BLG 212E
Microprocessor Systems
Recitation 5

Sadettin Fidan, Alparslan Tavukçu

December 16, 2024

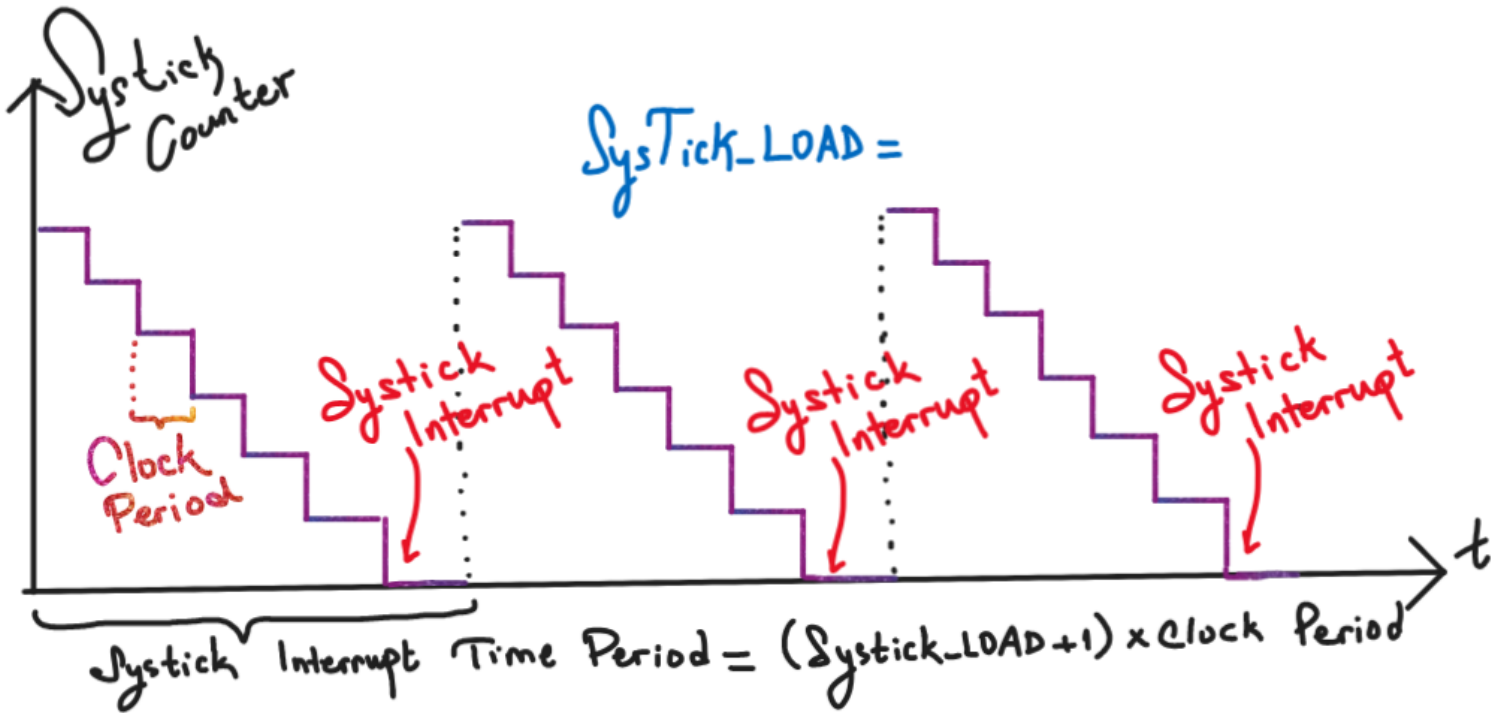
Introduction

System Timer, "SysTick"

- ⋆ Each ARM Cortex-M processor core is equipped with a timer called the System tick timer, abbreviated as Systick timer.
- ⋆ The SysTick timer allows the system to initiate a cyclic action.
- ⋆ The SysTick timer is a 24-bit down-counter with automatic reloading of the count value.
- ⋆ It is controlled by the system clock signal or the internal oscillator of the microcontroller chip.

