



# Chapter 6

## The 8051 Microcontroller

### Timer/Counter

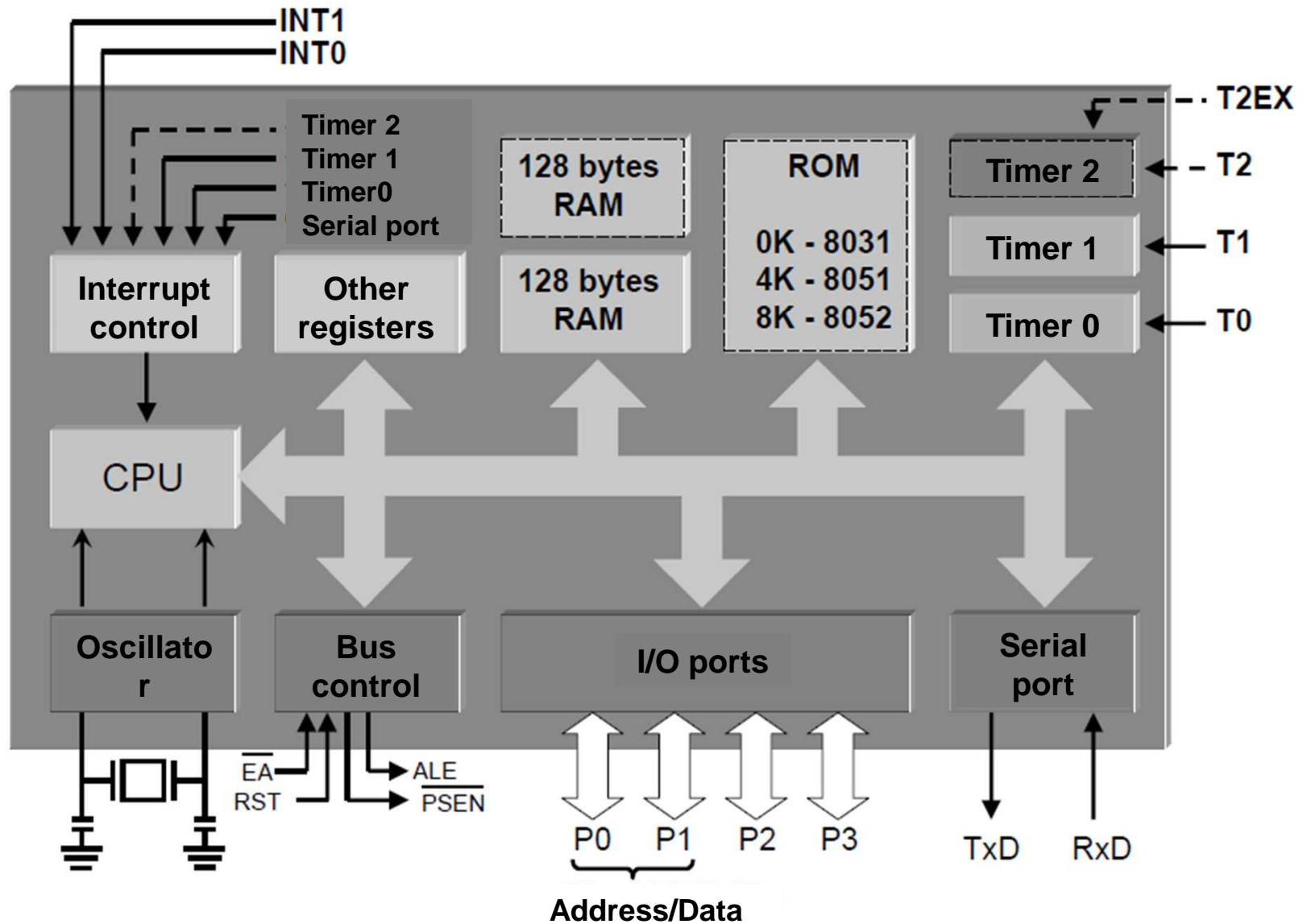


Faculty of Computer Science and Engineering  
Department of Computer Engineering

Nguyen Quang Huy

[huynguyen@cse.hcmut.edu.vn](mailto:huynguyen@cse.hcmut.edu.vn)

