VIT
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF ELECTRONICS ENGINEERING (SENSE)

Continuous Assessment Test – II

Winter Semester -2023-24

Programme Name & Branch : B.Tech ECE

Course Name & Code : BECE204L- Microprocessors and Microcontrollers

Slot: B1 Answer all the questions Exam Duration:50 Minutes

<table>
<tr><th>S.No</th><th>Question</th><th>Marks</th><th>CO</th><th>BL</th></tr>
<tr><td>1</td><td>Assume a switch is connected to port P2.0, LED is connected to P1.1 and buzzer is connected to P2.2 of 8051 microcontroller. Write a program to perform the following operations
<table>
<tr><th>Input Switch(P2.0)</th><th>Output</th></tr>
<tr><td>0</td><td>Buzzer is ON for 15ms</td></tr>
<tr><td>1</td><td>LED is ON for 20 ms</td></tr>
</table></td><td>10</td><td>CO4</td><td>L2</td></tr>
<tr><td>2</td><td>Pin P1.0 of 8051 is connected to a switch. If the switch is closed (logic 0), then write an 8051 assembly language program to transmit “CLOSED” serially at the baud rate 19200. If the switch is open (logic 1), then serially transmit “OPEN” at the baud rate 9600. Assume “CLOSED” is stored in ROM location 200H onwards and “OPEN” in 300H location onwards.</td><td>10</td><td>CO4</td><td>L3</td></tr>
<tr><td>3</td><td>Write an assembly program using interrupt of 8051 microcontroller to simultaneously create 7 KHz and 500 Hz square waves on port pin P1.7 and P1.6.</td><td>10</td><td>CO4</td><td>L3</td></tr>
<tr><td>4.</td><td>8051 microcontroller is connected to water level sensor through ADC0808 which gives 5V when the water tank is 100% full. Develop an assembly program with suitable interfacing diagram to interface ADC0808 with 8051 to compare the water level stored in accumulator(A) with threshold value “E6H”. If A>E6H send a message ‘HIGH LEVEL’ through serial port at a baud rate of 2400. Assume ‘HIGH LEVEL’ is stored in ROM location 500H onwards.</td><td>10</td><td>CO4</td><td>L2</td></tr>
</table>

5.	Write an 8051 assembly program to interface 16x2 LCD display with suitable diagram. Assume the string "HELLO" stored in ROM location from 300H onwards. A switch is connected to port pin P2.7. If the switch input is 1, display the string in line 1 otherwise display the string in reverse order in line 2 as shown below. If switch P2.7=1 display If switch P2.7=0 display	10	CO4	L3

						H	**E**	**L**	**L**	**O**					

						O	**L**	**L**	**E**	**H**					

CAT-2 B1 Key

(2)

```
    SETB P1.0
    JNB P1.0 OPEN
XX: MOV DPTR, #200H
    MOV TMOD, #20H
    MOV TH1
    MOV A, PCON
    SETB ACC.7
    MOV PCON, A
    MOV TMOD, #20H
    MOV TH1, #-3
    MOV SCON, #50H
    SETB TR1
    MOV R1, #6
Again: CLR A
    MOV C A, @A+DPTR
    MOV SBUF, A
Here: JNB TI, Here
    CLR TI
    INC DPTR
    DJNZ R1, Again
    JMP XX
OPEN: MOV DPTR, #300H
    MOV TMOD, #20H
    MOV TH1, #-3
    MOV SCON, #50H
    SETB TR1
    MOV R2, #4
Again1: CLR
    MOVC A, @A+DPTR
    MOV SBUF, A
Here1: JNB TI, Here1
```

CLR TI

INC DPTR

DJNZ R2, OPEN

JJmp XX.

(3) $f = 5kHz$ I.V = FC66 TIM1

$f = 7kHz$ Count = 66 TOM2

MOV TMOD 12H

MOV TH0, #-66

MOV TH1, # FC

MOV TL1, # 66

SETB TR1 → MOV IE, #8'H

Here. JJmp Here

ORG 000BH

CPL P1.7

RETI

ORG 001BH

CPL P1.6

RETI

(4) MOV P1, #0FFH

BACK: CLR P2.6

SETB P2.6

Here. JB P2.7, Here

CLR P2.5

MOV A, P1

CJNE A, #0E6H, NE

L1 ACALL DISPLAY

JJmp Back

NE: INC L1

JJmp Back

```
SETB P2.5
~~JMP BACK~~ ~~MOV~~ MOV DPTR, #500H
DISPLAY: MOV TMOD, #20H
         MOV TH1, #-3
         MOV SCON, #50H
         SETB TR1
         MOV R1, #10
Again1:  CLR A
         MOVC A, @A+DPTR
         MOV SBUF, A
Here1:   JNB TI, Here1
         CLR TI
         INC DPTR
         DJNZ R1, Again1
~~STOP~~
         RET
```

Date: / /

① delay = 15ms

Count = $\frac{15ms}{1.085\mu}$ = C9FF (Butt) TIM1

delay = 80ms ⇒ Count : B7FF (LED) TOM1

```
        SETB P2.0
Again:  JNB P2.0, BUTT
DELAY:  MOV TMOD, #01H
        MOV TH0, #B7H
        MOV TL0, #0FFH
        SETB TR0
BACK:   JNB TF0, BACK
        CLR TR0
        CLR TF0
        RET

BUTT:   SETB P2.2
        ACALL DELAY1
        CLR P2.2
        JMP Again

        SETB P1.1
        ACALL DELAY
        CLR P1.1
        JMP Again

DELAY1: MOV TMOD, #01H
        MOV TH1, #0C9H
        MOV TL1, #0FFH
        SETB TR1
BACK1:  JNB TF1, BACK1
        CLR TR1
        CLR TF1
        RET
```