REG.NO.:

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

SCHOOL OF ELECTRONICS ENGINEERING
CONTINUOUS ASSESSMENT TEST - 1
WINTER SEMESTER 2025-2026

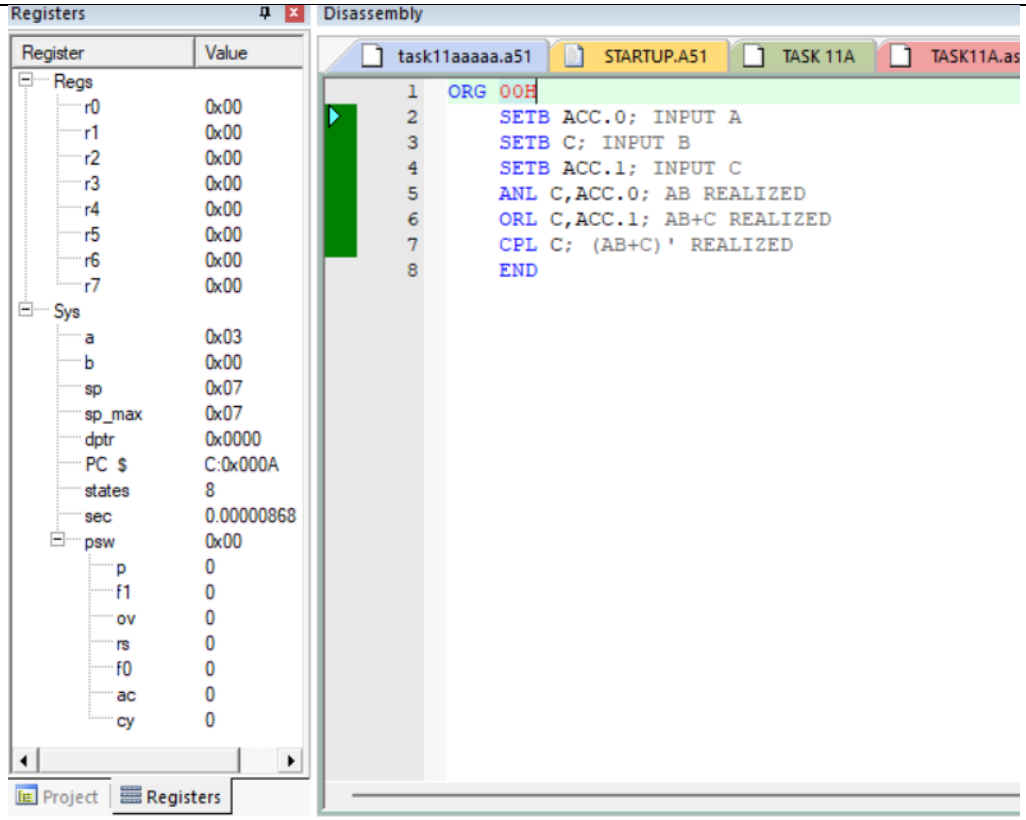
SLOT: C2+TC2

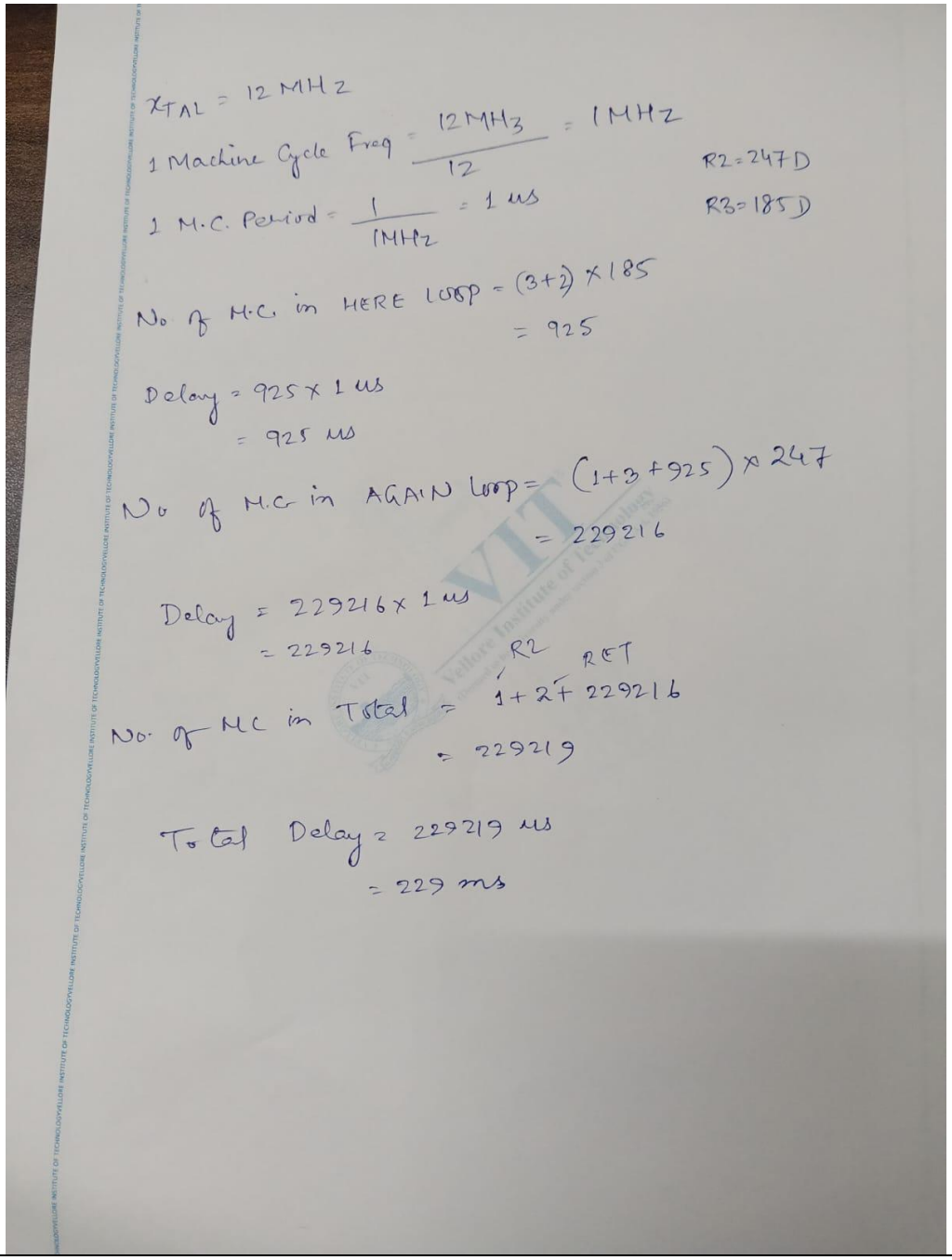
Programme Name & Branch : B.TECH CSE
Course Code and Course Name : BECE204L- MICROPROCESSORS & MICRCONTROLLERS
Faculty Name(s) : DR. VETRIVEERAN RAJAMANI (COURSE COORDINATOR)
Class Number(s) : VL2025260501054/1016/1021/1019/1023/1056/1025/1058/103/1041/1029/1060/1044/1031/1047/1062/1065/1034/1069/1071/1073/1067/1075/1077/1051
Date of Examination : 29.1.2026
Exam Duration : 90 minutes **Maximum Marks: 50**

