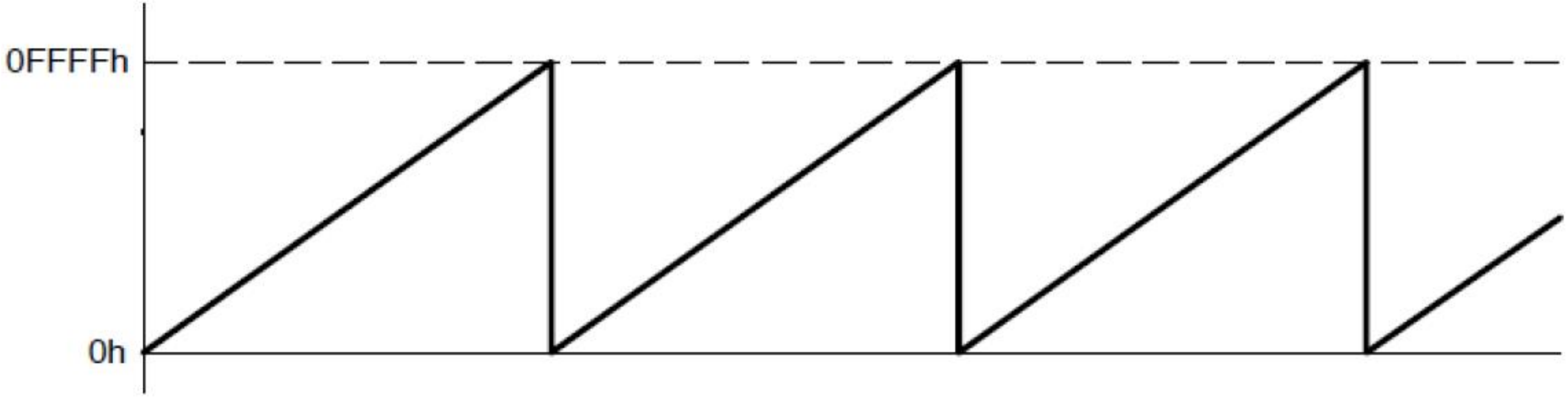
- The TACCR0 CCIFG interrupt flag is set when the timer counts to the TACCR0 value.
- The TAIFG interrupt flag is set when the timer counts from TACCR0 to zero.





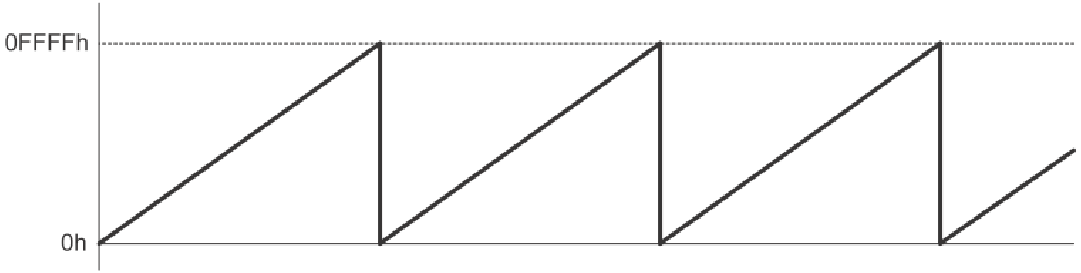
Continuous Mode

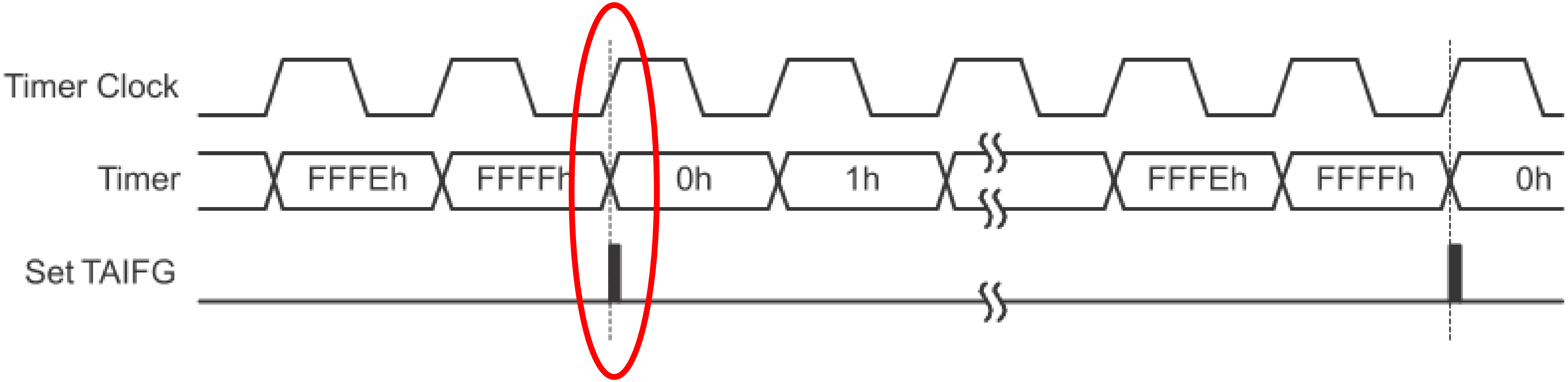
- In the continuous mode, the timer repeatedly counts up to 0FFFFh and restarts from zero as shown in the figure.
- The number of timer counts in the period is 0xFFFF + 1
- The capture/compare register TAxCCR0 works the same way as the other capture/compare registers.



Continuous Mode Flag setting

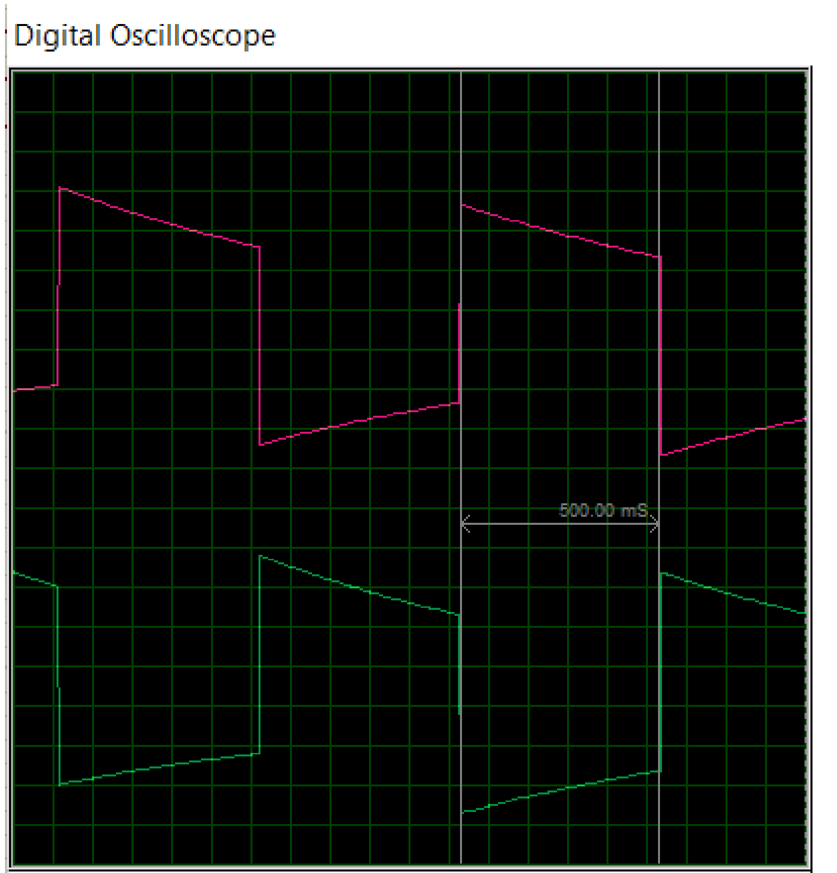
- The TAIFG interrupt flag is set when the timer counts from 0FFFFh to zero

