بسم الله الرحمن الرحيم

* SysTick Timer:

SysTickis 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.

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

* NVIC_ST [NVIC --> nested vector interrupt control & ST --> sys tick]

① CTRL_R [control] has bit(@ 0) for enable or disable timer itself & bit(@ 1) for interrupt

Bit	Name	Description
0	Enable	Enable 0: the counter is disabled 1: enables SysTick to begin counting Down
1	INTEN	Interrupt Enable 0: Interrupt generation is disabled 1: when SysTick counts to 0 an interrupt is generated
2	CLK_SRC	Clock Source 0: Precision internal oscillator (PIOSC) divided by 4 [مش مهم دلوقتي] 0: no clock 1: System clock
16	COUNT	Count Flag 0: the SysTick has not counted Down to zero since the last time this bit was read 1: the SysTick has counted Down to Zero **Note**: this flag is cleared by reading the STRCTL or writing to STCURRENT Register

② RELOAD_R --> has the initial value that counter will decrease from it and it has 24-bits so, the max value is 2^24 which in hexa = 0x00FFFFFF

③ CURRENT_R has the current value of the counter after decreament through certain time

---

There are four steps to initialize the SysTick timer.

1. Clear the ENABLE (NVIC_ST_CTRL_R) bit to turn off SysTick during initialization.
2. Set the RELOAD (NVIC_ST_RELOAD_R) register.
3. Write to the NVIC_ST_CURRENT_R value to clear the counter.
4. Write the desired mode to the control register, NVIC_ST_CTRL_R.
   - Set the CLK_SRC bit specifying the core clock will be used.
   - We must set CLK_SRC=1 bit specifying the core clock will be used, because CLK_SRC=0 external clock mode is not implemented on the LM3S/TM4C family.
   - Set INTEN (NVIC_ST_CTRL_R) to enable interrupts, but in this first example we clear INTEN so interrupts will not be requested.
   - Set the ENABLE (NVIC_ST_CTRL_R) bit so the counter will run.

When the CURRENT (NVIC_ST_CURRENT_R) 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.

The COUNT flag could be configured to trigger an interrupt. However, in this first example interrupts will not be generated.

SysTick Timer

- Timer/Counter operation
  - 24-bit counter *decrements* at bus clock frequency
    - With 80 MHz bus clock, decrements every 12.5 ns
  - Counting is from $n \rightarrow 0$

→ has 24-bits 0x00FFFFFF → 2^{24} [max initial value]

