

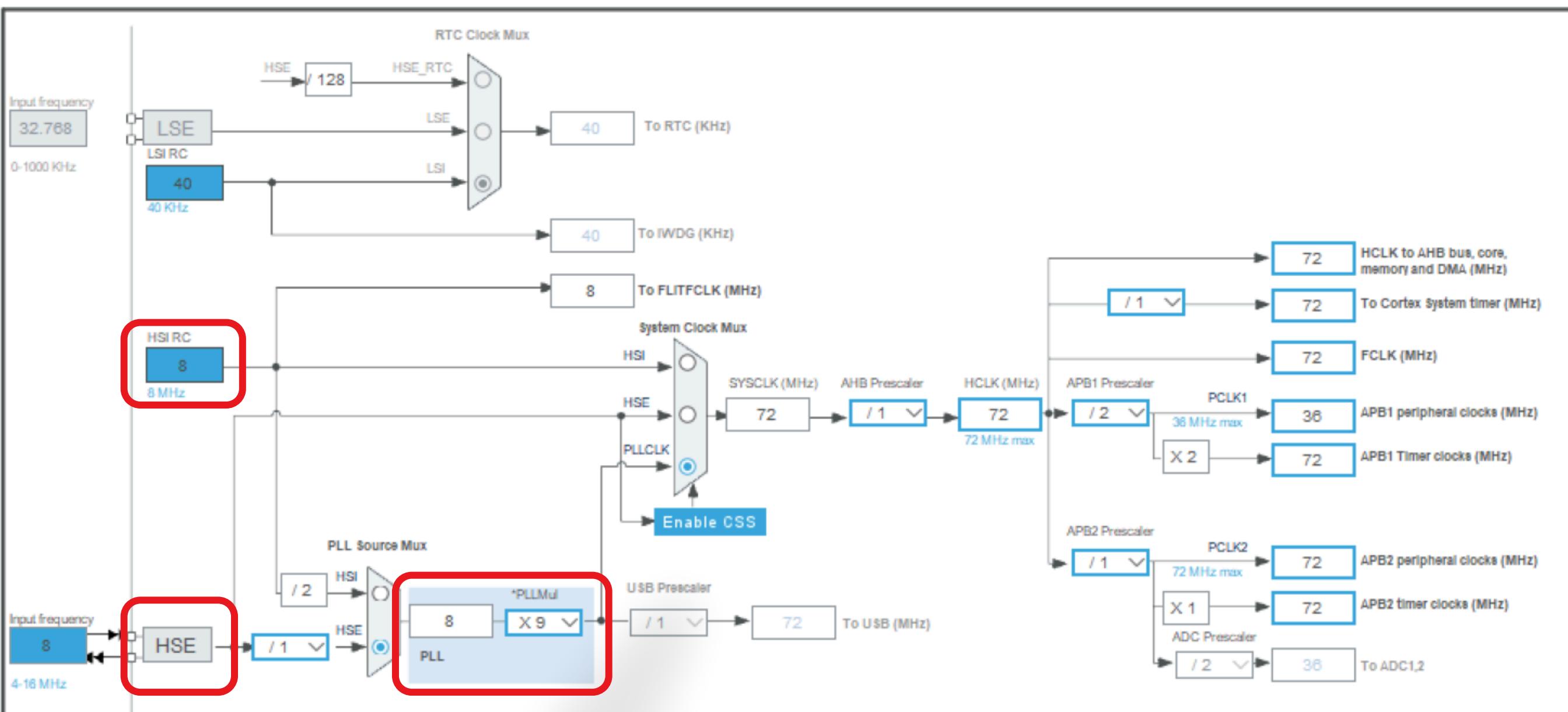
# LẬP TRÌNH ARM - STM32

STM32 - TIMER

Date: Mar 09, 2025



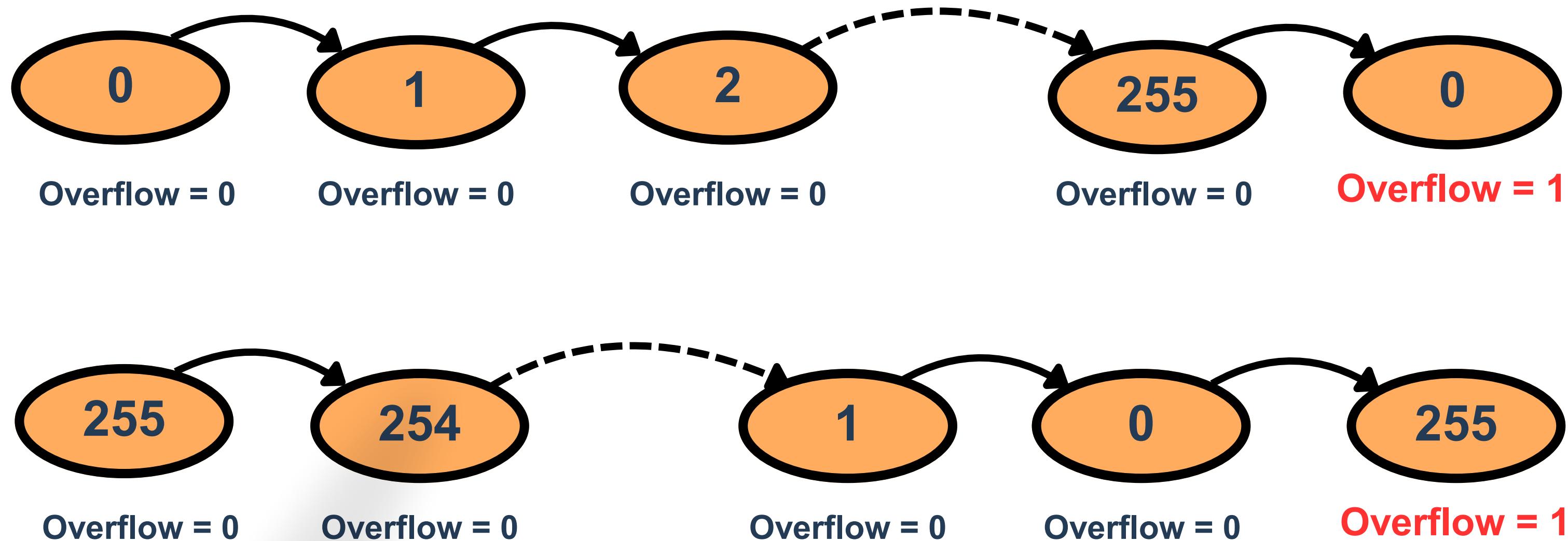
# System Clock



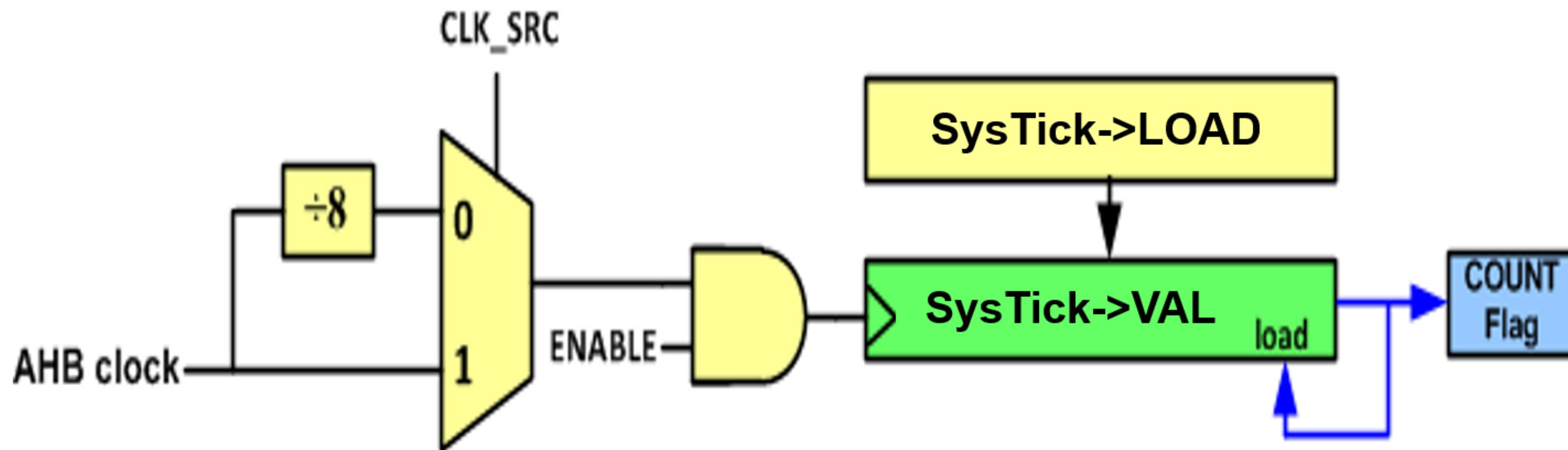
Nguồn clock hệ thống (**SYSCLK**)

- **HSI** (High Speed Internal – thường là 8 MHz hoặc 16 MHz)