ANSWER KEY

Q. No	Question	M	CO	BL
1.	a) Explain the usage of branching instructions in 8051 microcontroller and describe the role of the stack in their implementation. Ans: Branching instructions allow the program to change the flow of execution instead of running sequentially. In the 8051 microcontroller, branching is essential for decision-making, looping, and subroutine calls. Types of Branching Instructions 1. **Unconditional Branching** o **SJMP (Short Jump)**: Jumps to a target address within ±128 bytes. o **LJMP (Long Jump)**: Jumps anywhere in the 64 KB program memory. o **AJMP (Absolute Jump)**: Jumps within the same 2 KB page. 2. **Conditional Branching** o **JZ (Jump if Accumulator = 0)** o **JNZ (Jump if Accumulator ≠ 0)** o **JC (Jump if Carry = 1)** o **JNC (Jump if Carry = 0)** o **DJNZ (Decrement and Jump if Not Zero)** These instructions allow decisions based on flags or register values. 2. **Subroutine Calls** o **ACALL (Absolute Call)**: Calls a subroutine within the same 2 KB page. o **LCALL (Long Call)**: Calls a subroutine anywhere in program memory. o **RET / RETI**: Return from subroutine or interrupt. The stack is a reserved area in RAM used for temporary storage. It plays a crucial role in subroutine calls and interrupt handling. How the Stack Works in Branching When a CALL instruction (ACALL/LCALL) is executed: ▪ The return address (the address of the next instruction after CALL) is pushed onto the stack. ▪ The Program Counter (PC) is loaded with the subroutine's starting address. ▪ When a RET instruction is executed: ▪ The return address is popped from the stack. ▪ The PC resumes execution from that address	5	3	2

	b) For the instructions given below, check if they are correct and rectify if they are incorrect. Justify your answers i. MOV PC,@A+PC: INCORRECT AS DATA CANNOT BE COPIED TO ROM & MOVC SHOULD BE USED IN INDEXED ADDRESSING MODE CORRECT: MOVC A,@A+DPTR, MOVC A,@A+PC ii. PUSH R5: INCORRECT AS STACK WORKS ON DIRECT ADDRESSING MODE. CORRECT: PUSH 5 iii. ADD A,@R5: INCORRECT AS REGISTER INDIRECT ADDRESSING MODE WORKS ONLY WITH R0,R1 CORRECT: ADD A,@R0 OR ADD A,@R1 iv. MOV P1,R0: INCORRECT AS DATA TRANSFER TO PORTS HAS TO BE DONE THORUGH ACCUMULATOR AND NOT GPR CORRECT: MOV P1,A v. MUL A,B: INCORRECT SYNTAX FOR MULTIPLICATION CORRECT: MUL AB	5	3	2
2.	a) For the following program, write the output and status of PSW after execution of each line of the program Ans: SETB PSW.3 **PSW=08H** MOV A,#25H **A=25H PSW= 09H** MOV B,#52H **B=52H A=25H PSW= 09H** MUL AB **B=0BH A=DAH PSW=0DH** SETB PSW.4 **B=0BH A=DAH PSW=1DH** RRC **B=0BH A=6DH PSW=1DH** SWAP A **B=0BH A=D6H PSW=1D H** END **B=0BH A= D6H PSW=1D H**	5	3	3
	b) Write an 8051 Assembly Language Program (ALP) to realize $Y = \left(\overline{AB + C}\right)$ Ans:	5	3	3

	 **NOTE: THE ABOVE PROGRAM IS FOR INPUTS A=1,B=1 AND C =1 TO GIVE OUTPUT Y=0. ANY COMBINATION OF A & B CAN BE REALIZED BY THE STUDENTS I.E. THEY CAN USE SETB OR CLR AND ANY BITS OF ACCUMULATOR IN THE ANSWER.**			
3.	Calculate the amount of delay caused by HERE loop, AGAIN loop, and the total delay for the below given subroutine if the system has an 8051 with frequency of 12 MHz. DELAY: MOV R2, # 11110111B (1) AGAIN: MOV R3, #0B9H (1) HERE: NOP (1) NOP (1) NOP (1) DJNZ R3, HERE (2) DJNZ R2, AGAIN (2) RET (2) Note: The No. of Machine Cycles for every instruction is given in brackets ( ).	10	3	3

