

JOURNAL

NAME:

ROLL NO :

SUBJECT :

EMBEDDED SYSTEM

Practical No-1

AIM : Program to add 8-bit number.

Memory Address	Mnemonics	Operand	Hex code
C000			3E
C001	MVI	A, 05	05
C002			06
C003	MVI	B, 04	04
C004			80
C005	ADD	B	32
C006	STA	C300	00
C007			C3
C008	RST 1		CF

Practical No-2

AIM : Program to Subtract two 8-bit number.

Memory Address	Mnemonics	operand	Hex code
C000	MVI	A, 05	3E
C001			05
C002	MVI	B, 03	06
C003			03
C004	SUB	B	80
C005	STA	C300	32
C006			00
C007			C3
C008	HLT		76

Practical No-3

AIM : Program to Store data byte 32H into memory location 4000H

Memory location	Mnemonics	Operand	Hex Code
C000	MVI	A, 32H	3E
C001			32
C002	STA	4000H	32
C003			00
C004			40
C005	HLT		76

Practical No-4

AIM : Design and develop a reprogrammable embedded computer using 8051 microcontrollers and to show the following aspects.

a. Programming

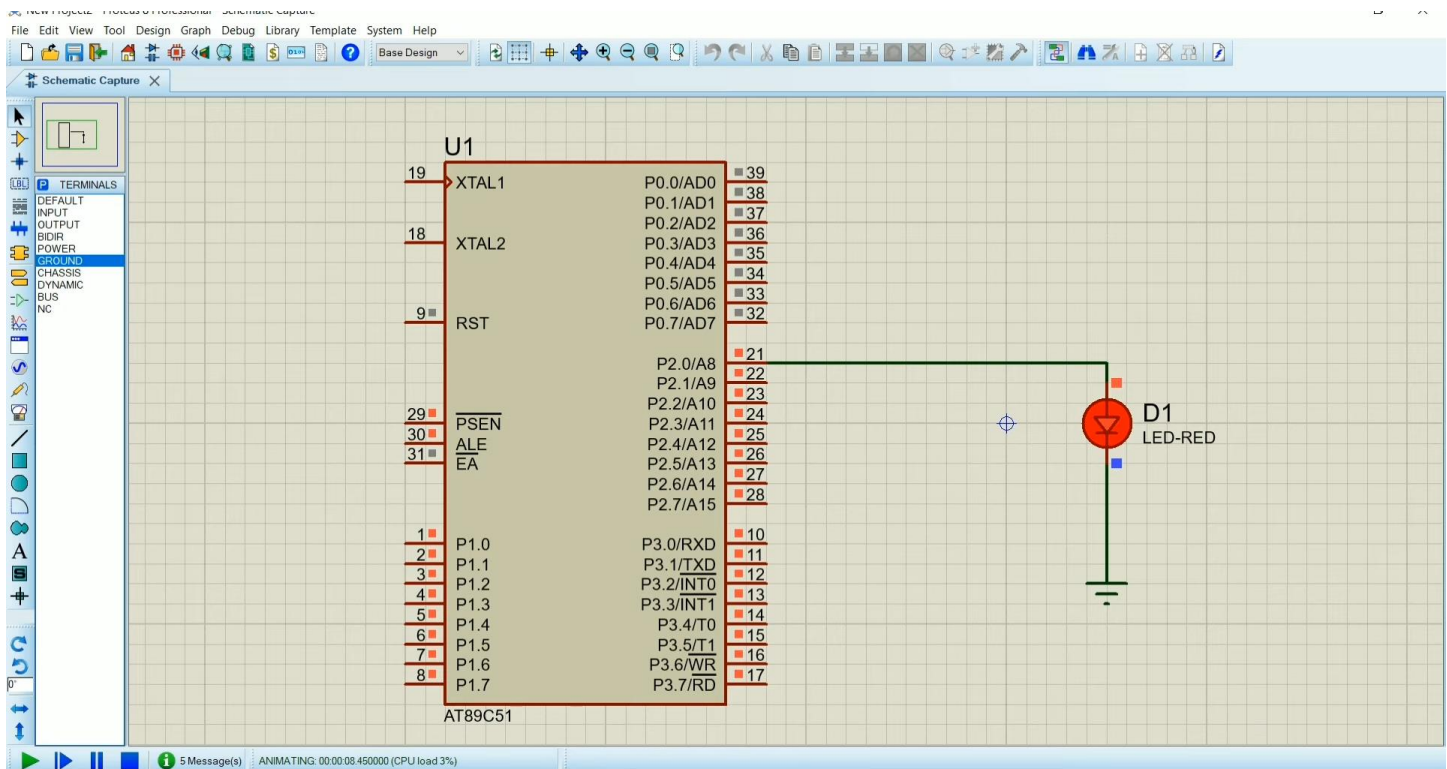
b. Execution

c. Debugging

Source code:

```
#include<reg52.H>
sbit LED = P2^0;
void Delay(void);
void main(void)
{
    while(1)
    {
        LED = 0;
        Delay();
        LED = 1;
        Delay();
    }
}
void Delay(void)
{
    int j;
    int i;
    for(i = 0; i < 10; i++)
    {
        for(j = 0; j < 10000; j++)
        {
        }
    }
}
```

