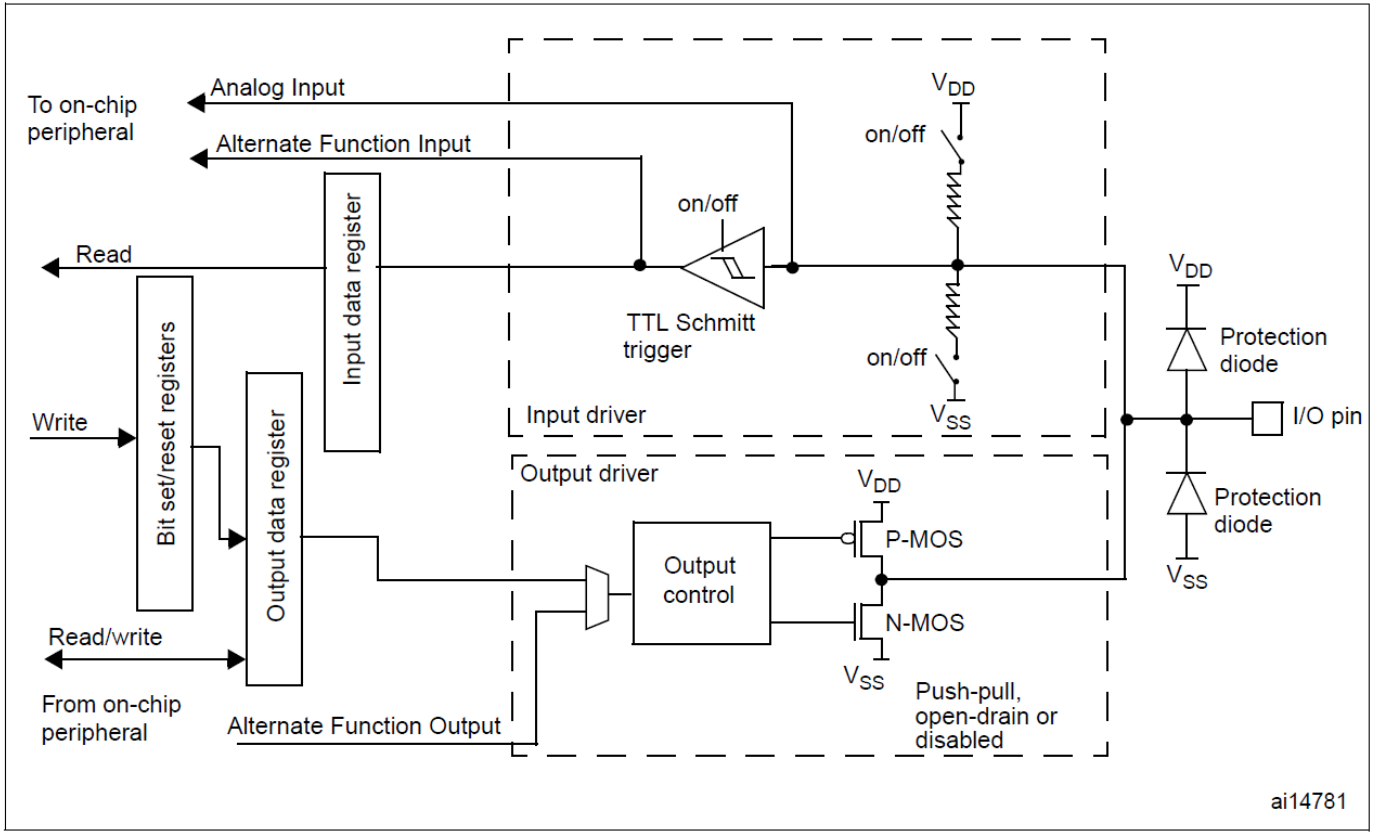

Iran University of
Science and Technology

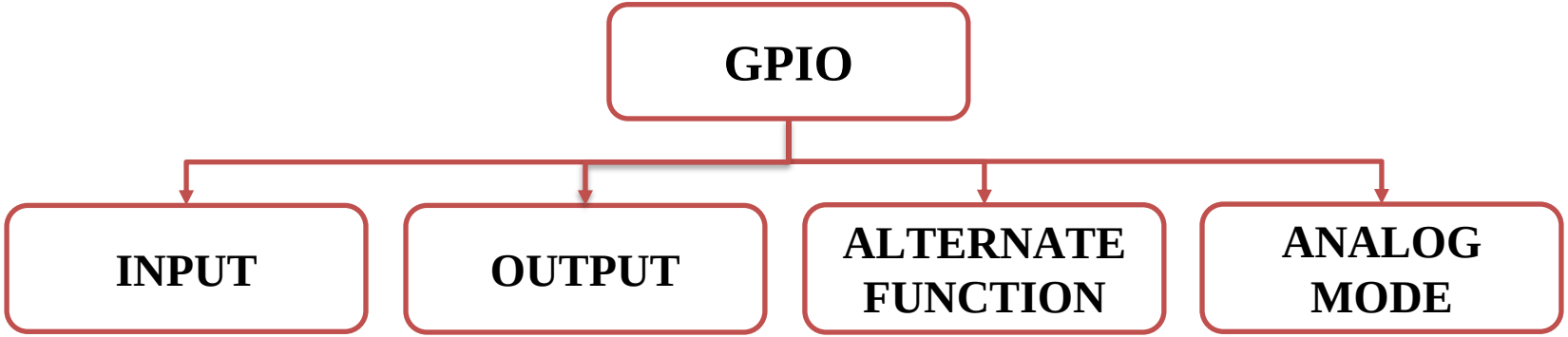
# Microcontroller

Department of Electrical Engineering
Iran University of Science and Technology

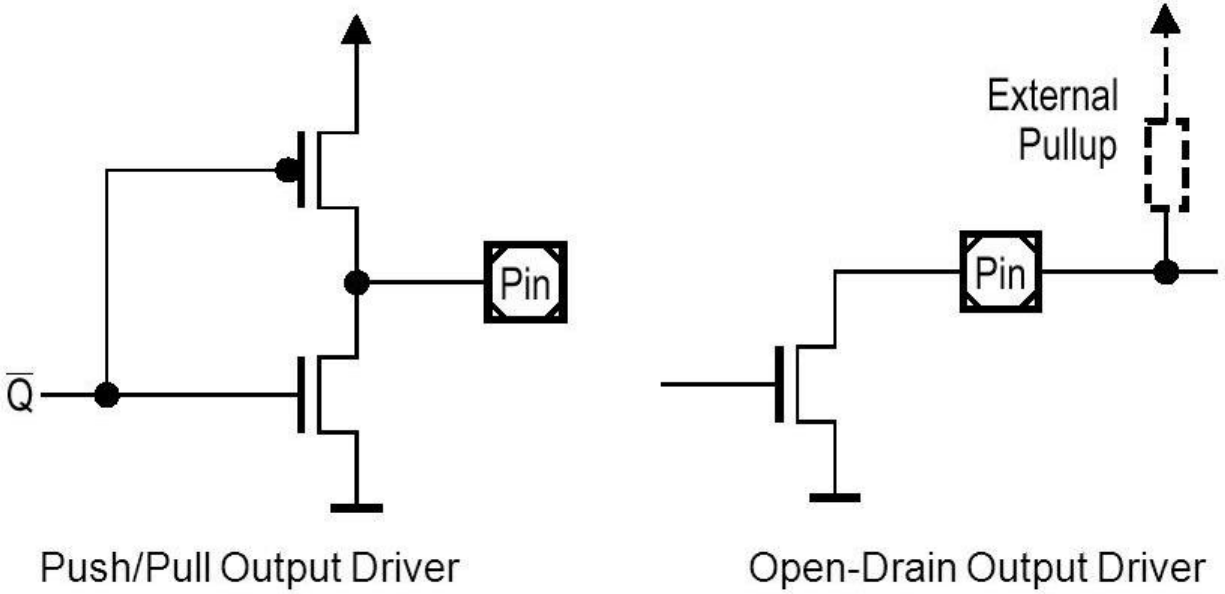
# GPIO MAP


To on-chip peripheral
Analog Input
Alternate Function Input
Read
Write
Read/write
From on-chip peripheral
Bit set/reset registers
Input data register
Output data register
Alternate Function Output
on/off
TTL Schmitt trigger
on/off
on/off
VDD
VSS
Input driver
Output driver
VDD
P-MOS
Output control
N-MOS
VSS
Push-pull, open-drain or disabled
VDD
Protection diode
I/O pin
Protection diode
VSS
ai14781