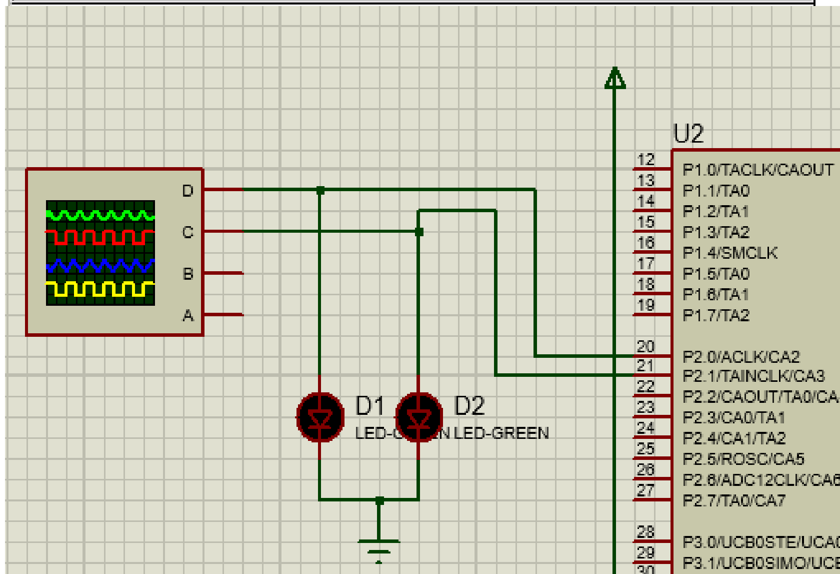




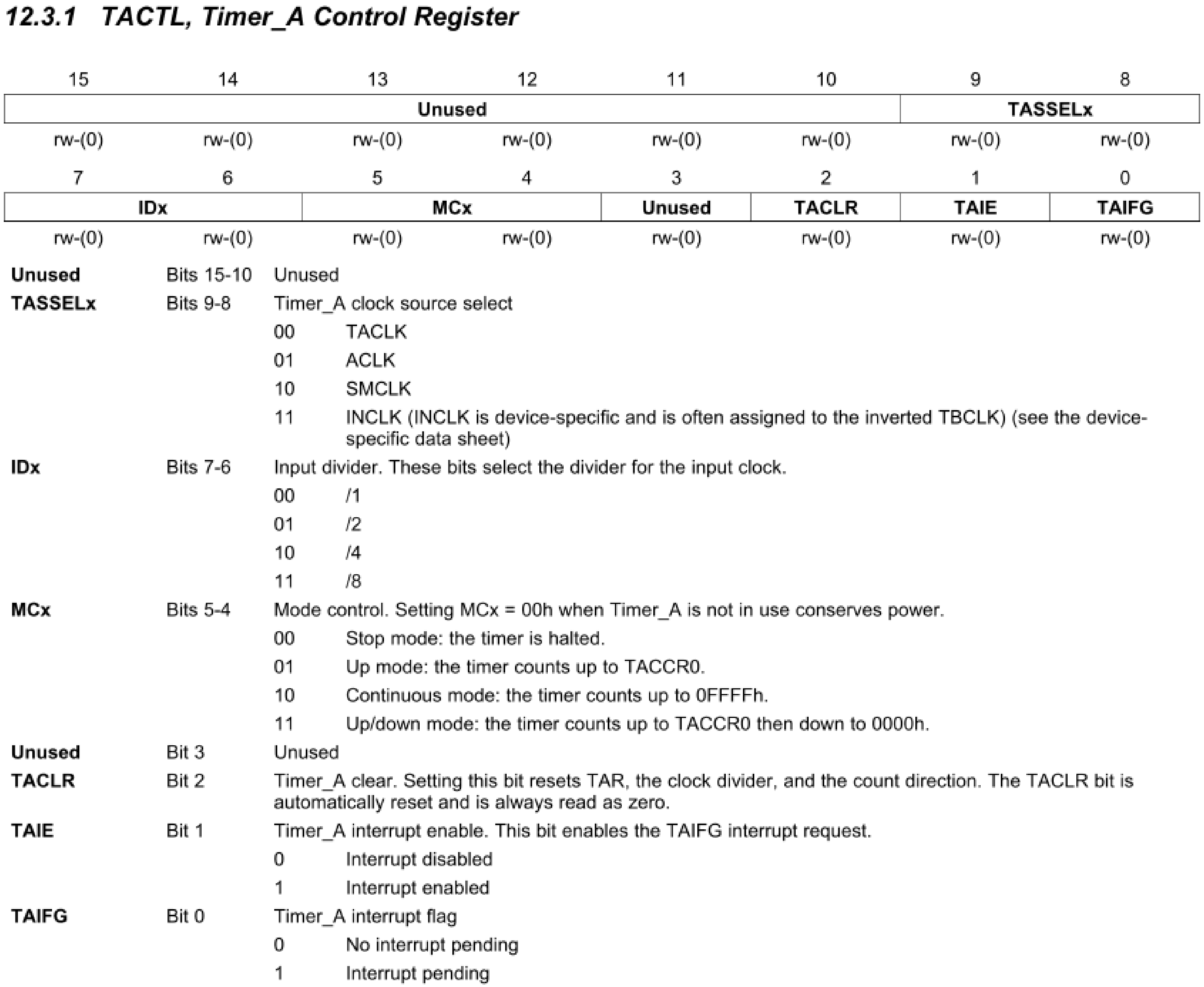
One Sec Delay using Timer

```
// Toggles LEDs in ISR using interrupts from timer A CCR0
// in Up mode with a period of 0.5s
#include <MSP430F2418.h>
#include <stdint.h>
#define LED1 BIT0
#define LED2 BIT1
uint16_t x=0;
void main (void)
{
   WDTCTL = WDTPW | WDTHOLD;    // Stop watchdog timer
   BCSCTL1 = CALBC1_1MHZ;        // Set range to calibrated 1MHz
   DCOCTL  = CALDCO_1MHZ;        // Set DCO step and modulation to calibrated 1MHz
   P2OUT = ~LED1;               // Preload LED1 on, LED2 off
   P2DIR = LED1 | LED2; // Set pins with LED1,2 to output
   TACCR0 = 50000; // Upper limit of count for TAR
   TACCTL0 = CCIE; // Enable interrupts on Compare 0
   TACTL = MC_1 | ID_0 | TASSEL_2 | TACLR; // Set up and start Timer A
   // “Up to CCR0” mode, divide by 1, clock SMCLK, clear timer

   for (;;) { // Loop forever doing nothing
    __bis_SR_register(LPM4_bits | GIE);
   } // Interrupts do the work
}
// Interrupt Service Routine for Timer A channel 0
#pragma vector = TIMERA0_VECTOR // Assoc. the funct. w/ an interrupt vector
__interrupt void TA0_ISR (void) // name of the interrupt function (can be anything)
{
   x++;
   if(x==10)
   {
   x=0;
   P2OUT ^= LED1 | LED2; // Toggle LEDs
   }
}
```





TACTL Register Configuration



12.3.1 TACTL, Timer_A Control Register

15	14	13	12	11	10	9	8
Unused						**TASSELx**	
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

7	6	5	4	3	2	1	0
IDx		**MCx**		**Unused**	**TACLR**	**TAIE**	**TAIFG**
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

Unused Bits 15-10 Unused

TASSELx Bits 9-8 Timer_A clock source select

- 00 TACLK
- 01 ACLK
- 10 SMCLK
- 11 INCLK (INCLK is device-specific and is often assigned to the inverted TBCLK) (see the device-specific data sheet)

IDx Bits 7-6 Input divider. These bits select the divider for the input clock.