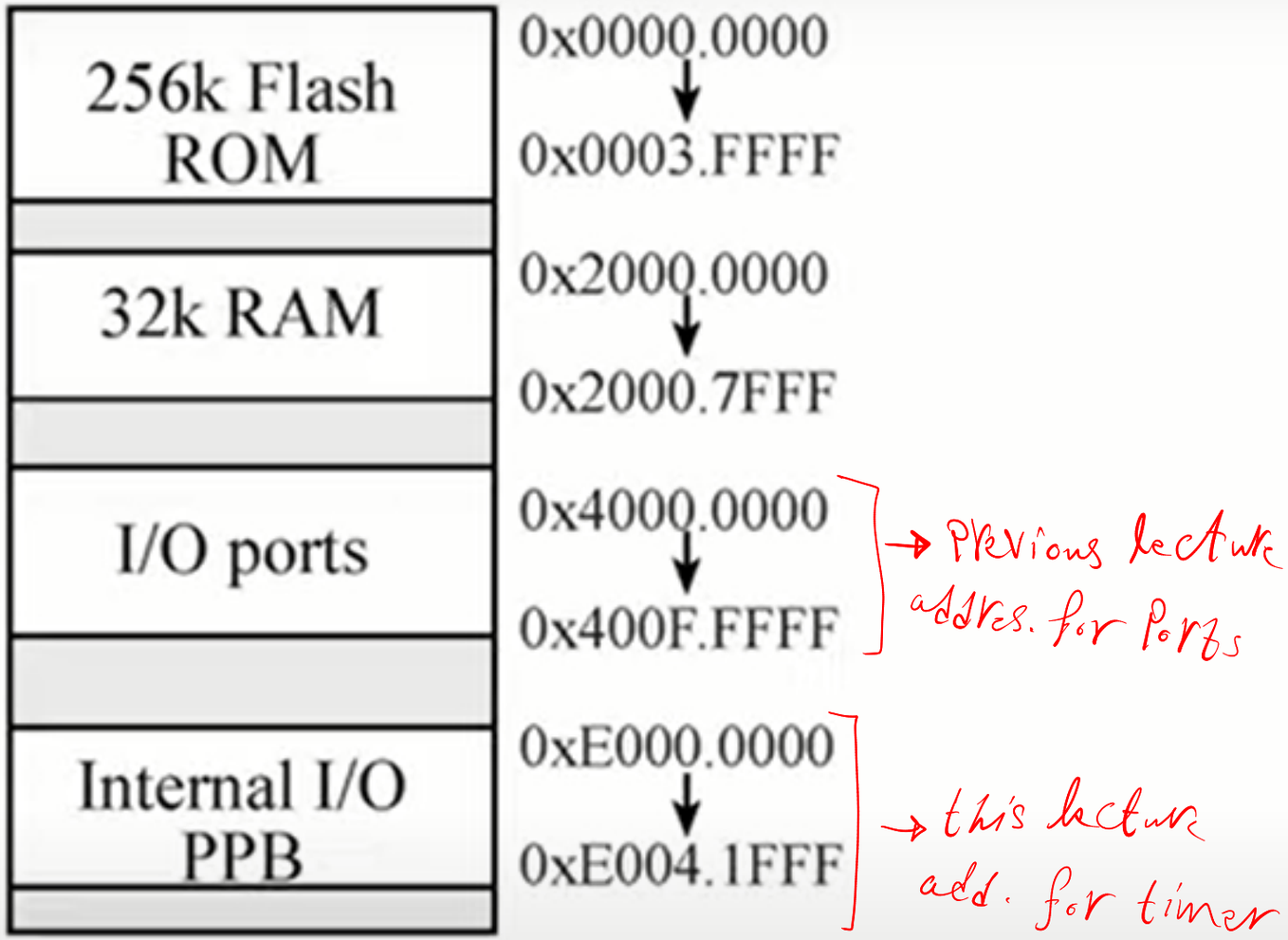
↳ Counter → decrement value of reload timer every clock cycle

Processor frequency

1 → clock cycle means 1 Period time = $\frac{1}{f} = \frac{1}{80\,MHz} = 12.5\,ns$

[0xFFFFFF]

ARM Memory-map



TI TM4C123
Microcontroller

SysTick Timer

Address	31-24	23-17	16	15-3	2	1	0	Name
SE000E010	0	0	COUNT	0	CLK_SRC	INTEN	ENABLE	NVIC_ST_CTRL_R
SE000E014	0	24-bit RELOAD value						NVIC_ST_RELOAD_R
SE000E018	0	24-bit CURRENT value of SysTick counter						NVIC_ST_CURRENT_R

- Initialization (4 steps)
  - ① – Step1: Clear ENABLE to stop counter
  - – Step2: Specify the RELOAD value
  - – Step3: Clear the counter via NVIC_ST_CURRENT_R
  - – Step4: Set NVIC_ST_CTRL_R
    - CLK_SRC = 1 (bus clock is the only option)
    - INTEN = 0 for no interrupts
    - ENABLE = 1 to enable

