INDEX

Experiment No.:-1

Objective:- WAP to perform all arithmetic operations in Assembly.

Tools Required:-

1. Keil version 4 and ECE Flash
2. JTAG Cable
3. Application board of 8051
4. Connecting wires and Power supply

Steps for Executing Program:-

1. Open Keil version 4 and create new project and save it.
2. Create new file and type a program(in C or Assembly language) in the editor and save it with **.ASM or .C** extension.
3. Compile the program file using built target, if there are errors then correct them and again click on built target.
4. Run the program by clicking on Debug and see the output. Perform the step by step execution by pressing F11 key or execute the whole program by pressing F5 key. After this execution .hex file is generated with name same as project name.
5. Dump the generated .hex file of the program into the 8051 board by using ECE Flash with USB cable. Now check the output of the Dump program on the 8051 board.

Program:-

```
ORG 0000H

MOV A, #10H

MOV B, #05H

ADD A,B

MOV R0,A

SUBB A,B

MOV R1,A

MUL AB

MOV R2,A

MOV B, #05H

DIV AB

MOV R3,A

END
```

Result:-

Content of R0 =0FH, Content of R1 =10H, Content of R2 =32H

Content of R3 =19H, Content of Accumulator =19H

Observations:-

First 010H and 05H are moved into register A and B respectively and then perform arithmetic operations on both register A and B. The final result is store in R0, R1, R2 and R3.

Experiment No.:-2

Objective:- WAP to perform logical operations in Assembly.

Tools Required:-

1. Keil version 4 and ECE Flash
2. JTAG Cable
3. Application board of 8051
4. Connecting wires and Power supply

Steps for Executing Program:-

1. Open Keil version 4 and create new project and save it.
2. Create new file and type a program(in C or Assembly language) in the editor and save it with .**ASM or .C** extension.
3. Compile the program file using built target, if there are errors then correct them and again click on built target.
4. Run the program by clicking on Debug and see the output. Perform the step by step execution by pressing F11 key or execute the whole program by pressing F5 key. After this execution .hex file is generated with name same as project name.
5. Dump the generated .hex file of the program into the 8051 board by using ECE Flash with USB cable. Now check the output of the Dump program on the 8051 board.

Program:-

```
ORG 0000H

MOV A, #2AH

MOV B, #0A8H

ANL A,B

MOV R0,A

ORL A,B

MOV R1,A

XRL A,B

MOV R2,A

CPL A

MOV R3,A

END
```

Result:-

Content of R0 =2BH, Content of R1 =A8H, Content of R2 =00H

Content of R3 =FFH, Content of Accumulator =FFH

Observations:-

First 2AH and A8H are moved into register A and B respectively and then perform logical operations on both register A and B. The final result is store in R0, R1, R2 and R3.

Experiment No.:-3

Objective:- WAP to add three 10 times to accumulator in Assembly.

Tools Required:-

1. Keil version 4 and ECE Flash
2. JTAG Cable
3. Application board of 8051
4. Connecting wires and Power supply

Steps for Executing Program:-

1. Open Keil version 4 and create new project and save it.
2. Create new file and type a program(in C or Assembly language) in the editor and save it with **.ASM or .C** extension.
3. Compile the program file using built target, if there are errors then correct them and again click on built target.
4. Run the program by clicking on Debug and see the output. Perform the step by step execution by pressing F11 key or execute the whole program by pressing F5 key. After this execution .hex file is generated with name same as project name.
5. Dump the generated .hex file of the program into the 8051 board by using ECE Flash with USB cable. Now check the output of the Dump program on the 8051 board.

Program:-

```
ORG 0000H

MOV A, #00H

MOV R1, #0AH

Label: ADD A, #03H

       DJNZ R1, Label

END
```

Result:-

Content of Accumulator =1EH

Content of R1 =00H

Observations:-

The above program is started from the address 0000H. First copy 00H and 0AH in register A and register R1 respectively. Add contents of accumulator with 03H till the content of the register has become 00H. The content of the register R1 is reduced by 01H after every execution of subprogram Label and add 03H with accumulator content.

Experiment No.:-4

Objective:- WAP to perform stack operations in Assembly.

