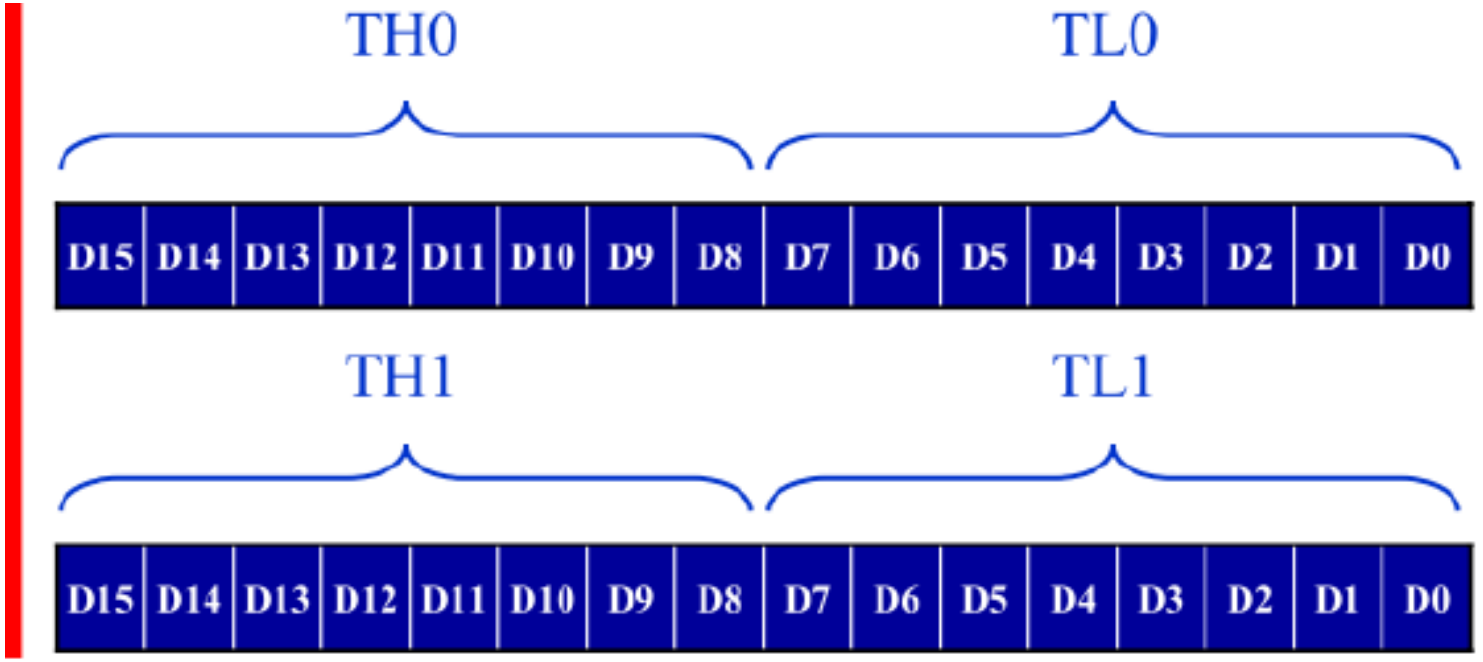
# Lab No.6
# Implementation of Timer using 8051


- Engr. Shahzada Fahim Jan

# Timers:

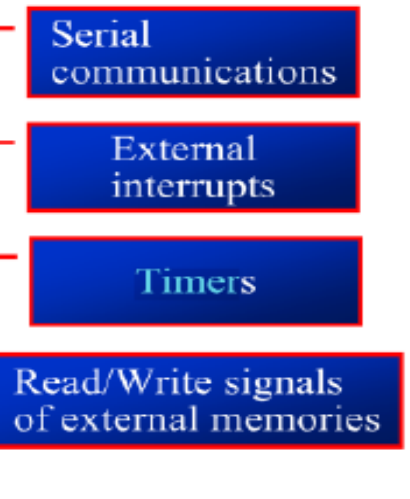
- 8051 has two timers/counters, They can be used either as
  - ➢ Timer to generate a timer delay
  - ➢ Event counter to count events happening outside the microcontroller
- Both timer 0 and timer 1 are 16 bit wide
- Since 8051 has 8-bit architecture, each 16 bit timer is accessed as two separate registers of low byte and high byte

- Accessed as low byte and high byte
- Low byte register is called TL0/TL1
- High byte register is called TH0/TH1