SysTick Timer Registers

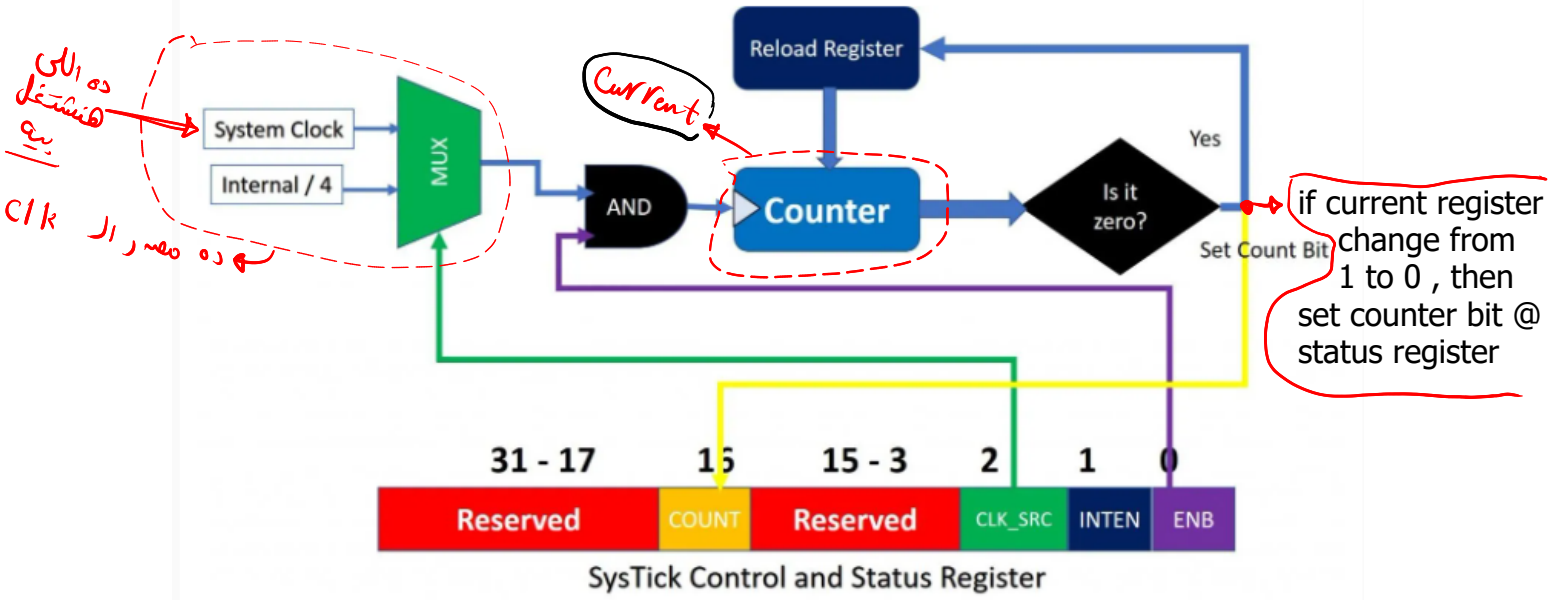
address is constant but content can be modified

```
#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))
```

* دلوقتي متحاول اللغيطة أنت عندك حاجه إسمها Counter Flag ← ده عبارة عن إشارة من الأخر
إنما ال Counter Value ← هو اللي بتبقى متخزنه في Current Register ← ده اللي أنا بقعد أنقص منه لحد أول ما يوصل لصفر ساعتها ال Counter Flag يحصله Setting ويبقى بواحد .

