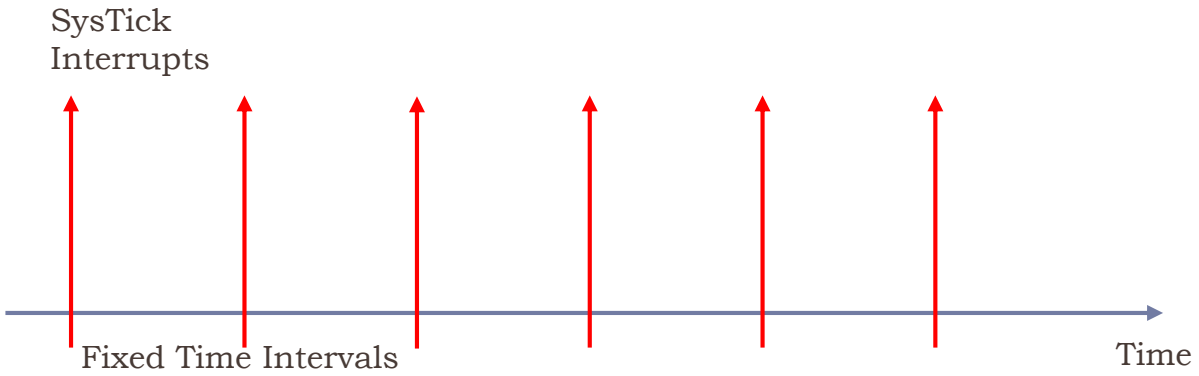
System Timer (SysTick)

TM4C123GH6PM7

SysTick Timer

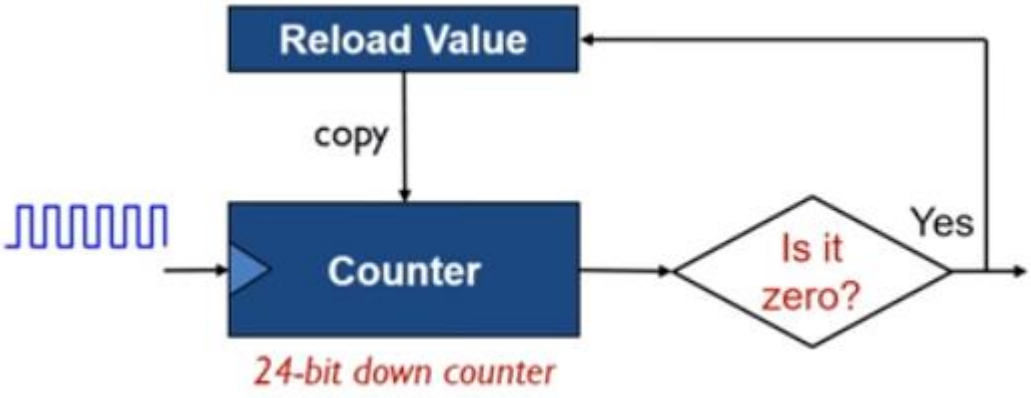
- **SysTick Timer** is a simple counter that can be used to create time delays and generate periodic interrupts.
- SysTick Timer is a standard timer integrated into ARM Cortex-M microcontrollers.
- SysTick is a **24-bit down counter**.
  - For the Tiva LaunchPad, SysTick timer runs on either the bus clock frequency or precision internal oscillator (PIOSC).



