**Experiment No: 01 and 02**

1. Simple Assembly language programming.

a. Write assembly language program for addition of two 8-bit numbers.

b. Write assembly language program for addition of N 8-bit numbers. Take the input numbers

from memory and store result in memory.

2. Complex Assembly language programming.

a. Find square of a number using DPTR.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// 1 - Addition of n 8-bit numbers

MOV R0,#30H; pointer to memory

MOV R1,#05H; counter

MOV 36H,#00H; initialize sum=0

MOV 35H,#00H; initialize carry=0

MOV A,#00H;

L2:ADD A,@R0; A=A+@R0

DA A

JNC L1

INC 35H; store carry

L1: INC RO; point to next loc

DJNZ R1,L2

MOV 36H,A;

L3:SJMP L3;

END

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// 2 - Square of a number with DPTR

MOV R0,#30H;

MOV A,@R0;

MOV DPTR,#2000H;

MOVC A,@A+DPTR;

MOV 31H,A;

L1:SJMP L1

ORG 2000H

    DATA1: DB 0,1,4,9,16,25,36,49,64,81,100

END

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// 3- Addition of two 16-bit numbers

MOV A,30H; A=LB1

ADD A,31H; A=A+LB2=LBR

MOV 36H,A; LBR STR

CLR A;

MOV A,32H; A=HB1

ADDC A,33H; A=A+HB2=HBR

MOV 35H,A; HBR STR

JNC L1;

INC 34H; INCREMENT CARRY

L1:SJMP L1;

END

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// 4 - Finding No.of ODD and EVEN Nos. from set of 10 numbers

MOV R0,#30H

MOV R1,30AH

MOV 0BH,R1

MOV R2,OBH

MOV 3AH,#00H

MOV 3BH,#00H

L2:MOV A,@R0

RRC A

JNC L1

INC 3AH

L1:INC R0

DJNZ R1,L2

CLR C

MOV A,R2

SUBB A,3AH

MOV 3BH,A

L3:SJMP L3

END

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//5 - Finding No.of Positive and Negative Integers from set of 10 numbers

MOV R0,#30H

MOV R1,#OAH

MOV OBH,R1

MOV R2,OBH

MOV 3AH,#00H

MOV 3BH,#00H

L2:MOV A,@RO

RLC A

JNC L1

INC 3AH

L1:INC R0

DJNZ R1,L2

CLR C

MOV A,R2

SUBB A,3AH

MOV 3BH,A

L3:SJMP L3

END

**EXP-3 Interfacing of LED, Buzzer, Relay and Switch with C8051F340**

Write C program for interfacing of LED, Buzzer, Relay and Switch with C8051F340 to turn it ON when key is pressed

**LED Interfacing**

#include"C8051F340.h"

#define LED P4

sbit key3=P1^2;

sbit key4=P1^3;

sbit Relay=P1^4;

sbit Buzzer=P3^3;

void main()

{

XBR1=0X40;

P1MDIN=0X0C;

P3MDOUT=0X80;

P1MDOUT=0X10;

P4MDOUT=0XFF;

while(1){

    if(key3==0){

        while(1){

            Relay = 0;

            Buzzer = 1;

            LED = 0x00;

            if(key4==0){

                    break;

                    }

            }

        }

    if(key3==0){

        while(1){

                LED = 0XFF;

                Relay = 1;

                Buzzer = 0;

                if(key4==0){

                    break;

                    }

            }

        }

    }

}

**EXP-4 Interfacing of LCD**

Write C program for interfacing of 16x2 LCD with C8051F340 in 8-bit mode

**Basic LCD interfacing**

#include"c8051f340.h"

void DelayMs(unsigned int Ms);

void Write\_command\_LCD(unsigned char character);

void Write\_Data\_LCD(unsigned char name);

sbit LCD\_RS=P1^5;

sbit LCD\_RW=P1^6;

sbit LCD\_EN=P1^7;

void main()

{

    XBR1=0x40;

    P2MDOUT=0xFF;

    P1MDOUT=0xE0;