# Introduction



CPU

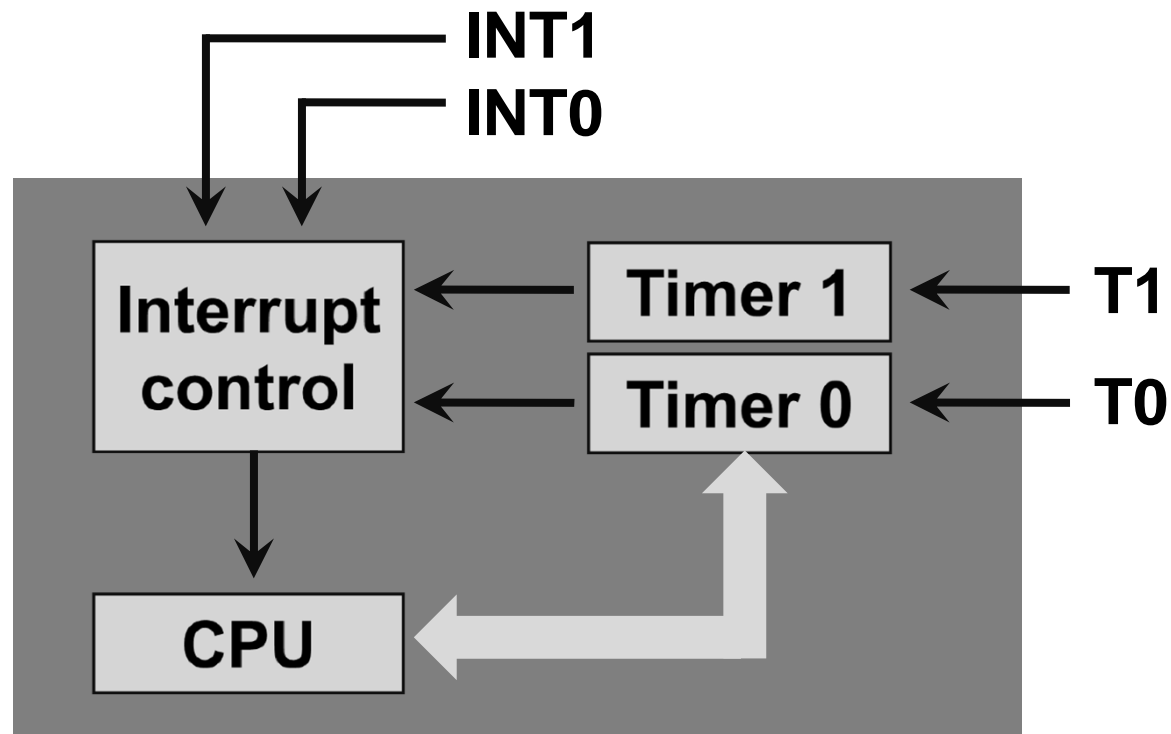
I/O

Timer

Serial

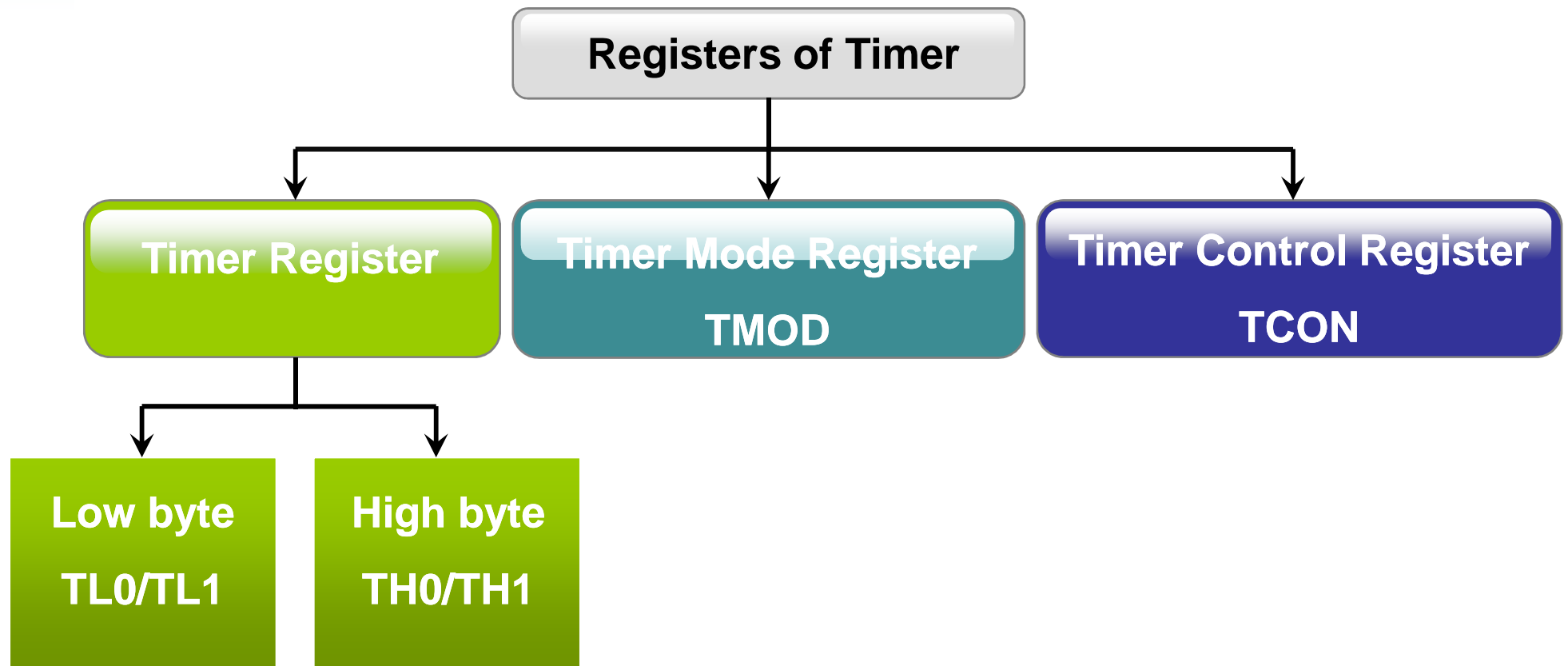
Interrupt

# Timer overview



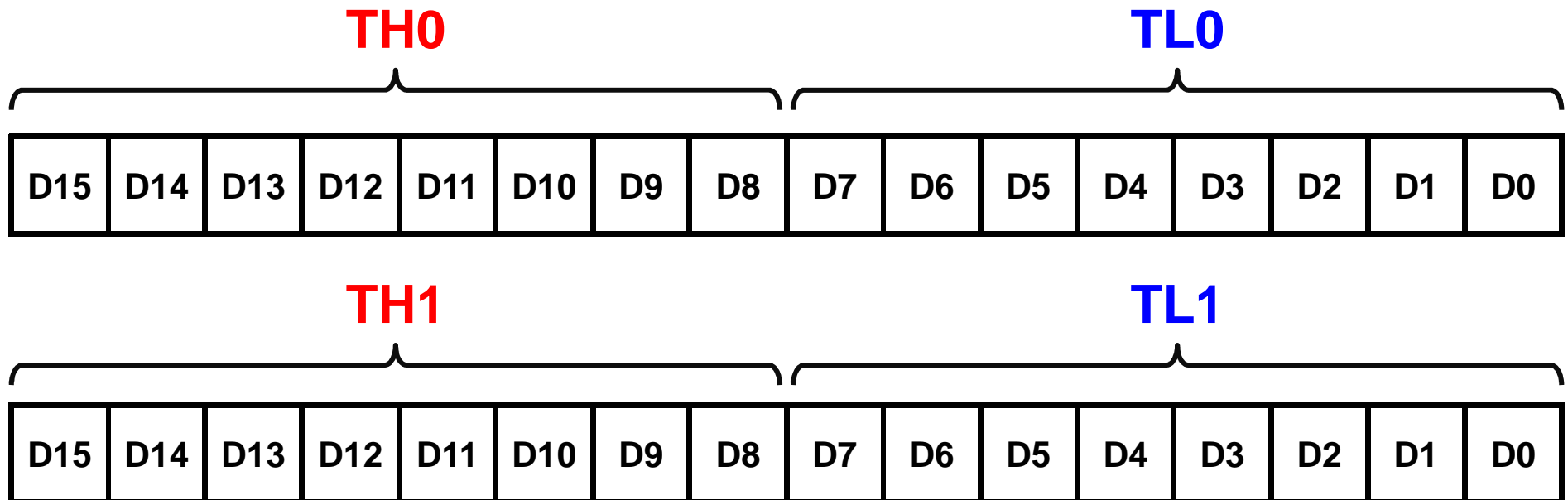
- Two **16-bit** timers/counters
  - Generate a time delay
  - Count events happening outside the microcontroller