- 00 /1
- 01 /2
- 10 /4
- 11 /8

MCx Bits 5-4 Mode control. Setting MCx = 00h when Timer_A is not in use conserves power.

- 00 Stop mode: the timer is halted.
- 01 Up mode: the timer counts up to TACCR0.
- 10 Continuous mode: the timer counts up to 0FFFFh.
- 11 Up/down mode: the timer counts up to TACCR0 then down to 0000h.

Unused Bit 3 Unused

TACLR Bit 2 Timer_A clear. Setting this bit resets TAR, the clock divider, and the count direction. The TACLR bit is automatically reset and is always read as zero.

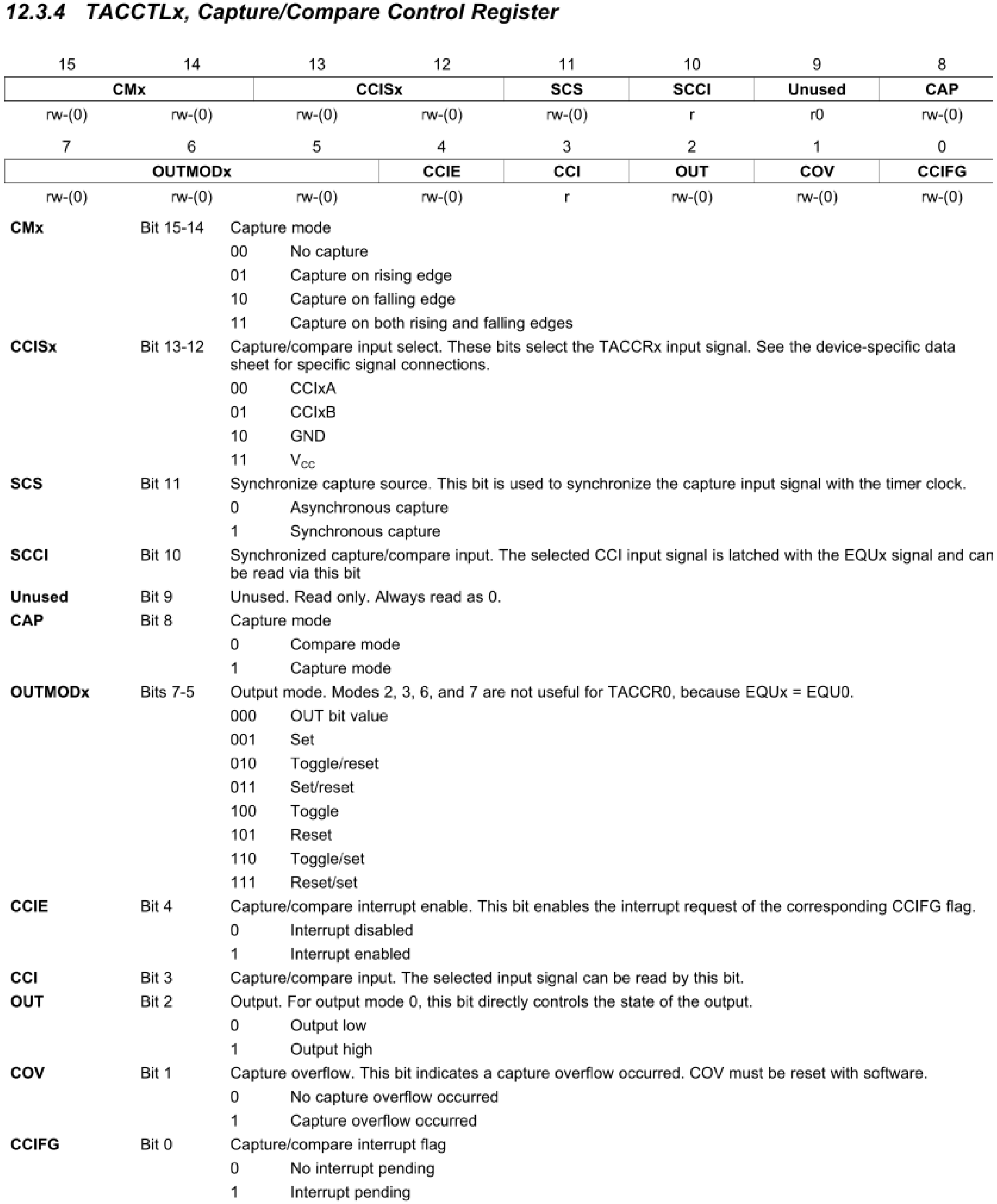
TAIE Bit 1 Timer_A interrupt enable. This bit enables the TAIFG interrupt request.

- 0 Interrupt disabled
- 1 Interrupt enabled

TAIFG Bit 0 Timer_A interrupt flag

- 0 No interrupt pending
- 1 Interrupt pending

TACCTLx Register Configuration



12.3.4 TACCTLx, Capture/Compare Control Register

15	14	13	12	11	10	9	8
CMx		**CCISx**		**SCS**	**SCCI**	**Unused**	**CAP**
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	r	r0	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)

CMx Bit 15-14 Capture mode

- 00 No capture
- 01 Capture on rising edge
- 10 Capture on falling edge
- 11 Capture on both rising and falling edges

CCISx Bit 13-12 Capture/compare input select. These bits select the TACCRx input signal. See the device-specific data sheet for specific signal connections.

- 00 CCIxA
- 01 CCIxB
- 10 GND
- 11 V_{CC}

SCS Bit 11 Synchronize capture source. This bit is used to synchronize the capture input signal with the timer clock.

- 0 Asynchronous capture
- 1 Synchronous capture

SCCI Bit 10 Synchronized capture/compare input. The selected CCI input signal is latched with the EQUx signal and can be read via this bit

Unused Bit 9 Unused. Read only. Always read as 0.

CAP Bit 8 Capture mode

- 0 Compare mode
- 1 Capture mode

OUTMODx Bits 7-5 Output mode. Modes 2, 3, 6, and 7 are not useful for TACCR0, because EQUx = EQU0.

- 000 OUT bit value
- 001 Set
- 010 Toggle/reset
- 011 Set/reset
- 100 Toggle
- 101 Reset
- 110 Toggle/set
- 111 Reset/set

CCIE Bit 4 Capture/compare interrupt enable. This bit enables the interrupt request of the corresponding CCIFG flag.

- 0 Interrupt disabled
- 1 Interrupt enabled

CCI Bit 3 Capture/compare input. The selected input signal can be read by this bit.

OUT Bit 2 Output. For output mode 0, this bit directly controls the state of the output.

- 0 Output low
- 1 Output high

COV Bit 1 Capture overflow. This bit indicates a capture overflow occurred. COV must be reset with software.