# GPIO MODE



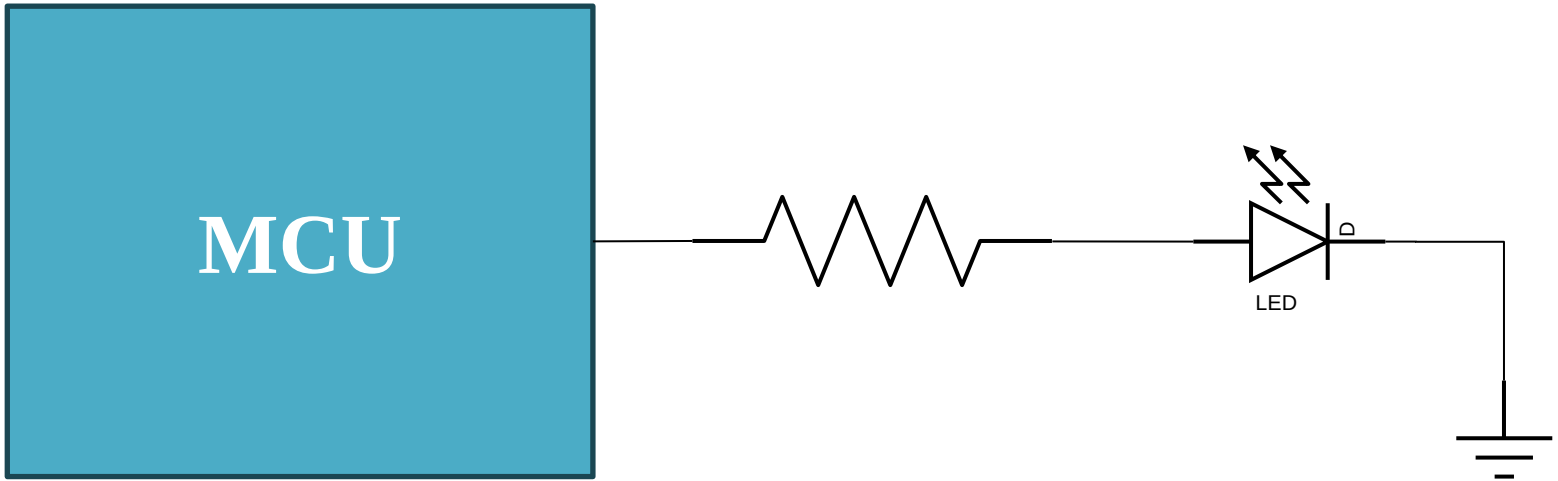
# PUSH PULL / OPEN DRAIN




Push/Pull Output Driver

Open-Drain Output Driver

# LED



# RCC_APB2ENR :IOPCEN

IO port C clock enable
0: IO port C clock disabled
1: IO port C clock enabled

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | TIM11 EN | TIM10 EN | TIM9 EN | Reserved | | |
| | | | | | | | | | | rw | rw | rw | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADC3 EN | USART 1EN | TIM8 EN | SPI1 EN | TIM1 EN | ADC2 EN | ADC1 EN | IOPG EN | IOPF EN | IOPE EN | IOPD EN | IOPC EN | IOPB EN | IOPA EN | Res. | AFIO EN |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | | rw |

# RCC_APB2ENR :IOPCEN

```
RCC->APB2ENR |= RCC_APB2ENR_IOPCEN
```

# GPIOx_CRH :MODEy[1:0]

Port x mode bits (y= 8 .. 15)
00: Input mode (reset state)
01: Output mode, max speed 10 MHz.
10: Output mode, max speed 2 MHz.
11: Output mode, max speed 50 MHz.

