

Practical 2 Arithmetic Instructions-Addition, Subtraction, Multiplication and Division using 8051

ORG 0000H

MOV A,#27H

MOV B,#49H

ADD A,B

MOV P0,A

END

Program 3: Write a program to generate a pattern on LEDs

```
#include <reg51.h>
```

```
void MSDelay(unsigned int itime);
```

```
void main(void) {
```

```
    while(1) {
```

```
        P0 = 0x55;
```

```
        P2 = 0x55;
```

```
        MSDelay(250);
```

```
        P0 = 0xAA;
```

```
        P2 = 0xAA;
```

```
        MSDelay(250);
```

```
    }
```

```
}
```

```
void MSDelay(unsigned int itime) {
```

```
    unsigned int i, j;
```

```
    for(i = 0; i < itime; i++)
```

```
        for(j = 0; j < 1275; j++);
```

```
}
```

Program 4: Write and test the program to display “0” to “9” on seven segment display

Experiment No.4

Display number “0” to “9” on seven segment display(Single digit)

Program in Keil:

```
#include<reg51.h>
```

```
void msdelay(unsigned int time) // Function for creating delay in milliseconds.
```

```
{  
    unsigned i,j ;  
    for(i=0;i<time;i++)  
        for(j=0;j<1275;j++);  
}
```

```
void main()
```

```
{  
    P1=0x01;  
    unsigned char Segment_code[ ]={0xC0,0xF9,0xA4,0xB0,0x99,0x92,0x82,0xF8,0x80,0x90};  
    int k;  
    while(1)  
    {  
        for(k=0;k<10;k++)  
        {  
            P0=Segment_code[k];  
            msdelay(100);  
        }  
    }  
}
```

Display digit “00” to “99” on seven segment display(Multiple digit)

Program (in Keil C)

```

#include <reg51.h>

sbit DIS1=P1^0;
sbit DIS2=P1^1;

void M_Delay(unsigned char);

unsigned char Dig,Dec;

unsigned int Delay;

unsigned char Pattern[10]={0xC0,0XF9,0xA4,0xB0,0x99,0x92,0x82,0xF8,0x80,0x90};

void main(void)
{
    P0=0xff;
    while(1)
    {
        for(Dec=0;Dec<10;Dec++)
        {
            for(Dig=0;Dig<10;Dig++)
            {
                for(Delay=0;Delay<2000;Delay++)
                {
                    DIS1=1;

                    DIS2=0;

                    P0=Pattern[Dec];

                    M_Delay(30);

                    DIS2=1;

                    DIS1=0;

                    P0=Pattern[Dig];

                    M_Delay(30);

                }
            }
        }
    }
}

```

```
        }  
    }  
}  
void M_Delay(unsigned char MyTime)  
{  
    unsigned char x;  
    for(x=0;x<MyTime;x++);  
}
```

Program 5 : Write a program to display message on LCD

```
#include <reg51.h>
```

```
sbit RS = P2^0;
```

```
sbit EN = P2^1;
```

```
void delay_ms(unsigned int ms) {  
    unsigned int i, j;  
    for (i = 0; i < ms; i++)  
        for (j = 0; j < 120; j++);  
}
```

```
void delay_us(unsigned int us) {  
    unsigned int i;  
    for (i = 0; i < us; i++) {  
        nop(); nop(); nop();  
    }  
}
```

```
void lcd_send_nibble(unsigned char nibble) {  
    unsigned char tmp;  
    nibble &= 0x0F;  
    tmp = P2 & 0xC3;  
    tmp |= (nibble << 2);  
    P2 = tmp;  
    EN = 1;  
    delay_us(2);  
    EN = 0;  
    delay_us(50);  
}
```

```
void lcd_cmd(unsigned char cmd) {  
    RS = 0;  
    lcd_send_nibble(cmd >> 4);  
    lcd_send_nibble(cmd & 0x0F);  
    if (cmd == 0x01 || cmd == 0x02)  
        delay_ms(2);  
    else  
        delay_us(50);  
}
```

```
void lcd_data(unsigned char dat) {  
    RS = 1;  
    lcd_send_nibble(dat >> 4);  
    lcd_send_nibble(dat & 0x0F);  
    delay_us(50);  
}
```

```
void lcd_init(void) {  
    delay_ms(20);  
    RS = 0;  
    EN = 0;  
    lcd_send_nibble(0x03);  
    delay_ms(5);  
    lcd_send_nibble(0x03);  
    delay_us(200);  
    lcd_send_nibble(0x03);  
    delay_us(200);  
    lcd_send_nibble(0x02);  
    delay_us(50);  
    lcd_cmd(0x28);  
    lcd_cmd(0x0C);  
}
```

```
    lcd_cmd(0x01);  
    delay_ms(2);  
    lcd_cmd(0x06);  
}
```

```
void lcd_puts(const char *s) {  
    while (*s) {  
        lcd_data(*s++);  
    }  
}
```

```
void lcd_gotoxy(unsigned char col, unsigned char row) {  
    unsigned char addr;  
    if (row == 0)  
        addr = 0x00 + col;  
    else  
        addr = 0x40 + col;  
    lcd_cmd(0x80 | addr);  
}
```

```
void main(void) {  
    lcd_init();  
    lcd_gotoxy(0, 0);  
    lcd_puts("Hello, World!");  
    lcd_gotoxy(0, 1);  
    lcd_puts("Keil + 8051 LCD");  
    while (1);  
}
```