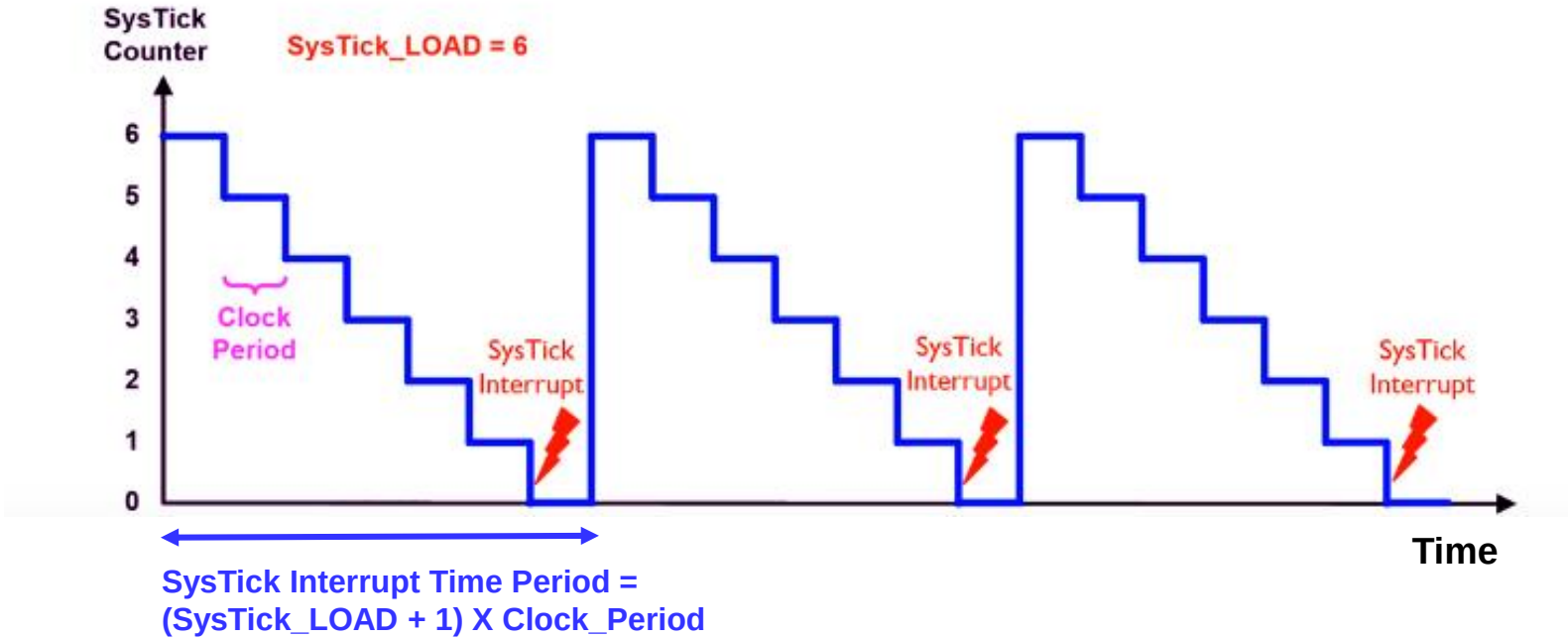
- Common Uses of SysTick:
  - Executing tasks periodically, like polling, RTOS tick.
  - Measuring time periods.

SysTick Timer – How it Works

- SysTick Timer is a **24-bit DOWN counter** that counts down on every clock cycle.
- There is a **RELOAD** value & **CURRENT** value.
- CURRENT value counts down on every clock cycle.
- When CURRENT value reaches 0, the COUNT bit (STCTRL register) is set and an interrupt is generated (*if enabled)*.
- SysTick timer is then automatically reloaded at the next clock cycle with the RELOAD value and counting continues.



SysTick Timer - How it Works



- If SysTick Timer Reload (SysTick_LOAD) value is 6, count down starts from 6.
- SysTick counts down in the following sequence: 6 → 5 → 4 → 3 → 2 → 1 → 0.
- SysTick Interrupt occurs when SysTick Counter = 0.
- Thus, Interrupt Period = 7 × Clock_Periods.
- At the next clock cycle, CURRENT counter is loaded to 6 again.
- Cycle then repeats ….

SysTick Timer ISR

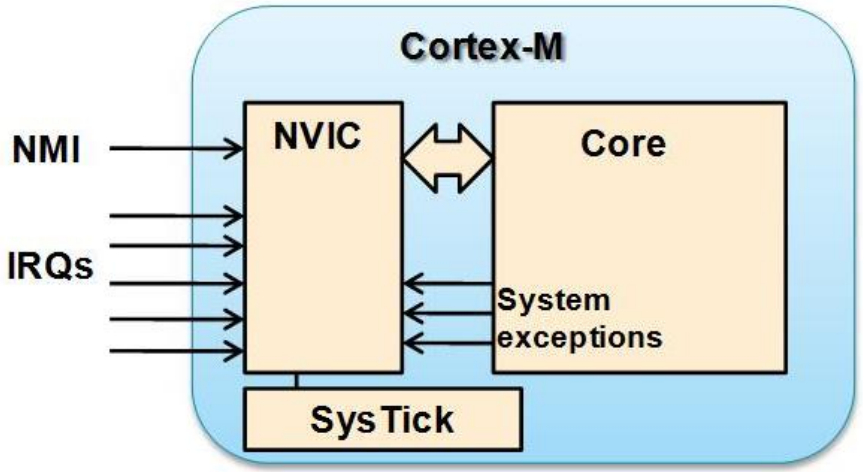
If interrupt is enabled, the SysTick Timer Interrupt Service Routine (ISR) is executed at fixed time intervals.