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CNF15[1:0] | | MODE15[1:0] | | CNF14[1:0] | | MODE14[1:0] | | CNF13[1:0] | | MODE13[1:0] | | CNF12[1:0] | | MODE12[1:0] | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CNF11[1:0] | | MODE11[1:0] | | CNF10[1:0] | | MODE10[1:0] | | CNF9[1:0] | | MODE9[1:0] | | CNF8[1:0] | | MODE8[1:0] | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

# GPIOx_CRH :MODEy[1:0]

```
RCC->APB2ENR |= RCC_APB2ENR_IOPCEN;

GPIOC->CRH |= GPIO_CRH_MODE9;
```

# GPIOx_CRH :CNFy[1:0]

Port x configuration bits (y= 8 .. 15)
In output mode (MODE[1:0] > 00):
00: General purpose output push-pull
01: General purpose output Open-drain
10: Alternate function output Push-pull
11: Alternate function output Open-drain

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CNF15[1:0] | | MODE15[1:0] | | CNF14[1:0] | | MODE14[1:0] | | CNF13[1:0] | | MODE13[1:0] | | CNF12[1:0] | | MODE12[1:0] | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CNF11[1:0] | | MODE11[1:0] | | CNF10[1:0] | | MODE10[1:0] | | CNF9[1:0] | | MODE9[1:0] | | CNF8[1:0] | | MODE8[1:0] | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

# GPIOx_CRH :CNFy[1:0]

```
RCC->APB2ENR |= RCC_APB2ENR_IOPCEN;

GPIOC->CRH |= GPIO_CRH_MODE9;
GPIOC->CRH &= ~GPIO_CRH_CNF9;
```

# GPIOx_ODR

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ODR15 | ODR14 | ODR13 | ODR12 | ODR11 | ODR10 | ODR9 | ODR8 | ODR7 | ODR6 | ODR5 | ODR4 | ODR3 | ODR2 | ODR1 | ODR0 |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

# GPIOx_ODR

```
GPIOC->ODR |= GPIO_ODR_ODR13;

GPIOC->ODR &= ~ GPIO_ODR_ODR13;
```

# GPIOx_BSRR

Bits 15:0 **Set** bit y
0: No action on the corresponding ODRx bit
1: Set the corresponding ODRx bit

Bits 31:16 **Reset** bit y
0: No action on the corresponding ODRx bit
1: Reset the corresponding ODRx bit

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BR15 | BR14 | BR13 | BR12 | BR11 | BR10 | BR9 | BR8 | BR7 | BR6 | BR5 | BR4 | BR3 | BR2 | BR1 | BR0 |
| w | w | w | w | w | w | w | w | w | w | w | w | w | w | w | w |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BS15 | BS14 | BS13 | BS12 | BS11 | BS10 | BS9 | BS8 | BS7 | BS6 | BS5 | BS4 | BS3 | BS2 | BS1 | BS0 |
| w | w | w | w | w | w | w | w | w | w | w | w | w | w | w | w |