# Basic Registers



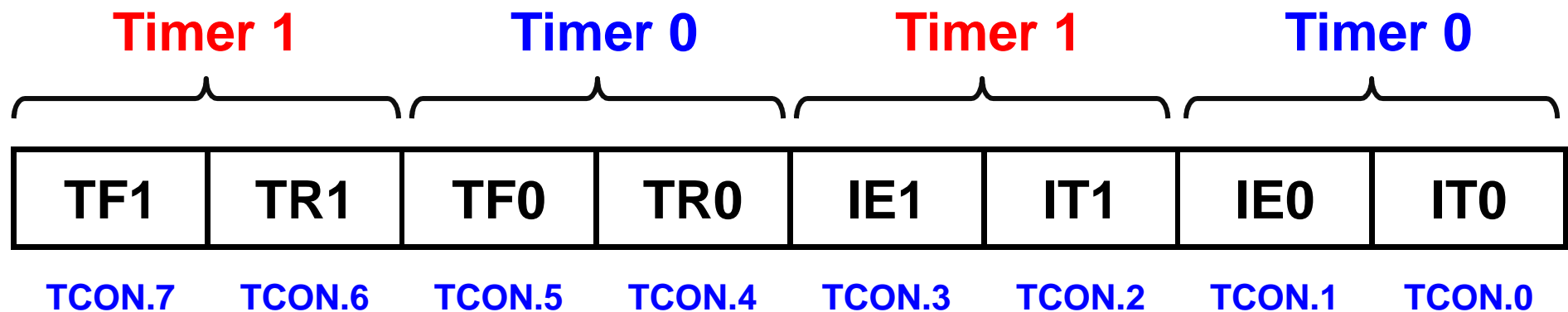
# Timer Register (Timer Counter)

- 16-bit register (**byte-addressable** only)
  - Accessed as LOW byte (**TLx**) and HIGH byte (**THx**)
  - Example:  $10000_{10} = 2710_H \longrightarrow \text{TL0} = 0x10;$   
 $\text{TH0} = 0x27;$



