INSTITUTE OF COMPUTER TECHNOLOGY

B-TECH COMPUTER SCIENCE ENGINEERING 2025-26

SUBJECT: MICROCONTROLLER & APPLICATION

NAME: Rahul Prajapati

ENRLL. NO: 23162171020

BRANCH: CYBER SECURITY

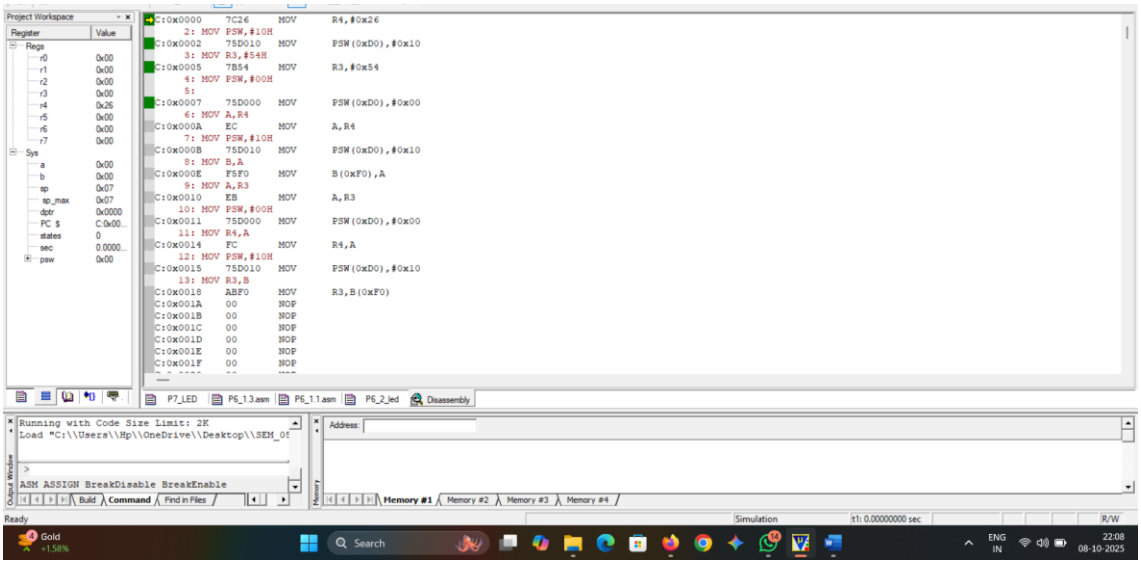
BATCH: 52

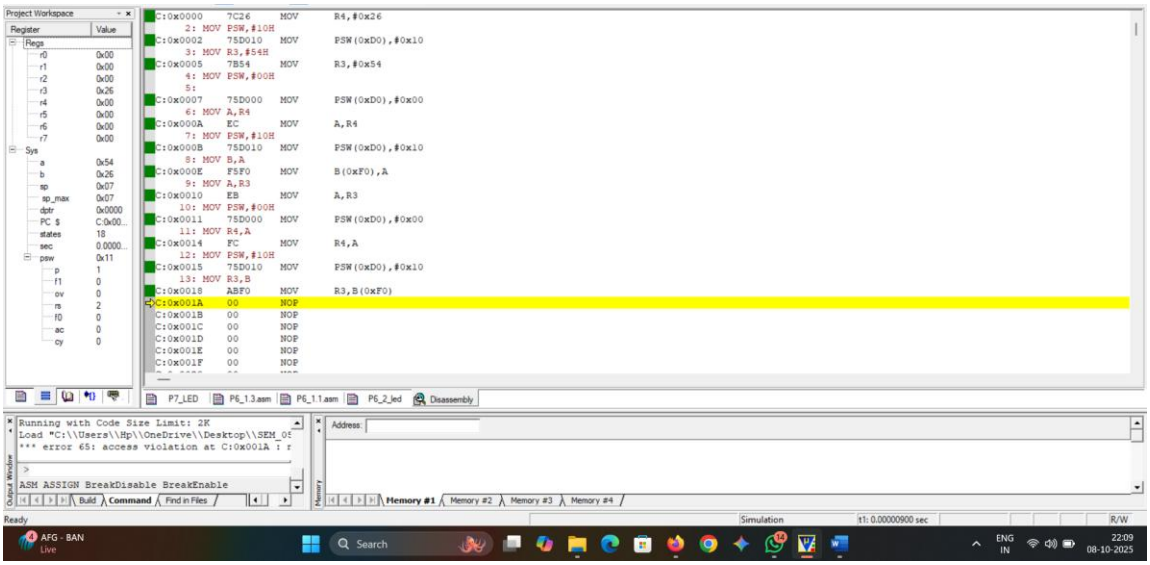
PRACTICAL_6

