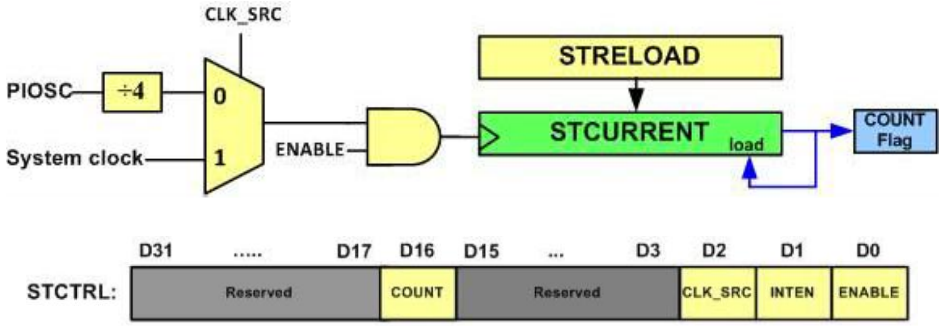
# System Tick Timer (SysTick)

The Tiva-C system timer ( SysTick ) is a simple 24-bit decrementing timer, The timer can be used as an RTOS tick timer, high-speed alarm timer, or a simple counter.

### The Timer Registers

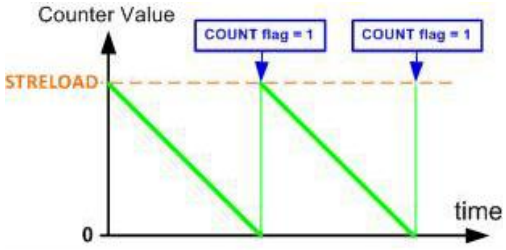
The Timer consists of three registers:

- SysTick Control and Status (**STCTRL**): used to enable the counter, enable the interrupt, select the clock source and determine the counter status. Use **`SysTick->CTRL`** to access it.
- SysTick Reload Value (**STRELOAD**): used to hold the reload value for the counter. Use **`SysTick->LOAD`** to access it.
- SysTick Current Value (**STCURRENT**): holds the current value of the counter. Use **`SysTick->VAL`** to access it.




### How it works

After Starting, the timer counts down on each clock from the reload value in the **STRELOAD** register to zero, then starts again counting from the value in the **STRELOAD** register on the next clock. When it reaches 0, in the next clock, it underflows and it raises a flag called **COUNT** (field in the **STCTRL** register).




If **STRELOAD** register is loaded with N, then the SysTick needs N+1 clock cycles to set **COUNT** flag. Hence, the time needed to set **COUNT** is **`(N+1)/CLK`**. The following function makes a delay of 1ms. Assume sysclk = 16MHz

```
void delay(void)
{
      SysTick->LOAD = 15999;
      SysTick->CTRL = 0x5;                    /*Enable the timer and choose sysclk */
      while((SysTick->CTRL & 0x10000) == 0)   /*wait until the Count flag is set */
      { }
      SysTick->CTRL = 0;                      /*Stop the timer (Enable = 0) */
}
```