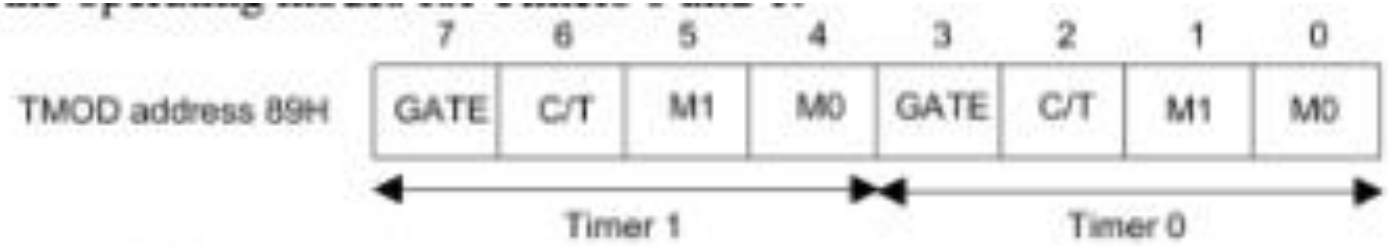
- PORT 3 can be used as input or output
- PORT 3 does not need any Pull-up resistor
- PORT 3 has additional function of providing some extremely important signals

| P3 Bit | Function | Pin |
|---|---|---|
| P3.0 | RxD | 10 |
| P3.1 | TxD | 11 |
| P3.2 | $\overline{\text{INT0}}$ | 12 |
| P3.3 | $\overline{\text{INT1}}$ | 13 |
| P3.4 | T0 | 14 |
| P3.5 | T1 | 15 |
| P3.6 | $\overline{\text{WR}}$ | 16 |
| P3.7 | $\overline{\text{RD}}$ | 17 |



# TMOD (timer mode) register

- Both timer 0 and timer 1 use the same register, called TMOD to set the various timer operation modes.
- TMOD is an 8-bit register in which the lower 4 bits are set aside for Timer 0 and the upper 4 bits for Timer 1.
- In each case the lower 2 bits are used to set the timer mode and the upper 2 bits to specify the operation.




Gate: 0 = timer runs whenever TR0/TR1 set
1 = timer runs only when INT0/INT1 is high along with TR0/TR1

C/T:counter/timer select
0 = input from system clock,1 = input from TX0/TX1

| M1 | M0 | Mode | |
|---|---|---|---|
| 0 | 0 | 0 | 13-bit counter, lower 5 bits of TL0 and all 8 bits of TH0 |
| 0 | 1 | 1 | 16-bit counter |
| 1 | 0 | 2 | 8-bit auto-reload timer/counter |
| 1 | 1 | 3 | TL0 is a 8-bit timer/counter controlled by Timer 0 control bits. TH0 is an 8-bit timer controlled by Timer 1 control bits. Timer 1 is off. |

# M1,M0

- M0 and M1 select the timer mode.
- There are three modes 0, 1 and 2
- Mode 0 is a 13-bit timer
- Mode 1 is a 16-bit timer
- Mode 2 is an 8-bit timer
- We will focus on Mode 1 and Mode 2 since they are the ones used most widely

# C/T(clock/timer)

- This bit in the TMOD register is used to decide whether the timer is used as a delay generator or an event counter. If C/T=0, it is used as a timer for time delay generation

# Mode 1 programming

- Characteristics and operations of mode 1
- It is a 16-bit timer, therefore, it allows values of 0000 to FFFFH to be loaded into the timers register TL and TH.
- After TH and TL are loaded with a 16-bit initial value, the timer must be started. This is done by “SETB TR0” for timer 0 and “SETB TR1” for timer1.
- After the timer is started, it start to count up. It counts up until it reaches its limit of FFFFH. When it rolls over from FFFFH to 0000, it sets high a flag bit called timer flag. This timer flag can be monitored.

- When this timer flag is raised, one option would be to stop the timer with  the instructions “CLR TR0” for timer 0 and “CLR TR1” for timer1.
- After the timer reaches its limit and rolls over, in order to repeat the process the registers TH and TL must be reloaded with the original values, and TF must be reset to 0.