- 0 No capture overflow occurred
- 1 Capture overflow occurred

CCIFG Bit 0 Capture/compare interrupt flag

- 0 No interrupt pending
- 1 Interrupt pending

Output Compare Operation

- Common Applications
  - Generating PWM signals
  - Producing interrupts at specific time intervals
- Events Triggered by TAR Reaching TACCRx Value
  - Interrupt flag CCIFG is set and EQUx = 1
  - EQUx affects the output according to the output mode
  - The input signal CCI is latched into SCCI
- Output Unit (OU)
  - Makes timed signal available on I/O pins
  - One output unit per capture/compare block
  - Eight configurable output modes per output

Output Unit Modes

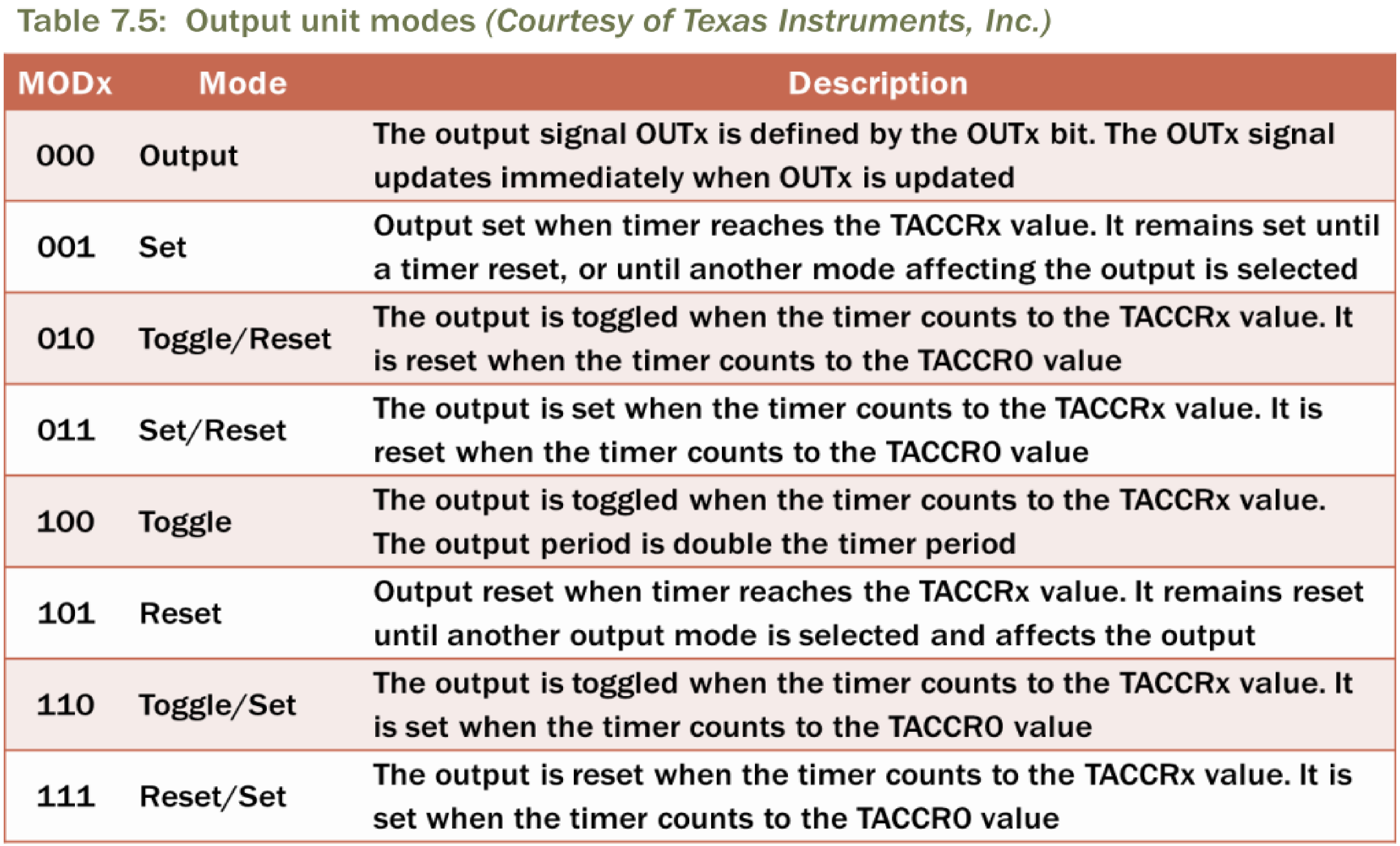
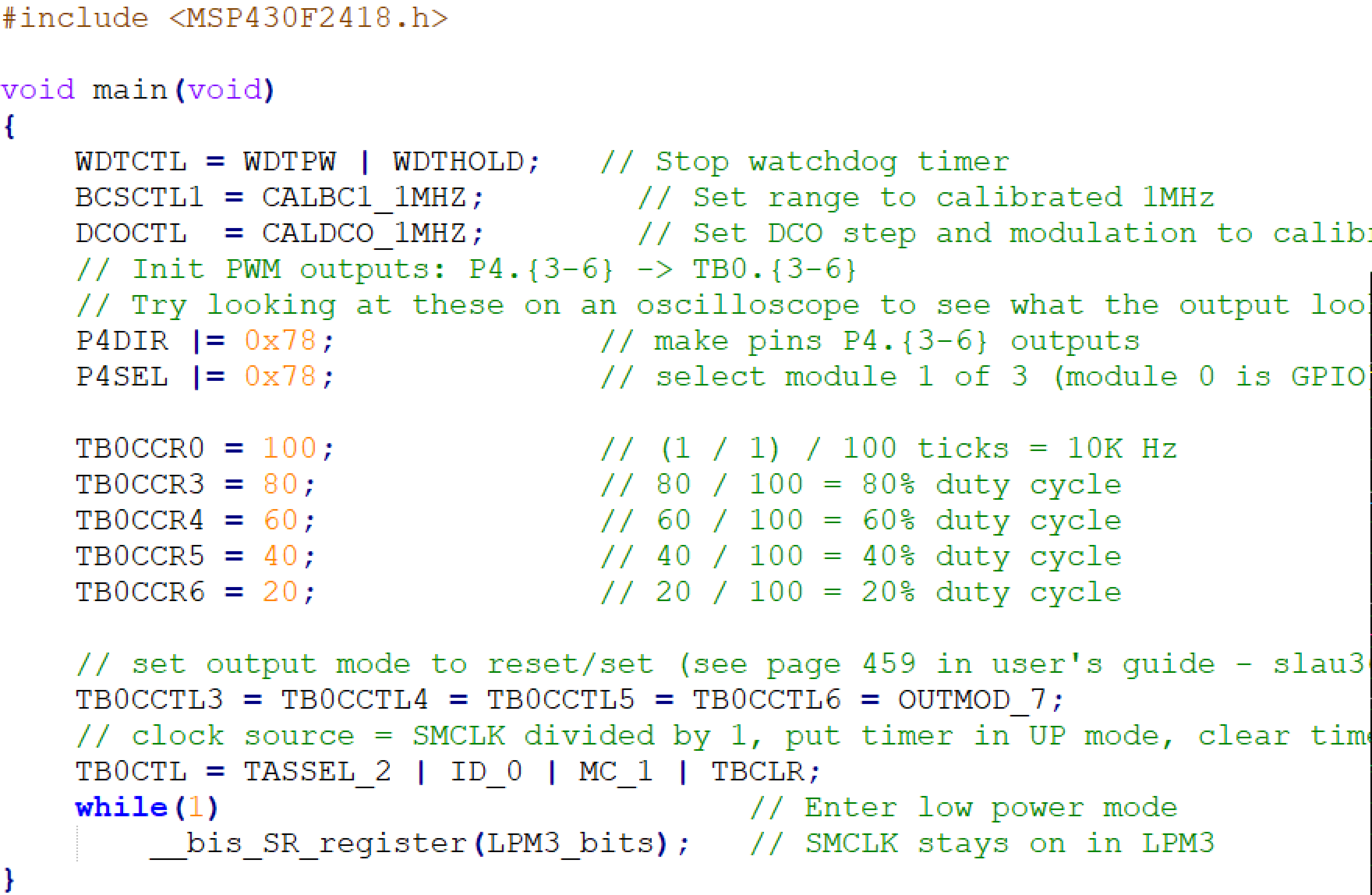