SysTick interrupt is routed through the NVIC.

(NVIC: Nested Vector Interrupt Controller prioritizes & handles all interrupts).

When ISR is triggered, the processor executes the **SysTick_Handler ISR**.

Note that the name of the ISR is predefined – its name should not be changed.




```
/**  SysTick ISR   **/
void SysTick_Handler (void)
{
  /* SysTick Handler code  */
}
```

System Timer (SysTick) Registers

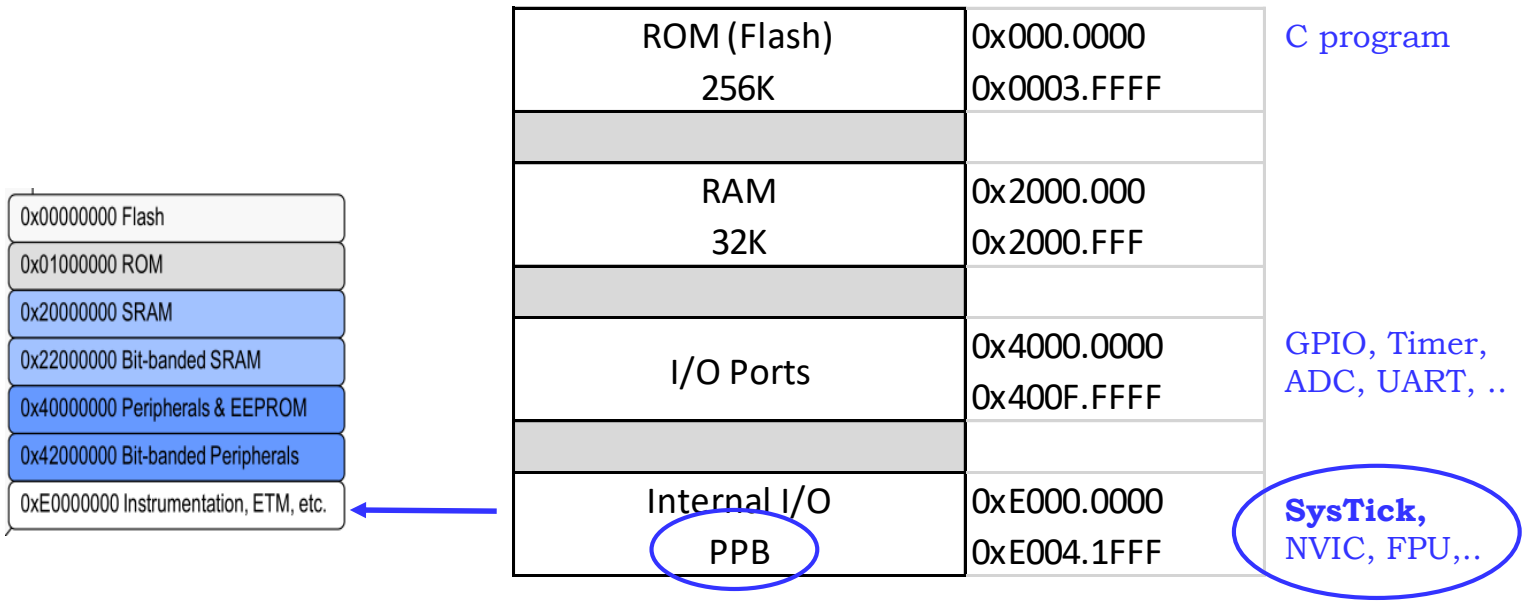
- There are 3 registers to program the SysTick Timer.
  - **Control Register** (STCTRL)
  - **Reload Register** (STRELOAD)
  - **Current Value Register** (STCURRENT)
- SysTick registers are 32-bit wide.
- Not all bits of the registers are used. E.g. Only 24 bits of the Reload & Current registers are used.