Aim:- Learning programs based on various Addressing modes of 8051 & learning use of Proteus Hardware Simulator for 8051 application programs.

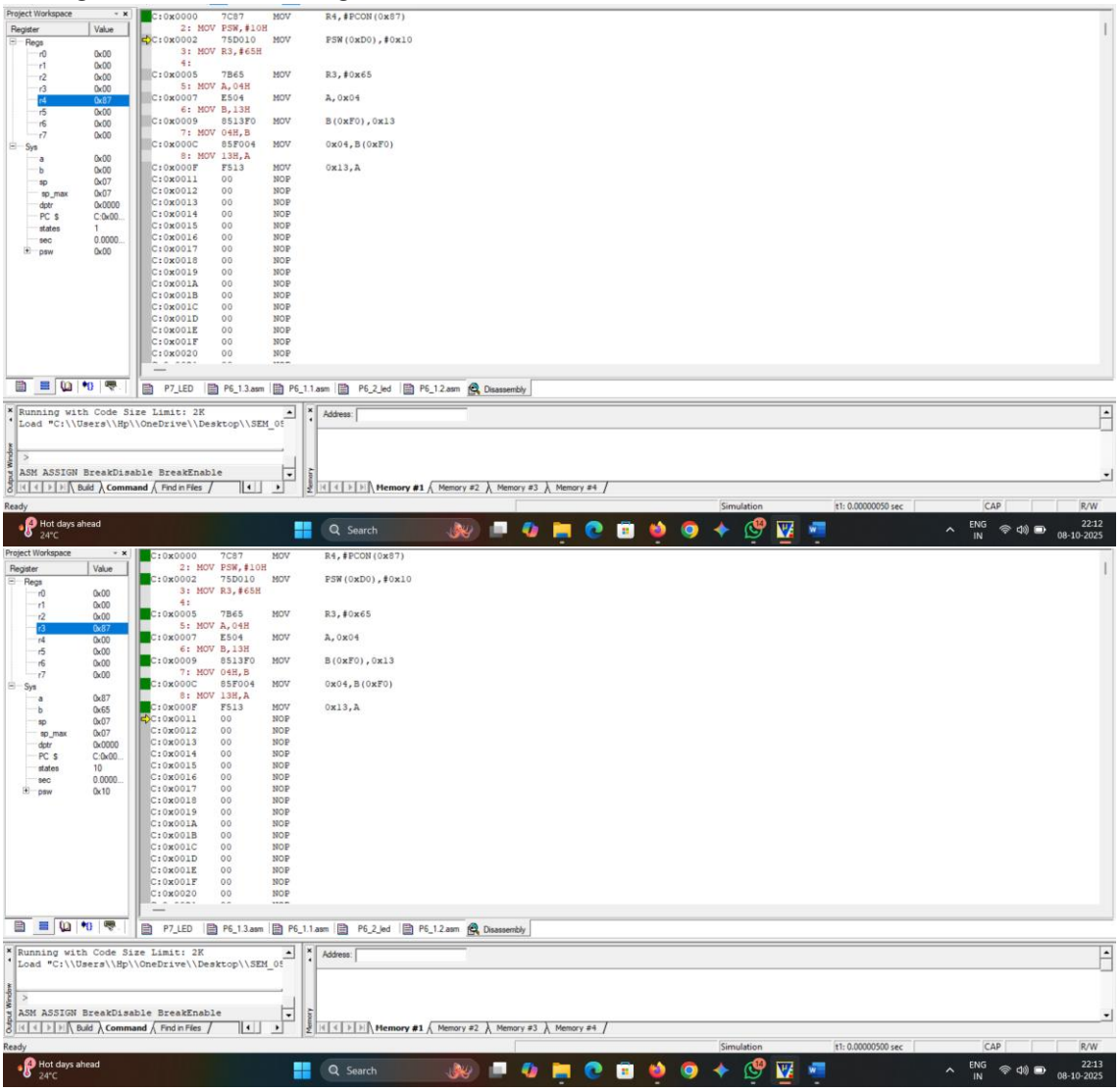
1. Swap the contents of R4 of register bank 0 and R3 of register bank 2 using all possible addressing modes.