# GPIOx_BSRR

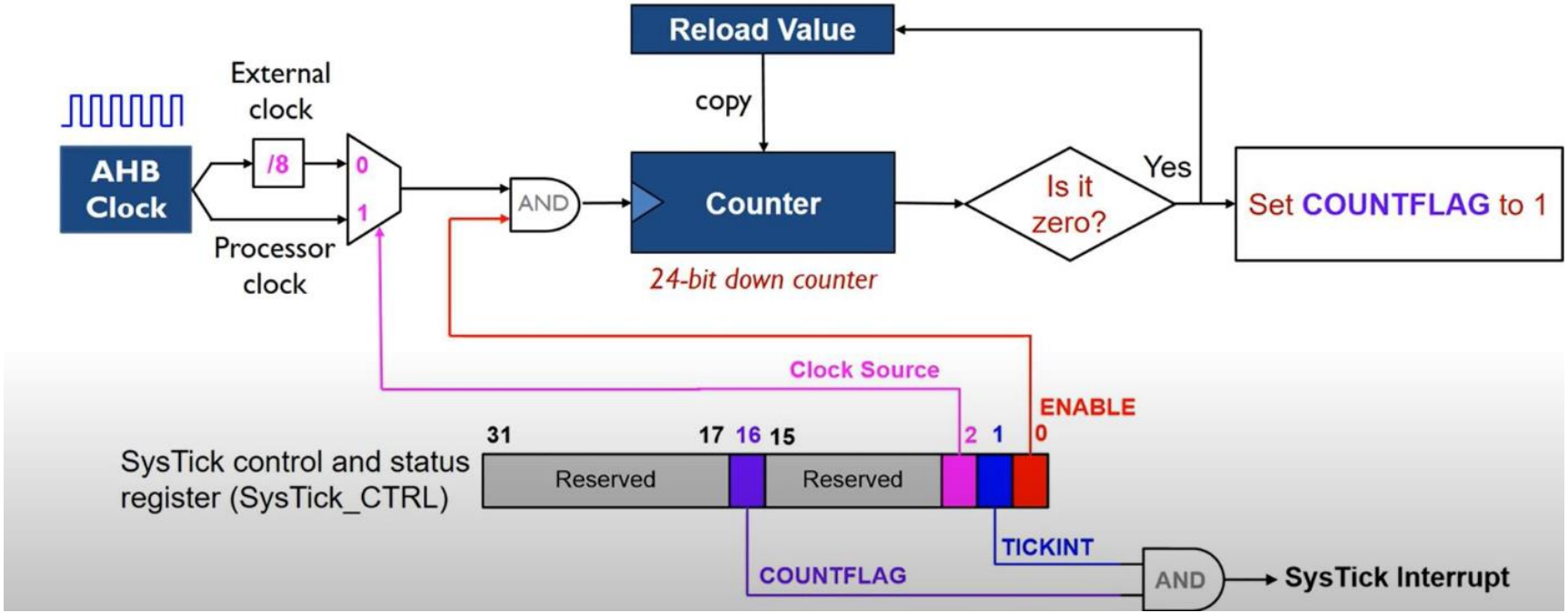
```
GPIOC->BSRR = GPIO_BSRR_BS13;

GPIOC->BSRR = GPIO_BSRR_BR13;
```

# SYSTICK

The processor has a 24-bit system timer, SysTick, that counts down from the reload value to zero, reloads (wraps to) the value in the LOAD register on the next clock edge, then counts down on subsequent clocks.

# SYSTICK


Reload Value
copy
External
clock
AHB
Clock
/8
0
1
Processor
clock
AND
Counter
24-bit down counter
Is it
zero?
Yes
Set COUNTFLAG to 1
Clock Source
ENABLE
31
17 16 15
2 1
0
SysTick control and status
register (SysTick_CTRL)
Reserved
Reserved
TICKINT
COUNTFLAG
AND
SysTick Interrupt

# STK_LOAD

RELOAD[23:0]: RELOAD value
The LOAD register specifies the start value to load into the VAL register when the counter is enabled and when it reaches 0.

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | RELOAD[23:16] | | | | | | | |
| | | | | | | | | rw | rw | rw | rw | rw | rw | rw | rw |

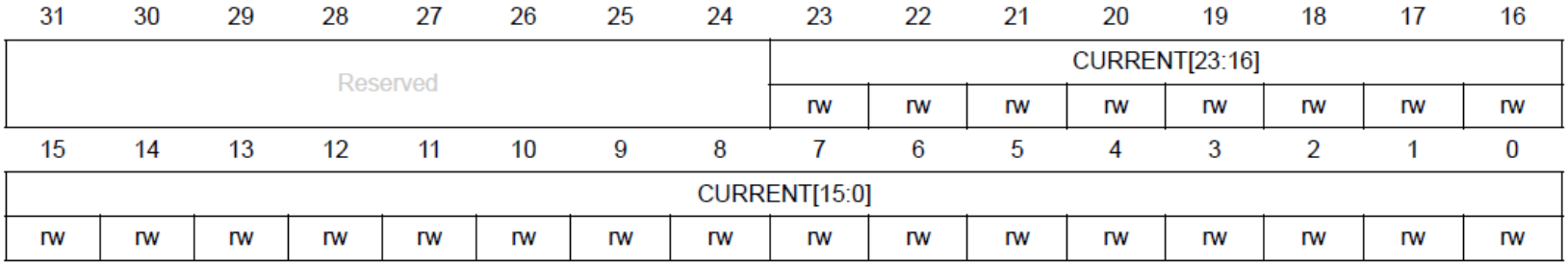
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RELOAD[15:0] | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

# STK_LOAD

```
uint32_t  ticks = SystemCoreClock / 1000;
   SysTick->LOAD = ticks - 1;
```

# STK_VAL

CURRENT[23:0]: Current counter value
The VAL register contains the current value of the SysTick counter.