Offset	Name	Type	Reset	Description	See page
System Timer (SysTick) Registers					
0x010	STCTRL	RW	0x0000.0004	SysTick Control and Status Register	138
0x014	STRELOAD	RW	-	SysTick Reload Value Register	140
0x018	STCURRENT	RWC	-	SysTick Current Value Register	141

RW = Read/Write
RWC = Read/Write-Clear

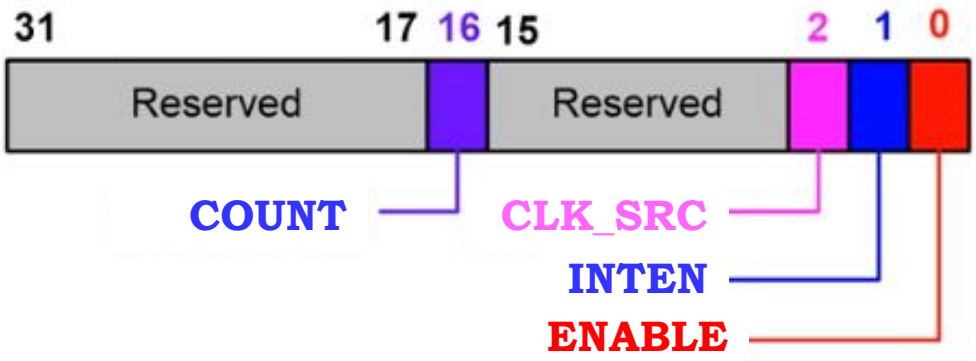
Source: Tiva TM4C123GH6PM Data Sheet (spmu376e.pdf, p134)

SysTick Address Map



- SysTick Registers starts from base address of **0xE000.E000**.
- It occupies part of the **Private Peripheral Bus** (PPB) address space that included the NVIC *(separate from the GPIO peripherals)*.

SysTick Control Register (STCTRL)



(RW register)

- **ENABLE:**
  - 0 (SysTick disabled);
  - 1 (SysTick enabled).
- **INTEN:**
  - 0 (SysTick interrupt disabled);
  - 1 (SysTick interrupt generated when count reached 0)
- **CLK_SRC:**
  - 0 (uses the internal oscillator/4);
  - 1 (uses system clock)
- **COUNT:**
  - 1 (when SysTick reached count of 0);
  - 0 (0 count not reached);
  - Bit is cleared by a WRITE to STCURRENT register.

Source: Tiva TM4C123GH6PM Data Sheet (spmu376e.pdf, p138)

SysTick Reload Register (STRELOAD)

(RW register)

- 24-bit SysTick RELOAD counter.
- SysTick counts down from RELOAD value.
- Value from 0x0000.0000 to 0x00FF.FFFF (0 to 16,777,215). → *Note: 2^{24} = 16,777,216 counts.*
- Writing a 0 to STRELOAD effectively disables the Timer.
- Time interval between 2 SysTick interrupts:

$$Interval = (RELOAD + 1) \times Clock\_Period$$

- ❑ If clock frequency is 10MHz, what is the interval between SysTick Interrupts if RELOAD is 100?

  Clock period = 1/10MHz = 100ns

  Interval = (100 + 1) ×100ns = 10.1ms

- ❑ If 100 clock periods are needed between SysTick Interrupts, what is the value of RELOAD?

  RELOAD = 100 – 1 = 99

Source: Tiva TM4C123GH6PM Data Sheet (spmu376e.pdf, p140)