   A. Using Register Addressing Mode:

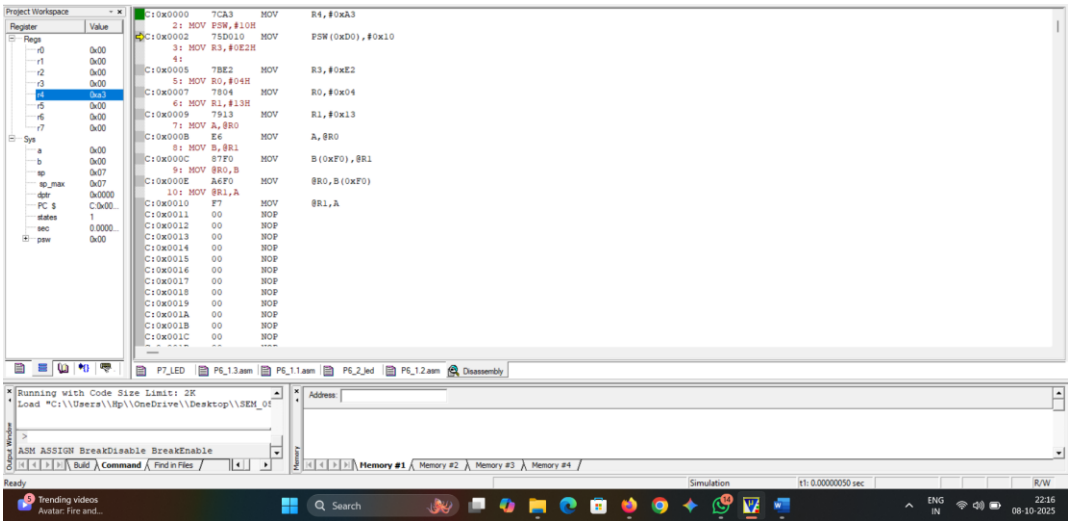


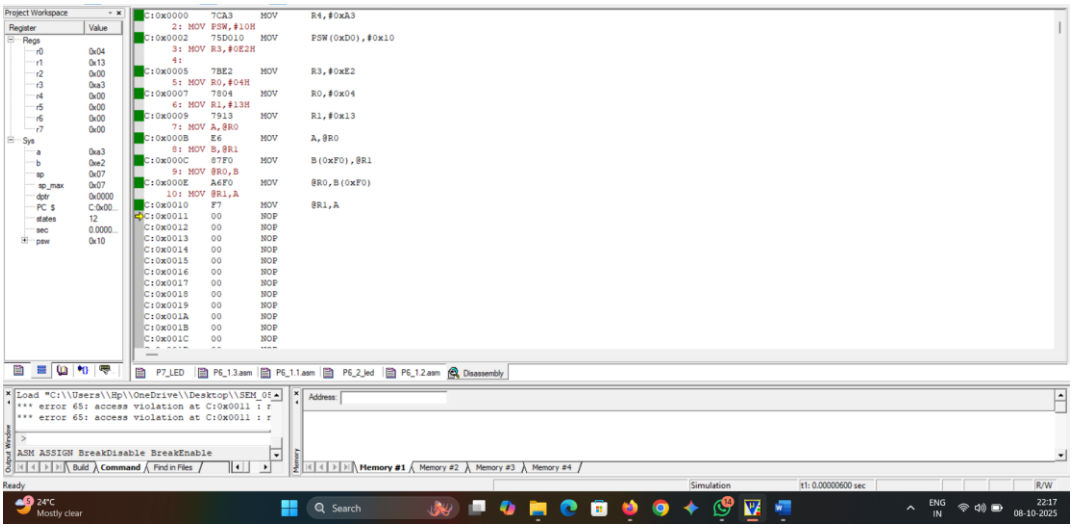