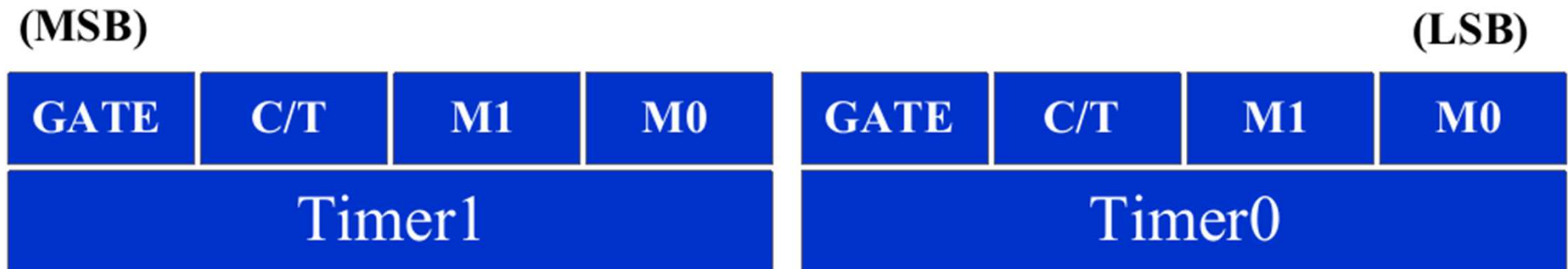
# Timer Control Register (TCON)

- 8-bit register (**bit-addressable** access)
  - IT0/IT1: used for timer interrupt
  - IE0/IE1: used for external interrupt
  - **TR0/TR1: start/stop timer 0/1**
    - $TR_x = 1 \rightarrow$  start timer
  - **TF0/TF1: overflow flag of timer 0/1**
    - $TF_x = 1 \rightarrow$  overflow



# Timer Mode Register (TMOD)

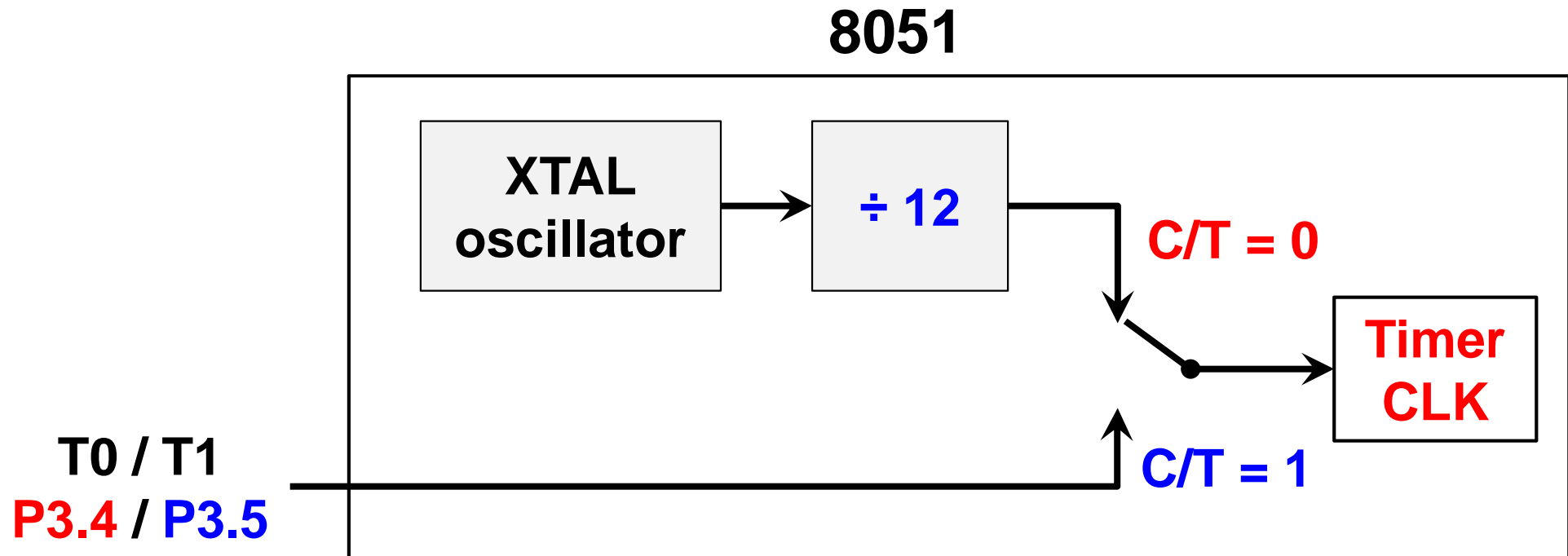
- TMOD is 8-bit register (**byte-addressable** only)
- Set the various timer operation modes
- Both Timer 0 and 1 use the same register
  - The lower 4 bits: Timer 0
  - The upper 4 bits: Timer 1