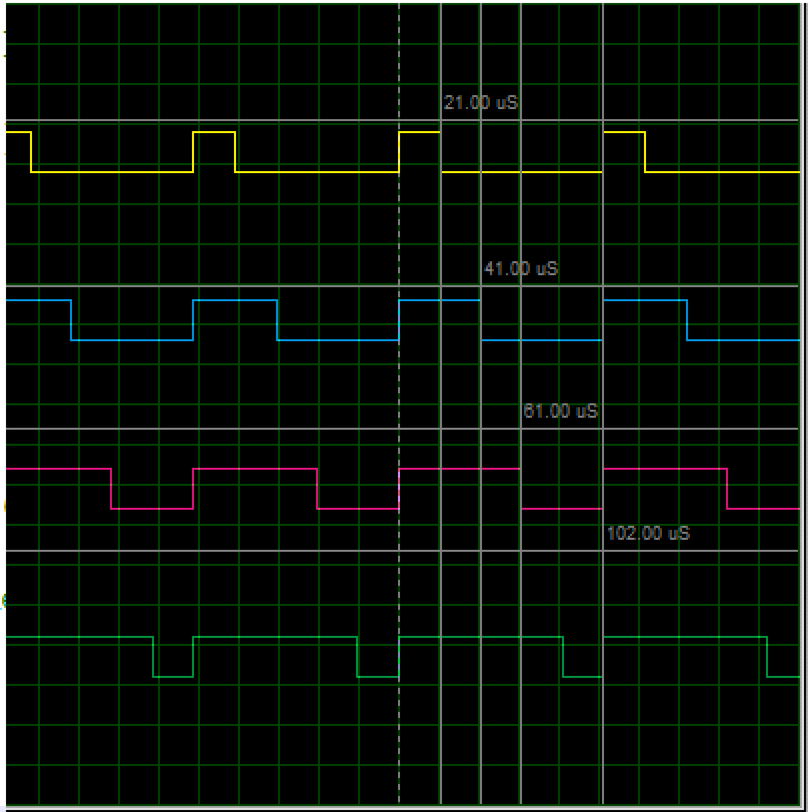
Table 7.5: Output unit modes *(Courtesy of Texas Instruments, Inc.)*

MODx	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	Output set when timer reaches the TACCRx value. It remains set until a timer reset, or until another mode affecting the output is selected
010	Toggle/Reset	The output is toggled when the timer counts to the TACCRx value. It is reset when the timer counts to the TACCR0 value
011	Set/Reset	The output is set when the timer counts to the TACCRx value. It is reset when the timer counts to the TACCR0 value
100	Toggle	The output is toggled when the timer counts to the TACCRx value. The output period is double the timer period
101	Reset	Output reset when timer reaches the TACCRx value. It remains reset until another output mode is selected and affects the output
110	Toggle/Set	The output is toggled when the timer counts to the TACCRx value. It is set when the timer counts to the TACCR0 value
111	Reset/Set	The output is reset when the timer counts to the TACCRx value. It is set when the timer counts to the TACCR0 value

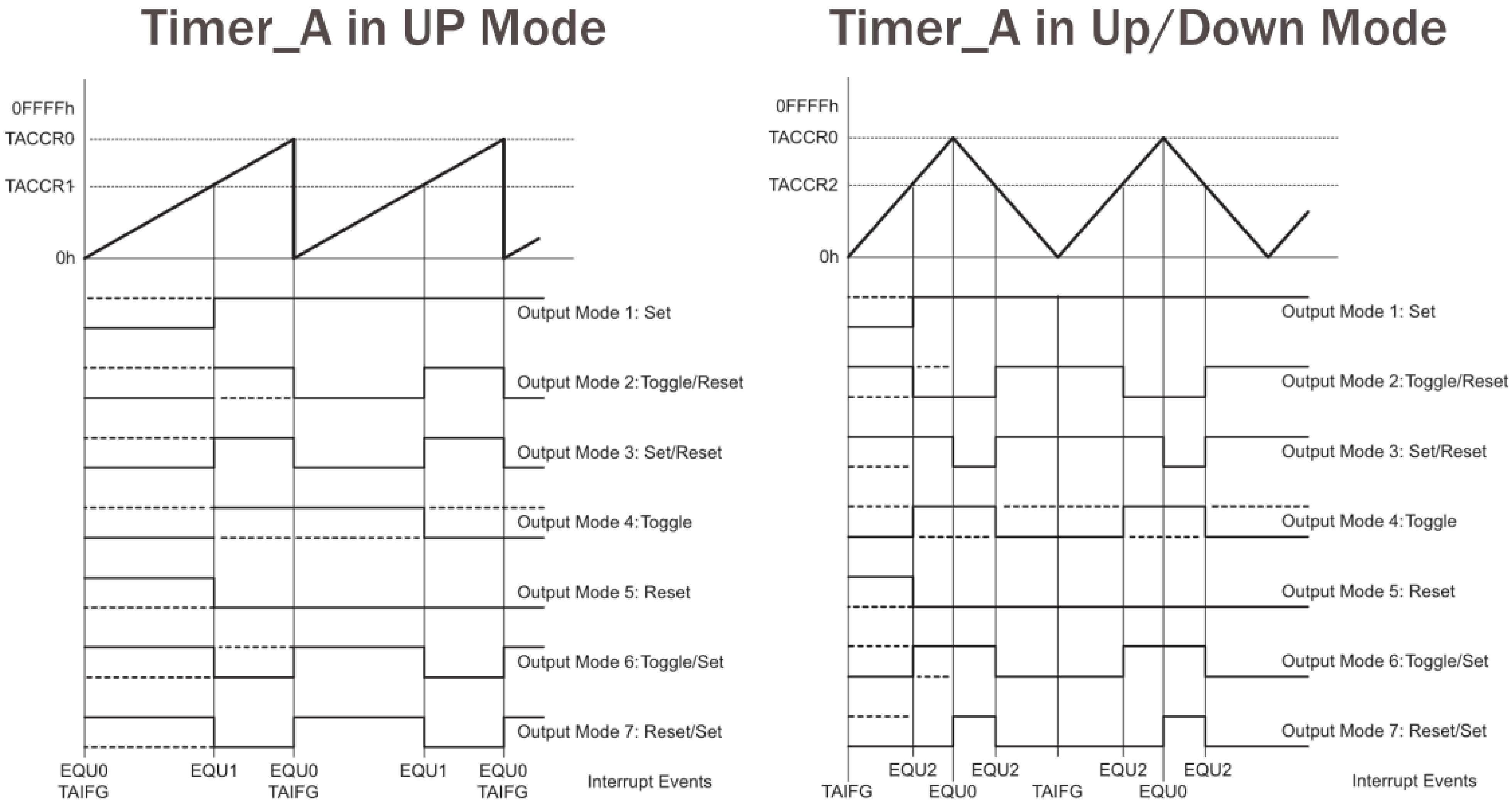
PWM Example: Output Modes



```
#include <MSP430F2418.h>

void main(void)
{
    WDTCTL = WDTPW | WDTHOLD;   // Stop watchdog timer
    BCSCTL1 = CALBC1_1MHZ;         // Set range to calibrated 1MHz
    DCOCTL  = CALDCO_1MHZ;         // Set DCO step and modulation to calibrated 1MHz
    // Init PWM outputs: P4.{3-6} -> TB0.{3-6}
    // Try looking at these on an oscilloscope to see what the output loo
    P4DIR |= 0x78;              // make pins P4.{3-6} outputs
    P4SEL |= 0x78;              // select module 1 of 3 (module 0 is GPIO

    TB0CCR0 = 100;              // (1 / 1) / 100 ticks = 10K Hz
    TB0CCR3 = 80;               // 80 / 100 = 80% duty cycle
    TB0CCR4 = 60;               // 60 / 100 = 60% duty cycle
    TB0CCR5 = 40;               // 40 / 100 = 40% duty cycle
    TB0CCR6 = 20;               // 20 / 100 = 20% duty cycle

    // set output mode to reset/set (see page 459 in user's guide - slau3
    TB0CCTL3 = TB0CCTL4 = TB0CCTL5 = TB0CCTL6 = OUTMOD_7;
    // clock source = SMCLK divided by 1, put timer in UP mode, clear time
    TB0CTL = TASSEL_2 | ID_0 | MC_1 | TBCLR;
    while(1)                            // Enter low power mode
        __bis_SR_register(LPM3_bits);   // SMCLK stays on in LPM3
}
```



