Timer and Counter
Timer and Counter

- Timer and Counter
  - Measure the time/frequency of input signal.
  - Generate outputs with variable frequency and pulse width.
  - Can be controlled with software.
  - Can be accessed thru interrupt and software.

- Structure of Timer/Counter (a series of divide-by-two flip flop:)

- 4 bit counter
Timer and Counter

• Not a power of 2 multiple of the clock freq.
  – Q1: Count-by-10 counter

• Timer: when the incoming clock frequency is known.
  – Get clock from Oscillator frequency (1/12), then by setting the pre-load value. We can generate a fixed period of time known to the designers.

• Counter: when the incoming clock is ‘irregular’
  – Get clock from external pin, and we are only interested in the number of occurrence of this pulse. This is called counter (counting events)
Timer 0 and Timer 1

- Two 16-bit timers: Timer 0 and Timer 1
- Timer 0 (16-bit)

\[
\begin{array}{|c|c|}
\hline
TH0 & TL0 \\
\hline
\end{array}
\]

- Timer 1 (16-bit)

\[
\begin{array}{|c|c|}
\hline
TH1 & TL1 \\
\hline
\end{array}
\]
Setup the Timer and Counter

• 2 registers: TMOD and TCON
• TMOD: Timer Mode Register ➔ Set the various timer operation mode.
  – TMOD is an 8-bit register where the lower 4 bits are set aside for timer 0 and the upper 4 bits are set aside for timer 1.
TMOD Register

- **GATE:** To start and stop the timer
  - GATE=1 ➔ HW control: is enabled only while INTx pin is ‘1’ and TRx control pin (in TCON) is set.
  
  ![Diagram](image)

  ![INT1 — P3.3]
  ![INT0 — P3.2]

  - GATE=0 ➔ SW control (used frequently)
TMOD Register

- C/!T: Timer or counter selection
  - C/!T = 0 ➔ Timer (input from internal system clock)
    the Osc. Crystal (1/12) is used to trigger the timer.
  - C/!T = 1 ➔ Counter (input from Tx input pin)

- M1 and M0: Mode selection for timer and counter

<table>
<thead>
<tr>
<th>Mode</th>
<th>M1</th>
<th>M0</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>➔ 13-bit timer/counter mode</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>➔ 16-bit timer/counter mode</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>➔ 8-bit auto reload timer/counter mode</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>➔ split timer/counter mode</td>
</tr>
</tbody>
</table>
TCON Register

- **TF1**: Timer 1 overflow flag
  - TF1=1: Timer/counter 1 overflows.
  - TF1=0: processor vectors to the interrupt services.

- **TR1**: Timer 1 run control bit
  - TR1=1: turn Timer 1 ON
  - TR1=0: turn Timer 1 OFF
### TCON Register

**MSB**

<table>
<thead>
<tr>
<th>TF1</th>
<th>TR1</th>
<th>TF0</th>
<th>TR0</th>
<th>IE1</th>
<th>IT1</th>
<th>IE0</th>
<th>IT0</th>
</tr>
</thead>
</table>

- **Timer 1**
  - **TF1**: External interrupt 1 edge flag
    - IE1=1: external interrupt is detected.
    - IE1=0: when interrupt is processed.
- **Timer 0**
  - **IT1**: Interrupt 1 type control bit
    - IT1=1: falling edge.
    - IT1=0: low level triggered external interrupt.
Run Timer (SW Control)

- Gate=0, SETB TR1 ➞ Run Timer 1
  SETB TR0 ➞ Run Timer 0
- Gate=0, CLR TR1 ➞ OFF Timer 1
  CLR TR0 ➞ OFF Timer 0
Timer and Counter

- Q2: Indicate which timer and which mode are selected for each of the following (a) MOV TMOD, #01H, (b) MOV TMOD, #20H, and (c) MOV TMOD, #12H.

- Q3: Find the timers’ clock frequency and its period for various 8051-based systems with the following crystal frequencies. (a) 12MHz, (b) 16MHz, and (c) 11.0592MHz.

- Q4: Find the value for TMOD if we want to program Timer 0 in mode 2, use 8051 XTAL for the clock source, and use instructions to start and stop the timer.
Timer Mode 1 Program

- Mode 1: 16-bit timer operation
- Steps
  - 0000 ~ FFFFH to be loaded into the timers’ registers TH and TL.
  - Start timer: SETB TR0 (i.e. Run Timer 0)
  - Timer overflow flag (TF1 or TF0) should be monitored

```
 0000
 0001
:
 FFFF OV
 0000
```
  - Stop timer: CLR TR0 (OFF Timer 0)
  - Repeat: TH and TL are loaded again
  - TF reset to 0
Steps to Program for
Timer 0, Mode 1 Example

• Steps:
  1. Load the TMOD value (e.g. MOV TMOD, #21H)
  2. Load TH0 and TL0 (e.g. MOV TL0, #9CH & MOV TH0, 0FFH)
  3. Start the timer (SETB TR0)
  4. Keep monitor the timer flag (TF0)

    **target:** *JNB TF0, target*  ➔ Jump out only if TF0=1

  5. Stop the timer 0 (CLR TR0)
  6. CLR TF0
  7. Repeat Step 2.
Timer Delay Calculation

- For XTAL=11.0592MHz

\[(FFFF-YYXX+1) \times 1.085 \mu s\]

\[(65535-NNNNN) \times 1.085 \mu s\]

Hexdecimal

TH TH TL

OR

NNNNN is a decimal value derived from YYXXH
Examples

- Q5: In the following, we are creating a square wave of 50% duty cycle (with equal portions high and low) on the P1.5 bit. Timer 0 is used to generate the time delay.
  