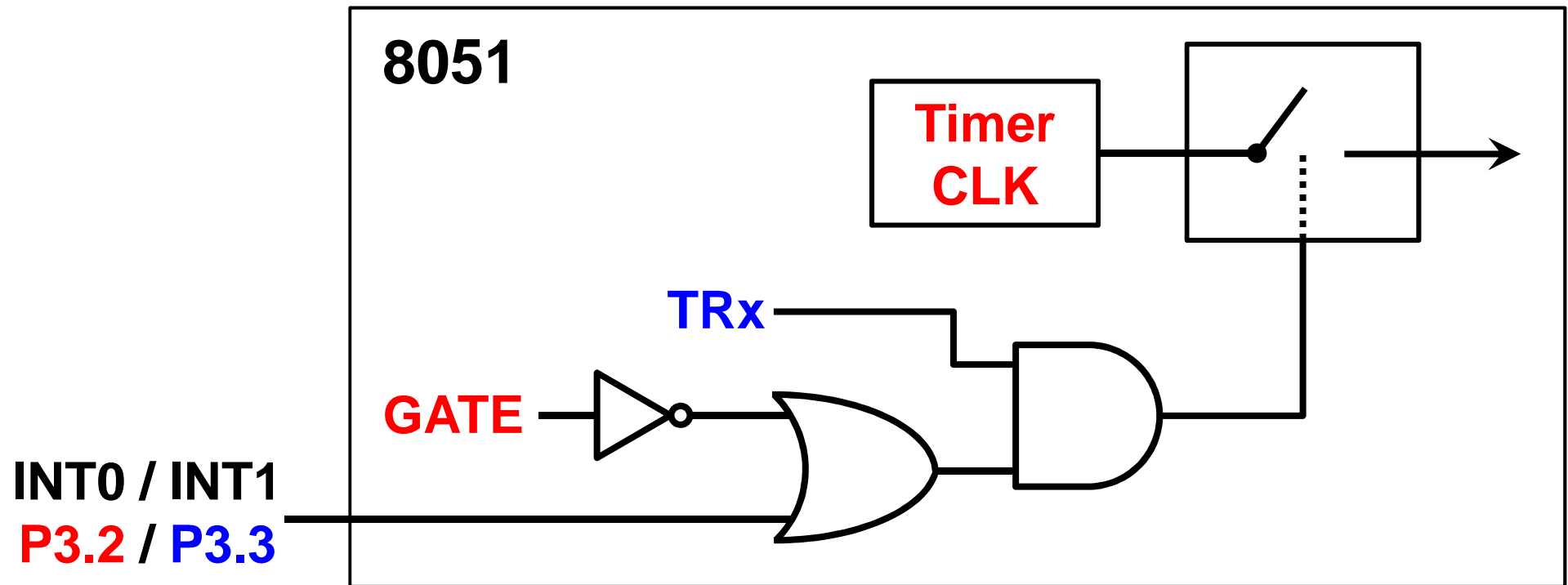
# Timer Mode Register (TMOD)

- C/T (Counter/Timer)
  - C/T = 0 → Timer Mode
  - C/T = 1 → Counter Mode



# Timer Mode Register (TMOD)

- GATE: start/stop control
  - GATE = 0: **only** software control
  - GATE = 1: **enable** hardware control



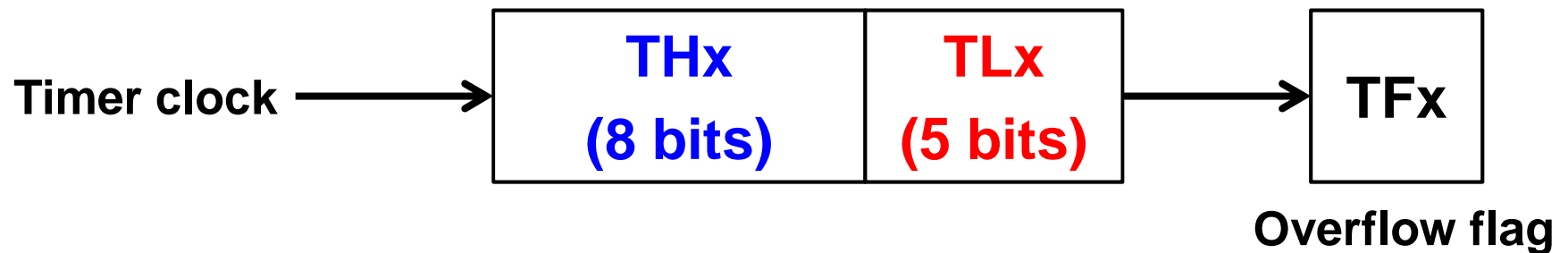
# Timer Mode Register (TMOD)

M1	M0	Mode	Operating Mode
0	0	0	<b>13-bit timer mode</b> 8-bit timer/counter THx with TLx as 5-bit prescaler
0	1	1	<b>16-bit timer mode</b> 16-bit timer/counter THx and TLx are cascaded; there is no prescaler
1	0	2	<b>8-bit auto reload</b> 8-bit auto reload timer/counter; THx holds a value which is to be reloaded TLx each time it overflows
1	1	3	<b>Split timer mode</b>

# Timer Modes

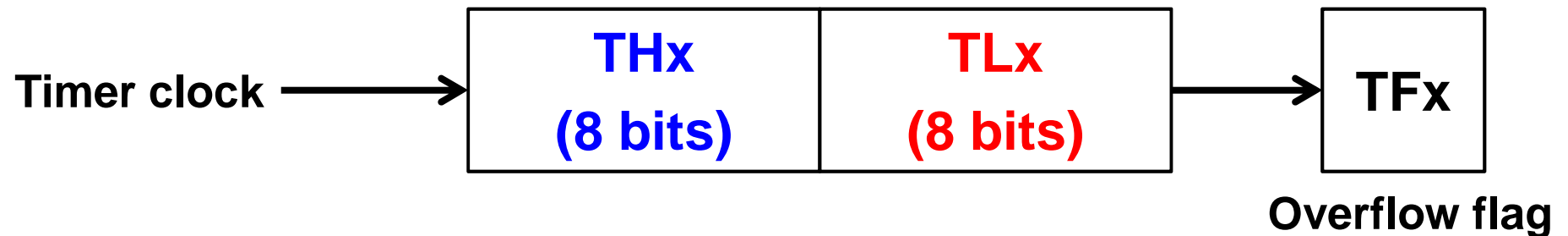
- **Mode 0: 13-bit Timer**

- High byte (TH0/TH1) + 5 bits of lower byte (TL0/TL1)