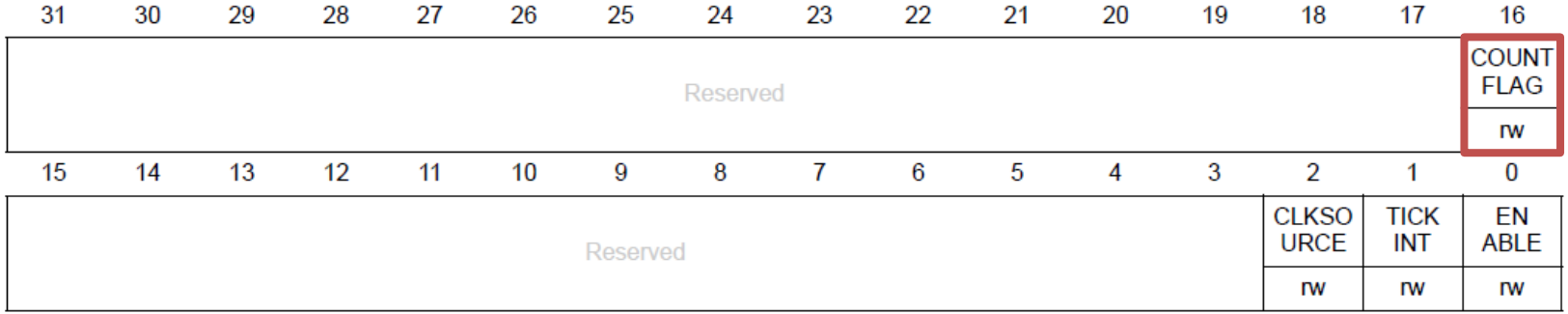
# STK_LOAD

```
uint32_t  ticks = SystemCoreClock / 1000;
  SysTick->LOAD = ticks - 1;

  SysTick->VAL = 0;
```

