Ex. No : 11 LED BLINKING

AIM

To write a code to blink the LED using MCU 8051 IDE

Procedure

// Write procedure here

Code

ORG 0000H ; Set the origin of the program to address 0x0000

SETB P2.0 ; Set Port 2, Pin 0 (P2.0) high (turn LED on)

BLINK: CPL P2.0 ; Toggle the state of P2.0 (turn LED on/off)

LCALL DELAY ; Call the delay subroutine

SJMP BLINK ; Jump back to the BLINK label (create an infinite loop)

DELAY: MOV R7, #01H ; Load the value 0x01 into register R7 (outer loop counter)

LOOP: MOV R6, #01H ; Load the value 0x01 into register R6 (inner loop counter)

DJNZ R6, $ ; Decrement R6 and jump to the same instruction until R6 is zero (small delay)

DJNZ R7, LOOP; Decrement R7 and jump back to the LOOP label until R7 is zero (larger delay)

RET ; Return from the delay subroutine to the main program

END ; End of the program

Steps to Execute

1. Click on the simulator
2. Open the menu Virtual HW 🡪 Open LED Panel
3. In the LED Panel click on the Tools icon and select Windows Always on top.
4. Turn on the LED
5. Set the value of Port to 2 and Bin to 0
6. Click on the Animate

**Explanation of Comments**:

ORG 0000H: Sets the starting address of the program.

SETB P2.0: Turns the LED on by setting P2.0 high.

BLINK: Label marking the start of the main loop.

CPL P2.0: Toggles the LED state.

LCALL DELAY: Calls the delay subroutine to create a delay.

SJMP BLINK: Jumps back to the BLINK label, creating an infinite loop.

DELAY: Label marking the start of the delay subroutine.

MOV R7, #01H: Loads 0x01 into R7 for the outer loop counter.

LOOP: Label marking the start of the inner loop.

MOV R6, #01H: Loads 0x01 into R6 for the inner loop counter.

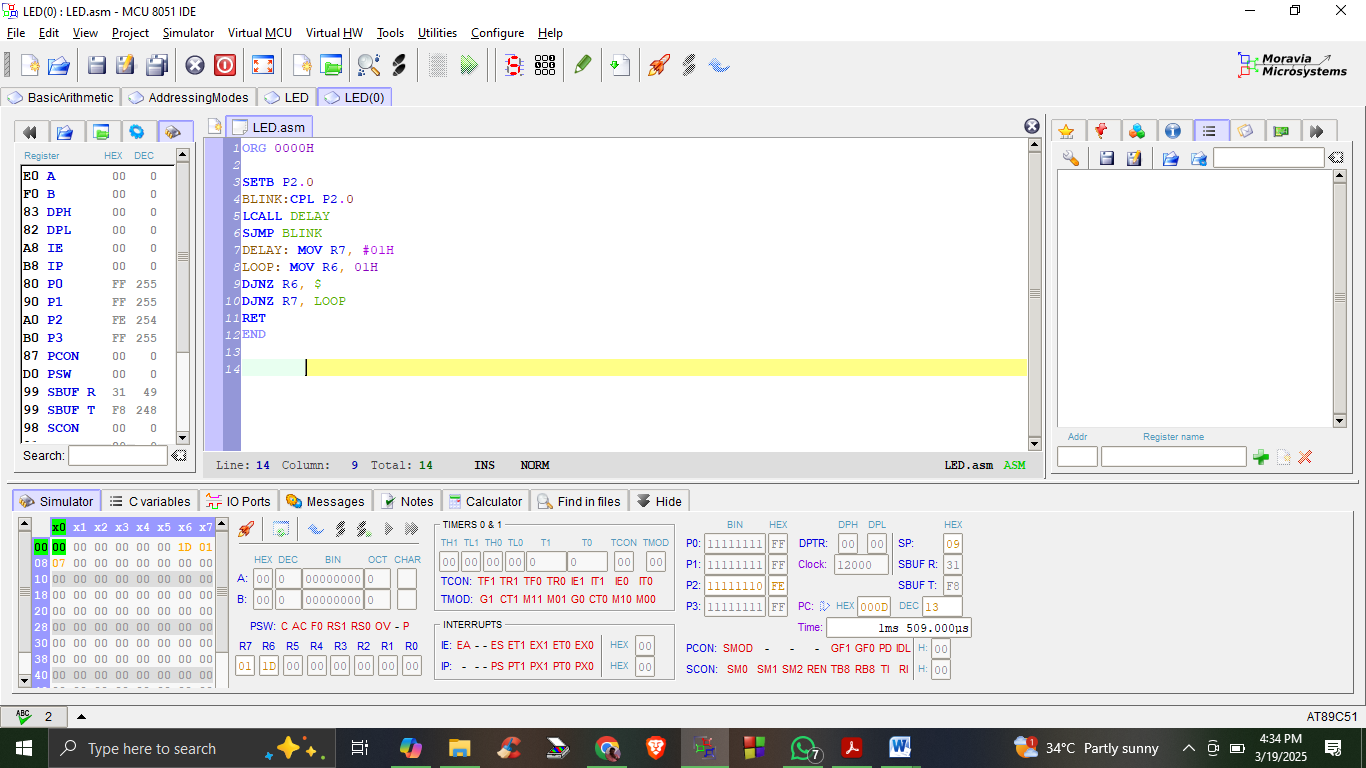
DJNZ R6, $: Decrements R6 and loops until R6 is zero, creating a small delay.

DJNZ R7, LOOP: Decrements R7 and loops back to the inner loop until R7 is zero, creating a larger delay.

RET: Returns from the delay subroutine to the main program.

END: Marks the end of the program.

Output



LED Panel