- **HSE** (High Speed External – thường là 8 MHz hoặc dùng thạch anh ngoài)
- **PLL** (Phase Locked Loop – có thể nhân lên từ HSI/HSE)

# 8-bit counter Stages

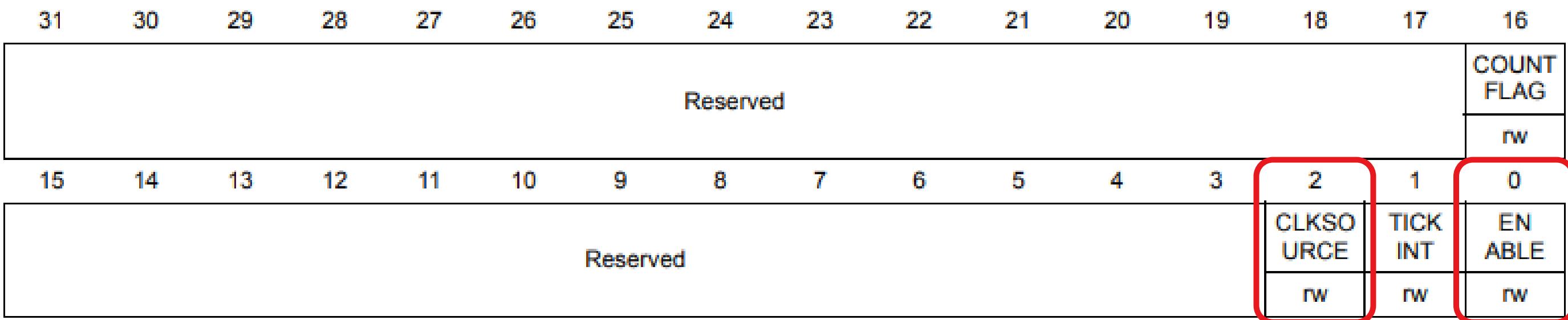


# System Tick Timer



- Available in all Cortex-M MCUs
- It is a 24-bit down counter. It counts down from an initial value to 0
- Used to initiate an action on a periodic basis

# STM32F1 - SysTick Registers



STK\_CTRL

## Bit 0 ENABLE: Counter enable

- 0: Counter disabled
- 1: Counter enabled

## Bit 16 COUNTFLAG:

Returns 1 if timer counted to 0 since last time this was read.