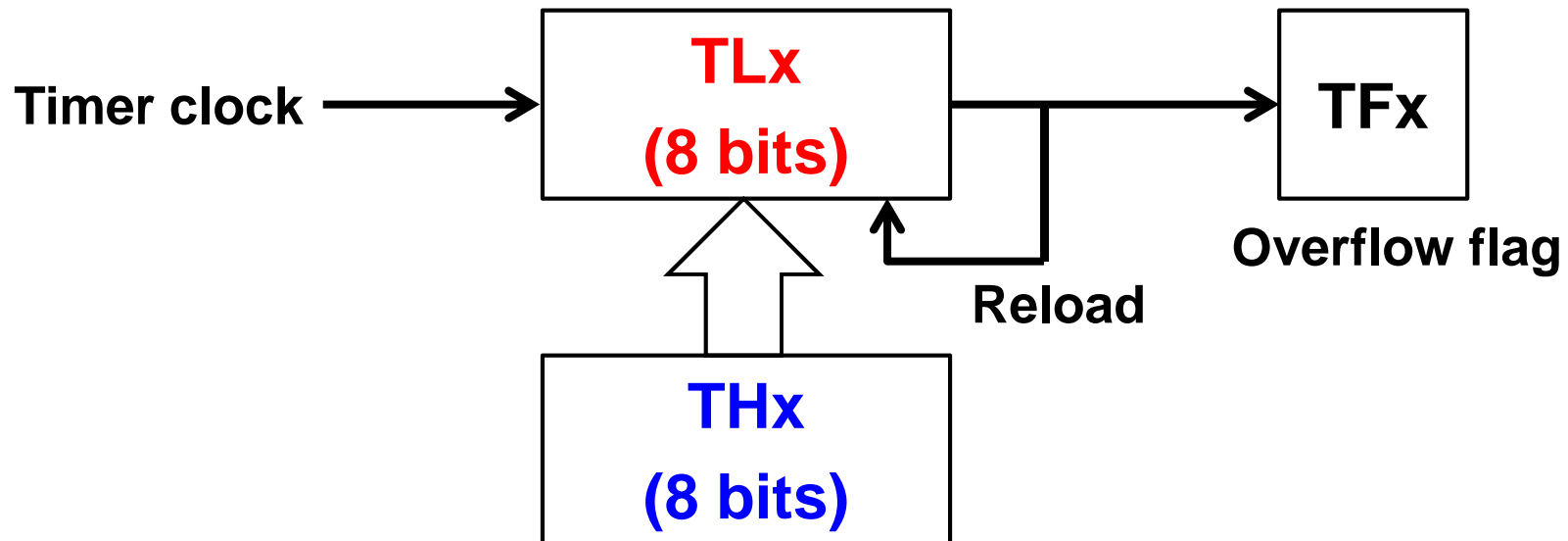
- **Mode 1: 16-bit Timer**

- All 16 bits of timer are used (TL0/TL1 + TH0/TH1)



# Timer Modes

- **Mode 2: 8-bit Auto Reload**
  - Only the lower byte (TLx) is used for counting
  - Upper byte (THx) holds the value to reload into TLx after an overflow

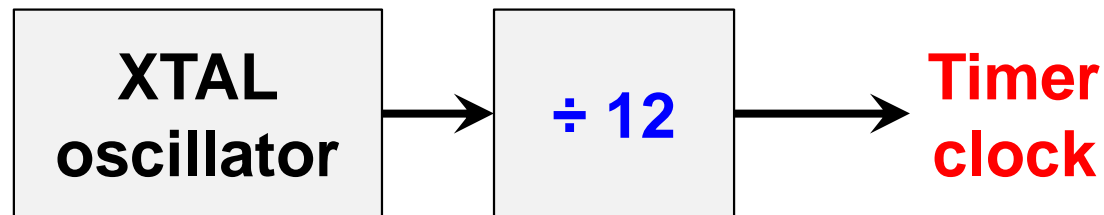


# Example

- Indicate which mode and which timer are selected
  - a)  $\text{TMOD} = 0\text{x}01\text{H}$
  - b)  $\text{TMOD} = 0\text{x}12\text{H}$
- Solution
  - a)  $01\text{H} \rightarrow 0000\ 0001\text{B}$   
**Mode 1** of **timer 0** is selected
  - b)  $12\text{H} \rightarrow 0001\ 0010\text{B}$   
**Mode 1** of **timer 1** and **mode 2** of **timer 0** are selected

# Timer Delay & Timer Loaded Value

- Timer Delay =  $D \times T$ 
  - $D$  : Delay value  $\rightarrow$  how many counts (clocks) before overflow flag **TFx is set**
  - $T$  : Timer clock cycle duration