Output

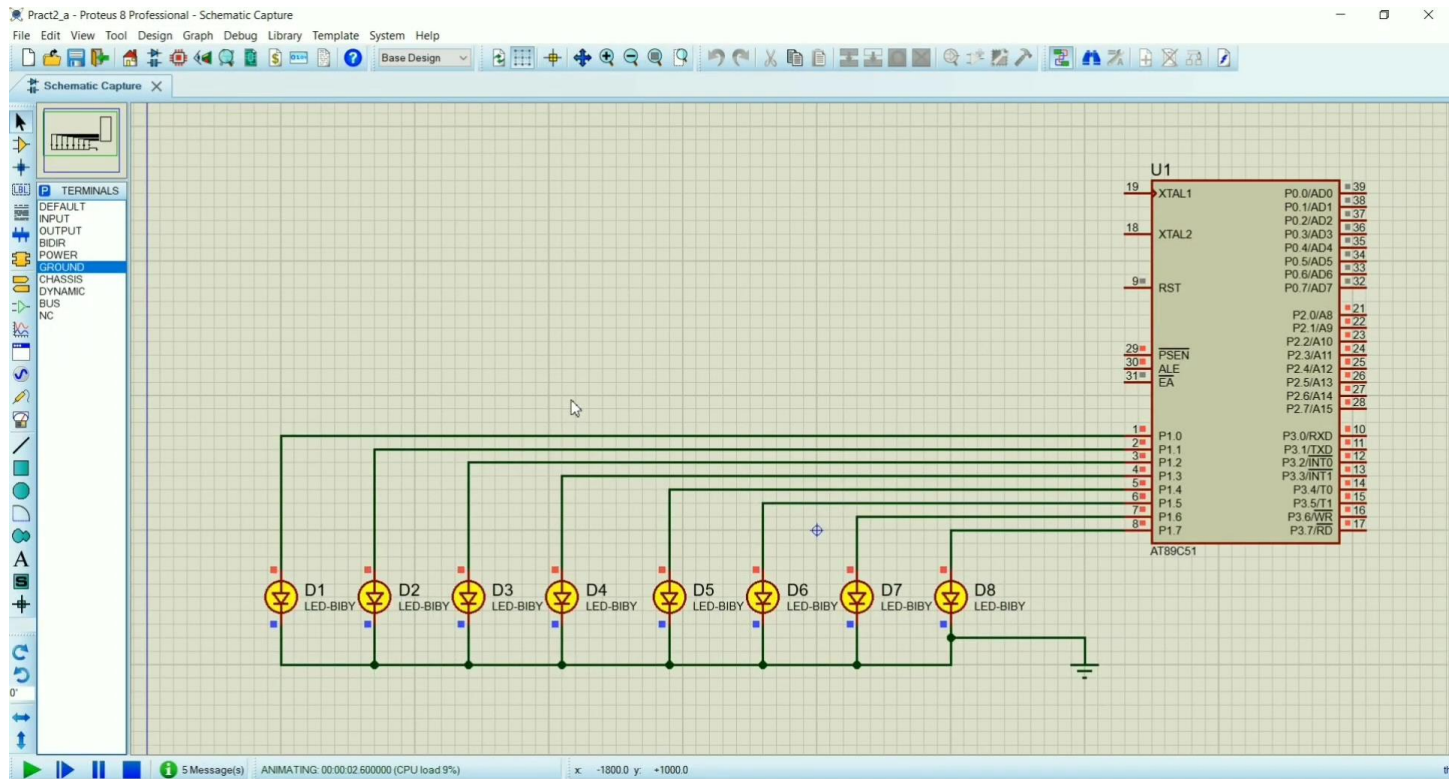


Practical No-5

AIM : Configure timer control register of 8051 and develop a program to generate given time delay.

Source code:

```
#include<reg51.H>
void Delay(void);
void main(void)
{
    while(1)
    {
        P1 = 0xFF; // Make all bits of P1 high
        Delay();
        P1 = 0x00; // Make all bits of P1 low
        Delay();
    }
}
void Delay(void)
{
    int j;
    int i;
    for(i = 0; i < 1000; i++)
    {
    }
    for(j = 0; j < 1000; j++)
    {
    }
}
```


Output :

