CMR INSTITUTE OF TECHNOLOGY

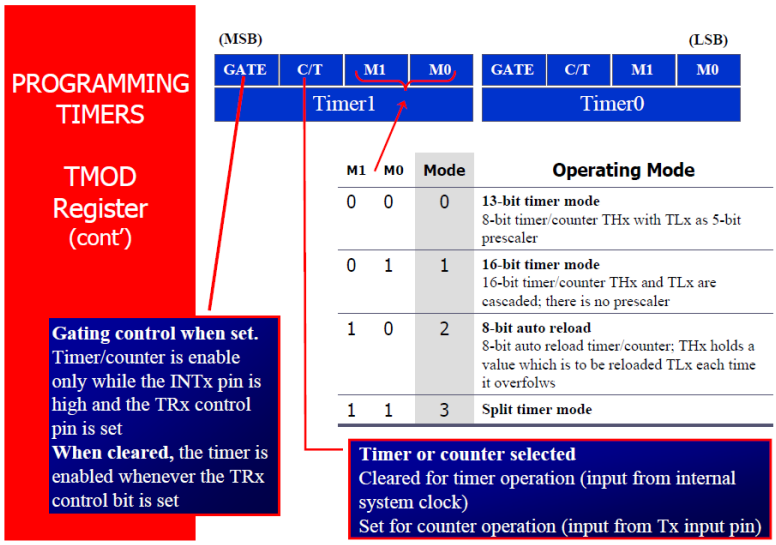
USN | | | | | | | | | | |

Internal Assesment Test - II

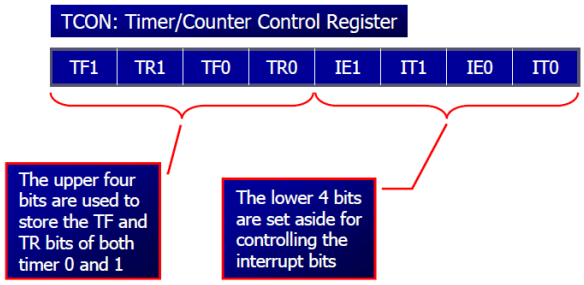
Sub:	Microcontroller							Code:	17EE52
Date:	12/10/2019	Duration:	90 mins	Max Marks:	50	Sem:	5	Branch:	EEE
Answer FIVE Questions Out of the Choices									

SOLUTION

1. Explain TCON and TMOD registers of 8051, with appropriate diagrams.



- TCON (timer control) register is an 8-bit register



2. Write an 8051 C program to toggle all the bits of P1, P2 and P0 continuously with a 250 ns delay. Use sfr keyword to declare port addresses and timer 0 in mode 2.

2. Given 250 ns delay.

(i) Divide delay given by 1.085 μs.

$$\frac{250 \times 10^{-9}}{1.085 \times 10^{-6}} = 0.230$$

(ii) Subtract above answer from 65536/256.

65536 − 0.230 = 65535.77

256 − 0.230 = 255.77 = FF H

(iii) Load TH0 = FF H

```
#include <reg51.h>
sfr Port0 = 0x80;
sfr Port1 = 0x90;
sfr Port2 = 0xA0;
void delay_sub(unsigned char);
void main()
{
 while(1)
 {
 Port0 = 0x55 ; Port1 = 0x55 ; Port2 = 0x55;
 delay_sub(255);
 Port0 = 0xAA ; Port1 = 0xAA ; Port2 = 0xAA ;
 delay_sub(255);
 }
 delay_sub(unsigned char time1)
 {
  TMOD = 0x02; TH0 = time1; TR0 = 1; while(TF0==1); TF0=0;
  TR0=0; }
```

3. a. Explain with an example, bit-wise logic operators in 8051 C.

- Bit-wise operators
  - AND (&), OR (|), EX-OR (^), Inverter (~), Shift Right (>>), and Shift Left (<<)
    - These operators are widely used in software engineering for embedded systems and control

Bit-wise Logic Operators for C

		AND	OR	EX-OR	Inverter
A	B	A&B	A|B	A^B	~B
0	0	0	0	0	1
0	1	0	1	1	0
1	0	0	1	1	
1	1	1	1	0	

b. Explain different data types in 8051 C.

- Unsigned char
- Signed char
- Unsigned int
- Signed int
- Sbit (single bit)
- Bit and sfr

Brief explanation on each one of them.