## Bit 2 CLKSOURCE: Clock source selection

- 0: AHB/8
- 1: Processor clock (AHB)

# STM32F1 - SysTick Registers

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								RELOAD[23:16]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RELOAD[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

STK\_LOAD

► Bits 23:0 RELOAD: RELOAD value

# STM32F1 - SysTick Registers

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								CURRENT[23:16]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CURRENT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

**STK\_VAL**

## ► Bits 23:0 CURRENT: Current counter value

The VAL register contains the current value of the SysTick counter. Reads return the current value of the SysTick counter. A write of any value clears the field to 0, and also clears the COUNTFLAG bit in the STK\_CTRL register to 0.

# STM32F1 - Timer

## ► Họ STM32F10x có 3 loại Timer:

- Các Timer điều khiển nâng cao (Advanced control timers – TIM1&TIM8)
- Các Timer cơ bản (Basic timers – TIM6&TIM7)
- Các Timer sử dụng cho mục đích thông thường(General purpose timers –timer còn lại)

## ► Một số đặc điểm của Timer:

- Bộ đếm 16 bit, có thể đếm tiến, lùi, tiến/lùi, tự động nạp lại.
- Hỗ trợ bộ chia tần 16 bit, có thể lập trình được.
- Hỗ trợ 4 kênh hoạt động độc lập cho các chức năng:
  - Input Capture (Capture đầu vào)
  - Output Compare (So sánh đầu ra)
  - PWM (Điều chế độ rộng xung)
  - One-pulse mode output (Ngõ ra của chế độ 1 xung)
- Hỗ trợ kích hoạt các sự kiện DMA.
- Hỗ trợ kích hoạt ngắn.
- Hỗ trợ trigger cho các sự kiện.

# STM32F1 - Timer Library

```
TIM_TimeBaseInitTypeDef A;  
(Khai báo biến A thuộc kiểu  
TIM_TimeBaseInitTypeDef)
```

```
A.TIM_Period = B;  
(Lệnh chọn chu kỳ đếm cho timer - giá  
trị tự động nạp lại)  
B: Giá trị cài đặt giới hạn [0; 65535]
```

```
A.TIM_Prescaler = C;  
(Lệnh chọn bộ chia(xung) trước khi vào bộ đếm)  
C: Giá trị cài đặt giới hạn [0; 65535]
```

```
TIM_Cmd (A, B);  
(Lệnh cho phép hoặc cấm timer)
```

<b>A: TIMx</b>	<b>A:</b>
TIM1	Sử dụng bộ TIMER1
TIM2	Sử dụng bộ TIMER2
...	...
<b>B:</b>	<b>B:</b>
ENABLE	Cho phép
DISABLE	Không cho phép

# STM32F1 - Timer Library

```
TIM_TimeBaseInitTypeDef A;  
(Khai báo biến A thuộc kiểu  
TIM_TimeBaseInitTypeDef)
```

```
uint16_t B;  
B = TIM_GetCounter(A);  
(Lệnh đọc giá trị đếm của timer)
```

