



VIT

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

REG. NO.:

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
CONTINUOUS ASSESSMENT TEST - II
WINTER SEMESTER 2024-2025**

SLOT: B1+TB1

Programme Name & Branch : B.Tech CSE
Course Code and Course Name : BECE204L- Microprocessors and Microcontrollers
Faculty Name(s) : Gerardine Immaculate Mary (Course Co-ordinator)
Class Number(s) : VL2024250504038/4040/4042/4045/4047/4049/4051/5054/
 4057/4059/4061/4063/4065/4068/4071/4073/4075/4077/4078
 4080/4082/4084/4086/4088/4191/4199
Date of Examination : 17-03-2025
Exam Duration : 90 minutes **Maximum Marks: 50**

General instruction(s):

- Answer All Questions
- M - Max mark; CO – Course Outcome; BL – Blooms Taxonomy Level (1 – Remember, 2 – Understand, 3 – Apply, 4 – Analyse, 5 – Evaluate, 6 – Create)
- Course Outcome
CO4-Deploy the implementation of various peripherals such as general purpose input/output, timers, serial communication, LCD, keypad and ADC with 8051 microcontroller
- Comment the program instructions, explain the format of Special Function Registers (SFRs) used, and provide any necessary calculations.

Q. No	Question	M	CO	BL
1.	Develop an 8051 Assembly Language Program (ALP) for the following scenario. The port P1 receives the data from the user. Based on the nature of the received data do the following: <ul style="list-style-type: none"> • If the received data at P1 is EVEN then send it to P2 by inverting bits that are in the even position. • If the received data at P1 is ODD then send it to P3 by inverting bits that are in the odd position. 	10	4	3
2.	Develop an 8051 ALP to interface a capacitive touch sensor with Port P3.4 and use Timer 0 to detect touch events. When the sensor is touched, increment the count on Port P1 and display the touch count on Port P2 every 1 second using Timer 1.	10	4	3
3.	Develop an 8051 ALP to receive data via the serial port at a baud rate of 9600. The received data should be buffered in Port P1, and the program should send an acknowledgment message ("Data Received") back to the sender using interrupt.	10	4	3
4.	Develop an 8051 ALP to interface 16x2 LCD display with suitable diagram. Assume the string "VIT" is stored in ROM location from 200H onwards and the string "Vellore" is stored in ROM location from 300H onwards. A switch is connected to port pin P0.4. If the switch input is 1, display the string "VIT" in line 1 of the LCD display otherwise display the string "Vellore" in line 2 of the LCD display.	10	4	3
5.	An 8051 microcontroller is connected to linear temperature sensor through ADC0804 which gives 5V when the temperature of the water in boiler at industry is 100°C. Develop an ALP with suitable interfacing diagram to interface ADC0804 with 8051 and compare the value of temperature stored in Accumulator (A) with threshold value T=50°C. If A> 50°C then turn ON buzzer connected to port P1.3 to alarm for 60 ms using timer 0.	10	4	3

Q1: Develop an 8051 Assembly Language Program (ALP) for the following scenario. The port P1 receives the data from the user. Based on the nature of the received data do the following:

- If the received data at P1 is EVEN then send it to P2 by inverting bits that are in the even position.
- If the received data at P1 is ODD then send it to P3 by inverting bits that are in the odd position.

```

ORG 0000H
MOV P1, #0FFH } to make a Port I/P Port
MOV A, P1 }
MOV R0, A } Putting I/P data in a register
ANL A, #01H } Logic: for any odd no. the LSB
                should be 1. So we are
                checking that bit here.

JNZ odd

MOV A, R0 } Putting I/P data again in R0
MOV B, #55H } this will invert the even positions
XRL A, B } of the I/P data
MOV P2, A } for O/P if incoming data is even
SJMP back

odd: MOV A, R0 } Putting I/P data in R0 again
MOV B, #0AAH } This will invert the odd positions of
XRL A, B } the I/P data
MOV P3, A } Putting odd data to O/P.

back: SJMP back

END

```

Q 2: Develop an 8051 ALP to interface a capacitive touch sensor with Port P3.4 and use Timer 0 to detect touch events. When the sensor is touched, increment the count on Port P1 and display the touch count on Port P2 every 1 second using Timer 1.

Mode 1 can generate maximum 71.1 msec. delay.
 $65536 \times 1.085 \mu\text{sec} = 71.1 \text{ msec.}$

to generate 1 sec. delay.

$$\frac{1 \text{ Sec.}}{71.1 \mu\text{sec.}} = 14.06 \approx 14 \text{ times a loop has to run. (14 overflow)}$$

Starting point of the timer

$$\frac{1 \text{ Sec}}{1.085 \mu\text{sec}} = 921659 \text{ (total No.)}$$

$$921659 - (14 \times 65536)$$

$$= 4155 \text{ (Extra)} = \underline{103B} \text{ H}$$

time mode 1 \Rightarrow 16 bit timer will be loaded with initial value 103B.

