

ECE 271 Microcomputer Architecture and Applications**Lab 5: System Timer (SysTick)****Instructor: Prof. Yifeng Zhu****Spring 2015****Goals**

1. Understand the clock tree of STM32L processor
2. Understand the basic concept of system timer
3. Gain knowledge of using SysTick to create a time delay function
4. Understand the basic procedure of interrupt handling
5. Understand the auto stacking and un-stacking of interrupt handling process

Pre-Lab Assignment

1. Read Chapter 12.4 System Timer
2. Complete the pre-assignment

In-Lab Demo

1. Change the processor default clock from MSI 2.097 MHz to MSI 4.194 MHz.
2. Use SysTick to generate an interrupt every 1ms. Implement a delay function to toggle a LED with a time interval of one second by calling *delay* (1000). Use an oscilloscope to measure the period of the LED pin signal. Calculate the accuracy of your MSI clock.
3. Something cool. Note that controlling another LED is not accounted as something cool. The following gives a few examples.
 - a. Use the delay function to accurately control the rotation speed of a stepper motor. If you make your motor runs 60 rounds per minute, your motor is a second clock.
 - b. Display a second counter on the LCD.
 - c. Calibrate the MSI or HSI clock if your clock has an error larger than 0.5%.
 - d. Use HSE, instead of HSI or MSI, to drive the system timer
 - e. Use the delay function to send out a Morse code of SOS (··· - - - ···) via the LED light.

Post-Lab Assignment

1. Complete the post-lab assignment report
2. Write your answers in the file *Readme.md* and submit to the gitlab server

Processor Clock

Four different clock sources can be used to drive the system clock (SYSCLK):

1. 16 MHz HSI (high-speed internal) oscillator clock
2. 1 to 24 MHz HSE (high-speed external) oscillator clock
3. 2 to 24 MHz PLL (phase-locked loop) clock that is clocked by HSI or HSE
4. 65.5 kHz to 4.2MHz MSI (multispeed internal) oscillator clock

The MSI is selected as the system clock source after startup from reset, wake-up from the stop mode or the standby low-power mode. The maximum frequency of the AHB, APB1 and the APB2 domains is 32 MHz. All the peripheral clocks are derived from the system clock (SYSCLK) except USB, SDIO, RTC, and IWDG.

The MSI clock signal is generated from an internal RC oscillator. Seven frequency ranges are available:

- 65.536 kHz,
- 131.072 kHz,
- 262.144 kHz,
- 524.288 kHz,
- 1.048 MHz,
- **2.097 MHz** (default value) and
- 4.194 MHz.

The MSIRDY flag in the RCC_CR register indicates whether the MSI RC oscillator (a combination of resistors and capacitors) is stable or not. At startup, the signal output of the MSI RC oscillator is not released until this bit is set by hardware. The MSI RC oscillator can be turned on or off by using the MSION bit in the RCC_CR register.

The SysTick timer clock is not stopped during the stop mode debug (DBG_STOP bit set). The counter keeps on being decremented and can generate interrupts if they are enabled.

The CLKSOURCE of the SYS_CTRL register selects the clock source for SysTick.

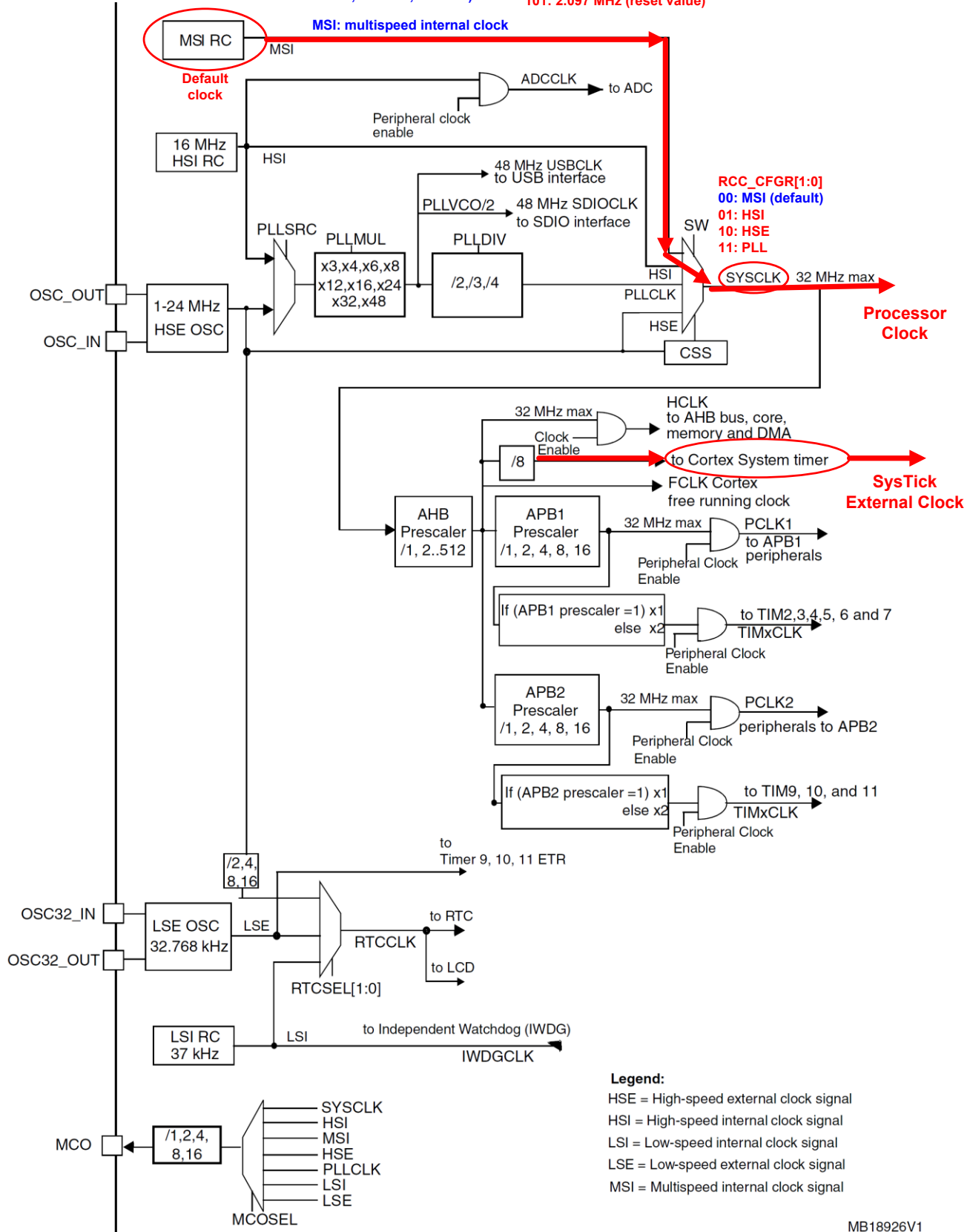
- If CLKSOURCE = 0, the external clock is used. The frequency of SysTick clock is the frequency of the AHB clock divided by 8.
- If CLKSOURCE = 1, the processor clock is used. The default processor clock is MSI **2.097 MHz**.