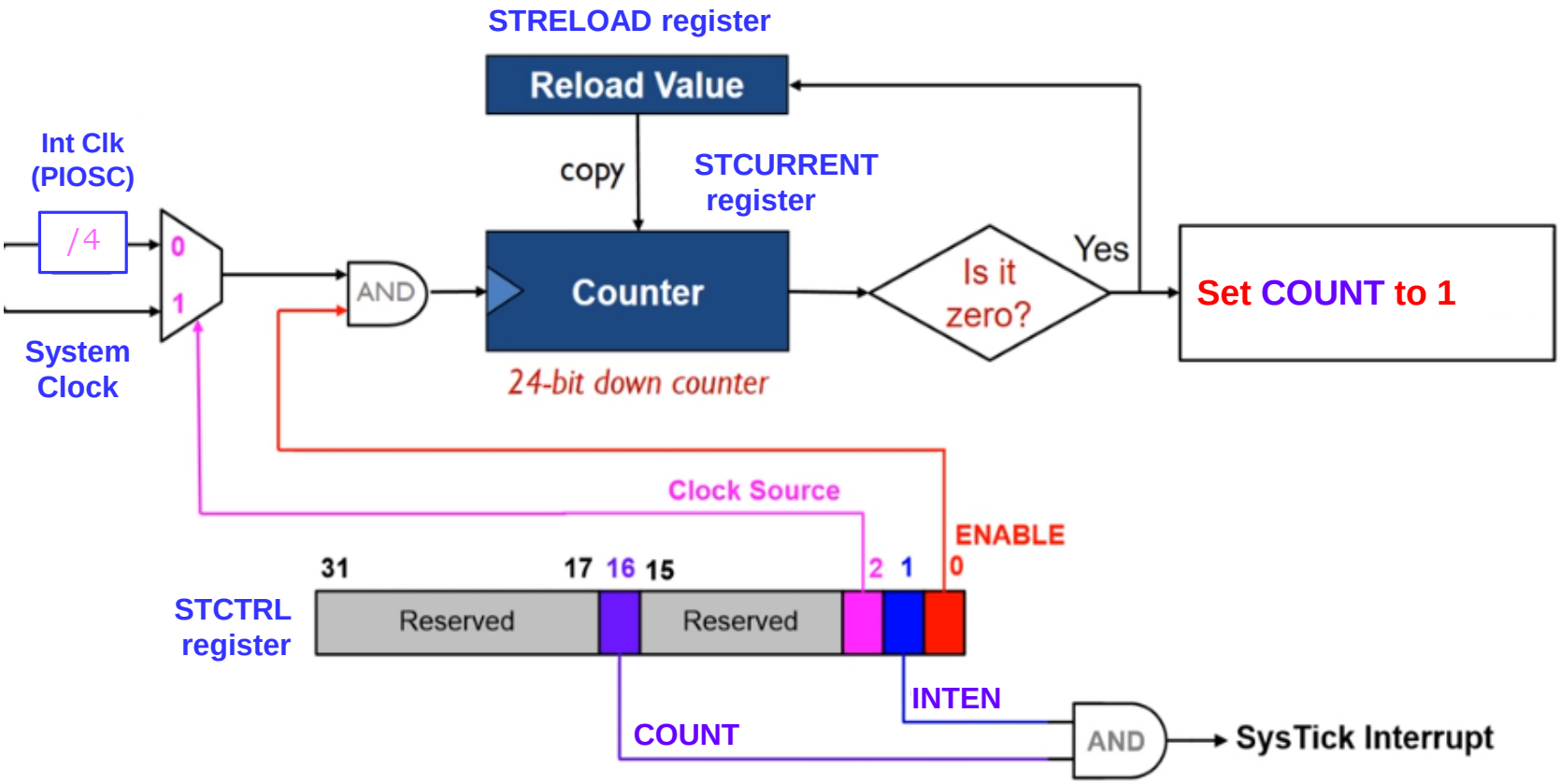
SysTick Current Registers (STCURRENT)

(RWC register)

- When register is read, register returns a 24-bit CURRENT value of SysTick counter at time of read.
- When CURRENT goes from 1 to 0, it generates a SysTick Interrupt (*if interrupt is enabled)* & counter is reloaded at the next clock cycle.
- As this is a Read, Write-Clear (RWC) register. Writing to this register:
  - clears the register & resets COUNT bit to '0' *(COUNT bit is in STCTRL register)*.
  - It causes the counter to reload on the next timer clock.
  - However it does not generate a SysTick interrupt.

Note: This register has a random value upon reset. Always clear it before enabling the timer. Source: Tiva TM4C123GH6PM Data Sheet (spmu376e.pdf, p141)

System Timer (SysTick) Registers



Ex 1: SysTick Reload Value

If system bus frequency = 80MHz, and the required SysTick Interval = 10ms.

What is the RELOAD value?

$$clock\_period = \frac{1}{80MHz} = 12.5ns$$

$$\text{RELOAD} = \frac{10ms}{clock\_period} - 1$$

$$= \frac{10ms}{12.5ns} - 1$$

$$= 800{,}000 - 1 = 799{,}999$$

Ex 2: SysTick Reload Value

If system bus frequency = 40MHz, and the required SysTick Interval = 0.5s.

What is the RELOAD value?

$$clock\_period = \frac{1}{40MHz} = 25ns$$

$$RELOAD = \frac{0.5s}{clock\_period} - 1 \quad = \frac{0.5s}{25ns} - 1$$

$$= 20{,}000{,}000 - 1 = 19{,}999{,}999$$

Remember:

- RELOAD is a 24-bit value.
- Maximum value is 0x00FF.FFFF (decimal 16,777,215)!
- Thus above value would not work.