Output Waveform Examples



Timer_A in UP Mode

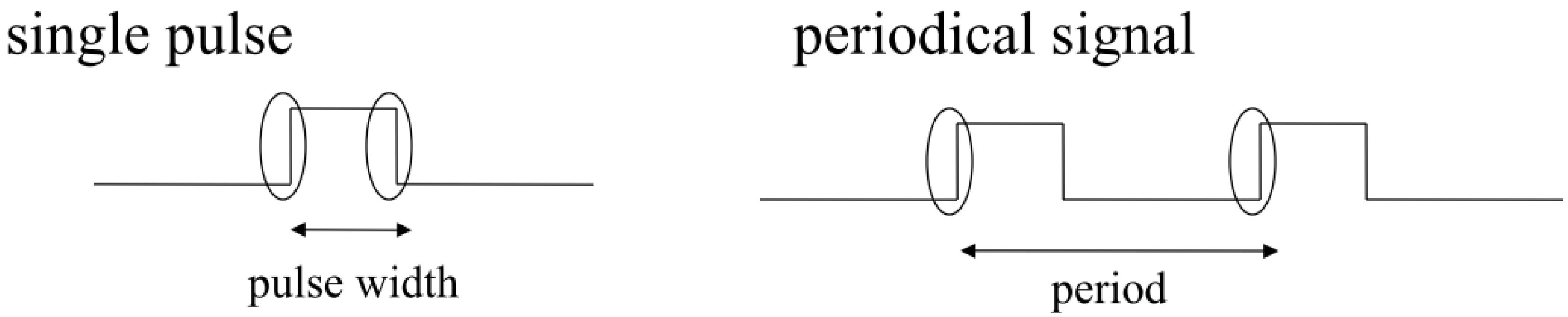
Timer_A in Up/Down Mode

Figure 7.28: Output unit waveforms *(Courtesy of Texas Instruments, Inc.)*

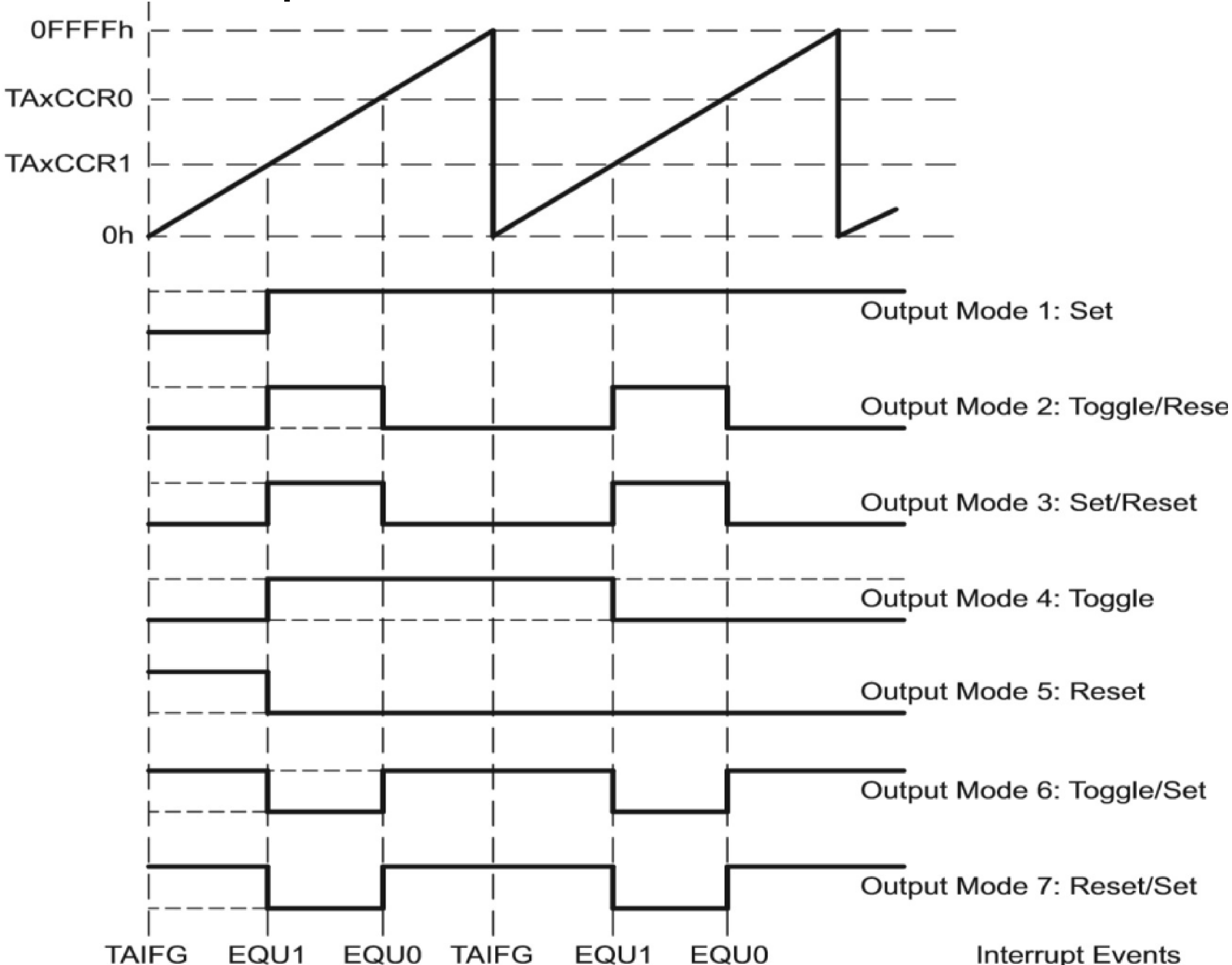
Other MSP430 Timer Resources

- Basic Timer
  - Available in legacy x3xx and x4xx devices
  - Two independent, cascadable, interrupt-driven 8-bit timers
- Timer_B
  - Same as Timer_A with seven CCRs
- Watchdog Timer +
  - Similar to WDT discussed earlier
- Real-time Clock
  - Available in x4xx, x5xx, and x6xx devices
  - 32-bit counter with calendar function
  - Seconds, minutes, hours, DOW, DOM, month, year (w/leap)
  - Selectable BCD output