# Time delay

- To generate a time delay, Follow these steps

1. Load TMOD value register indicating which timer (timer 0 or timer 1) is to be used and which timer mode (0 or 1) is selected.
2. Load registers TL and TH with initial count values
3. Start the timer.
4. Keep monitoring the timer flag (TF) with go out of the loop when TF becomes high
5. Stop the timer.
6. Clear the TF flag for the next round
7. Go to step 2 to load TH and TL again

# IE(Interrupt Enable)

## IE: Interrupt Enable

**IE : Interrupt Enable Register (Bit Addressable)**

If the bit is 0, the corresponding interrupt is disabled. If the bit is 1, the corresponding interrupt is enabled.

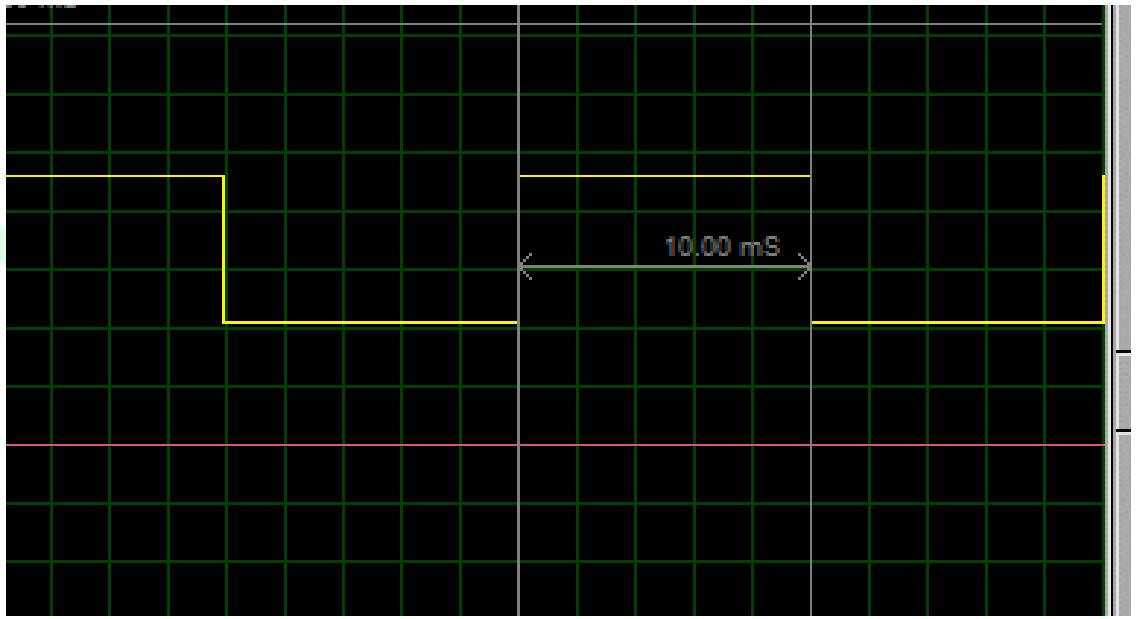
| EA | - | - | ES | ET1 | EX1 | ET0 | EX0 |
|---|---|---|---|---|---|---|---|

| | | |
|---|---|---|
| EA | IE.7 | Disables all interrupts. If EA = 0, no interrupt will be acknowledged. If EA = 1, interrupt source is individually enable or disabled by setting or clearing its enable bit. |
| - | IE.6 | Not implemented, reserved for future use*. |
| - | IE.5 | Not implemented, reserved for future use*. |
| ES | IE.4 | Enable or disable the Serial port interrupt. |
| ET1 | IE.3 | Enable or disable the Timer 1 overflow interrupt. |
| EX1 | IE.2 | Enable or disable External interrupt 1. |
| ET0 | IE.1 | Enable or disable the Timer 0 overflow interrupt. |
| EX0 | IE.0 | Enable or disable External Interrupt 0. |

```c
#include<reg51.h>
#include<stdio.h>
sbit pin = P3^0;
void start_timer0(){
TR0 = 1;
}
void timer0() interrupt 1
{
TH0 = 0xD8;
TL0 = 0xEF; //10 ms delay
}
void start_timer(){
  TMOD = 0x01;
  IE=0x82;
}
void main(){
start_timer0();
start_timer();
while(1)
{
while(TF0==0);
pin = ~pin;

}

}
```



# Tasks

1. Use timer to Generate a delay of 15ms
2. Use timer to Generate a delay of 50ms
3. Use timer to Generate a delay of 200ms
4. Use timer to Generate a delay of 200us