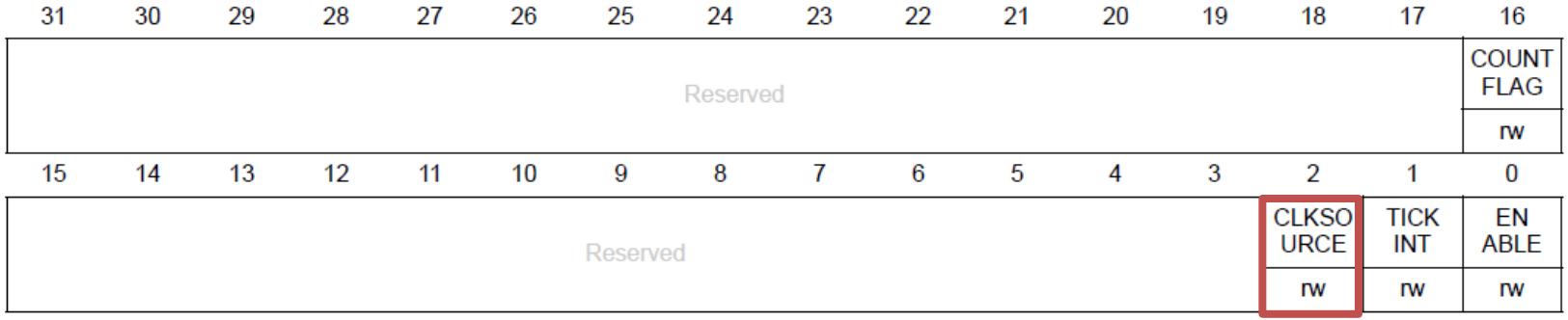
# STK_CTRL

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



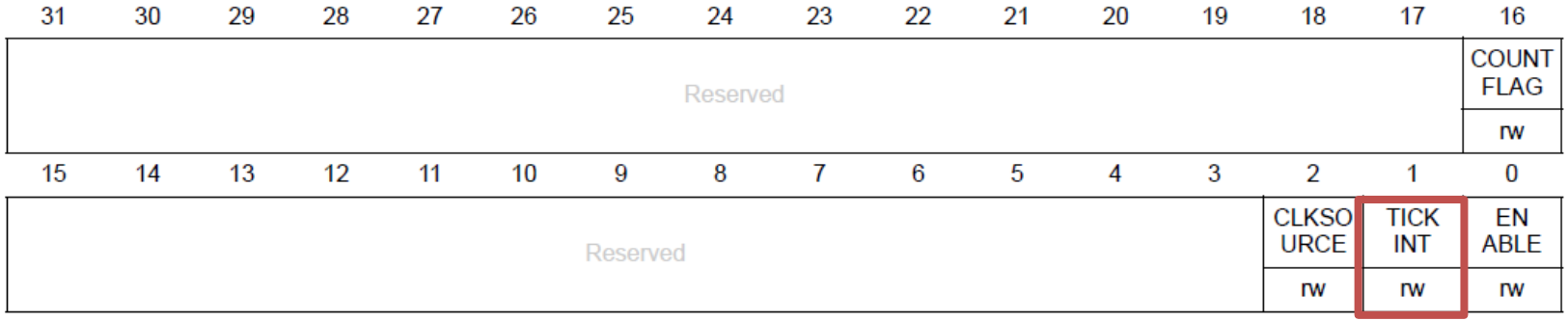
# STK_CTRL

CLKSOURCE: Clock source selection
Selects the clock source.
0: AHB/8
1: Processor clock (AHB)



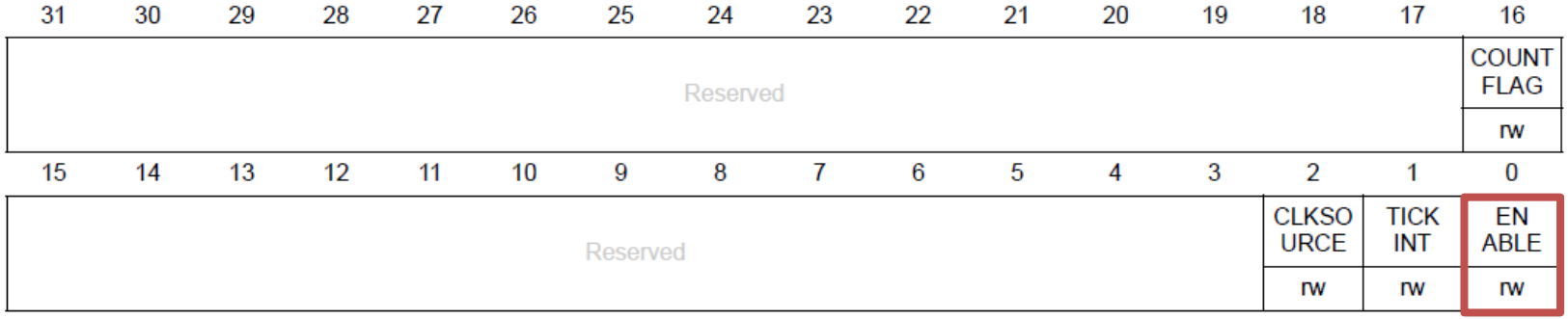
# STK_CTRL

TICKINT: SysTick exception request enable
0: Counting down to zero does not assert the SysTick exception request
1: Counting down to zero to asserts the SysTick exception request.



# STK_CTRL

ENABLE: Counter enable



# STK_LOAD

```
uint32_t  ticks = SystemCoreClock / 1000;
   SysTick->LOAD = ticks - 1;

   SysTick->VAL = 0;

   SysTick->CTRL |= SysTick_CTRL_CLKSOURCE |
                    SysTick_CTRL_TICKINT |
                    SysTick_CTRL_ENABLE;
```

# Delay function

```
volatile uint32_t msTicks = 0;
void SysTick_Handler(void) {
    if (msTicks > 0) {
        msTicks--;
    }
}
void delay_ms(uint32_t ms) {
    msTicks = ms;
    while (msTicks > 0) { }
}
```

# Contact us


tarasarpoolaki@gmail.com

amin.feizi751381@gmail.com

www.linkedin.com/in/Tara-Sarpoolaki

www.linkedin.com/in/aminfeizishahri