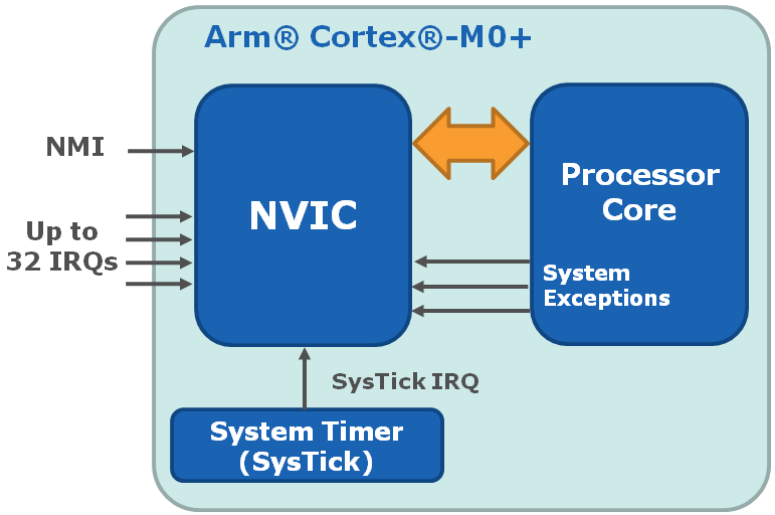
System Timer: SysTick

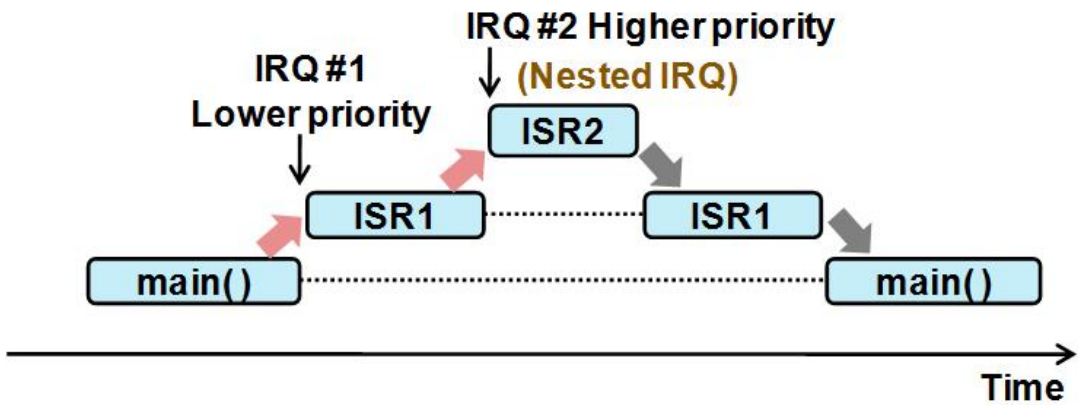
- ⋆ It counts down from an initial value to 0. When the count reaches 0, the counter sets the COUNT status flag and generates an interrupt (interrupt number 15) to reload the initial value and repeat the counting process. We can set the initial value to any value in the range 0x000000 to 0xFFFFFF.
- ⋆ Maximum interrupt period = 16.777.216 x Clock Period



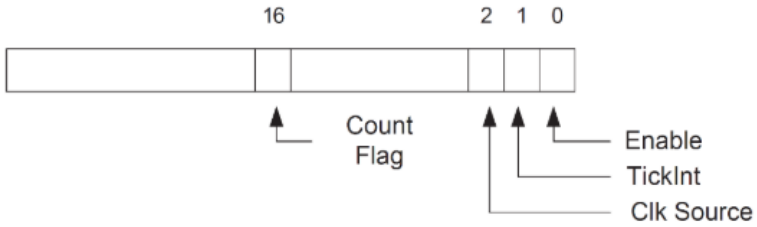
Arm Cortex M0+ Diagram

⋆ The interrupt lines from the SysTick timer and all peripheral lines of the microcontroller are connected to the Nested Vectored Interrupt Controller (NVIC).





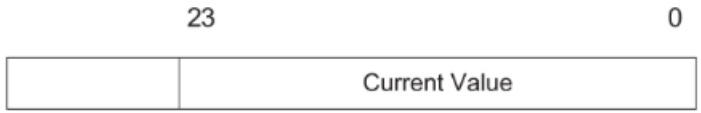
Systick Timer Registers (STCTRL)

Symbol	Format	Address
STCTRL		Base address + 0x010
SysTick®CTRL		

Field	Description
Bit 0: Enable	Systick timer trigger signal - 0: Prevent Systick timer from working - 1: Allow Systick timer to work
Bit 1: INTEN (Interrupt Enable)	- 0: Prevent systick timer from creating interrupts - 1: Allows the systick timer to generate an interrupt request when it counts to 0
Bit 2: CLK_SRC (Clock Source)	Choose clock source: - 0: Clock from high precision internal oscillator (PIOSC) divided by 4 - 1: System clock
Bit 16: Count (Counting Flags)	- 0: Systick timer has not counted to 0 since the last time this bit was read - 1: Systick timer has counted to 0

