SysTick is a simple counter that we can use to create time delays and generate periodic interrupts. It exists on all Cortex-M microcontrollers, so using SysTick means the system will be easy to port to other microcontrollers. Table 9.1 shows the register definitions for SysTick. The basis of SysTick is a 24-bit down counter that runs at the bus clock frequency. There are four steps to initialize the SysTick timer. First, we clear the **ENABLE** bit to turn off SysTick during initialization. Second, we set the **RELOAD** register. Third, we write to the **NVIC_ST_CURRENT_R** value to clear the counter. Lastly, we write the desired mode to the control register,**NVIC_ST_CTRL_R**. The **CLK_SRC** bit specifies which clock source to use. On the TM4C123, there is no option, we must set **CLK_SRC**=1, so the counter runs off the system clock. In Chapter 12, we will set **INTEN** to enable interrupts, but in this first example we clear **INTEN** so interrupts will not be requested. We need to set the **ENABLE** bit so the counter will run. When the **CURRENT** value counts down from 1 to 0, the **COUNT** flag is set. On the next clock, the **CURRENT** is loaded with the **RELOAD** value. In this way, the SysTick counter (**CURRENT**) is continuously decrementing. If the **RELOAD** value is *n*, then the SysTick counter operates at modulo *n*+1 (…*n*, *n*-1, *n*-2 … 1, 0, *n*, *n*-1, …). In other words, it rolls over every *n*+1 counts. In this chapter we set **RELOAD** to 0x00FFFFFF, so the **CURRENT** value is a simple indicator of what time it is now.

Address	31-24	23-17	16	15-3	2	1	0	Name
$E000E010	0	0	COUNT	0	CLK_SRC	INTEN	ENABLE	NVIC_ST_CTRL_R
$E000E014	0	24-bit RELOAD value						NVIC_ST_RELOAD_R
$E000E018	0	24-bit CURRENT value of SysTick counter						NVIC_ST_CURRENT_R

Table 9.1. SysTick registers.

Without activating the phase-lock-loop (PLL), our TM4C123 LaunchPad will run at 16 MHz, meaning the SysTick counter decrements every 62.5 ns. If we activate the PLL to run the microcontroller at 80 MHz, then the SysTick counter decrements every 12.5 ns. In general, if the period of the core bus clock is *t*, then the **COUNT** flag will be set every (*n*+1)*t*. Reading the **NVIC_ST_CTRL_R** control register will return the **COUNT** flag in bit 16 and then clear the flag. Also, writing any value to the **CURRENT** register will reset the counter to zero and clear the **COUNT** flag. Program 9.1 initializes the SysTick. To determine the time, one simply reads the **CURRENT** register.

With the **RELOAD** set to 0x00FFFFFF, SysTick behaves like a clock. Every *t* time it counts down by one, and when it reaches zero, the counter is reloaded with 0x00FFFFFF and it continues to count.

```
#define NVIC_ST_CTRL_R      (*((volatile unsigned long *)0xE000E010))
#define NVIC_ST_RELOAD_R    (*((volatile unsigned long *)0xE000E014))
#define NVIC_ST_CURRENT_R   (*((volatile unsigned long *)0xE000E018))
void SysTick_Init(void){
  NVIC_ST_CTRL_R = 0;               // 1) disable SysTick during setup
  NVIC_ST_RELOAD_R = 0x00FFFFFF;    // 2) maximum reload value
  NVIC_ST_CURRENT_R = 0;            // 3) any write to current clears it
  NVIC_ST_CTRL_R = 0x00000005;      // 4) enable SysTick with core clock
}
```

Program 9.1. Initialization of SysTick.

Help