4. Differentiate between a counter and timer. Explain the timer modes of operation in 8051.

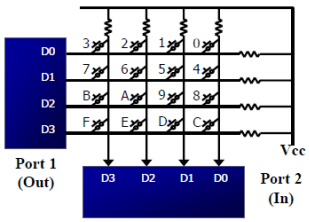
(LSB)

C/T	M1	M0	GATE	C/T	M1	M0
Timer1			Timer0			

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

Timer or counter selected
Cleared for timer operation (input from internal system clock)
Set for counter operation (input from Tx input pin)

5. Show a simple keyboard interface with a port of 8051 and explain its operation.



1. To make sure that the preceding key has been released, 0s are output to all rows at once, and the columns are read and checked repeatedly until all the columns are high
   - When all columns are found to be high, the program waits for a short amount of time before it goes to the next stage of waiting for a key to be pressed
2. To see if any key is pressed, the columns are scanned over and over in an infinite loop until one of them has a 0 on it
   - Remember that the output latches connected to rows still have their initial zeros (provided in stage 1), making them grounded
   - After the key press detection, it waits 20 ms for the bounce and then scans the columns again
     (a) it ensures that the first key press detection was not an erroneous one due a spike noise
     (b) the key press. If after the 20-ms delay the key is still pressed, it goes back into the loop to detect a real key press
3. To detect which row key press belongs to, it grounds one row at a time, reading the columns each time
   - If it finds that all columns are high, this means that the key press cannot belong to that row
     - Therefore, it grounds the next row and continues until it finds the row the key press belongs to
   - Upon finding the row that the key press belongs to, it sets up the starting address for the look-up table holding the scan codes (or ASCII) for that row
4. To identify the key press, it rotates the column bits, one bit at a time, into the carry flag and checks to see if it is low
   - Upon finding the zero, it pulls out the ASCII code for that key from the look-up table
   - otherwise, it increments the pointer to point to the next element of the look-up table

6. Interface an LCD display unit to 8051 and write an ALP to display the message ‘DONE’.

```
        ORG   0
        MOV   DPTR,#MYCOM
C1:     CLR   A
        MOVC  A,@A+DPTR
        ACALL COMNWRT  ;call command subroutine
        ACALL DELAY    ;give LCD some time
        INC   DPTR
        JZ    SEND_DAT
        SJMP  C1
SEND_DAT:
        MOV   DPTR,#MYDATA
D1:     CLR   A
        MOVC  A,@A+DPTR
        ACALL DATAWRT  ;call command subroutine
        ACALL DELAY    ;give LCD some time
        INC   DPTR
        JZ    AGAIN
        SJMP  D1
AGAIN:  SJMP  AGAIN    ;stay here

COMNWRT:              ;send command to LCD
        MOV   P1,A    ;copy reg A to P1
        CLR   P2.0    ;RS=0 for command
        CLR   P2.1    ;R/W=0 for write
        SETB  P2.2    ;E=1 for high pulse
        ACALL DELAY   ;give LCD some time
        CLR   P2.2    ;E=0 for H-to-L pulse
        RET
DATAWRT:              ;write data to LCD
        MOV   P1,A    ;copy reg A to port 1
        SETB  P2.0    ;RS=1 for data
        CLR   P2.1    ;R/W=0 for write
        SETB  P2.2    ;E=1 for high pulse
        ACALL DELAY   ;give LCD some time
        CLR   P2.2    ;E=0 for H-to-L pulse
        RET
DELAY:  MOV   R3,#250 ;50 or higher for fast CPUs
HERE2:  MOV   R4,#255 ;R4 = 255
HERE:   DJNZ  R4,HERE ;stay until R4 becomes 0
        DJNZ  R3,HERE2
        RET
        ORG   300H
MYCOM:  DB    38H,0EH,01,06,84H,0 ; commands and null
MYDATA: DB    "HELLO",0
        END
```

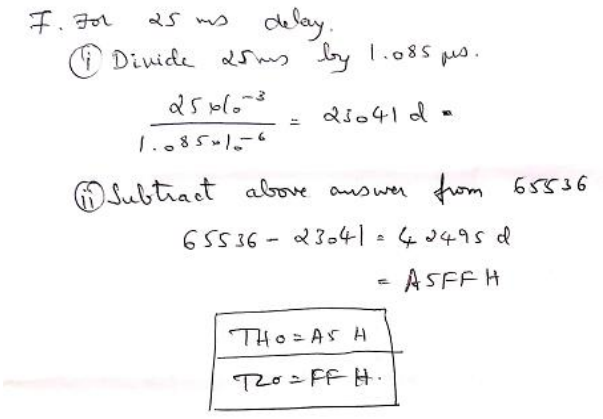
NOTE: Instead of HELLO, DONE should be used.

7. Find the delay generated by timer 0 in the following code. Calculate the delay generated. What count has to be loaded in TL0 and TH0 if delay has to be increased to 25 ms?

