**6th Lab - Simulating Binary Counter (8-bit) on LEDs**

**Implementation:**

**Keil:**

Open Keil µVision → Go to Project → New µVision Project → Set a project name and save it → In the Select Device for Target window, search and select AT89C51 → Click OK → When prompted, click No → In Project window, right-click Source Group 1 under Target 1 → Select Add New Item to Group 'Source Group 1' → Create and save a new C file → Write the code below and save:

**Source Code:**

#include <reg51.h>

void Delay(int timer)

{

int i, j;

for (i = 0; i < timer; i++)

{

for (j = 0; j < 1000; j++)

{}

}

}

void main()

{

P2 = 0x00;

while (1)

{

P2++;

Delay(200);

}

}

→ Right-click Source Group 1 → Add Existing Files to Group → Select the saved .c file → Click Add and Close → Right-click Target 1 → Options for Target → Go to Output tab → Check "Create Hex File" → Go to Target tab → Set Xtal (MHz) to 11.0592 → Click OK → Press F7 to build the project → If there are no errors or warnings, your code is okay.

**Proteus:**

Open Proteus → Click New Project → Set project name → Keep clicking Next until Finish → Click Finish → From left sidebar, select Component Mode → Click P → Search and add AT89C51 → Add LED ×8 → From the sidebar, select Terminals Mode → Add Ground

**Component Placement:**

Place AT89C51 MCU on the workspace → Place 8 LEDs (D1–D8) horizontally at the bottom → Place Ground symbol near LED cathode side

**Wiring:**

Connect P2.0–P2.7 (pins 21–28) to anode side of 8 LEDs → Connect cathode side of all LEDs to Ground

**Programming:**

Double-click U1 → Set clock frequency = 11.0592 MHz → Browse and select U1's HEX file → Click OK

**Simulation:**

Click the Play button → LEDs blinking in binary counting pattern → The LED output will go from 00000000 → 00000001 → 00000010 → ... → 11111111 and loop → Save the project

A computer screen shot of a circuit

AI-generated content may be incorrect.**Diagram:**