ESET 269 – Embedded Systems Development in C

Timers

(Delays using Timer32)

Dr. Garth V. Crosby

Timer 32

- Similar in use to SysTick
- Timer32 is 32 bit
- Can be used as 16-bit or 32-bit timer
- To slow down the speed of timer counting, it has a prescaler
- Has the following additional options:
  - Frequency divide (prescale) system clock
  - Run mode (continuously count)
  - One-shot mode (one iteration of count)

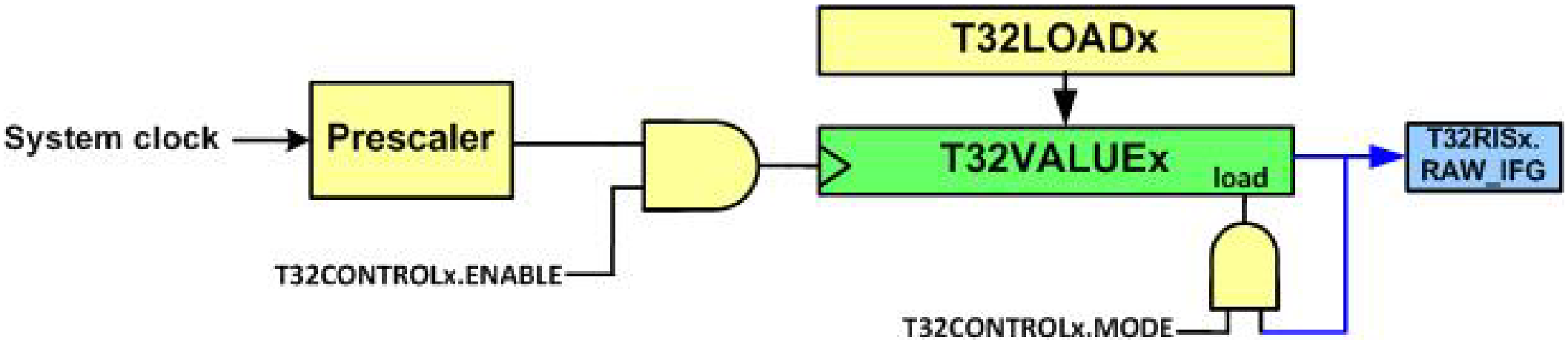
Timer 32

- MSP432 has two Timer32 timers which are identical
  - Timer 1
  - Timer 2
- Registers of Timer 32:
  - T32LOADx : Timer x Load Register
  - T32VALUEx: Timer x Current Value Register
  - T32CONTROLx: Timer x Control Register
  - T32INTCLRx: Timer x Interrupt Clear Register
  - T32RISx: Timer x Raw Interrupt Status Register
  - T32MISx: Timer x Interrupt Status Register
  - T32BGLOADx: Timer x Background Load Register

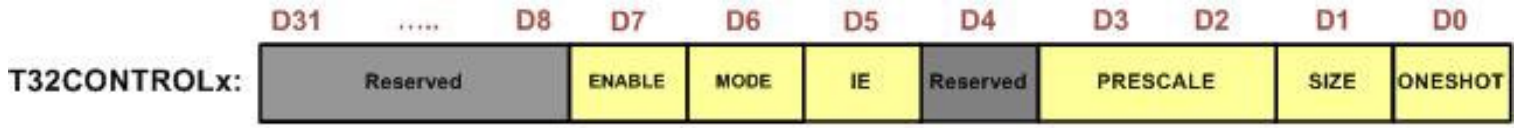
Note that ‘x’ is or 1 or 2 depending on the specific Timer32 being used.

Timer 32

- Timer32 is a 32-bit down-counter driven by the master clock
- It counts down from an initial value to 0
- When it reaches 0, the flag RAW_IFG in T32RISx register is raised
- The down counter is named T32VALUEx
- The clock is ANDed with the ENABLE bit of T32CONTROL register
  - Therefore, it counts down when the ENABLE bit is set (1).



