**Lab:2**

**Interfacing LEDs to 8051and displaying Binary Patterns**

A logo of a university of engineering and technology

Description automatically generated

**MBSD Lab**

**Spring 2023**

**Submitted by:**

**Maaz Habib**

**Reg no:**

**20Pwcse1952**

“On my honour, as a student of University of Engineering and Technology Peshawar, I have neither nor received unauthorized assistance on this academic work”

**Submitted to:**

**Dr. Amaad Khalil**

**Task 1: Implement the assembly code for toggling Port 1 pins and test it on Keil IDE**

ORG 0H ; Start address of the program

MOV P1, #0FFH ; Set all Port 1 pins as high initially

MOV R1, #0 ; Initialize R1 to 0

LOOP:

CPL P1.0 ; Toggle the state of Port 1 pin 0

CPL P1.1 ; Toggle the state of Port 1 pin 1

CPL P1.2 ; Toggle the state of Port 1 pin 2

CPL P1.3 ; Toggle the state of Port 1 pin 3

CPL P1.4 ; Toggle the state of Port 1 pin 4

CPL P1.5 ; Toggle the state of Port 1 pin 5

CPL P1.6 ; Toggle the state of Port 1 pin 6

CPL P1.7 ; Toggle the state of Port 1 pin 7

ACALL DELAY ; Call the delay subroutine

INC R1 ; Increment R1

CJNE R1, #10H, LOOP ; Repeat the loop until R1 = 0x10

END ; End of the program

DELAY:

MOV R2, #0FFH ; Initialize R2 to 0xFF

MOV R3, #0FFH ; Initialize R3 to 0xFF

DELAY\_LOOP:

DJNZ R2, DELAY\_LOOP ; Decrement R2 and repeat the loop until R2 = 0

DJNZ R3, DELAY\_LOOP ; Decrement R3 and repeat the loop until R3 = 0

RET ; Return from the subroutine

To test this code in the Keil IDE, follow these steps:

1. Open the Keil IDE and create a new project for the 8051 microcontroller.
2. Copy the above assembly code and paste it into a new source file in the Keil IDE.
3. Build the project by clicking on the "Build" button or pressing Ctrl + F7.
4. If there are no syntax errors, proceed to the next step.
5. Connect the microcontroller board to your computer and configure the appropriate settings in the Keil IDE for flashing the program onto the microcontroller.
6. Flash the program onto the microcontroller using the "Flash" or "Download" option in the Keil IDE.
7. Once the program is successfully flashed, you can observe the pins of Port 1 toggling at the specified delay in the code.

**Task 2: Implement the C code for toggling Port 2 pins and test it on Keil IDE.**

#include <reg51.h>

void delay() {

unsigned int i, j;

for (i = 0; i < 500; i++) {

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

}

}

}

void main() {

unsigned char i;

P2 = 0xFF; // Set all Port 2 pins as high initially

while (1) {

for (i = 0; i < 8; i++) {

P2 ^= (1 << i); // Toggle the state of Port 2 pin i

delay(); // Delay for some time

}

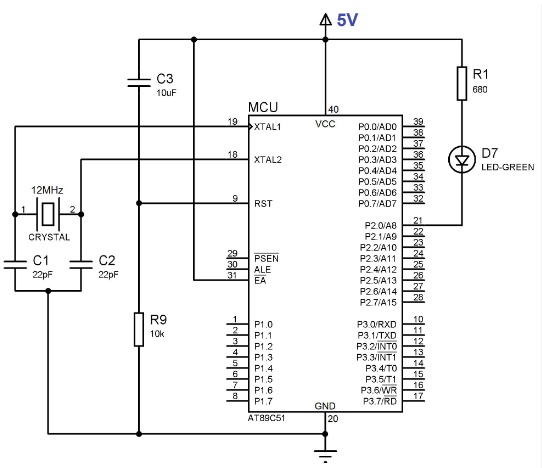
}

}

To test this code in the Keil IDE, follow these steps:

1. Open the Keil IDE and create a new project for the 8051 microcontroller.
2. Copy the above C code and paste it into a new source file in the Keil IDE.
3. Build the project by clicking on the "Build" button or pressing Ctrl + F7.
4. If there are no syntax errors, proceed to the next step.
5. Connect the microcontroller board to your computer and configure the appropriate settings in the Keil IDE for flashing the program onto the microcontroller.
6. Flash the program onto the microcontroller using the "Flash" or "Download" option in the Keil IDE.
7. Once the program is successfully flashed, you can observe the pins of Port 2 toggling at a certain delay defined in the **delay ()** function.

