School of Electronics Engineering (SENSE)

Fall Semester 2024-25

Continuous Assessment Test – II

SLOT: G2+TG2

Programme Name & Branch: B. Tech & CSE

Course Name & Code: Microprocessor & Microcontrollers (BECE204L)

Class Number (s):~~VL2024250104321, VL2024250104315, VL2024250104323, VL2024250104343, VL2024250104331, VL2024250104318, VL2024250104334, VL2024250104328, VL2024250104326~~

Faculty Name (s): Debashish Dash~~, A D D Dwivedi,~~ G Ragunath, Vikas Vijayavargiya, Henridass, Abhishek Tripathi, Naushad H Laskar, Vishal Gupta

Exam Duration: 90 Min. **Maximum Marks: 50**

General instruction(s):

Specify if any printed material may be permitted

Any other specific instruction

Q. No.	Question	Max Marks	CO	BL
1.	Using Timer 0 auto reload mode of the 8051-microcontroller having 12Mhz crystal frequency, write an Assembly Language Program (ALP) to generate a 2 kHz square wave on pin P1.0. Additionally, implement the following features: (i) The program should count how many times the square wave has toggled (both rising and falling edges). After 50 toggles, the frequency of the square wave should automatically double. (ii) After another 50 toggles (total of 100 toggles), the program should stop generating the square wave	10	CO4	BL4
2.	Assume that a switch (SW) is connected to pin P1.7. Write an assembly language program for 8051 to monitor the switch and perform the following: (a) If SW=0, send the message "HELLO" serially with baud rate of 9600. (b) If SW=1, send the message "GOODBYE" serially with baud rate of 4800.	10	CO4	BL5
3.	Develop an assembly language program for the following scenario: Imagine a sensor is associated with pin P3.2 (INT0) to count the number of entries inside the hall and a buzzer is connected to pin P1.5 of 8051. For sensor, you can feed the input by connecting a switch at P3.2. Normally, the switch is high, and buzzer is OFF. Sensor will count the number of incoming persons entering the hall only when it goes low and makes the buzzer ON.	10	CO4	BL5

4.	Write an assembly language program for 8086 to compute the function $y = (X^3+X+1)/2$. Assume the values of X are the 5 initial odd numbers. Also, store the result at 3000H onwards.	10	CO2	BL4

5.	a) The following segment and offset values are provided for an 8086 microprocessor:
	Code Segment: 2A3FH, Data Segment: 3B20H, Instruction Pointer: 0042H, Extra Segment: 4C10H, Stack Segment: 3000H, Base Register: 015AH,
	Destination Index: 00E5H, Source Index: 00D0H, Stack Pointer: 1000H
	Calculate the physical address for the following:
	MOV AX, [CS:IP]
	MOV AX, [DS:BX]
	MOV [ES:DI], AX
	MOVSB
	PUSH BX
	b) Find the errors in the following instructions for 8086 (if so). Please correct these.
	i). MOV DS, 5000H
	ii). ADD [AX], 4000H
	iii). DIV BX, CX
	iv). INC 4500H
	v). CMP BX
		5+5	CO2	BL4

CAT – II Answer Key

Answer 1:

```
ORG 0000H        ; Start of code
MOV TMOD, #02H    ; Set Timer 0 in Mode 2 (Auto-Reload mode)
MOV TH0, #06H     ; Load Timer 0 high byte with 6 for 250 µs delay
MOV TL0, #06H     ; Load Timer 0 low byte with 6 for the first cycle
SETB TR0          ; Start Timer 0
MOV R1, #0        ; Toggle counter (count rising + falling edges)

MAIN_LOOP:
   JNB TF0, MAIN_LOOP   ; Wait for Timer 0 overflow (TF0 flag)
   CLR TF0           ; Clear the Timer 0 overflow flag
   CPL P1.0          ; Toggle P1.0 (square wave toggle)
   INC R1            ; Increment toggle counter

   CJNE R1, #50, CHECK_100_TOGGLES   ; Check if 50 toggles are completed
   ; After 50 toggles, double the frequency (4 kHz square wave)
   MOV TH0, #83H      ; Load Timer 0 with 131 for 125 µs delay (4 kHz)
   MOV TL0, #83H      ; Reload Timer 0

CHECK_100_TOGGLES:
   CJNE R1, #100, MAIN_LOOP  ; If not yet 100 toggles, continue

   ; After 100 toggles, stop the square wave generation
   CLR TR0           ; Stop Timer 0
   CLR P1.0          ; Ensure P1.0 is low
```

```
    SJMP $          ; End of program, stay in infinite loop

END
```