Timer 32 Control Register



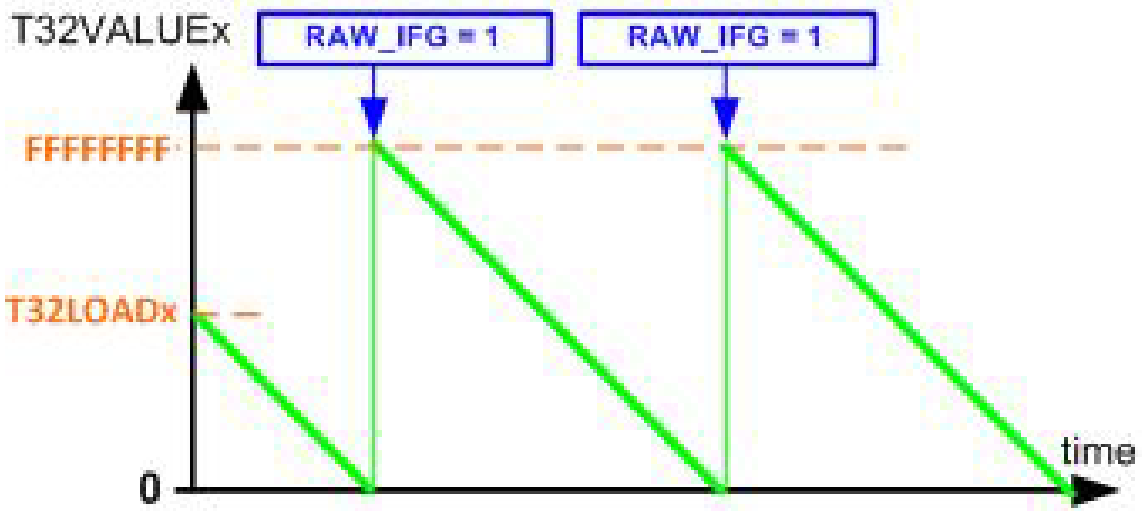
bit	Name	Description
7	ENABLE	Enable (0: the timer is disabled, 1: enables timer to begin counting down)
6	MODE	Mode bit 0: Free-running mode (The timer rolls over to its maximum value) 1: Periodic mode (The timer is reloaded with the value of the T32LOADx register)
5	IE	Interrupt Enable bit 0: Timer interrupt disabled 1: Timer interrupt enabled
3-2	PRESCALE	Prescale bits 00: clock is divided by 1 01: clock is divided by 16 10: clock is divided by 256 11: Reserved
1	SIZE	Selects 16-bit or 32-bit counter operation 0: 16-bit counter 1: 32-bit counter
0	ONESHOT	Selects one-shot or wrapping counter mode: 0: wrapping mode (The timer continues counting when it reaches to zero) 1: one-shot (The timer stops when it reaches to zero)

```
TIMER32_1 ->CONTROL
TIMER32_2 ->CONTROL
```

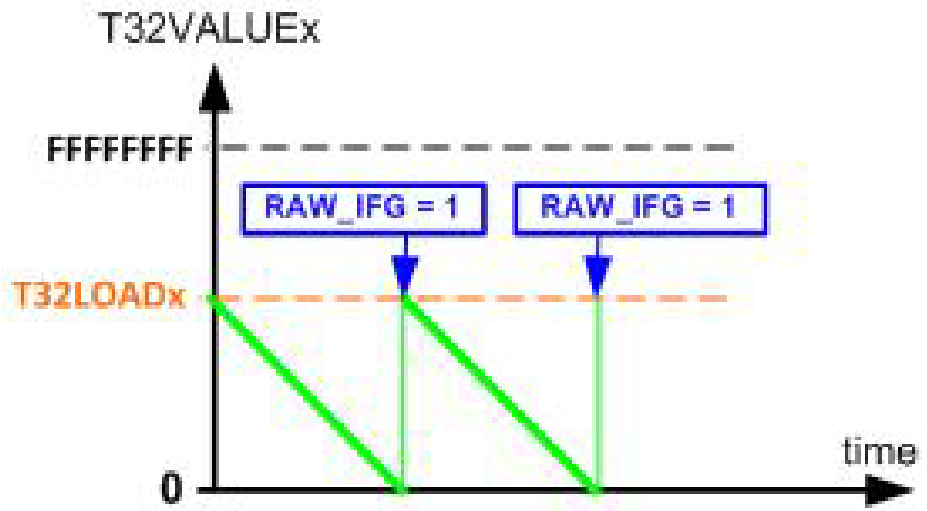
An interrupt occurs when the RAW_IFG is set