Designing a PWM

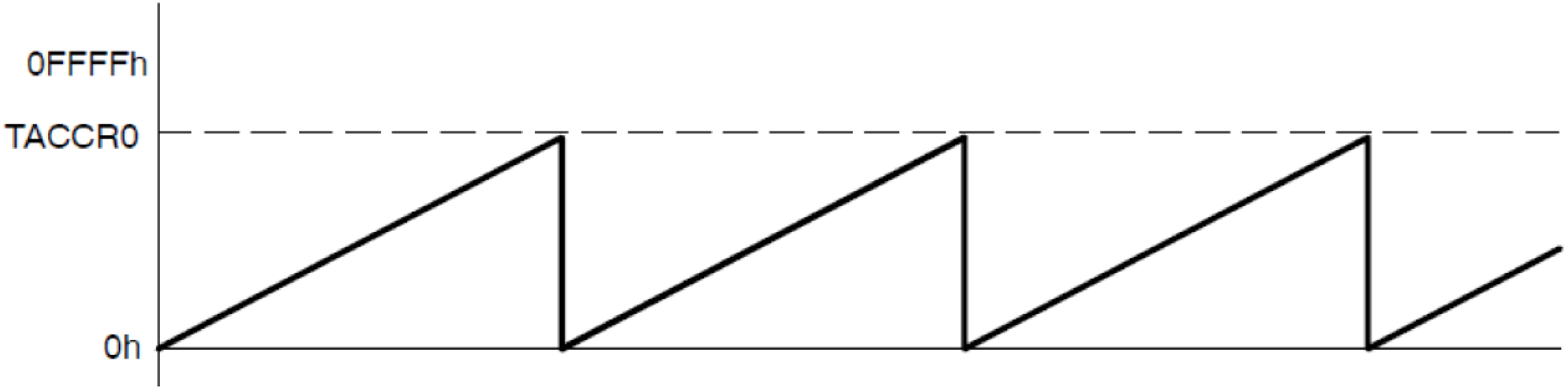


Output Example: Continuous Mode



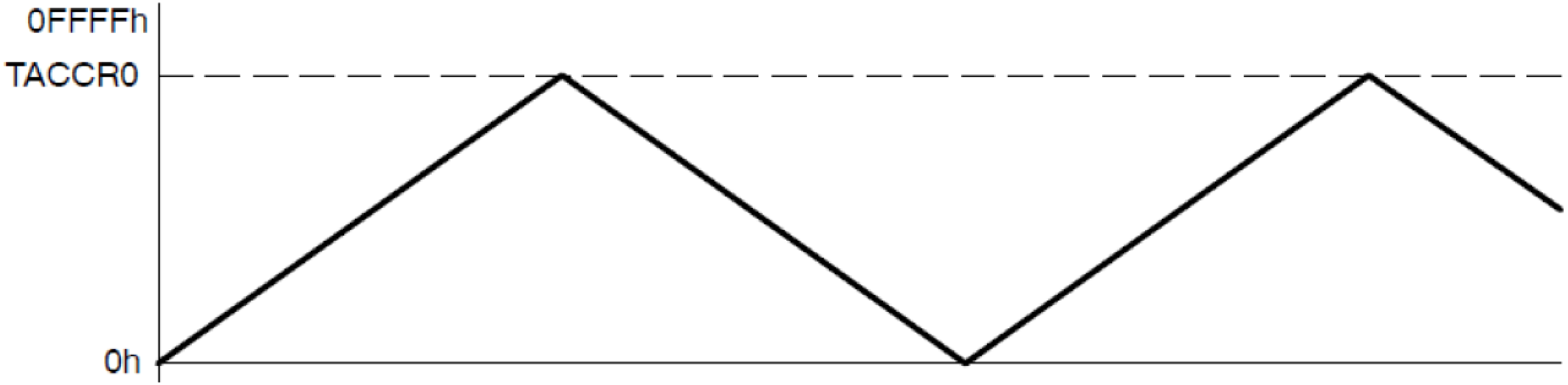
Output: Up Mode

- The timer repeatedly counts up to the value of compare register TAxCCR0, which defines the period, as shown in the figure.
- The number of timer counts in the period is TAxCCR0+1.
- When the timer value equals TAxCCR0 the timer restarts counting from zero.