Ex 3: SysTick Reload Value

If processor clock = 80MHz, ticks = 1,000, what is the SysTick interrupt interval? What is the RELOAD value?

$$RELOAD = ticks\ - 1 = 1000 - 1$$
$$Interval = (RELOAD + 1) \times clock\_period$$
$$= 1000 \times 12.5ns = 12.5us$$

If processor clock = 80MHz, ticks = 8,000, what is the SysTick interrupt interval? What is the RELOAD value?

$$RELOAD = ticks\ - 1 = 8000 - 1$$
$$Interval = (RELOAD + 1) \times clock\_period$$
$$= 8000 \times \frac{1}{80{,}000{,}000} = 0.1ms$$

SysTick Registers

(Check datasheet for more details)

STRELOAD, STCURRENT, STCOUNT, SYSPRI3

SysTick Control and Status (STCTRL)

SysTick Control and Status Register (STCTRL)

Base 0xE000.E000
Offset 0x010
Type RW, reset 0x0000.0004

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	reserved															COUNT
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	reserved													CLK_SRC	INTEN	ENABLE
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

- **ENABLE:**
  - 0 (SysTick disabled);
  - 1 (SysTick enabled).
- **INTEN:**
  - 0 (SysTick interrupt disabled);
  - 1 (SysTick interrupt generated when count reached 0)
- **CLK_SRC:**
  - 0 (uses the internal oscillator/4);
  - 1 (uses system clock)
- **COUNT flag:**
  - 1 (when SysTick reached count of 0);
  - 0 (0 count not reached);
  - Cleared by a Write to STCURRENT register.

Source: Tiva TM4C123GH6PM Data Sheet (spmu376e.pdf, p138)

SysTick Current Value Register (STCURRENT)

SysTick Current Value Register (STCURRENT)

Base 0xE000.E000
Offset 0x018
Type RWC, reset -

RWC: Read, Write-Clear

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	reserved								CURRENT							
Type	RO	RO	RO	RO	RO	RO	RO	RO	RWC	RWC	RWC	RWC	RWC	RWC	RWC	RWC
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CURRENT															
Type	RWC	RWC	RWC	RWC	RWC	RWC	RWC	RWC	RWC	RWC	RWC	RWC	RWC	RWC	RWC	RWC
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- **CURRENT:**
  - 24-bit SysTick counter current value.

Note: Writing to this register clears the register & COUNT bit (STCTRL register).

Source: Tiva TM4C123GH6PM Data Sheet (spmu376e.pdf, p141)

SysTick Reload Register (STRELOAD)

SysTick Reload Value Register (STRELOAD)

Base 0xE000.E000
Offset 0x014
Type RW, reset -

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	reserved								RELOAD							
Type	RO	RO	RO	RO	RO	RO	RO	RO	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RELOAD															
Type	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- **RELOAD:**
  - 24-bit SysTick counter start (reload) value.
  - Value from 0x0000.0001 to 0x00FF.FFFF.
  - Start value of 0 is possible but has no effect because the SysTick interrupt & COUNT bit are activated when counting from 1 to 0.

Note: To access this register correctly, system clock must be at 8MHz minimum.

Source: Tiva TM4C123GH6PM Data Sheet (spmu376e.pdf, p140)

SysTick System Handler Priority

System Handler Priority 3 (SYSPRI3)

Base 0xE000.E000
Offset 0xD20
Type RW, reset 0x0000.0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TICK			reserved					PENDSV			reserved				
Type	RW	RW	RW	RO	RO	RO	RO	RO	RW	RW	RW	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	reserved								DEBUG			reserved				
Type	RO	RO	RO	RO	RO	RO	RO	RO	RW	RW	RW	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit/Field	Name	Type	Reset	Description
31:29	TICK	RW	0x0	SysTick Exception Priority This field configures the priority level of the SysTick exception. Configurable priority values are in the range 0-7, with lower values having higher priority.

- SysTick system handler priority is set in bits 31-29.
- 8 levels of priority from 0 to 7; 0 being highest priority.
- Default priority is 0 (highest).

Source: Tiva TM4C123GH6PM Data Sheet (spmu376e.pdf, p172)

SysTick Setup

STRELOAD, STCURRENT, STCOUNT, SYSPRI3

SysTick Timer - Setup

Steps to setup the SysTick timer:

1. Clear ENABLE bit to turn off SysTick timer during initialization;
2. Set the **RELOAD** register.
3. Write to **STCURRENT** register to clear the value.
4. Set SysTick timer interrupt priority through **NVIC** register.
5. Specify clock sources to use through CLK_SRC bit in **STCTRL** register.
6. Enable SysTick interrupt through INTEN bit in the **STCTRL** register
7. Set ENABLE bit in the **STCTRL** register to start timer.