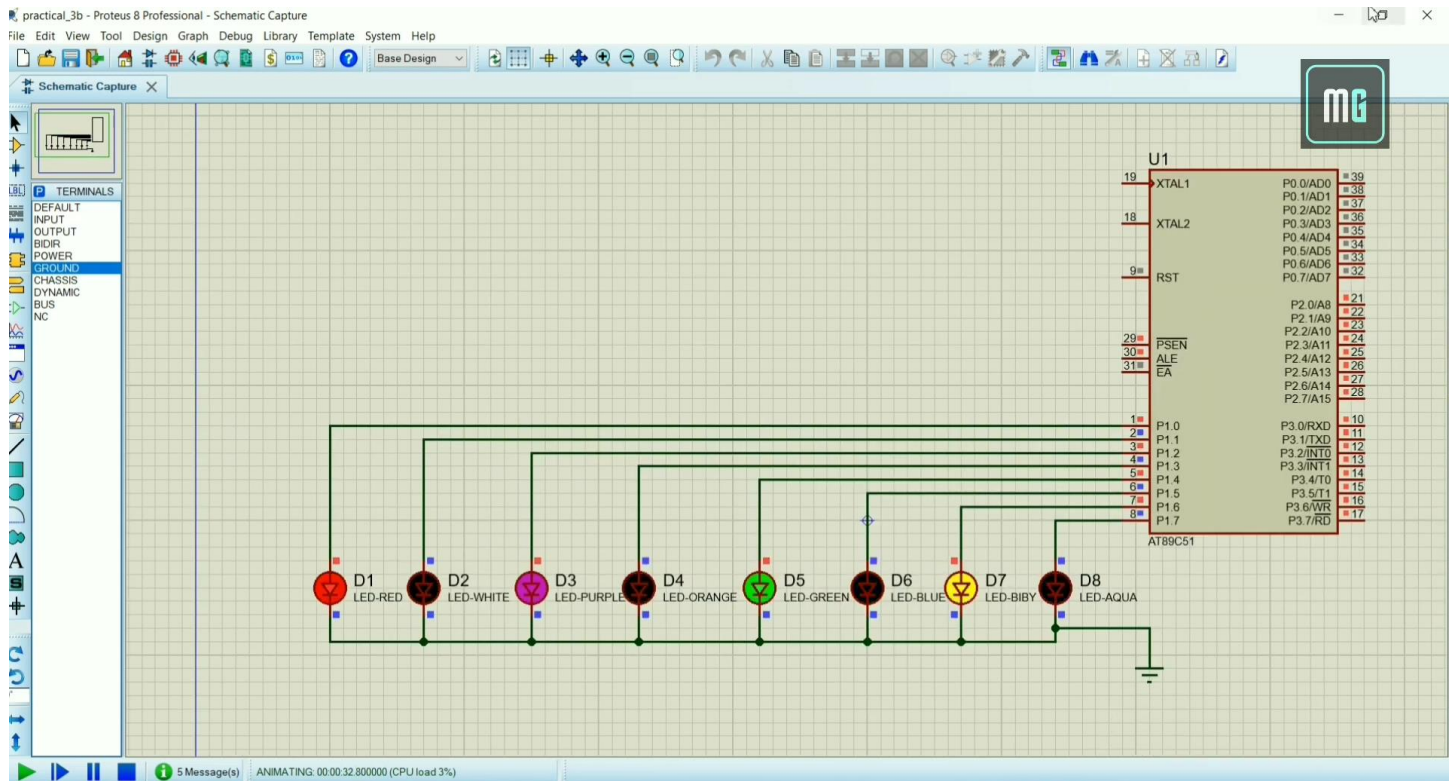
Practical No-6

AIM : To interface 8 LED's at Input-output port and create different patterns

Source code:

```
#include<reg51.h>
void delay();
void main()
{
    while(1)
    {
        P1 = 0xAA; // Pattern to turn on alternate LEDs -
10101010
        delay();
        P1 = 0x55; // Reverse the pattern
        delay();
    }
}
void delay()
{
    unsigned int i, j;
    for(i = 0; i < 23; i++)
    {
        for(j = 0; j < 1000; j++)
        {
        }
    }
}
```

Output :



