Experiment-9

a. Programming for Delay Generation and Effect of CPU Clock

Delay Generation in 8051

Think of the 8051 microcontroller like a kid clapping hands.

The kid claps once every few moments (that's the CPU clock).

If you tell the kid:

"Clap 200 times before moving on" → that's a delay loop.

The faster the kid claps (higher clock speed), the shorter the delay.

The slower the clap (lower clock speed), the longer the delay.

So, a delay program just makes the microcontroller "waste time" by looping again and again before doing the next thing.

Effect of CPU Clock:

CPU Clock = the rhythm of the microcontroller's heartbeat.

If the heartbeat is fast (e.g., 24 MHz), all actions (loops, delays) finish sooner.

If the heartbeat is slow (e.g., 6 MHz), the same loop takes longer.

Same program, different crystal → different delay length.

```
        ORG 0000H
START:  SETB P1.0     ; LED ON
        ACALL DELAY
        CLR  P1.0     ; LED OFF
        ACALL DELAY
        SJMP START    ; repeat forever


; --- Delay subroutine ---
DELAY:  MOV  R3,#200  ; repeat 200 times
LOOP:   DJNZ R3,LOOP  ; loop until R3=0
        RET
        END
```

How to See It in Keil (like a toy oscilloscope)

1. Build & run your program in **Keil Simulator**.

2. Open **I/O Ports → Port 1** → you'll see **P1.0 changing 1/0**.
3. Open **Logic Analyzer**, add PORT1.0 → you'll see a **square wave**.
4. Change the **crystal frequency** (e.g., 6 MHz → 12 MHz) and rebuild:
    - At **12 MHz** the LED blinks faster.
    - At **6 MHz** the LED blinks slower.