Table 1: Systick Timer Configuration

Systick Timer Registers (STRELOAD, STCURRENT)

Symbol	Format	Address
STRELOAD SysTick®LOAD		Base address + 0x014
STCURRENT SysTick®VAL		Base address + 0x018

- ⋆ The STRELOAD register is located at 0xE000E014.
- ⋆ The STRELOAD register is used to program the starting value of the systick counter - the STCURRENT.
- ⋆ If we want to get a period corresponding to 1000 clock cycles, we set STRELOAD = 999.
- ⋆ Although these are 32-bit registers, only the lower 24 bits are used. Therefore, the highest value can be loaded into the register is 0xFFFFFF (16,777,216)

Question 1

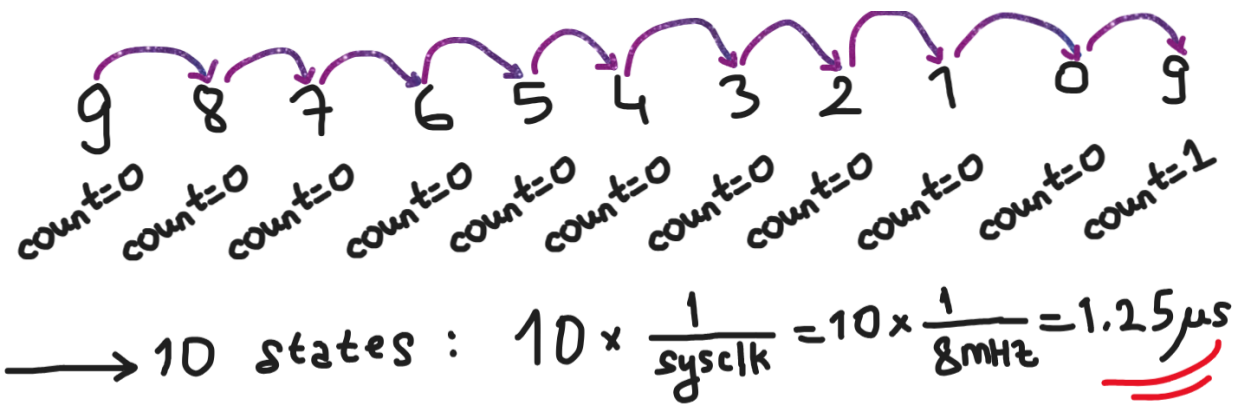
Consider an ARM microcontroller with a system clock of 8 MHz. Calculate the delay generated by the following function:

```
1 void      delay(void)
2 {
3       SysTick->LOAD = 9;
4       SysTick->CTRL = 5; /*Enable timer and select system
            clock as clock source */
5       while((SysTick->CTRL &0x10000) == 0) /*Wait until Count
            flag is set */
6       {
7       }
8       SysTick->CTRL = 0; /*Stop timer (Enable = 0) */
9 }
```

delay.c

Solution 1

- ⋆ The function call and its execution also consume a few clock cycles. If we want to calculate the exact amount of latency, we must include this extra time. However, in this book, we do not consider it because in most cases the time is negligible.
- ⋆ So here is the final solution:



Question 2

Question Statement: Consider an ARM microcontroller with the system clock selected as the clock source for the Systick timer. Calculate the delay generated by the timer if the STRELOAD register is loaded with the value N.

Answer:

Since the timer is initialized with value N, it will go through N+1 states.
Since the system clock is used as the clock source for the counter, each pulse lasts for a period equal to 1/sysclk .
So the program will generate a delay equal to (N+1)/sysclk.

Question 3

Question Statement: Write a Systick timer control function that generates a 1ms delay. Assume that the system clock frequency is sysclk = 41.94MHz

Answer:

Delay = (N + 1) / sysclk

hence: N = Delay x sysclick + 1 = 41939, therefore

```
1  void      delay1ms(void)
2  {
3       SysTick->LOAD = 41939;
4       SysTick->CTRL = 0x5; /* Enable timer and select
            sysclk as clock source */
5       while ((SysTick->CTRL & 0x10000) == 0) /* wait
            until COUNT flag is set */
6       {
7
8       }
9       SysTick->CTRL = 0; /* Stop timer (Enable = 0) */
10 }
```

References

https://documentation-service.arm.com/static/60411750ee937942ba301773?token=
https://dayhocstem.com/blog/2023/03/frdm-kl46z-8-lap-trinh-systick-timer-tren-cac-vi-xu-ly-cortex-m.html
https://community.arm.com/arm-community-blogs/b/architectures-and-processors-blog/posts/beginner-guide-on-interrupt-latency-and-interrupt-latency-of-the-arm-cortex-m-processors