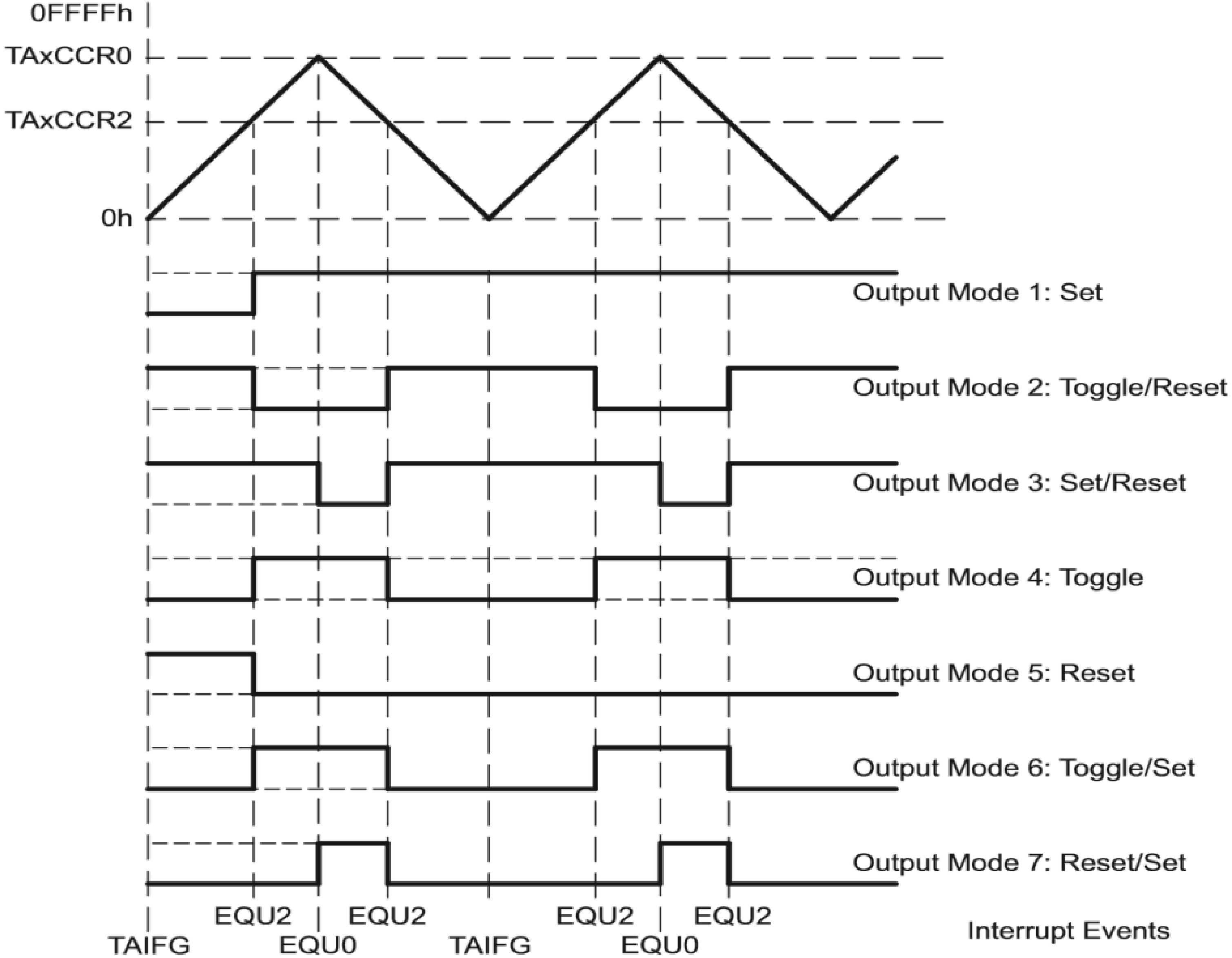


Up/Down Mode

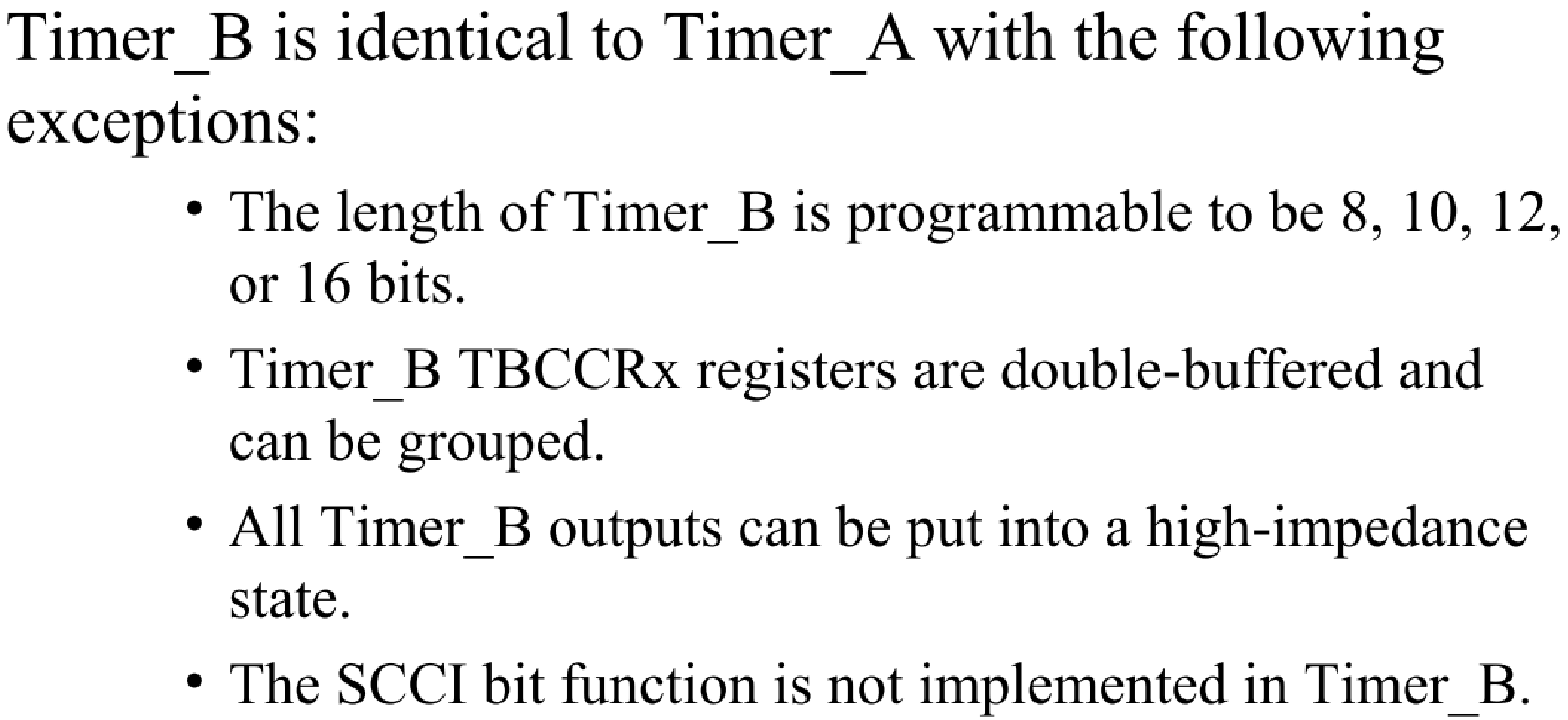
- The timer repeatedly counts up to the value of compare register TAxCCR0 and back down to zero, as shown in the figure.
- The period is twice the value in TAxCCR0.



Up/Down Mode


0FFFFh
TAxCCR0
TAxCCR2
0h
Output Mode 1: Set
Output Mode 2: Toggle/Reset
Output Mode 3: Set/Reset
Output Mode 4: Toggle
Output Mode 5: Reset
Output Mode 6: Toggle/Set
Output Mode 7: Reset/Set
TAIFG
EQU2
EQU0
EQU2
TAIFG
EQU2
EQU0
EQU2
Interrupt Events

Timer_B: Additional Features



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 TBCCRx 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.

Timer_B Schematic

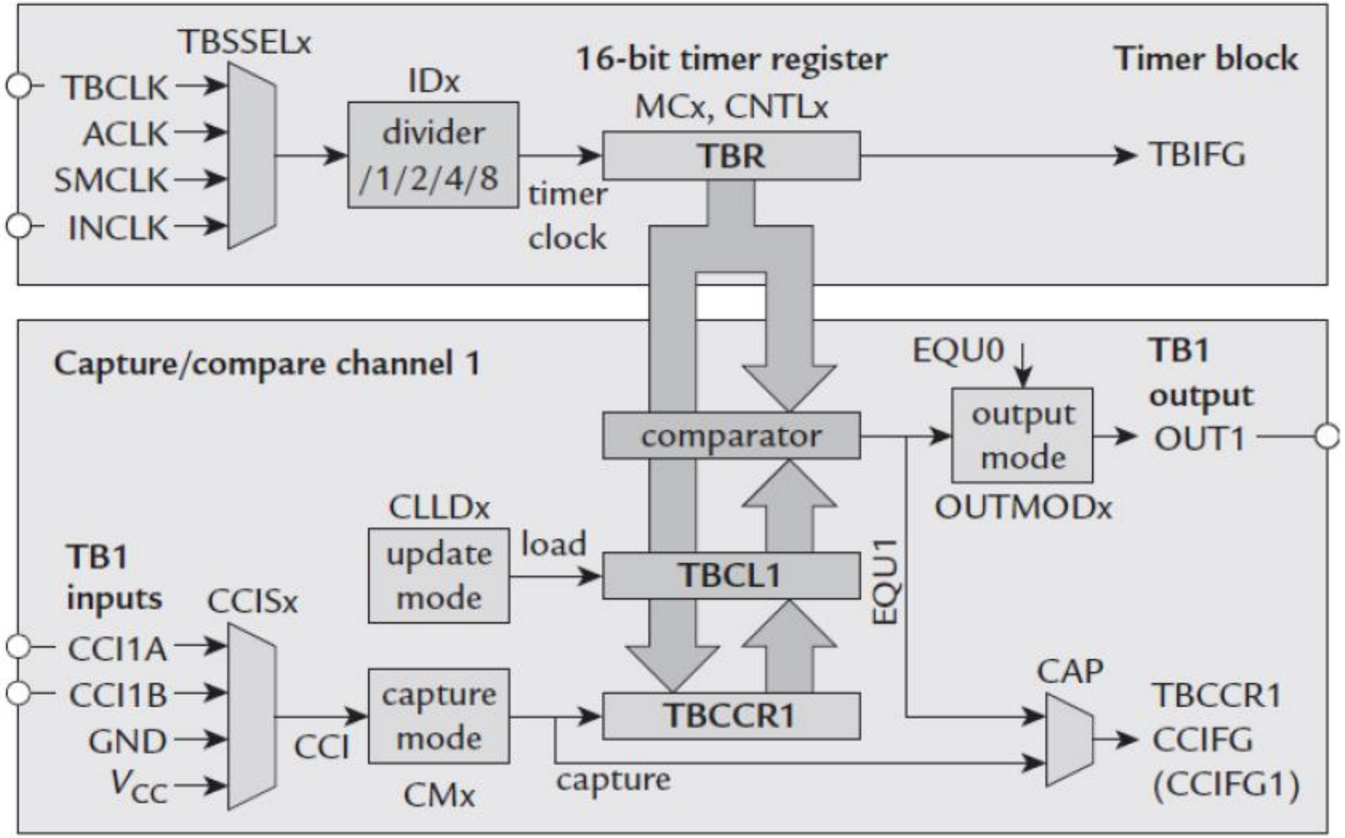


Figure 8.16: Simplified block diagram of Timer_B showing the timer block and capture/compare channel 1. Note the latch TBCL1 between the capture/compare register TBCCR1 and the comparator for the Compare mode.