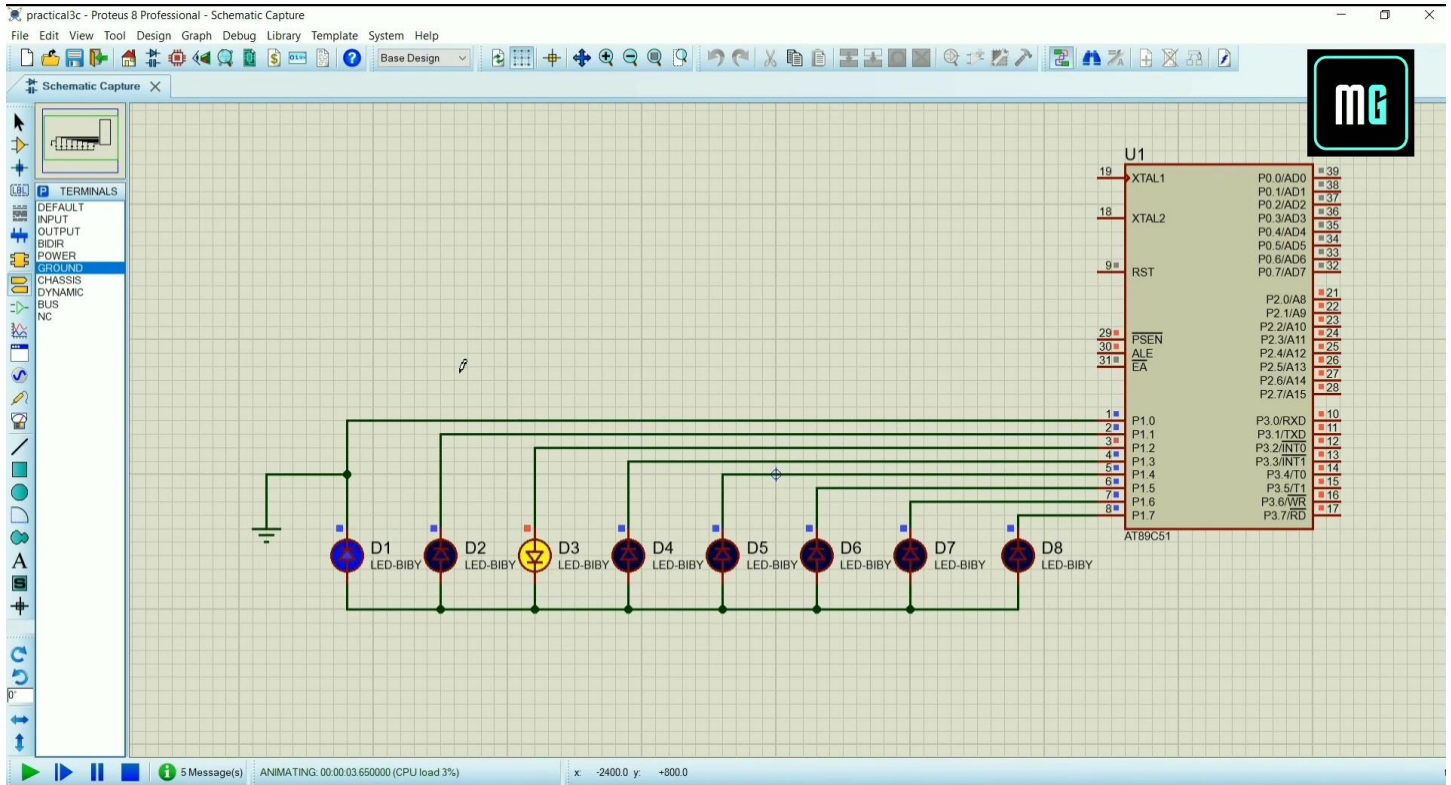
Practical No-7

AIM : To demonstrate timer working in timer mode and blink LED without using any loop delay routine.

Source code:

```
#include<reg51.h>
int i = 0;
void timer_ISR(void)interrupt 1
{
    i++;
    if(i == 10)
    {
        i=0;
        P1++;
    }
}
void main(void)
{
    TMOD = 0x01;
    ET0 = 1;
    TR0 = 1;
    EA = 1;
    while(1);
}
```

Output :