**Schematic:**



**Task3: Write a code for Interfacing LED's to 8051 MC and display glowing LED's in different Patterns.**

**1) Even and Odd LEDs glow**

**2) All ON and OFF LEDs Pattern**

**3) First Four LEDs ON and Last Four** **OFF**

**Task 01:** Even and Odd LEDs glow.

**Source Code:**

ORG 0000H

Start:

 MOV R1,#0AAH

 MOV P1,R1

 ACALL Delay

 MOV R2,#55H

 MOV P1,R2

 ACALL Delay

 SJMP Start

Delay:

 MOV R3,#250  ;2 MC

AGAIN:

 MOV R4,#250  ;2MC

HERE:

 NOP          ;1 MC

 NOP          ;1MC

 DJNZ R4,HERE  ;(1+1+2)x250=1000Mc internal loop (here)

 DJNZ R3,AGAIN ;(2+2+1000) x 250= 251000+2+2=251004=251 msec

 RET            ;2 MC

 END

ORG 0000H

Start:

 MOV R1,#0AAH

 MOV P1,R1

 ACALL Delay

 MOV R2,#55H

 MOV P1,R2

 ACALL Delay

 SJMP Start

Delay:

 MOV R3,#250  ;2 MC

AGAIN:

 MOV R4,#250  ;2MC

HERE:

 NOP          ;1 MC

 NOP          ;1MC

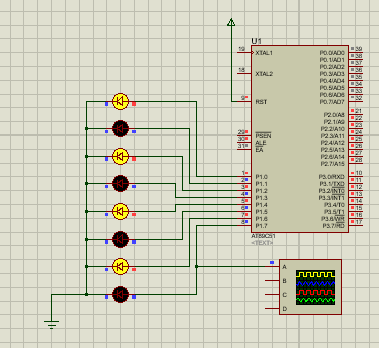
 DJNZ R4,HERE  ;(1+1+2)x250=1000Mc internal loop (here)

 DJNZ R3,AGAIN ;(2+2+1000) x 250= 251000+2+2=251004=251 msec

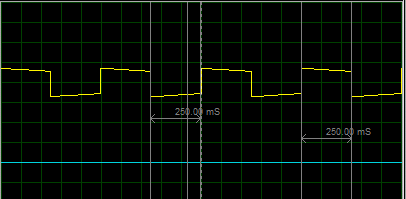
RET            ;2 MC

 END

**Output:**

****

**Graph:**

****

**Task02:** All ON and OFF LEDs pattern.

**Source Code:**

ORG 0000H

Start:

 MOV R1,#00H

 MOV P1,R1

 ACALL Delay

 MOV R2,#0FFH

 MOV P1,R2

 ACALL Delay

 SJMP Start

Delay:

 MOV R3,#250

AGAIN:

 MOV R4,#250

HERE:

 NOP

 NOP

 DJNZ R4,HERE

 DJNZ R3,AGAIN

 RET

 END

ORG 0000H

Start:

 MOV R1,#00H

 MOV P1,R1

 ACALL Delay

 MOV R2,#0FFH

 MOV P1,R2

 ACALL Delay

 SJMP Start

Delay:

 MOV R3,#250

AGAIN:

 MOV R4,#250

HERE:

 NOP

 NOP

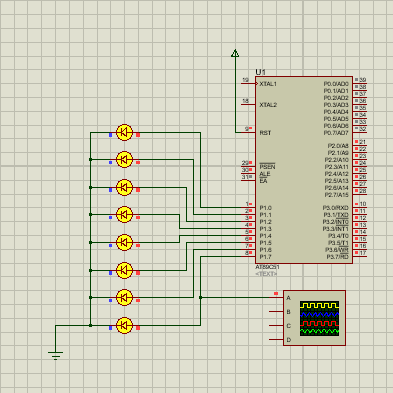
 DJNZ R4,HERE

DJNZ R3,AGAIN

 RET

 END

**Output:**



**Task03:** First Four LEDs ON and Last Four OFF.

**Source Code:**

ORG 0000H

Start:

 MOV R1,#0F0H

 MOV P1,R1

 ACALL Delay

MOV R2,#0FH

 MOV P1,R2

 ACALL Delay

 SJMP Start

Delay:

 MOV R3,#250

AGAIN:

 MOV R4,#250

HERE:

 NOP

 NOP

 DJNZ R4,HERE

 DJNZ R3,AGAIN

 RET

 END

ORG 0000H

Start:

 MOV R1,#0F0H

 MOV P1,R1

 ACALL Delay

 MOV R2,#0FH

 MOV P1,R2

 ACALL Delay

 SJMP Start

Delay:

 MOV R3,#250

AGAIN:

 MOV R4,#250

HERE:

 NOP

 NOP

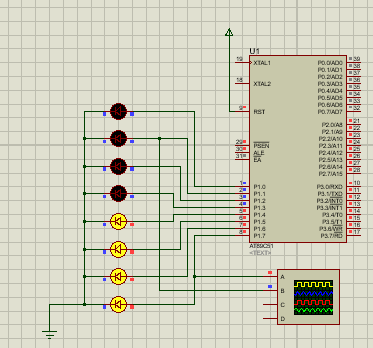
 DJNZ R4,HERE

 DJNZ R3,AGAIN

 RET

 END

**Output:**



**Graph:**