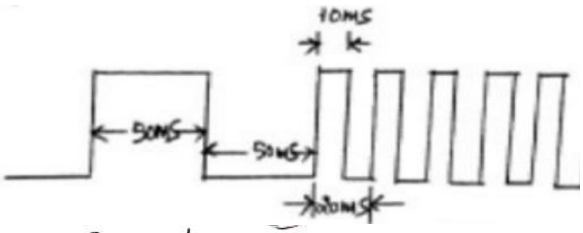
```
        CLR P2.3
HERE:   MOV TMOD, #01
        MOV TL0, #3EH
        MOV TH0, #0B8H
        SETB P2.3
        SETB TR0
AGAIN:  JNB TF0, AGAIN
        CLR TF0
        CLR TR0
        CLR P2.3
```

(a) (FFFFH – B83E + 1) = 47C2H = 18370 in decimal and 18370 × 1.085 us = 19.93145 ms

(b) Since TH – TL = B83EH = 47166 (in decimal) we have 65536 – 47166 = 18370. This means that the timer counts from B38EH to FFFF. This plus Rolling over to 0 goes through a total of 18370 clock cycles, where each clock is 1.085 us in duration. Therefore, we have 18370 × 1.085 us = 19.93145 ms as the width of the pulse.



8. Using P1.5, timer – 1 in mode – 1, write a program to generate the following waveform as shown in the figure below, assume that the system clock is 11.0592 MHz, show the delay calculations. This waveform should be generated continuously.



8. Given two time delays, 50 ms & 10ms

(i) For 50ms

→Divide given no. by 1.085 μs

$$\frac{50ms}{1.085\mu s} = 27027$$

→Subtract above answer from 65536

65536 − 27027 = 38509 d
= 966D H

TH1 = 96 H

TL1 = 6D H

(ii) For 10 ms

→Divide given no. by 1.085 μs

$$\frac{10ms}{1.085\mu s} = 9216$$

→Subtract above answer from 65536

65536 − 9216 = 56320 d
= DC00 H

TH1 = DC H

TL1 = 00H

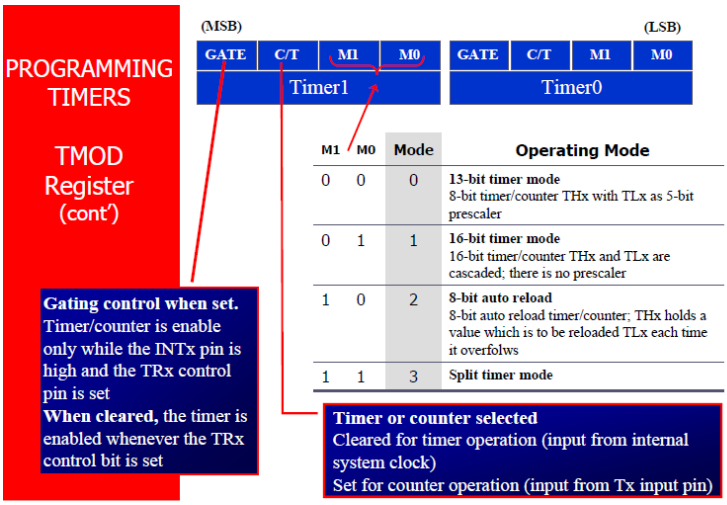
```
        MOV TMOD, #10H
Again:  MOV TH1, #96H
        MOV TL1, #6DH
        SETB P1.5
        SETB TR1
STAY:   JB TF1, STAY
        CLR TR1
        CLR TF1
        CLR P1.5
        MOV R3, #05H
Repeat: MOV TH1, #DCH
        MOV TL1, #00H
        SETB P1.5
        SETB TR1
STAY1:  JB TF1, STAY1
        CLR TR1
        CLR TF1
        CLR P1.5
        DJNZ R3, Repeat  SJMP Again
```

9. With regards to timer of 8051,

(i)explain briefly the difference between the timer and counter operation modes.

(ii) Indicate how to start or stop the timer if GATE control is also used.

(iii) Explain mode-2 operation.



10. Write an 8051 C program to toggle all bits of P3 continuously every 500 ms. Use Timer 1 in mode 1 to create delay.

Solution:

```
//tested for DS89C420, XTAL = 11.0592 MHz
#include <reg51.h>
void T1M1Delay(void);
void main(void){
  unsigned char x;
  P2=0x55;
  while (1) {
    P2=~P2;
    for (x=0;x<20;x++)
        T1M1Delay();
  }
}
void T1M1Delay(void){
  TMOD=0x10;
  TL1=0xFE;
  TH1=0xA5;
  TR1=1;
  while (TF1==0);
  TR1=0;
  TF1=0;
}
```

A5FEH = 42494 in decimal
65536 – 42494 = 23042
23042 × 1.085 µs = 25 ms and
20 × 25 ms = 500 ms