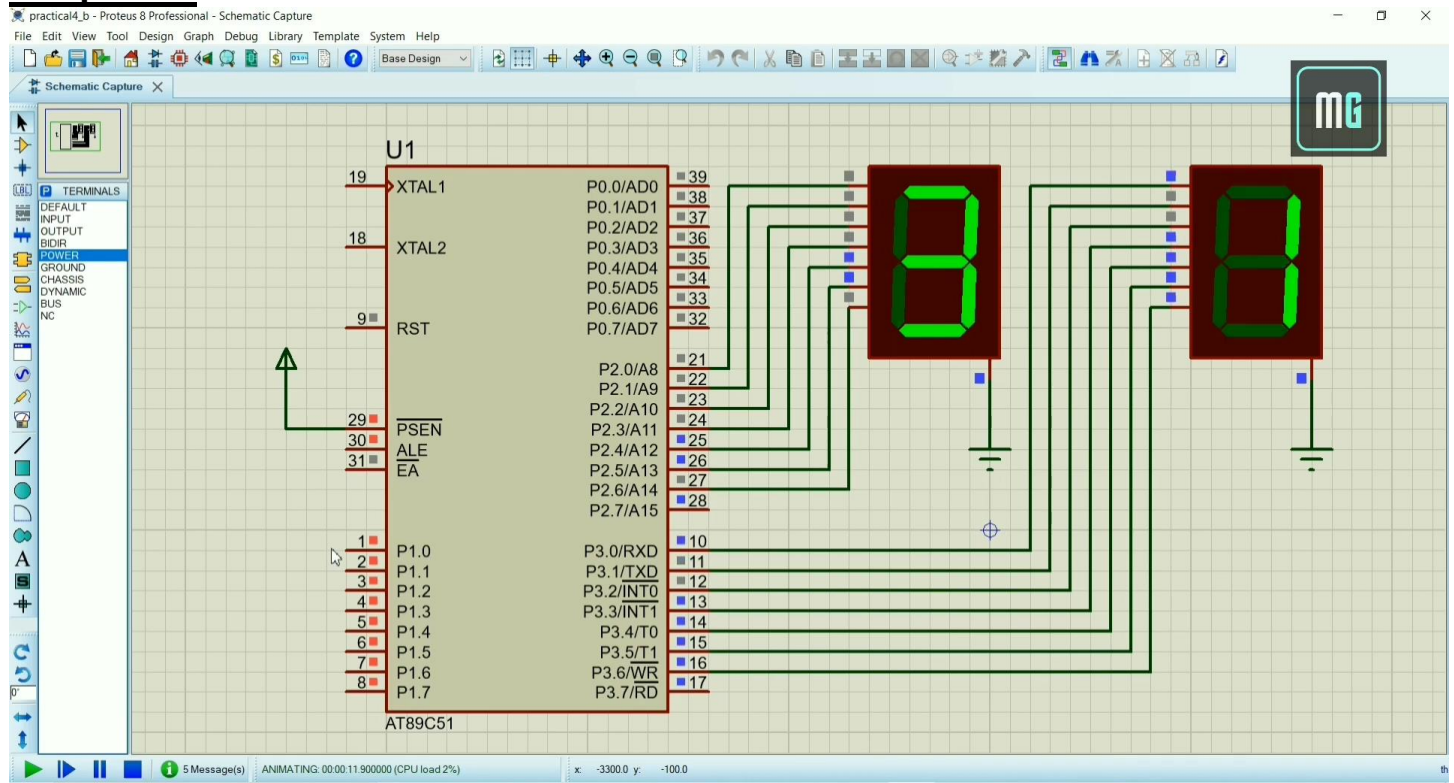
Practical No-8

AIM :To demonstrate interfacing of seven-segment LED display and generate counting from 0 to 99 with fixed time delay.

Source code:

```
#include<reg51.h>
void delay(unsigned int ms)
{
    unsigned int i, j;
    for(i = 0; i<ms; i++)
    {
        for(j = 0; j<=1275; j++)
        {
        }
    }
}
void main(void)
{
    char number[] = {0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07, 0x7F, 0x6F};
    int i, j;
    P2 = 0x00;
    P3 = 0x00;
    while(1)
    {
        for(i = 0; i<=9; i++)
        {
            P2 = number[i];
            for(j = 0; j<=9; j++)
            {
                P3 = number[j];
                delay(50);
            }
        }
    }
}
```

}
}
Output :