Timer 32 Modes

- Free running resets to 0xFFFFFFFF when counter reaches 0
- Periodic resets the TIMER32 Load register value when counter reaches 0



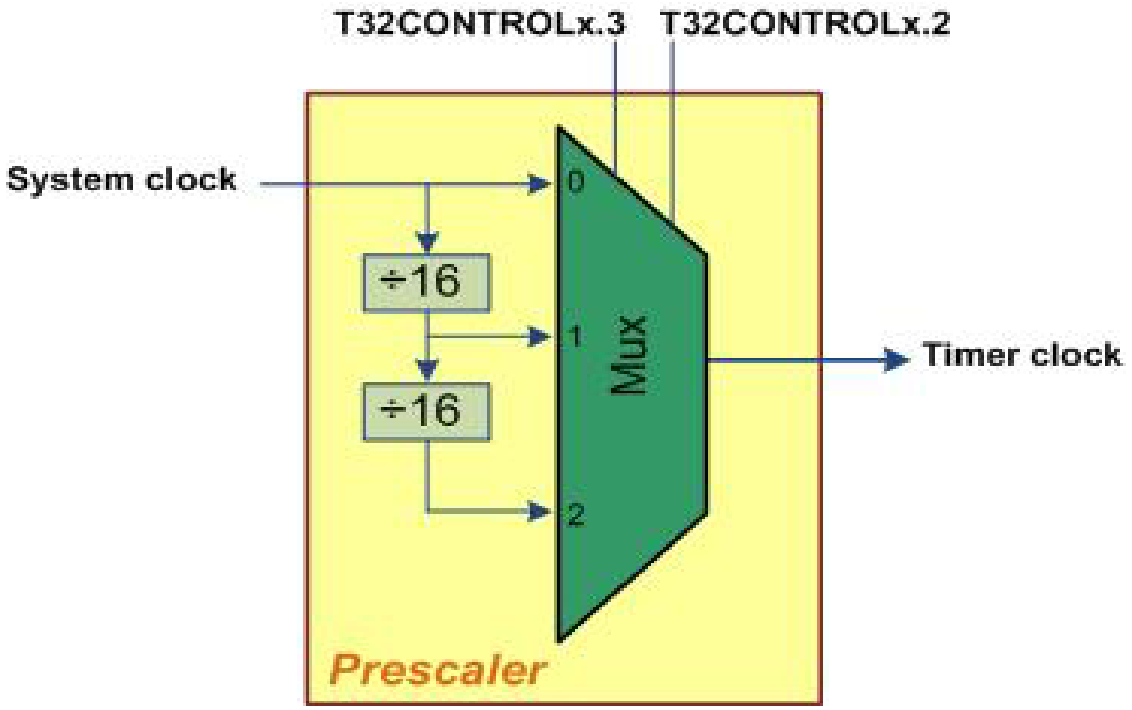
(a) Free running (MODE = 0)



(b) Periodic (MODE = 1)