    Write\_command\_LCD(0x38);

    DelayMs(50);

    Write\_command\_LCD(0x01);

    DelayMs(50);

    Write\_command\_LCD(0x0C);

    DelayMs(50);

    Write\_command\_LCD(0x80);

    DelayMs(50);

    Write\_Data\_LCD('W');

    DelayMs(50);

    Write\_Data\_LCD('P');

    DelayMs(50);

    Write\_Data\_LCD('U');

    DelayMs(50);

    while(1);

}

void DelayMs(unsigned int Ms)

{

    unsigned int n;

    unsigned int i;

    for(n=0;n<Ms;n++)

    {

        for(i=0;i<65;i++);

    }

}

void Write\_Command\_Lcd(unsigned char command)

{

    LCD\_RS=0;

    LCD\_RW=0;

    P2=command;

    LCD\_EN=1;

    DelayMs(15);

    LCD\_EN=0;

}

void Write\_Data\_LCD(unsigned char character)

{

    LCD\_RS=1;

    LCD\_RW=0;

    P2=character;

    LCD\_EN=1;

    DelayMs(15);

    LCD\_EN=0;

}

**LCD interfacing/displaying name Shreerang in 8-bit**

#include"c8051f340.h"

void DelayMs(unsigned int Ms);

void Write\_command\_LCD(unsigned char character);

void Write\_Data\_LCD(unsigned char name);

sbit LCD\_RS=P1^5;

sbit LCD\_RW=P1^6;

sbit LCD\_EN=P1^7;

void main()

{

    unsigned char name[]={"SHREERANG"};

    int i;

    XBR1=0x40;

    P2MDOUT=0xFF;

    P1MDOUT=0xE0;

    Write\_command\_LCD(0x38);

    DelayMs(50);

    Write\_command\_LCD(0x01);

    DelayMs(50);

    Write\_command\_LCD(0x0C);

    DelayMs(50);

    Write\_command\_LCD(0x80);

    DelayMs(50);

    for(i=0;name[i]!='\0'; i++)

    {

        Write\_Data\_LCD(name[i]);

        DelayMs(50);

    }

    while(1);

}

void DelayMs(unsigned int Ms)

{

    unsigned int n;

    unsigned int i;

    for(n=0;n<Ms;n++)

    {

        for(i=0;i<65;i++);

    }

}

void Write\_Command\_Lcd(unsigned char command)

{

    LCD\_RS=0;

    LCD\_RW=0;

    P2=command;

    LCD\_EN=1;

    DelayMs(15);

    LCD\_EN=0;

}

void Write\_Data\_LCD(unsigned char character)

{

    LCD\_RS=1;

    LCD\_RW=0;

    P2=character;

    LCD\_EN=1;

    DelayMs(15);

    LCD\_EN=0;

}

**LCD interfacing/displaying name Shreerang in 4-bit**

#include"c8051f340.h"

void DelayMs(unsigned int Ms);

void Write\_command\_LCD(unsigned char character);

void Write\_Data\_LCD(unsigned char name);

sbit LCD\_RS=P1^5;

sbit LCD\_RW=P1^6;

sbit LCD\_EN=P1^7;

void main()

{

    unsigned char name[]={"SHREERANG"};

    int i;

    XBR1=0x40;

    P2MDOUT=0xFF;

    P1MDOUT=0xE0;

    Write\_command\_LCD(0x28);

    DelayMs(50);

    Write\_command\_LCD(0x01);

    DelayMs(50);

    Write\_command\_LCD(0x0C);

    DelayMs(50);

    Write\_command\_LCD(0x80);

    DelayMs(50);

    for(i=0;name[i]!='\0'; i++)

    {

        Write\_Data\_LCD(name[i]);

        DelayMs(50);

    }

    while(1);

}

void DelayMs(unsigned int Ms)

{

    unsigned int n;

    unsigned int i;

    for(n=0;n<Ms;n++)

    {

        for(i=0;i<65;i++);

    }

}

void Write\_Command\_Lcd(unsigned char command)