SysTick Timer Setup (NVIC.h)

```
/* define SysTick register addresses (file: NVIC.h  */
#define NVIC_ST_CTRL_R    (*((volatile uint32_t *)0xE000E010))
#define NVIC_ST_RELOAD_R  (*((volatile uint32_t *)0xE000E014))
#define NVIC_ST_CURRENT_R (*((volatile uint32_t *)0xE000E018))
#define NVIC_SYS_PRI3_R   (*((volatile uint32_t *)0xE000E40C))

/* defines for the bit fields in the NVIC_ST_CTRL register. */
#define NVIC_ST_CTRL_COUNT      0x00010000  // Count Flag
#define NVIC_ST_CTRL_CLK_SRC    0x00000004  // Clock Source
#define NVIC_ST_CTRL_INTEN      0x00000002  // Interrupt Enable
#define NVIC_ST_CTRL_ENABLE     0x00000001  // Enable

/* defines for the bit fields in the NVIC_ST_RELOAD register.  */
#define NVIC_ST_RELOAD_M        0x00FFFFFF  // Reload Value
#define NVIC_ST_RELOAD_S        0

/* defines for the bit fields in the NVIC_ST_CURRENT register.  */
#define NVIC_ST_CURRENT_M       0x00FFFFFF  // Current Value
#define NVIC_ST_CURRENT_S       0
```

SysTick Timer Setup (*Example Code*)

```
/* ticks = no of SysTick ticks between SysTick interrupts        */
void SysTick_Init(uint32_t ticks)
{
   /* 1. disable SysTick during setup                            */
   NVIC_ST_CTRL_R &= ~NVIC_ST_CTRL_ENABLE;
   NVIC_ST_RELOAD_R = ticks-1;  /* 2. reload value               */
   NVIC_ST_CURRENT_R = 0;       /* 3. any write to reg clears it */

   /* 4. set systick handler priority to 2                       */
   NVIC_SYS_PRI3_R = (NVIC_SYS_PRI3_R & NVIC_SYS_PRI3_TICK_M)|
                     (0x02 << NVIC_SYS_PRI3_TICK_S);
   /* 5,6,7: enable SysTick with core clock & interrupts         */
   NVIC_ST_CTRL_R = NVIC_ST_CTRL_INTEN |
                    NVIC_ST_CTRL_ENABLE |
                    NVIC_ST_CTRL_CLK_SRC;
 }
```

SysTick Timer - Limitations

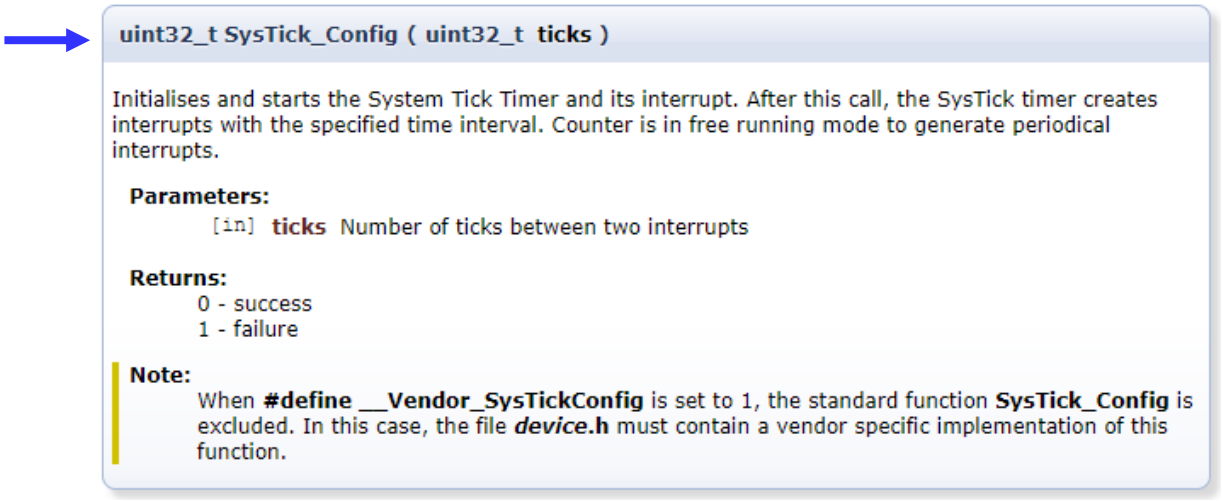
- System count values changes upon every clock cycle.
- Since timer count is a 24-bit number, the maximum count value is 0x00FF.FFFF (decimal 16,777,215).