	Ans:  			
4.	Write an 8051 Assembly Language Program (ALP) to monitor Pin 2 of Port 0 and create a waveform of 40% duty cycle on Pin 0 of Port 2 if Pin 2 of Port 0 is low. Ans:	10	4	3

```
1  ORG 00H
2      SETB P0.2          // Make P0.2 input (switch)
3    AGAIN:    JB  P0.2, HERE    //If switch = 1, skip pulse generation
4          SETB P2.0
5          ACALL DELAY
6          ACALL DELAY
7          ACALL DELAY
8          ACALL DELAY
9          CLR  P2.0
10         ACALL DELAY
11         ACALL DELAY
12         ACALL DELAY
13         ACALL DELAY
14         ACALL DELAY
15         ACALL DELAY
16
17 HERE:  SJMP AGAIN
18 DELAY: MOV R3,#20H
19 LOOP: DJNZ R3,LOOP
20 RET
21
22 END
23
```

```
ARTUP.obj (?C_STARTUP)
 MADE TO UNRESOLVED EXTERNAL

ARTUP.obj (?C_STARTUP)

ata=0 code=532
 0 Error(s), 3 Warning(s).
```

ALTERNATIVE SOLUTION:

```
ORG 00H
    SETB P0.2          // Make P0.2 input (switch)
 AGAIN:    JB  P0.2, HERE    //If switch = 1, skip pulse generation
       SETB P2.0
       ACALL DELAY
       ACALL DELAY
       CLR  P2.0
       ACALL DELAY
       ACALL DELAY
       ACALL DELAY

HERE:  SJMP AGAIN
DELAY: MOV R3,#20H
LOOP: DJNZ R3,LOOP
RET

END
```

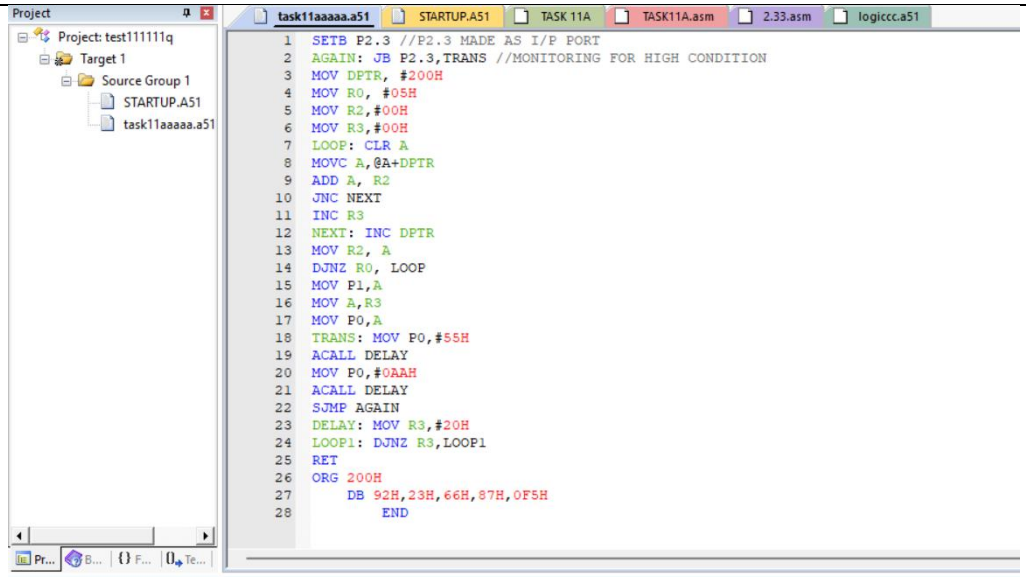
```
ARTUP.obj (?C_STARTUP)
  MADE TO UNRESOLVED EXTERNAL

ARTUP.obj (?C_STARTUP)

ata=0 code=532
  0 Error(s), 3 Warning(s).
```

NOTE: STUDENTS NEED NOT WRITE THE DELAY LOOP. STUDENTS MAY USE P0.1 ALSO AS MENTIONED PIN 2 OF PORT 1 INSTEAD OF P0.2. SO THAT CAN ALSO BE CONSIDERED FOR MARKING

5.	Write an 8051 Assembly Language Program (ALP) to monitor the status of P2.3 and perform the following i. If P2.3 is high, toggle the bits of P0 continuously with a delay ii. If P2.3 is low, add the numbers 92H,23H,66H,87H, F5H stored in ROM locations 200H onwards and send the result to P1, P0 (send Carry to P0). NOTE: You can exclude the delay sub-routine in the program in 5 i. Ans:	10	4	4



OUTPUT: P0=02H; P1=97H

****