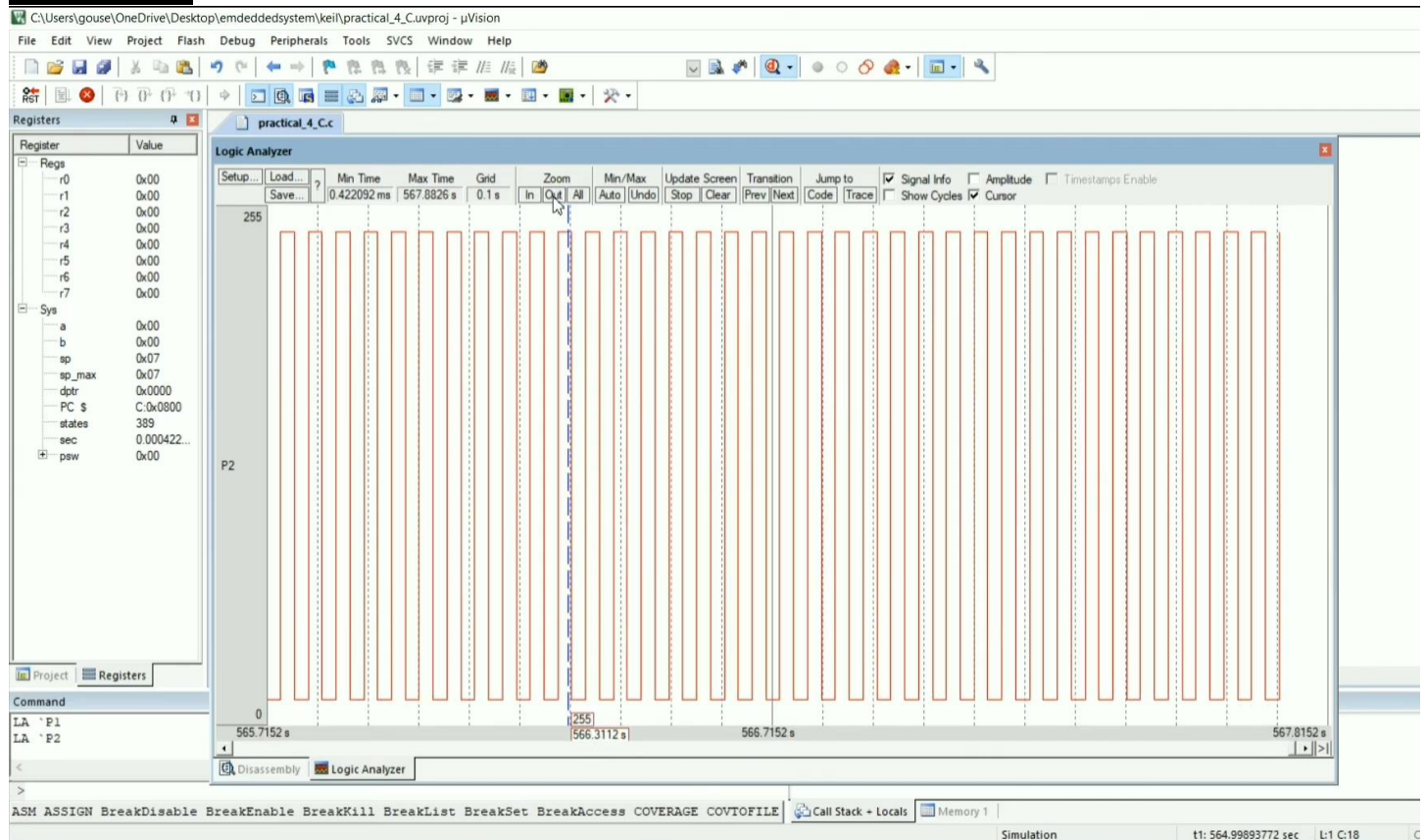
Practical No-9

AIM :To demonstrate interfacing of seven-segment LED display and generate counting from 0 to 99 with fixed time delay.

Source code:

```
#include<reg51.h>
void delay();
void main()
{
    P2 = 0x00;
    while(1)
    {
        P2 = 0xFF;
        delay();
        P2 = 0x00;
        delay();
    }
}
void delay()
{
    int i;
    for(i = 0; i <= 5000; i++)
    {
    }
}
```


Output :



Practical No-10

AIM : Interface 8051 with D/A converter and generate triangular wave of given frequency on oscilloscope.

Source code:

```
#include<reg51.h>
void main()
{
    P2= 0x00;
    while(1)
    {
        do
        {
            P2 += 0x05;
        }
        while(P2 < 0xFF);
        do
        {
            P2 -= 0x05;
        }
        while(P2 > 0x00);
    }
}
```

Output :