Timer 32 Prescale

- Divides the system clock by 16 or 256

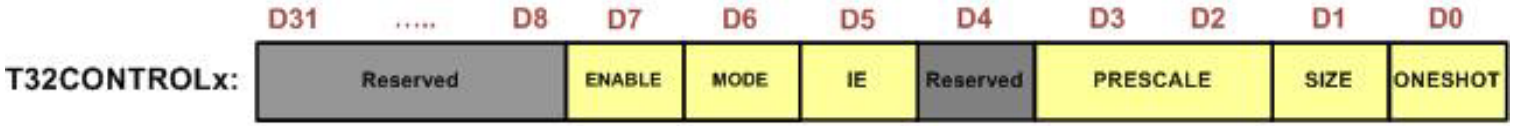


One Shot vs. Wrapping

- One shot will automatically stop the timer when it reaches 0
  - Will not reload and count down again
  - Do not need to reenable timer
- Wrapping will reload the timer and restart another countdown

Configuring Timer32

- Common configuration
  - Periodic Mode
  - No prescale
  - 32-bit size
  - Wrapping Mode



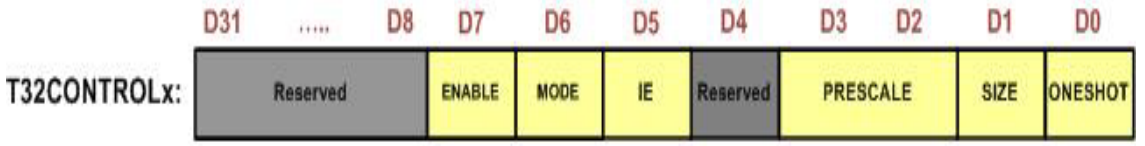
Determining if Count Reached

- When the timer reaches 0, the first bit in the T32RIS register goes to 1. It is 0 otherwise
  - This must be reset back to 0 using the T32INTCLR register

Using Timer32

1. Set the LOAD for the number of counts
2. Set the CONTROL for mode, prescale, bit size, and one-shot or wrapping. Initially disabled.
3. Whenever you want to count, enable Timer32
4. Poll the RIS if it is 1 to determine if count is finished
5. Set INTCLR to 0 when count is finished
6. Disable the timer (if not in one shot mode)
7. To reuse Timer32
   - Place LOAD value
   - Enable Timer

Example




```
//Set up Port 2 Pin 0
P2 ->SEL0 =0x00;
P2 ->SEL1 = 0x00;
P2 ->DIR |=0x01;

//set up Timer32
TIMER32_1 ->LOAD = 3000000-1;
TIMER32_1 ->CONTROL |=0x42; //periodic mode, no interurpt or prescale, 32-bit size, wrapping mode
P2 ->OUT |=0x01; //turn on pin 0
TIMER32_1 ->CONTROL |=0x80; //enable timer
while((TIMER32_1 ->RIS & 1)!=1) //while count not done
{
  //wait
}
TIMER32_1->INTCLR &=~0x01; //set INTCLR to 0 after count is reached
P2 ->OUT &=~0x01; //Turn off Pin 0
TIMER32_1 ->CONTROL &=~0x80; //disable Timer32
```

(0x42 = 0100 0010)

(0x80 = 1000 0000)

One Shot Example

(0x43 = 0100 0011)

```
TIMER32_1 ->CONTROL |= 0x43; //configure one shot mode
TIMER32_1 ->LOAD = 3000000-1; //load value into timer
TIMER32_1 ->CONTROL |=0x80; //start timer
while((TIMER32_1 ->RIS & 1) ==0) //wait
{
  //wait
}
TIMER32_1->INTCLR = 0; //clear count flag
```

*To reuse timer, place a value in the LOAD register.

Timer A

- Will not be covering TimerA
- Most advanced and versatile timer on the MSP432
- Common uses of TimerA
  - Capture and compare – determine frequency of digital signal
  - PWM (Pulse Width Modulation) for motor control
  - Count number of occurrences on digital pins