ASSIGNMENT EIGHT

2) Draw the bit configuration of the register TMOD. What's the function of all TMOD's bits?

<table>
<thead>
<tr>
<th>Timer Mode (TMOD)</th>
<th>Timer 1</th>
<th>Timer 0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Gate</td>
<td>C/T</td>
</tr>
<tr>
<td>Mode 0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Mode 1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Mode 2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Mode 3</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

M1 and Mo .... For assign the timer modes
C/T…. For selecting the source pulses to timer (either internal machine cycle or external clock pulse)

Gate …. For enable/disable the external interrupt to run/stop timer
4) Explain the Mode 1 of AT89C51's timers.

**Mode 1: 16-Bit Timer**

- **OSC** → +12
- **C/T = 0**
- **C/T = 1**
- **T1 PIN**
- **TR1**
- **Gate = 0**
- **INT1 PIN**

**Modes Selection:**
- **M1 M0 Mode**
  - 0 1 1
We must calculate the Initial value of Timer 1 from the overflow value

\[ 65536 - 400 = 65136 \Rightarrow \text{FE70 H} \]
We must assign the timer mode and counter/timer operation.

✓ \( C/\bar{T} \Rightarrow \) logic one to count the external pulses

✓ \( M1 \ M0 \Rightarrow \) “01” to assign Mode 1 (count 65136 pulses)

✓ Gate => disable external interrupt

**TMOD = 50h**

<table>
<thead>
<tr>
<th>Timer Mode (TMOD)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Timer 1</strong></td>
</tr>
<tr>
<td>Gate</td>
</tr>
<tr>
<td>0</td>
</tr>
</tbody>
</table>
Microprocessor Based Systems

**AT89C51**

- Controlling start alarm: P0.0
- Controlling Open/Close Gates: P0.1

Sensor → T1 → AT89C51
ORG 0h
ALARM EQU P0.0
CLOSE_DOORS EQU P0.1

MOV TMOD, #50H ; Timer 1 as counter in MODE 1
MOV TH1, #FEH ; Loads TH1 register
MOV TL1, #70H ; Loads TL1 register
SETB TR1 ; Run the Timer 1
HERE: JNB TF1, HERE ; Counting via (T1) until TF1 is set

SETB ALARM
SETB CLOSE_DOORS
CLR TR1 ; Stops Timer 1
CLR TF1 ; Clears TF1 flag
2) Describe the RS232 and its connector DB-9.

<table>
<thead>
<tr>
<th>Pin number</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CD - Carrier Detect</td>
</tr>
<tr>
<td>2</td>
<td>RXD - Receive Data</td>
</tr>
<tr>
<td>3</td>
<td>TXD - Transmit Data</td>
</tr>
<tr>
<td>4</td>
<td>DTR - Data Terminal Ready</td>
</tr>
<tr>
<td>5</td>
<td>GND - Signal Ground</td>
</tr>
<tr>
<td>6</td>
<td>DSR - Data Set Ready</td>
</tr>
<tr>
<td>7</td>
<td>RTS - Request To Send</td>
</tr>
<tr>
<td>8</td>
<td>CTS - Clear To Send</td>
</tr>
<tr>
<td>9</td>
<td>RI - Ring Indicator</td>
</tr>
</tbody>
</table>
4) Draw block diagram to indicate how AT89C51 can interface with computer through RS232 using MAX233 or MAX232. And why?
13) Draw hardware block diagram and write software assembly program code (without ISR) to design digital circuit for communicating between 2 computers through RS232 based on AT89C51. The first computer for sending data-bytes while the second computer for receiving them with baud rate 9600 (TH1=FDh).
ORG 0H
    SJMP MAIN
ORG 30H
MAIN:
    MOV TMOD, #20H ; timer 1 is auto reload (mode 2)
    MOV TH1, #FDH ; adjust 9600 baud rate
    MOV SCON, #50H ; Ser mode1 > 8-bit, 1 start, 1 stop
    SETB TR1 ; start timer 1

Recv_Chr:
WAIT_1: JNB RI, WAIT_1 ; stay in waiting until complete Receiving
    MOV A, SBUF ; Receiving data via serial port
    CLR RI ; clear Receive interrupt flag

SEND:    MOV SBUF, A ; sending data via serial port
WAIT_2: JNB TI, WAIT_2 ; stay in waiting until complete sending
    CLR TI ; clear transmit interrupt flag

SJMP Recev_Chr ; ANOTHER Chr.
Your Task

11) Draw block diagram and write assembly code (without ISR) for sending either words “Yes” or "No" from AT89C51 to computer via serial port with baud rate 9600 (TH1 = FDH) if P1.0 or P1.1 respectively equal logic zero.