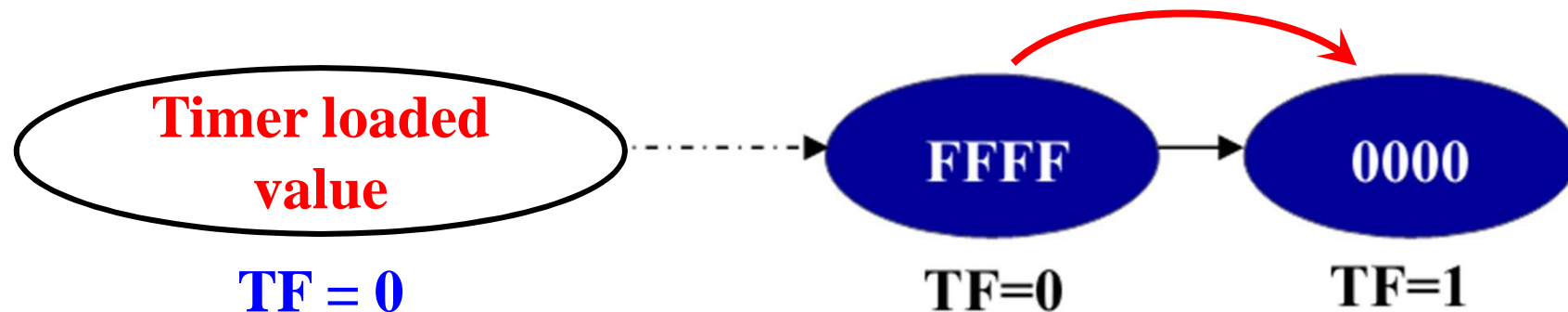
- XTAL = 11.0592MHz
- Timer clock frequency

$$f = 11.0592 \text{ MHz} / 12 = 921.6 \text{ kHz}$$

$\rightarrow$  Timer clock cycle:  $T = 1/f = 1/921.6\text{kHz} = 1.085\mu\text{s}$

# Timer Delay & Timer Loaded Value

- How to calculate *timer loaded value*?



$$\text{Delay value (D)} = \text{Maximum Count} - \text{Timer loaded value} + 1$$

$$\text{Timer loaded value} = \text{Maximum Count} - \text{Delay value} + 1$$



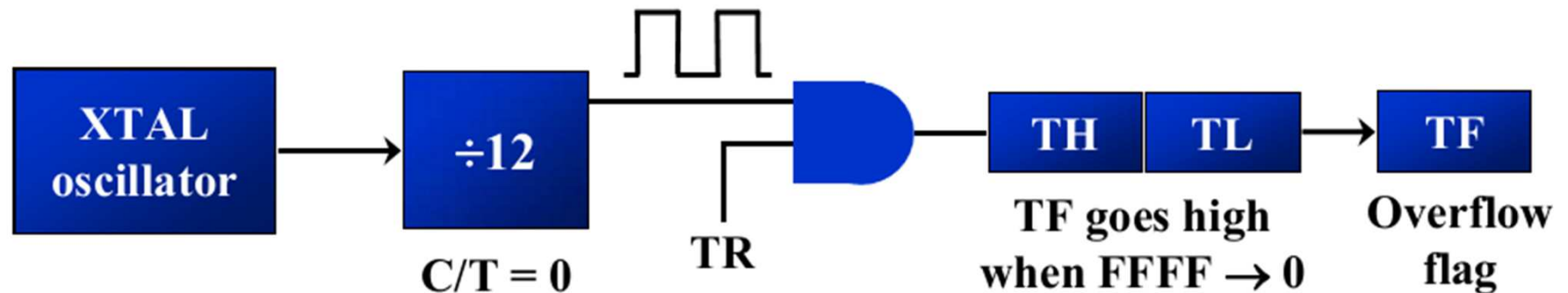
## Example

- Assume XTAL = 12 MHz. If we want to have a time delay of **5ms**, what value do we need to load into the timer register **TH** and **TL**?
- Solution
  - $T = 12/12\text{MHz} = 1 \text{ (us)}$
  - Delay value =  $5\text{ms} / 1 \text{ us} = 5000 \text{ (clocks)}$
  - Timer loaded value =  $65\,535 - 5000 + 1$   
 $= 60\,536_{10} = \text{EC}78_{16}$

→ **TH = 0xEC, TL = 0x78**

# Programming Timer – Mode 1

- **Mode 1: 16-bit timer**
  - Timer value: 0000 to FFFFh (THx & TLx)
  - **Set bit TRx** to start the timer
  - Timer starts to count up after bit TRx = 1
  - TFx (timer flag) is set when FFFF → 0 in timer register
  - THx and TLx **must be reloaded** with original value and TFx **must be cleared** with 0 to repeat the process



## Example – Timer Mode 1

- Write an 8051 C program to toggle only bit P1.5 continuously every 50 ms using Timer 0, mode 1 (16-bit) to create the delay. Assume XTAL = 11.0592 MHz

- $T = 12/11.0592\text{MHz} = 1.085 \text{ (us)}$
- Delay value =  $50\text{ms} / 1.085 \text{ us} \sim 46083 \text{ (clocks)}$
- Timer loaded value =  $65\,535 - 46083 + 1$   
 $= 19453_{10} = 4\text{BFD}_{16}$

→ **TH0 = 0x4B, TL0 = 0xFD**

# Example – Timer Mode 1

```
include <reg51.h>
sbit mybit = P1^5;
```

```
void T0M1Delay(void){
    TMOD = 0x01;
    TH0 = 0x4B;
    TL0 = 0xFD;
    TR0 = 1;
    while (TF0 == 0);
    TR0 = 0;
    TF0 = 0;
}
```