so 1 time it will run from 103B — FFFF(+1)
 and next 14 time it will run from 0000 — FFFF(+1)

```

ORG 0000H
SETB P3.4      → ZIP Pin
mov P1, #00H
mov P2, #00H
mov TMOD, #11H
mov R0, #14
mov TH1, #10H
mov TL0, #3BH
SETB TR1      → timer starting

HERE: JB P3.4, Touch } monitoring
      SJMP HERE      } the touch

Touch: INC P1
      mov A, P1
      mov P2, A
      lcall DELAY } Updating the
      SJMP HERE   } Display
                  } for 1 sec.
                  } Delay

L1: mov TH1, #00H
    mov TL1, #00H
    SETB TR1

DELAY:
L2: JNB TF1, L2
    CLR TR1
    CLR TF1
    DJNZ R0, L1
    RET

END
    
```


Q4: Develop an 8051 ALP to interface 16x2 LCD display with suitable diagram. Assume the string "VIT" is stored in ROM location from 200H onwards and the string "Vellore" is stored in ROM location from 300H onwards. A switch is connected to port pin P0.4. If the switch input is 1, display the string "VIT" in line 1 of the LCD display otherwise display the string "Vellore" in line 2 of the LCD display.

Sol. 4.

```

    let P1.0 - P1.7 Connected to LCD Pins D0-D7
    P2.0 = RS   P2.1 = R/W   P2.2 = E

    ORG 0000H
    ACALL SET_LCD

    HERE :   JNB P0.4, OFF [Checking the Condition of Switch]
            MOV R0, #3     [Size of VIT]
            MOV DPTR, #0200H [Location of VIT]
            MOV A, #80H
            ACALL COMN_WRT
            ACALL DELAY

    LOOP1:  CLR A
            MOV A, @A+DPTR
            INC DPTR
            ACALL DATA_WRT
            ACALL DELAY
            DJNZ R0, LOOP1
            SJMP HERE

    OFF :   MOV R0, #7     [Size of 'VELLORE']
            MOV DPTR, #300H [Location of 'VELLORE']
            MOV A, #C0H
            ACALL COMN_WRT
            ACALL DELAY

    LOOP2:  CLR A
            MOV A, @A+DPTR
            INC DPTR
            ACALL DATA_WRT
            ACALL DELAY
            DJNZ R0, LOOP2
            SJMP HERE
  
```

Sol. 4 Cont....

```
SET_LCD:  MOV A, #38H
          ACALL COMNWR1
          ACALL DELAY
          MOV A, #06H
          ACALL COMNWR1
          ACALL DELAY
          MOV A, #06H
          ACALL COMNWR1
          ACALL DELAY
          RET
```

```
COMNWR1:  MOV P1, A
          CLR P2.0
          CLR P2.1
          SETB P2.2
          ACALL DELAY
          CLR P2.2
          RET
```

```
DATAWR1: MOV P1, A
          SETB P2.0;
          CLR P2.1
          SETB P2.2
          ACALL DELAY
          CLR P2.2
          RET
```

```
DELAY:   MOV R4, #255
          H: DJNZ R4, H
          RET
          END
```

```
ORG 200H
DB "VIT"
ORG 300H
DB "VELLORE"
```

Q5: An 8051 microcontroller is connected to linear temperature sensor through ADC0804 which gives 5V when the temperature of the water in boiler at industry is 100°C. Develop an ALP with suitable interfacing diagram to interface ADC0804 with 8051 and compare the value of temperature stored in Accumulator (A) with threshold value $T=50^\circ\text{C}$. If $A > 50^\circ\text{C}$ then turn ON buzzer connected to port P1.3 to alarm for 60 ms using timer 0.

Sol. 5 \Rightarrow $100^\circ\text{C} \rightarrow 5\text{V}$ } 255 Range of ADC
 $50^\circ\text{C} \rightarrow 2.5\text{V}$ } 128 (Half of 255)

Logic: - If after ADC Conversion the generated no. is ≤ 128 buzzer is off. and if no. is > 128 buzzer is on.

So we will add 127 in it.

If after Conversion the generated no. is more than 128 we will add 127 in it and carry will be generated.

for Relay generation. $\frac{60 \times 10^{-3}}{1.085 \times 10^{-6}} = 55299 \approx 55300$

Timer will start = (27 FC)H

Program

Let P2.6 = WR P2.7 = INTK P2.5 = RD CS = 0
 & P1.0 - P1.7 = D0 - D7

ORG 0000H

MOV R0, #127;

MOV TMOD, #01H; [timer 0 mode 1]

MOV TH0, #27H } timer initialization

MOV TLO, #0FCH }

BACK: CLR P2.6; [WR=0]

SETB P2.6; [start conversion]

HERE: JB P2.7, HERE

CLR P2.5

MOV A, P1 [Converted data moved to Acc.]

ADD A, R0 [Implementing the logic]

JNC, NO-ALARM

SETB P1.3

ACALL DELAY

CLR P1.3

NO-ALARM: SETB P2.5

SJMP BACK

DELAY: SETB TR0 [timer start]

HERE: JNB TFO, HERE

CLR TR0

CLR TFO

MOV TH0, #27H

MOV TLO, #0FCH

RET

END