  ```assembly
  MOV TMOD, #01
  HERE:
  MOV TL0, #0F2H
  MOV TH0, 0FFH
  CPL P1.5
  ACALL Delay
  SJMP HERE

  Delay:
  SETB TR0

  Again:
  JNB TF0, Again
  CLR TR0
  CLR TF0
  RET
  ```
Examples

- Q6: See the example in Q5, calculate the amount of time delay in the DELAY subroutine generated by the timer. Assume XTAL = 11.0592MHz.

- Q7: Find the delay generated by timer 0 in the following code (Do not include the overhead due to instructions).

  CLR P2.3
  MOV TMOD, #01

  Here: MOV TL0, #3EH
  MOV TH0, #0B8H
  SETB P2.3
  SETB TR0

  Again: JNB TF0, Again
  CLR TR0
  CLR TF0
  CLR P2.3
Examples

- Q8: Modify TL and TH in Q7 to get the largest time delay possible. Find the delay in ms. (Do not include the overhead due to instructions)

- Q9: examples in Q7 and Q8 did not reload TH and TL, since it was a single pulse. The following program generates a square wave on P1.5 continuously using Timer 1 for a time delay. Find the frequency of the square wave if XTAL=11.0592MHz. (Not include overhead due to instructions)
Examples

• Assume we know the amount of timer delay, then how to find the values for TL and TH

• Q10: Assume the XTAL =11.0592MHz, what value do we need to load into the timers’ registers if we want to have a time delay of 5ms? Show the program for timer 0 to create a pulse width of 5ms on P2.3.

• Q11: Assume that XTAL =11.0592MHz, write a program to generate a square wave of 2 KHz frequency on P1.5.

• Q12: Assume that XTAL =11.0592MHz, write a program to generate a square wave of 50Hz frequency on P2.3.
Timer Mode 0

- Mode 0: 13-bit Timer/counter mode
- 0000 ~ 1FFFH
Timer Mode 2

- Mode 2: 8-bit auto reload Timer/counter mode (00 ~ FFH)
- In auto reload, TH is loaded with the initial count and a copy of it is given to TL. This reloading leaves TH unchanged still holding a copy of original values.
- This mode has many applications, including setting the baud rate in serial communication.
Timer Mode 2

• Mode 2 Programming
  – 8 bit ➔ 00 ~FFH
  – TH copy to TL
  – Start SETB TR0, or TR1
  – TL increased ➔ FFH (OV monitoring)
  – TH reload to TL

• TL is reloaded automatically with the original value kept by the TH register. To repeat the process, we must simply clear TF and let it go without any need by the programmer to reload the original value.
Steps to Programming for Timer Mode 2

1. Load TMOD (example for Timer 1)
2. Load TH1
3. Start timer: (SETB TR1)
4. Monitoring the OV: (loop: JNB TF1, loop)
5. CLR TF1
6. Go to step 4. (Note that: NOT Step 2)
Mode 2 Examples

- Q14: Assume that XTAL=11.0592MHz, find (a) the frequency of square wave generated on P1.0 in the following program, and (b) the smallest frequency achievable in this program, and the TH value to do that.

    MOV TMOD, #20H
    MOV TH1, #5
    SETB TR1
    Back: JNB TF1, Back
           CPL P1.0
           CLR TF1
           SJMP Back
Mode 2 Examples

- Q15: Assume that XTAL=11.0592MHz, find the frequency of a square wave generated on P1.0
  
  MOV TMOD, #02H
  MOV TH0, #0
  Again: MOV R5, #250
  ACALL Delay
  CPL P1.0
  SJMP Again
  Delay: SETB TR0
  Back: JNB TF0, Back
  CLR TR0
  CLR TF0
  DJNZ R5, Delay
  RET
Mode 2 Examples

- Q16: Assume that we are programming the timer for mode 2, find the value (in hex) loaded into TH for each of the following cases.
  
  MOV   TH1, #-200
  MOV   TH1, #-3
  MOV   TH0, #-48
  MOV   TH0, #-60
  MOV   TH1, #-12
Mode 2 Examples

- Q17: Find (a) the freq of the square wave generated in the following code and (b) the duty cycle of this wave.
  
  MOV TMOD, #02H  
  MOV TH0, #-150  
  Again: SETB P1.3  
  ACALL Delay  
  ACALL Delay  
  CLR P1.3  
  ACALL Delay  
  SJMP Again  
  Delay: SETB TR0  
  Back: JNB TF0, Back  
  CLR TR0  
  CLR TF0  
  RET
Mode 2 Examples

- Q18: Find (a) the freq of the square wave generated in the following code and (b) the duty cycle of this wave.
  
  ```
  MOV TMOD, #02H
  MOV TH0, #-150
  
  Again: SETB P1.3
  ACALL Delay
  ACALL Delay
  CLR P1.3
  ACALL Delay
  SJMP Again
  
  Delay: SETB TR0
  
  Back: JNB TF0, Back
  CLR TR0
  CLR TF0
  RET
  ```
Counter

- C/\!T=0: As Time, using 8051’s crystal as the source of the frequency.
- C/\!T=1: As counter, a pulse outside of the 8051 that increments the TH and TL registers
- When the C/\!T=1, the counter counts up as pulses are fed from Pins P3.4 (for counter 0) or P3.5 (for counter 1)
Counter Examples

- Q19: Assume that clock pulses are fed into pin T1, write a program for counter 1 in mode 2 to count the pulses and display the state of TL1 count on P2.
**Counter Examples**

- Q20: Assume that 1Hz freq pulse is connected to pin P3.4, write a program to display counter 0 on an LCD. Set the initial value of Th0 to –60.