- Example:
  - At 16MHz bus speed, clock period = (1/16MHz) = 62.5 ns. SysTick counter counts down every 62.5ns.
  - Maximum time delay before SysTick reloads = (62.5ns × 16,777,215) = ~1.05s.

- Example:
  - At the maximum 80MHz bus speed (for the Tiva LaunchPad), clock period = (1/80MHz) = 12.5 ns.
  - SysTick counter counts down every 12.5ns.
  - Maximum time delay before SysTick reloads = (12.5ns × 16,777,215) = ~210ms.

SysTick Timer Setup (core_cm4.h)

`SysTick_Config()` is a **CMSIS** function defined in file 'core_cm4.h'.
(You use this function in Labs/Assignments – after Lab 1).



uint32_t SysTick_Config (uint32_t ticks)

Initialises and starts the System Tick Timer and its interrupt. After this call, the SysTick timer creates interrupts with the specified time interval. Counter is in free running mode to generate periodical interrupts.

Parameters:

[in] **ticks** Number of ticks between two interrupts

Returns:

0 - success
1 - failure

Note:

When **#define __Vendor_SysTickConfig** is set to 1, the standard function **SysTick_Config** is excluded. In this case, the file ***device*.h** must contain a vendor specific implementation of this function.

CMSIS: Cortex Microcontroller Software Interface Standard is a vendor-independent hardware abstraction layer for the Cortex-M processor series.

SysTick Timer Setup (core_cm4.h)

```
__STATIC_INLINE uint32_t SysTick_Config (uint32_t ticks)
{
   if ((ticks - 1UL) > SysTick_LOAD_RELOAD_Msk)
   {
      return (1UL);  /* Reload value impossible */
   }
   SysTick->LOAD  = (uint32_t)(ticks - 1UL); /* set reload register */
   /* set Priority for Systick Interrupt */
   NVIC_SetPriority (SysTick_IRQn, (1UL << __NVIC_PRIO_BITS) - 1UL);
   SysTick->VAL   = 0UL;          /* Load the SysTick Counter Value */
   /* Enable SysTick IRQ and SysTick Timer */
   SysTick->CTRL  = SysTick_CTRL_CLKSOURCE_Msk |
                    SysTick_CTRL_TICKINT_Msk   |
                    SysTick_CTRL_ENABLE_Msk;
   return (0UL);   /* Function successful */
}
```

Review Questions *(Week 1)*

General:

1. What is an embedded system? Give a few examples of it.
2. List 2 differences in characteristics of microcontrollers versus microprocessors.
3. Differentiate between the Havard & Von-Neumann architectures of memory system. What architecture does the Cortex-M4 uses?
4. What is the name of the ARM-based microcontroller used on the Tiva LaunchPad TMC123G?
5. Explain the operating states & operating modes of the ARM Cortex-M4 CPU.

Tiva LaunchPad TMC123G:

6. What is the maximum CPU clock speed programmable on the Tiva LaunchPad?
7. What is the maximum program size possible on the Tiva LaunchPad?
8. Do the LaunchPad connects (J1 to J4) provide +5V and +3.3V outputs? Specify which ones.
9. How many GPIO ports do the LaunchPad has? List them.
10. Which GPIO pins are SW1 and SW2 connected to?
11. Which GPIO pins is the color LED connected to?
12. Is PC[3:0] available on the LaunchPad connectors (J1 to J4)? What are they used for?
13. Which GPIO pins can be used for analog inputs (Ain)? List all of them.

Review Questions *(Week 1)*

14. What is the Alternate Function for GPIO port pin PA0?
15. Explain how the GPIO Port Control (GPIOCTL) register bits are used to select an alternate GPIO function.
16. What is the difference between the APB and the AHB bus?
17. How do we select a GPIO port to be connected to the APB or the AHB bus?

SysTick Timer:

18. Name the 3 SysTick registers and briefly describe their functions.
19. List the base addresses of the 3 SysTick registers?
20. What is the maximum count for the SysTick timer? How does this value come about?
21. If the CURRENT SysTick count is 0x00FF.1234. What is the next count value?
22. If the CURRENT SysTick count is 0x0100.1000. What is the next count value?
23. What happens when we write to the STCURRENT register?
24. If 300 SysTicks are needed between SysTick interrupts, what is the value of RELOAD to use?
25. If system bus frequency is 50 MHz and we require a SysTick interrupt interval of 1ms, what is the RELOAD value to use?