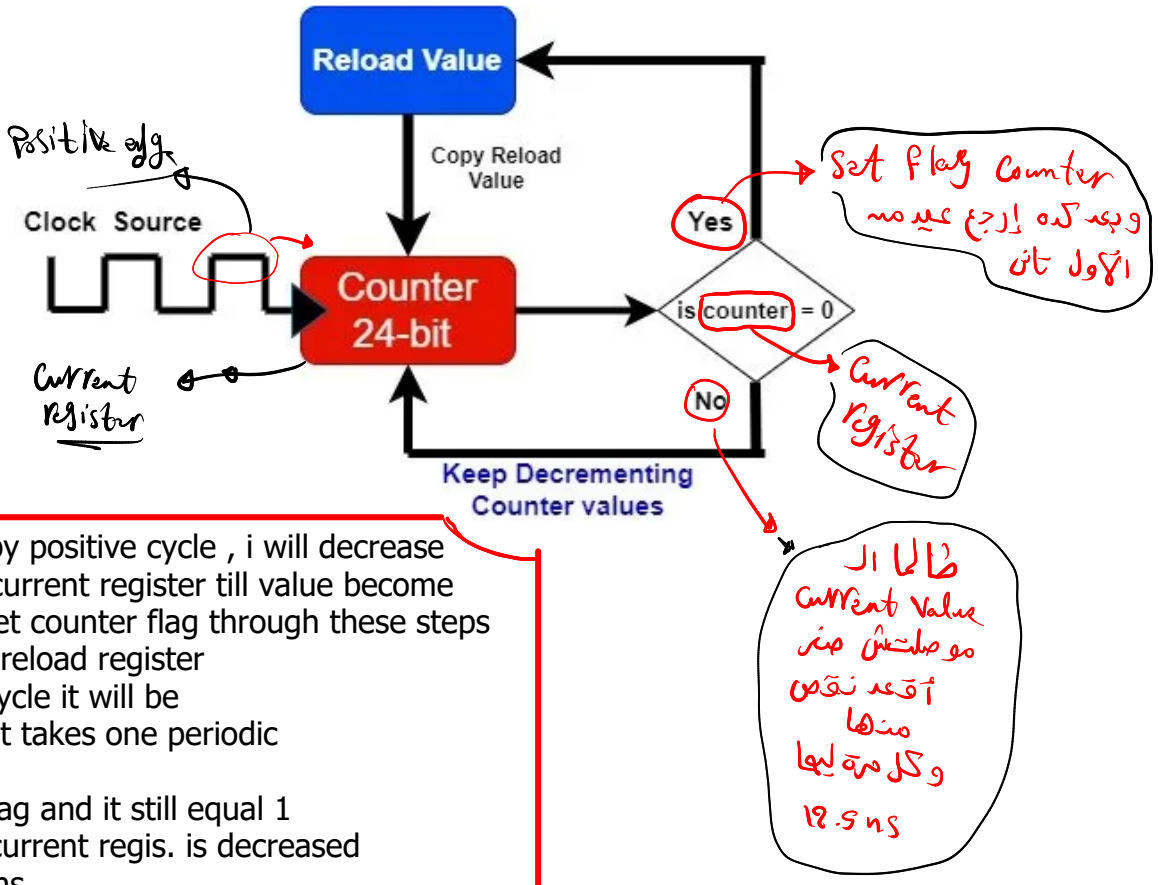
Imp. Notes

This is a counter flag or status bit for the counter. This bit becomes 1 when counter decrements from 0 to one.



---

* الرسمة دى فيها الحلاوة



counter affected only by positive cycle , i will decrease each clock cycle from current register till value become zero in tjis case i will set counter flag through these steps

1) give 0x00FFFFFF to reload register
2) through first clock cycle it will be decreased by 1 and it takes one periodic time (12.5 ns)
3) check the counter flag and it still equal 1 so, for next cycle , current regis. is decreased by 1 and take 12.5 ns
4) repeat this till counter = 0 , so it will repeat from the first value of reload register again
5) and for delay calc. simply 2^24 * 12.5 ns which 2^24 my initial value i will decrease from it through above diagram

SysTick Timer Example

initialization

```
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
}
```

max Value for 24-bit Reload

... 0	Clksrc	INTEN	enable
0	1	0	1

```
// The delay parameter is in units of the 80 MHz core clock(12.5 ns)
void SysTick_Wait(uint32_t delay){
  NVIC_ST_RELOAD_R = delay-1;  // number of counts
  NVIC_ST_CURRENT_R = 0;       // any value written to CURRENT clears
  while((NVIC_ST_CTRL_R&0x00010000)==0){ // wait for flag
  }
}
// Call this routine to wait for delay*10ms
void SysTick_Wait10ms(uint32_t delay){
 unsigned long i;
 for(i=0; i<delay; i++){
   SysTick_Wait(800000); // wait 10ms
 }
}
```

ارجع للdiagram اللى فوق

ده الbit رقم 16 ← هعمله AND مع الCount

& 1

Counter 0 → 0, 1 → 1

أنا اللى بيعت الرقم ده لما بنادى الfunc

ده لما بيروح بيبقى بدل ال(0x00FFFFFF) و بالتالى ال delay هيساوى 800000 * 12.5ns = 10 ms

once calling this func. it gets delay about 10ms , if i want delay to be 1 sec ,i should call this func about 100 times , in this ex. we used for loop and send parameter delay = 100 when we call this last func. by user

SysTick Timer

ده نفس كود السي اللي فوق

24-bit Countdown Timer

```
SysTick_Init
; disable SysTick during setup
    LDR R1, =NVIC_ST_CTRL_R
    MOV R0, #0                  ; Clear Enable
    STR R0, [R1]
; set reload to maximum reload value
    LDR R1, =NVIC_ST_RELOAD_R
    LDR R0, =0x00FFFFFF;        ; Specify RELOAD value
    STR R0, [R1]                ; reload at maximum
; writing any value to CURRENT clears it
    LDR R1, =NVIC_ST_CURRENT_R
    MOV R0, #0
    STR R0, [R1]                ; clear counter
; enable SysTick with core clock
    LDR R1, =NVIC_ST_CTRL_R
    MOV R0, #0x0005     ; Enable but no interrupts (later)
    STR R0, [R1]        ; ENABLE and CLK_SRC bits set
    BX  LR
```