Reset and Clock Control (RCC)

7 frequencies (65.5 kHz, 131 kHz, 262 kHz, 524 kHz, 1.05 MHz, 2.1 MHz, 4.2 MHz)

MSIRANGE[2:0] bits in the RCC_ICSCSR
101: 2.097 MHz (reset value)

MSI: multispeed internal clock



MB18926V1

ECE 271 Microcomputer Architecture and Applications
Lab 5: System Timer (SysTick)
Pre-Lab Report

Student Name: _____

In this lab, you generate a system ticker with a time period of 1ms second. When you call the *delay(1000)*, you will get a delay of one second.

The CLKSOURCE of the SYS_CTRL register indicates the clock source for SysTick.

- If CLKSOURCE = 0, the external clock is used. The frequency of SysTick clock is the frequency of the AHB clock divided by 8.
- If CLKSOURCE = 1, then processor clock is used. The default clock to the processor is a MSI of **2.097 MHz**.

In this lab, you are required to configure the AHB clock frequency as 4.194 MHz. In order to change the clock frequency, you need to

1. Select the internal clock source. This is achieved by setting MSIRANGE bits of RCC_ICSCR (Internal clock sources calibration register). Check the “STM32L Reference Manual” and find out what value should MSIRANGE bits should be.

MSIRANGE = _____

2. Set **MSION** (MSI Clock Enable) and wait for **MSIRDY** (MSI Clock Ready Flag) of RCC_CR (Clock Control Register).

- If CLKSOURCE = 0, what is the value of the SysTick Reload Value Register?

SysTick_LOAD = _____

- If CLKSOURCE = 1, what is the value of the SysTick Reload Value Register?

SysTick_LOAD = _____

ECE 271 Microcomputer Architecture and Applications
Lab 5: System Timer (SysTick)
Lab Demo

Student Name: _____

In this lab, you are required to use MSI 4.194 as the processor clock.

The frequency of internal clocks (RC oscillators) may vary from one chip to another due to the manufacturing process variations. In addition, the operating temperature has an impact on the accuracy of the RC oscillators. At 25 °C, the HSI and MSI oscillators have an accuracy of $\pm 1\%$ typically, but in the temperature range of -40 to 105 °C, the accuracy decreases.

Write a program to toggle a GPIO pin (specifically PB.4). Use an oscilloscope to measure the signal period of the PB.4 pin.

```
while(1){  
    delay(1000);      % delay 1 second  
    toggle_blue_led();  
}
```

1. What is the time delay you measure on the oscilloscope in room temperature? What is the percentage of error?
2. What is the time delay you measure on the oscilloscope in outdoor temperature? What is the percentage of error?

ECE 271 Microcomputer Architecture and Applications
Lab 5: System Timer (SysTick)
Post-Lab Assignment

Student Name: _____

Write your answer to the following questions in Readme.md file and submit it to the gitlab server.

1. What is the time delay you have measured on the oscilloscope at indoor and outdoor temperature? What is their percentage of error?
2. Is it possible to use the SysTick timer to generate an interrupt once every 100 seconds if there is only a clock 2.097 MHz? If not, what solutions could we have? If yes, how to set up the SysTick_LOAD register?