Program 6: Write a program to rotate stepper motor in clockwise

```
#include <reg51.h>
```

```
sbit SW = P2^7;   // Switch at P2.7
```

```
void MSDelay(unsigned int value);
```

```
void main() {
```

```
    while (1) {
```

```
        if (SW == 0) {   // If switch is pressed (active low)
```

```
            P1 = 0x66;
```

```
            MSDelay(100);
```

```
            P1 = 0xCC;
```

```
            MSDelay(100);
```

```
            P1 = 0x99;
```

```
            MSDelay(100);
```

```
            P1 = 0x33;
```

```
            MSDelay(100);
```

```
        } else {       // If switch NOT pressed
```

```
            MSDelay(100);
```

```
            P1 = 0x66;
```

```
MSDelay(100);
```

```
P1 = 0x33;
```

```
MSDelay(100);
```

```
P1 = 0x99;
```

```
MSDelay(100);
```

```
P1 = 0xCC;
```

```
MSDelay(100);
```

```
}
```

```
}
```

```
}
```

```
void MSDelay (unsigned int value) {
```

```
    unsigned int x, y;
```

```
    for (x = 0; x < 1275; x++)
```

```
        for (y = 0; y < value; y++);
```

```
}
```

Program 7: Write a program to transmit/receive a word using UART Serial Communication to/from another digital device OR PC

```
#include <reg51.h>
```

```
void SerTx(unsigned char x)
```

```
{  
    SBUF = x;  
    while(TI == 0);  
    TI = 0;  
}
```

```
void main(void)
```

```
{  
    TMOD = 0x20;  
    TH1 = 0xFD;  
    SCON = 0x50;  
    TR1 = 1;
```

```
    while(1)
```

```
{  
    SerTx('Y');  
    SerTx('E');  
    SerTx('S');  
}  
}
```

Program 8: Write a program to generate specific delay using Timer of 8051

```
#include <reg51.h>
```

```
sbit SW  = P1^7;
```

```
sbit IND = P1^0;
```

```
sbit WAVE = P2^5;
```

```
void timer0_ISR(void) interrupt 1
```

```
{
```

```
    WAVE = !WAVE;
```

```
}
```

```
void main()
```

```
{
```

```
    SW = 1;
```

```
    IND = 0;
```

```
    TMOD = 0x02;
```

```
    TH0 = 0xA4;
```

```
    TL0 = 0xA4;
```

```
    IE = 0x82;
```

```
    TR0 = 1;
```

```
    while (1)
```

```
    {
```

```
        IND = SW;
```

```
    }
```

```
}
```

Program 9: Write a program to generate a square wave using timer interrupt.

```
#include <reg51.h>
```

```
sbit SW  = P1^7;
```

```
sbit IND = P1^0;
```

```
sbit WAVE = P2^5;
```

```
void timer0_ISR(void) interrupt 1
```

```
{
```

```
    WAVE = !WAVE;
```

```
}
```

```
void main()
```

```
{
```

```
    SW = 1;
```

```
    IND = 0;
```

```
    TMOD = 0x02;
```

```
    TH0 = 0xA4;
```

```
    TL0 = 0xA4;
```

```
    IE = 0x82;
```

```
    TR0 = 1;
```

```
    while (1)
```

```
    {
```

```
        IND = SW;
```

```
    }
```

```
}
```

Program 10: DAC interfacing with waveform generation using microcontroller 8051

```
#include <reg51.h>
```

```
sfr DACDATA = P1;
```

```
void main() {
```

```
    unsigned char WAVEVALUE[12] = {
```

```
        128,192,238,255,
```

```
        238,192,128,64,
```

```
        17,0,17,64
```

```
    };
```

```
    unsigned char x;
```

```
    while (1) {
```

```
        for (x = 0; x < 12; x++) {
```

```
            DACDATA = WAVEVALUE[x];
```

```
        }
```

```
    }
```

```
}
```