{

    P2=(command & 0xF0);

    LCD\_RS=0;

    LCD\_RW=0;

    LCD\_EN=1;

    DelayMs(15);

    LCD\_EN=0;

    P2=(command & 0x0F)<<4;

    LCD\_RS=0;

    LCD\_RW=0;

    LCD\_EN=1;

    DelayMs(15);

    LCD\_EN=0;

}

void Write\_Data\_LCD(unsigned char character)

{

    P2=(character & 0xF0);

    LCD\_RS=1;

    LCD\_RW=0;

    LCD\_EN=1;

    DelayMs(15);

    LCD\_EN=0;

    P2=(character & 0x0F)<<4;

    LCD\_RS=1;

    LCD\_RW=0;

    LCD\_EN=1;

    DelayMs(15);

    LCD\_EN=0;

}

**EXP-5 Interfacing of 8-bit DAC**

Write C program for interfacing of 8 bit DAC with C8051F340 to generate

1. Square wave
2. Triangular wave
3. Sawtooth

**DAC Interfacing for Square Waveform**

#include"c8051f340.h"

void delay(unsigned int Ms);

void main(){

    P4MDOUT=0xff;

    while(1){

        P4=~P4;

        delay(50);

    }

}

void delay(unsigned int Ms){

    unsigned int n;

    unsigned int i;

    for(n=0;n<Ms;n++){

        for(i=0;i<65;i++);

    }

}

**DAC Interfacing for Triangular Waveform**

#include"c8051f340.h"

void main(){

    int i;

    P4MDOUT=0xff;

    while(1){

        for(i=0; i<=254;i++){

            P4=i;

        }

        for(i=255; i>=1; i--){

            P4=i;

        }

    }

}

void delay(unsigned int Ms){

    unsigned int n;

    unsigned int i;

    for(n=0;n<Ms;n++){

        for(i=0;i<65;i++);

    }

}

**DAC Interfacing for Sawtooth Waveform**

#include"c8051f340.h"

void main(){

    int i;

    P4MDOUT=0xff;

    while(1){

        for(i=0; i<=254;i++){

            P4=i;

        }

    }

}

void delay(unsigned int Ms){

    unsigned int n;

    unsigned int i;

    for(n=0;n<Ms;n++){

        for(i=0;i<65;i++);

    }

}

**EXP-6 Interfacing of 8-bit ADC**

Write C program for programming of on chip ADC of C8051F340

#include "C8051F340.h"

#define SYSClk 12000000

sbit Buzzer= P3^3;

void delay(unsigned int Ms);

void main()

{

    XBR1= 0X40;

    P4MDOUT= 0XFF;

    www.mitwpu.edu.in

    Buzzer= 0;

    P2SKIP= 0X20;

    P2MDIN= 0XD0;

    AMX0P= 0X04;

    AMX0N= 0x1F;

    ADC0CF= (((SYSClk/3000000)-1)<<3);

    REF0CN= 0x08;

    ADC0CN= 0x80;

    AD0EN= 1;

    {

        ADC0CN =0x90;

        while (AD0BUSY == 1);

        delay(50);

        P4= ~ ADC0L;

        delay(50);

        P4= ~ ADC0H;

        delay(50);

    }

while(1);

}

void delay(unsigned int Ms)

{

    unsigned int n;

    unsigned int i;

    for (n-0; n<Ms; n++)

    {

        for (i=0; i<65; i++);

        www.mitwpu.edu.in

    }

}

**EXP-7 Generation of PWM using C8051F340 to control speed of DC motor**

Write C program to generation PWM using C8051F340 to control speed of DC motor

#include "c8051f340.h"

#define SYSCLK 3000000

void main(){

    PCA0L=0x10;

    OSCICN=0x83;

    CLKSEL=0x00;

    XBR1=0x41;

    P2MDOUT=0x08;

    P0SKIP=0xff;

    P1SKIP=0xff;

    P2SKIP=0x07;

    while(1){

        PCA0MD=0x02;

        PCA0CPM0=0x42;