Answer 2:

```
ORG 0000H
MOV TMOD, #20H
SETB P1.7
AGAIN: JB P1.7, ON

MOV TH1, #-3
MOV SCON, #50H
SETB TR1
MOV A, #'H'
ACALL TRANSOFF
MOV A, #'E'
ACALL TRANSOFF
MOV A, #'L'
ACALL TRANSOFF
MOV A, #'L'
ACALL TRANSOFF
MOV A, #'O'
ACALL TRANSOFF
SJMP AGAIN

TRANSOFF: MOV SBUF, A
HEREOFF: JNB TI, HEREOFF
CLR TI
RET
```

```
ON: MOV TH1, #-6
MOV SCON, #50H
SETB TR1
MOV A, #'G'
ACALL TRANSON
MOV A, #'O'
ACALL TRANSON
MOV A, #'O'
ACALL TRANSON
MOV A, #'D'
ACALL TRANSON
MOV A, #'B'
ACALL TRANSON
MOV A, #'Y'
ACALL TRANSON
MOV A, #'E'
ACALL TRANSON
SJMP AGAIN
TRANSON: MOV SBUF, A
HEREON: JNB TI, HEREON
CLR TI
RET
END
```

Answer 3:

```
ORG 0000H
LJMP MAIN
ORG 0003H
```

```
SETB P1.5
MOV R2, #0FFH
LOOP1: DJNZ R2, LOOP1
CLR P1.5
RETI
ORG 0030H
MAIN: MOV IE, #10000001B
HERE: SJMP HERE
END
```

Answer 4:

```
ASSUME CS:CODE, DS:DATA
DATA SEGMENT
NUMLIST DB 01H, 02H, 03H, 04H, 05H, 06H, 07H, 08H, 09H, 0AH
NEXT DB 0FF00H DUP (00H)                    // REMAINING VALUES WILL BE ZEROS
COUNT EQU 0AH
DATA ENDS

CODE SEGMENT
START:
MOV AX, DATA                    ; get the segment address of DATA segment in DS register
MOV DS, AX
MOV CX, COUNT                   ; COUNT is the number of data has to be added
MOV SI, OFFSET NUMLIST          ; get the offset address of NUMLIST
MOV DI, 5000H
```

```
MOV AX, 0000H
MOV BX, 0000H
LP: MOV BL, [SI]
MOV AL, BL
MUL BL
MUL BL
ADD AL, BL
ADD AL, 01H
MOV [DI], AL
INC DI
INC SI
DEC CX
JNZ LP
MOV AH, 00H
INT 21H
CODE ENDS
END START
```

Answer 5:

a:

MOV AX, [CS:IP]

Physical Address = 2A432H

MOV AX, [DS:BX]

Physical Address = 3B35AH

MOV [ES:DI], AX

Physical Address = 4C1E5H

MOVSB

Source Physical Address = 3B2D0H, Destination Physical Address = 4C1E5H

PUSH BX

Physical Address=30000H+0FFEH=30FFEH

b. Summary of Corrections:

i) **MOV DS, 5000H**

- Corrected: MOV AX, 5000H, MOV DS, AX

ii) **ADD [AX], 4000H**

- Corrected: MOV BX, 4000H, ADD [AX], BX

iii) **DIV BX, CX**

- Corrected: MOV AX, BX, DIV CX

iv) **INC 4500H**

- Corrected: MOV SI, 4500H, INC [SI]

v) **CMP BX**

- Corrected: CMP BX, AX or CMP BX, 1000H