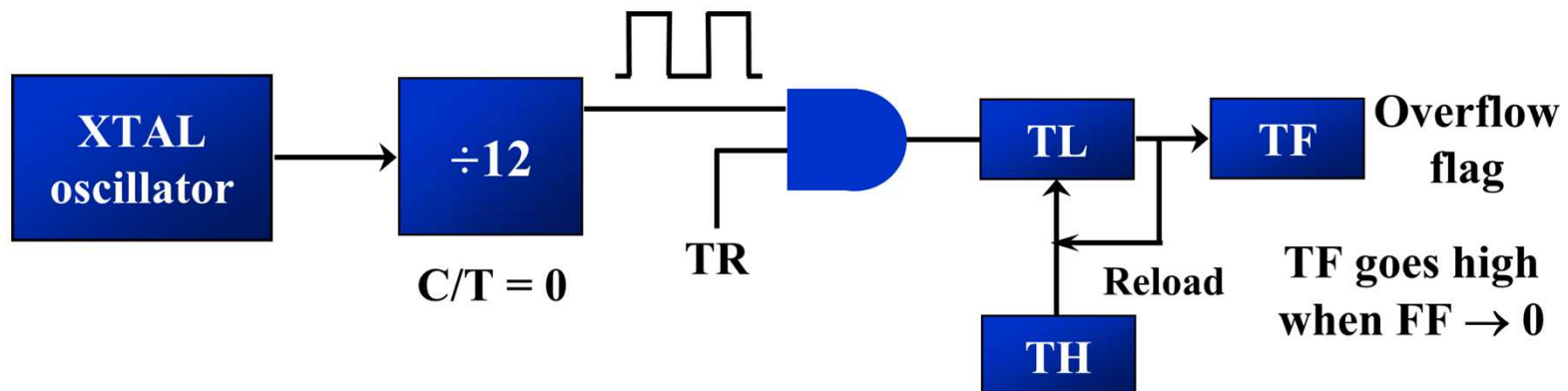
```
void main(void)
{
    while (1) {
        mybit = ~mybit;
        T0M1Delay();
    }
}
```

# Issue

- Maximum Timer Delay using Timer Mode 1?
- If *Time delay* > *Maximum Timer Delay*

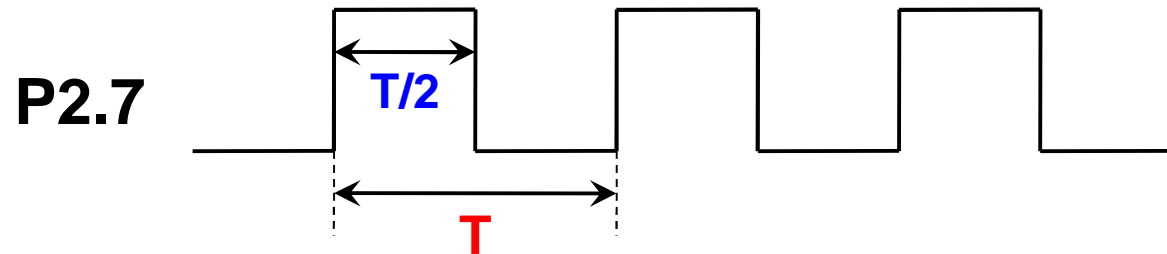
# Programming Timer – Mode 2

- Mode 2: 8-bit auto reload
  - Timer value: 00 to FFh  $\rightarrow$  THx
    - 8051 gives a copy of THx to TLx
  - Set bit TRx to start the timer
  - Timer starts to count up by TLx register after bit TRx = 1
  - TFx (timer flag) is set when FF  $\rightarrow$  0 in TLx register
  - TLx is reloaded **automatically** with the value in THx register
  - Clear bit TFx to repeat the process



## Example – Timer Mode 2

- Write an 8051 C program to create a frequency of **2500 Hz** on pin P2.7 using Timer 1, mode 2 to create delay. Assume XTAL = 11.0592 MHz



- $T = 1/2500\text{Hz} = 400 \text{ (us)} \rightarrow T/2 = 400/2 = 200 \text{ (us)}$
- Delay value =  $200 \text{ us} / 1.085 \text{ us} \sim 184 \text{ (clocks)}$
- Timer loaded value =  $255 - 184 + 1$   
 $= 72_{10} = 48_{16}$

→ **TH1 = 0x48**

or **TH1 = -184**

## Example – Timer Mode 2

```
#include <reg51.h>
sbit mybit = P2^7;

void T1M2Delay(void)
{
    TR1 = 1;
    while (TF1 == 0);
    TR1 = 0;
    TF1 = 0;
}
```

```
void main(void)
{
    TMOD = 0x20;
    TH1 = -184;

    while (1) {
        mybit = ~mybit;
        T1M2Delay();
    }
}
```



# Reference

- “*The 8051 Microcontroller and Embedded Systems Using Assembly and C – 2<sup>nd</sup>*” - Muhammad Ali Mazidi, Janice Gillispie Mazidi, Rolin D.McKinlay
- “*The 8051 Microcontroller - 2<sup>nd</sup>*” - I. Scott Mackenzie, Prentice-Hall 1995