        PCA0CPH0=(256-(256\*0.75));

        CR=1;

    }

}

**EXP-8 Implement UART with C8051F340**

Write a C program for serial communication using C8051F340 to transfer data from C8051F340 to PC

**Tranmission program for Uart with C8051F340**

#include"c8051f340.h"

#define sysclk 12000000

#define BR\_UART0 9600

void main()

{

    char ch[]=("SUGANDH");

    int i;

    OSCICN = 0X80;

    XBR0=0X01;

    XBR1=0X40;

    P0MDOUT=0X10;

    SCON0=0X00;

    CKCON=0X01;

    TH1=256-(sysclk/BR\_UART0/2/4);

    TH1=TL1;

    TMOD=0x20;

    TR1=1;

    while(1)

    {

        for(i=0;ch[i]!='\0';i++)

        {

            SBUF0=ch[i];

            while(TI0==0);

        }

    }

}

**Reciving program for Uart with C8051F340**

#include"c8051f340.h"

#define sysclk 12000000

#define BR\_UART0 9600

sbit BUZZER=P3^3;

void main()

{

    OSCICN = 0X83;

    XBR0=0X01;

    XBR1=0X40;

    P3MDOUT=0X08;

    BUZZER=1;

    P0MDIN=0X20;

    P4MDOUT=0xFF;

    SCON0=0X10;

    CKCON=0X01;

    TH1=256-(sysclk/BR\_UART0/2/4);

    TH1=TL1;

    TMOD=0x20;

    TR1=1;

    while(RI0==0)

    {

        P4=~SBUF0;

        RI0=0;

    }

}

**EXP-9 Interfacing of Stepper motor with C8051F340**

**Code for Full Stepping in Clockwise Direction**

#include "C8051F340.h"

void delay(unsigned int Ms);

void main()

{

    char a[4]={0x02, 0x20, 0x80, 0x08};

    int i;

    P4MDOUT=0xFF;

    while(1)

    {

        for(i=0; i<4;i++)

        {

            P4=a[i];

            delay(60);

        }

    }

}

void delay(unsigned int Ms)

{

    unsigned int n;

    unsigned int j;

    for(n=0;n<Ms;n++)

    {

        for(j=0;j<65;j++);

    }

}

**Code for Full Stepping in Anti Clockwise Direction**

#include "C8051F340.h"

void delay(unsigned int Ms);

void main()

{

    char a[4]={0x02,0x08, 0x80, 0x20};

    int i;

    P4MDOUT=0xFF;

    while(1)

    {

        for(i=0; i<4;i++)

        {

            P4=a[i];

            delay(60);

        }

    }

}

void delay(unsigned int Ms)

{

    unsigned int n;

    unsigned int j;

    for(n=0;n<Ms;n++)

    {

        for(j=0;j<65;j++);

    }

}

**Code for Half Stepping in Clockwise Direction**

#include "c8051F340.h"

void delay(unsigned int Ms);

void main()

{

    char a[]={0x02, 0x22, 0x20, 0xA0,0x80,0x80,0x88,0x0C};

    int i;

    P4MDOUT=0xFF;

    while(1)

    {

        for(i=0; i<8;i++)

        {

            P4=a[i];

            delay(60);

        }

    }

}

void delay(unsigned int Ms)

{

    unsigned int n;

    unsigned int j;

    for(n=0;n<Ms;n++)

    {

        for(j=0;j<65;j++);

    }

}

**Code for Half Stepping in Anti Clockwise Direction**

#include "c8051F340.h"

void delay(unsigned int Ms);

void main()

{

    char a[]={0x02, 0x0A, 0x08, 0x88,0x80,0xA0,0x20,0x22};

    int i;

    P4MDOUT=0xFF;

    while(1)

    {

        for(i=0; i<8;i++)

        {

            P4=a[i];

            delay(60);

        }

    }

}

void delay(unsigned int Ms)

{

    unsigned int n;

    unsigned int j;

    for(n=0;n<Ms;n++)

    {

        for(j=0;j<65;j++);

    }

}