B. Using Direct Addressing Mode:



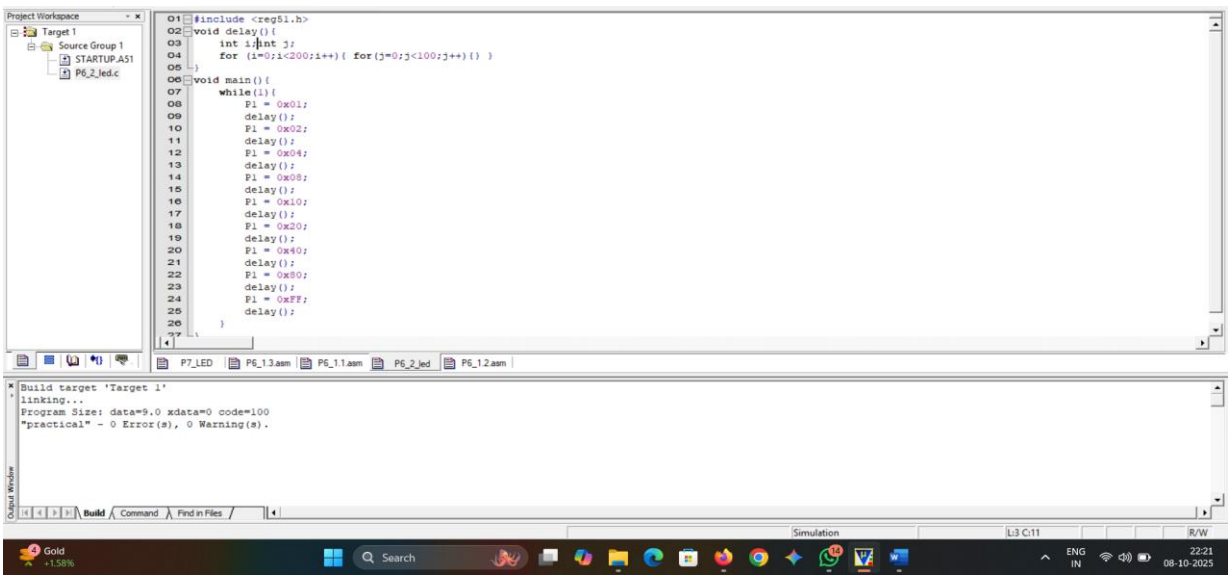
C. Using Indirect Addressing Mode:



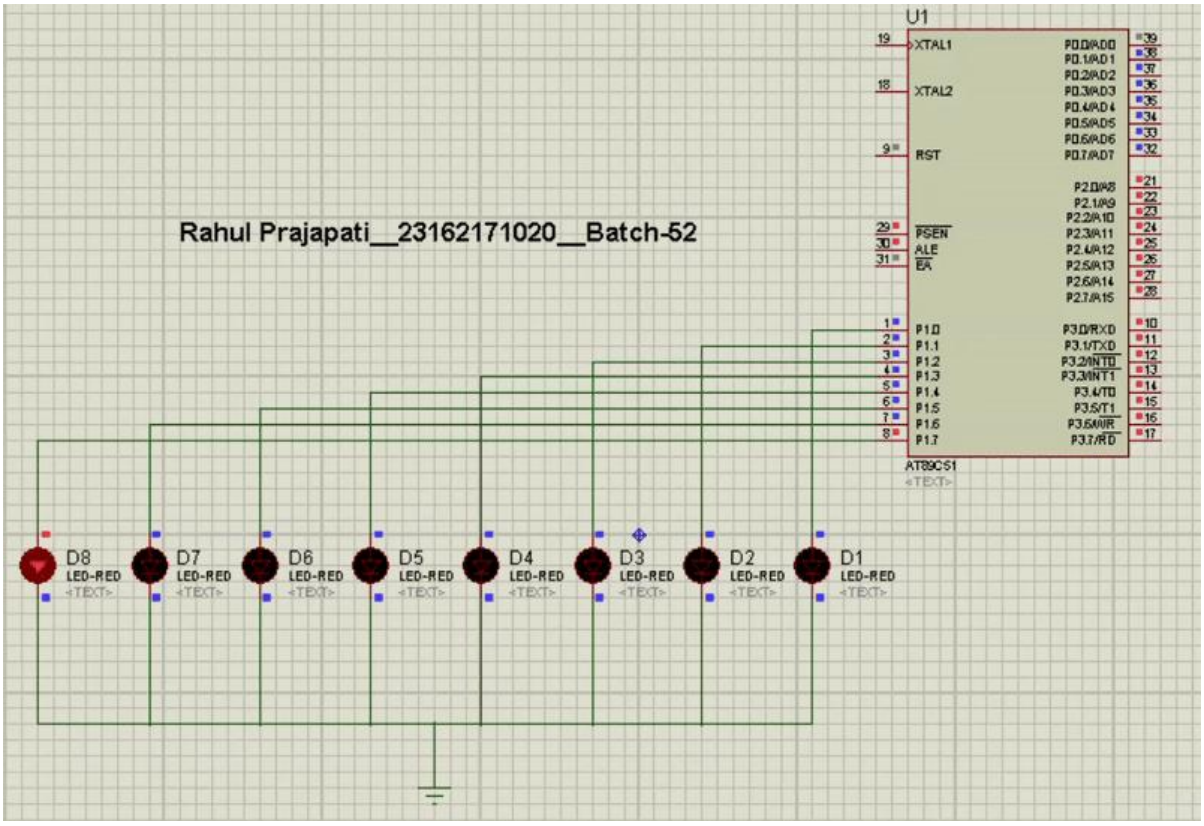


2. Learning to create hex file, loading it into hardware and observe the output on real hardware. Observing the same in the Proteus software.

(a) Write a sample program (using 'c' language) to switch ON the LEDs connected on port P1 one by one (without using bit level instructions) with some delay. After the successful execution in Keil software, create the hex file of it, load the hex file into 8051 IC using Proteus circuit simulator, do the necessary hardware connections and observe the output on it.



➔ SIMULATION: To see animation click on image 👇 .



CONCLUSION:

In this practical, we learned different addressing modes of the 8051 microcontroller by writing assembly programs to swap data between registers using register, direct, and indirect addressing. We also understood the process of creating a HEX file from C code in Keil, loading it into Proteus, and simulating real hardware behavior. This helped us practically understand 8051 programming, data handling, and hardware interfacing concepts.

Rahul Prajapati - 2316217020 - 52

Practical-6

1. Swap the contents of R4 of register bank 0 and R3 of register Bank 2 using all possible addressing modes.

→ A. Using Register Addressing Mode:

```
MOV R4, #26H ; load 26H into R4 of bank0
MOV PSW, #10H ; select reg. bank 2
MOV R3, #54H ; load 54H into R3 of bank 2
MOV PSW, #00H ; select bank0 0

MOV A, R4 ; MOV R4 (Bank0) to accumulator
MOV PSW, #10H ; Swith to bank 2
MOV B, A ; store acc. content 26H in B reg.
MOV A, R3 ; Move R3 (Bank 2) to accumulator.
MOV PSW, #00H ; switch to Bank 0
MOV R4, A ; store acc. content 54H into R4 (Bank0)
MOV PSW, #10H ; Switch to register bank 2.
MOV R3, B ; store B (26H) into R3 (bank 2)
END ; end of program.
```

→ B. Using Direct Addressing mode:

```
MOV R4, #87H ; load 87H into R4 of bank 0
MOV PSW, #10H ; switch to bank 2.
MOV R3, #65H ; load 65H into R3 of bank 2.
mov A, 04H ; move data from 04H to Accumulator
mov B, 13H ; move data from 13H to register B.
mov 04H, B ; store content of B into address 04H
mov 13H, A ; store content of A into address 13H
END ; End of program.
```

→ C. Using indirect Addressing mode:

```
MOV R4, #0A3H ; Load 0A3H into R4 (Bank 0)
Mov PSW, #10H ; Select register bank 2
mov R3, #0E2H ; Load 0E2H into R3 (bank 2).
MOV R0, #04H ; R0 Points to 04H
Mov R1, #13H ; R1 Points to 13H
mov A, @R0 ; move data from 04H to A
mov B, @R1 ; move data from 13H to B
mov @R0, B ; store B into 04H
mov @R1, A ; store A into 13H
END ; End of program.
```