Tools Required:-

1. Keil version 4 and ECE Flash
2. JTAG Cable
3. Application board of 8051
4. Connecting wires and Power supply

Steps for Executing Program:-

1. Open Keil version 4 and create new project and save it.
2. Create new file and type a program(in C or Assembly language) in the editor and save it with **.ASM or .C** extension.
3. Compile the program file using built target, if there are errors then correct them and again click on built target.
4. Run the program by clicking on Debug and see the output. Perform the step by step execution by pressing F11 key or execute the whole program by pressing F5 key. After this execution .hex file is generated with name same as project name.
5. Dump the generated .hex file of the program into the 8051 board by using ECE Flash with USB cable. Now check the output of the Dump program on the 8051 board.

Program:-

```
ORG 0000H

MOV R6, #025H

MOV R1, #12H

MOV R4, #0F3H

PUSH 6

PUSH 1

PUSH 4

POP 3

POP 5

END
```

Result:-
Content of R3 =F3H

Content of R5 =12H

Content of stack memory =25H

Observations:-

First copy 25H, 12H and F3 in register R6, R1 and R4 respectively. Content of register R6 is first move to stack memory by using PUSH 6. Similarly content of register R1 and R4 is moved to stack memory by using the PUSH 1 and PUSH 4 respectively. Then content from top of the stack is moved to register R3 and R5 by using POP 3 and POP 5 respectively.

Experiment No.:-5

Objective:- WAP to add 1E44H to 56CAH in Assembly.

Tools Required:-

1. Keil version 4 and ECE Flash
2. JTAG Cable
3. Application board of 8051
4. Connecting wires and Power supply

Steps for Executing Program:-

1. Open Keil version 4 and create new project and save it.
2. Create new file and type a program(in C or Assembly language) in the editor and save it with **.ASM or .C** extension.
3. Compile the program file using built target, if there are errors then correct them and again click on built target.
4. Run the program by clicking on Debug and see the output. Perform the step by step execution by pressing F11 key or execute the whole program by pressing F5 key. After this execution .hex file is generated with name same as project name.
5. Dump the generated .hex file of the program into the 8051 board by using ECE Flash with USB cable. Now check the output of the Dump program on the 8051 board.

Program:-

```
ORG 0000H

MOV R7, #44H

MOV R6, #1EH

MOV R5,#0CAH

MOV R4, #56H

MOV A, R5

ADD A , R7

MOV R3, A

MOV A, R6

ADDC A, R4

MOV R2,A

END
```

Result:-

Content of R3 =0EH

Content of R5 =75H

Observations:-

At first store lower bits of numbers in registers R7 and R5 and higher bits in R6 and R4. Then perform addition with carry and store resulted bits in R2 and R3.

Experiment No.:-6

Objective:- WAP for LED interfacing to toggle all bits of port P1 in C.

Tools Required:-

1. Keil version 4 and ECE Flash
2. JTAG Cable
3. Application board of 8051
4. Connecting wires and Power supply

Steps for Executing Program:-

1. Open Keil version 4 and create new project and save it.
2. Create new file and type a program(in C or Assembly language) in the editor and save it with **.ASM or .C** extension.
3. Compile the program file using built target, if there are errors then correct them and again click on built target.
4. Run the program by clicking on Debug and see the output. Perform the step by step execution by pressing F11 key or execute the whole program by pressing F5 key. After this execution .hex file is generated with name same as project name.
5. Dump the generated .hex file of the program into the 8051 board by using ECE Flash with USB cable. Now check the output of the Dump program on the 8051 board.

Program:-

```
#include<reg51.h>

void ms_delay(unsigned int);

void main(void)

{

    while(1)

         {

         P1=0x00;

         ms_delay(500);

         P1=0xFF;

         ms_delay(500);

         }

}

void ms_delay(unsigned int y)

{

               unsigned int x,z;

               for(x=0;x<=y;x++)

               for(z=0;z<=1275;z++)

               {}

}
```

Result:-

Content of Port P1 is first 00H and after 500ms it becomes FFH.

Observations:-

First delay function is assigned to ms_delay. In this delay function, two variables are used with for loop to provide delay. Variable “z” is incremented to 1275 which isd equivalent to 1 ms and this loop runs to 500 times to provide 500ms delay. Port P1 is first store 00H and store FFH after 500ms.

Experiment No.:-7

Objective:- WAP to display 00 to FF on port P0 in C.