Công thức chia tần

$$\frac{\text{SysClock} * T}{PSC} = PD$$

PSC: TIM\_Prescaler  
PD : TIM\_Period

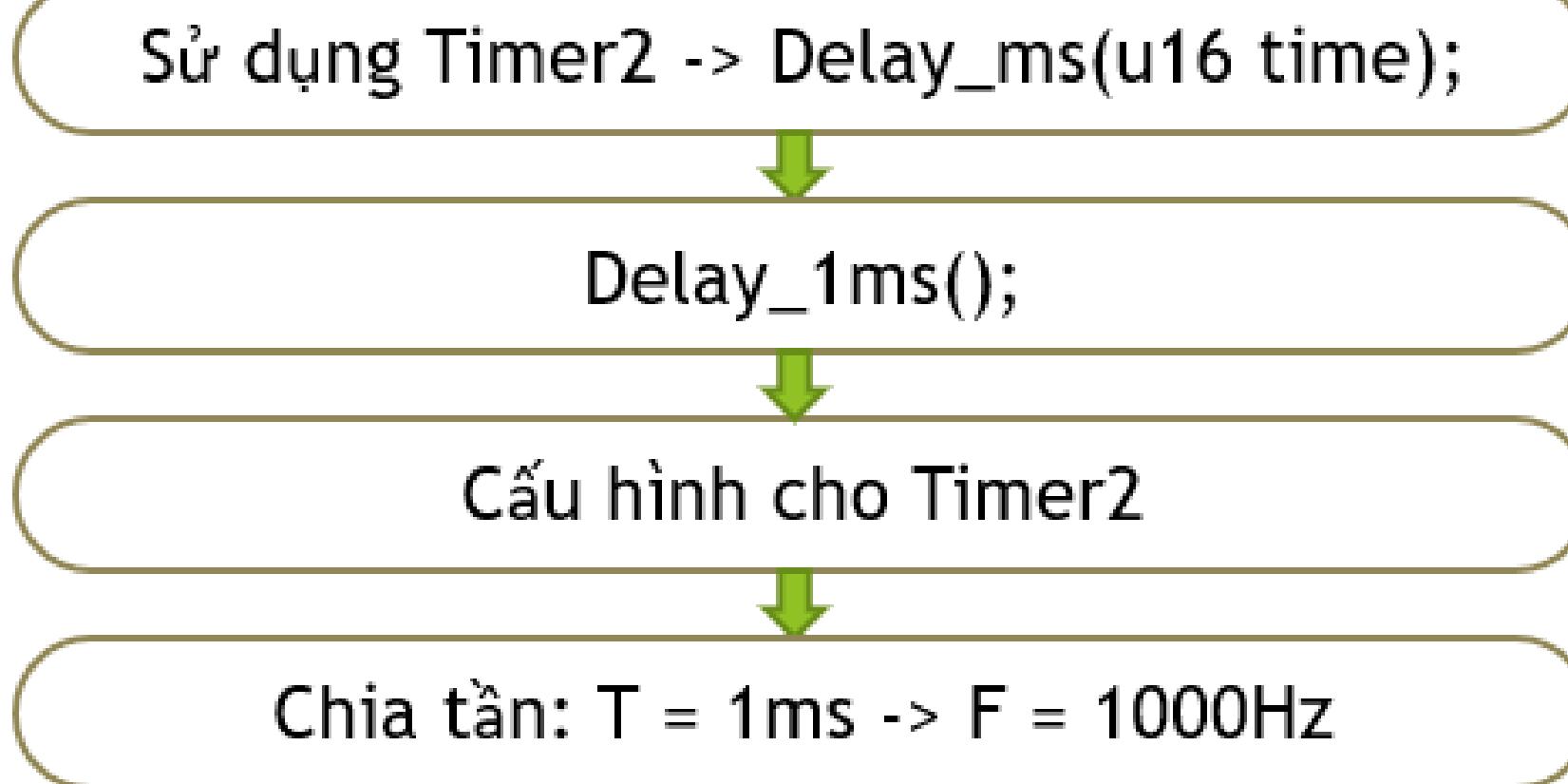
```
TIM_SetCounter(A, B);  
(Lệnh cài đặt giá trị đếm cho timer)  
A: Timer cần điều khiển  
      TIM1, TIM2, ...  
B: Giá trị cài đặt  
      [0, 65535]
```

# STM32F1 - Timer

**Bài tập áp dụng:** Sử dụng bộ Timer2, viết chương trình tạo hàm Delay ms.  
Sau đó thực hiện blink led PC13 với T=1s.

## Công thức chia tần

### ❖ Phân tích bài toán:



$$\frac{\text{SysClock} * T}{PSC} = PD$$

PSC: TIM\_Prescaler

PD : TIM\_Period

T = 0.001s, chọn giá trị PSC = 2

$$\frac{72 * 10^6 * 0.001}{2} = 36000$$

**TIM\_Prescaler = 2 – 1**

**TIM\_Period = 36000 – 1**