Tools Required:-

1. Keil version 4 and ECE Flash
2. JTAG Cable
3. Application board of 8051
4. Connecting wires and Power supply

Steps for Executing Program:-

1. Open Keil version 4 and create new project and save it.
2. Create new file and type a program(in C or Assembly language) in the editor and save it with **.ASM or .C** extension.
3. Compile the program file using built target, if there are errors then correct them and again click on built target.
4. Run the program by clicking on Debug and see the output. Perform the step by step execution by pressing F11 key or execute the whole program by pressing F5 key. After this execution .hex file is generated with name same as project name.
5. Dump the generated .hex file of the program into the 8051 board by using ECE Flash with USB cable. Now check the output of the Dump program on the 8051 board.

Program:-

```
#include<reg51.h>

void main(void)

{

    unsigned char x;

    for(x=0;x<=255;x++)

    {

            P0=x;

    }

}
```

Result:-

Content of Port P1=FFH

Observations:-

First Port P0 contains 00H. "For" loop is used to increment the value of variable x from 0to 255 in decimal and it shows FFH in port P0 in hexadecimal.

Experiment No.:-8

Objective:- WAP to toggle d_0 bit of port P2 50,000 times in C.

Tools Required:-

1. Keil version 4 and ECE Flash
2. JTAG Cable
3. Application board of 8051
4. Connecting wires and Power supply

Steps for Executing Program:-

1. Open Keil version 4 and create new project and save it.
2. Create new file and type a program(in C or Assembly language) in the editor and save it with **.ASM or .C** extension.
3. Compile the program file using built target, if there are errors then correct them and again click on built target.
4. Run the program by clicking on Debug and see the output. Perform the step by step execution by pressing F11 key or execute the whole program by pressing F5 key. After this execution .hex file is generated with name same as project name.
5. Dump the generated .hex file of the program into the 8051 board by using ECE Flash with USB cable. Now check the output of the Dump program on the 8051 board.

Program:-

```
#include<reg51.h>

sbit mybit=P2^0;

void ms_delay(unsigned int);

void main(void)

{

    unsigned int x;

    for(x=0;x<=50000;x++)

    {

            mybit=0;

            ms_delay(500);

            mybit=1;

            ms_delay(500);

    }

}

void ms_delay(unsigned int y)

{

            unsigned int x,z;

            for(x=0;x<=y;x++)

            for(z=0;z<=1275;z++)
            {}
}
```

Result:-

LSB bit of Port P2 is set to 0 for 500ms and set to 1 for 500ms. It toggles between 0 and 1 for 50000 times.

Observations:-

First assign LSB of Port P2 to Mybit by using Sbit. Use the delay function for providing 500ms delay between setting 0 and 1 of LSB. Use the variable “x” for optimizing the process till 50000 times. We are using “For” loop which is starting from 0 and ending at 50000 with increment of 1.

Experiment No.:-9

Objective:- WAP to display 0000 to 9999 on seven segment display in C.

Tools Required:-

1. Keil version 4 and ECE Flash
2. JTAG Cable
3. Application board of 8051
4. Connecting wires and Power supply

Steps for Executing Program:-

1. Open Keil version 4 and create new project and save it.
2. Create new file and type a program(in C or Assembly language) in the editor and save it with **.ASM or .C** extension.
3. Compile the program file using built target, if there are errors then correct them and again click on built target.
4. Run the program by clicking on Debug and see the output. Perform the step by step execution by pressing F11 key or execute the whole program by pressing F5 key. After this execution .hex file is generated with name same as project name.
5. Dump the generated .hex file of the program into the 8051 board by using ECE Flash with USB cable. Now check the output of the Dump program on the 8051 board.

Program:-

```
#include<reg51.h>

char num[10]={0xbf,0x86,0xdb,0xcf,0xeb,0xed,0xfd,0x87,0xff,0xef};

void delay(unsigned int);

void main(void)

{

    int x;

    for(x=0;x<=9;x++)

    {

            P2=0x0f;

            P0=num[x];

            delay(50);

    }

}

void delay(unsigned int y)

{

    unsigned int z,x;

            for(z=0;z<=y;z++)

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

            {}
}
```

Result:-

Displaying 0000 to 9999 numbers after every 50ms and repeat it from 0000 for infinite times on seven segment display.

Observations:-

First store the position of LED of seven segment display which we want to display in num[x]. Port P0 is used for data port and Port2 is used for control port. The lower nibble of Port P2 is used to control 4 seven segment display. P2 is stored with 0FH for activate all seven segment display. P0 is stored with num[x] data to display numbers from 0000 to9999. Delay function is used to provide the delay of 50ms between two numbers.

Experiment No.:-10

Objective:- WAP to activate buzzer by using port P1.

Tools Required:-

1. Keil version 4 and ECE Flash
2. JTAG Cable
3. Application board of 8051
4. Connecting wires and Power supply

Steps for Executing Program:-

1. Open Keil version 4 and create new project and save it.
2. Create new file and type a program(in C or Assembly language) in the editor and save it with **.ASM or .C** extension.
3. Compile the program file using built target, if there are errors then correct them and again click on built target.
4. Run the program by clicking on Debug and see the output. Perform the step by step execution by pressing F11 key or execute the whole program by pressing F5 key. After this execution .hex file is generated with name same as project name.
5. Dump the generated .hex file of the program into the 8051 board by using ECE Flash with USB cable. Now check the output of the Dump program on the 8051 board.

Program:-

```
#include<reg51.h>

void delay()

{

    int j;

            for(j=0;j<20000;j++)

            {}

}

void main()

{

    while(1)

    {

            P1=0x03;

            delay();

            delay();

    }

}
```

Result:-

Buzzer has been started after dumping the program in microcontroller.

Observations:-

For activating the buzzer, we need to set 2 LSB bits high. So, we need to assign 03H in the port P1.

Experiment No.:-11

Objective:- WAP for rotating the stepper motor clockwise or anticlockwise.

Tools Required:-

1. Keil version 4 and ECE Flash
2. JTAG Cable
3. Application board of 8051
4. Connecting wires and Power supply

Steps for Executing Program:-

1. Open Keil version 4 and create new project and save it.
2. Create new file and type a program(in C or Assembly language) in the editor and save it with **.ASM or .C** extension.
3. Compile the program file using built target, if there are errors then correct them and again click on built target.
4. Run the program by clicking on Debug and see the output. Perform the step by step execution by pressing F11 key or execute the whole program by pressing F5 key. After this execution .hex file is generated with name same as project name.
5. Dump the generated .hex file of the program into the 8051 board by using ECE Flash with USB cable. Now check the output of the Dump program on the 8051 board.

Program:-

```
#include<reg51.h>
sbit m1=P0^0;
sbit m2=P0^2;
sbit m3=P0^1;
sbit m4=P0^3;
void delay(unsigned int x)
{
unsigned int a;
for(a=0;a<=x;a++)
{}
}
void move_clk(int y;int v)
{
    int a;
    for(a=0;a<=y;a++)
    {
        m1=1;m2=0;m3=0;m4=0;
            delay(v);
            m1=0;m2=1;m3=0;m4=0;
            delay(v);
            m1=0;m2=0;m3=1;m4=0;
            delay(v);
            m1=0;m2=0;m3=0;m4=1;
            delay(v);
    }
}
void move_anticlk(int y;int v)
{
    int a;
    for(a=0;a<=y;a++)
    {
         m1=0;m2=0;m3=0;m4=1;
            delay(v);
            m1=0;m2=0;m3=1;m4=0;
            delay(v);
            m1=0;m2=1;m3=0;m4=0;
            delay(v);
            m1=1;m2=0;m3=0;m4=0;
            delay(v);
    }
}
void motor_stop()
{
    m1=0;m2=0;m3=0;m4=1;
}
int main()
{
    mov_clk(50,1000);
    motor_stop();
    delay(6000);
    mov_anticlk(50,1000);
    motor_stop();

    delay(6000);

    while(1)

}
```

Result:-

Stepper motor has started clockwise and anti-clockwise according to the input after dumping the program in Microcontroller.

Observations:-

Stepper motor is having two windings and four I/O pins. That pins are connected to the pins 0, 1, 2, 3 of Port P0. The lower nibble of the Port P0 is assigned to m1, m2, m3, m4 for storing the data. The stepper motor is started moving in clockwise direction when Port P0 is being stored by 0x01H, 0x04H, 0x02H, 0x08H and the stepper motor is started moving anti-clockwise direction when Port P0 is stored by 0x08H,0x02H,0x04H, 0x01H with some delay which is provided by delay function.