

MICROPROCESSOR LAB EXPERIMENT 4

GROUP - 18

Deenabandhan N ee23b021

Sai Harshith Gajendra ee23b069

Krutarth Patel ee23b137

04 September 2024

Introduction :

- In this experiment , we are going to learn how to program the micro controller **ATmega8**.
- This experiment involves ,
 - Introduction to assembly language.
 - Write a program in assembly language to display the maximum and minimum of 10 numbers stored in **FLASH** memory.
 - Write a program in assembly language to add 10 numbers stored in flash memory and store it in the register.
 - Sort 5 numbers stored in flash memory in arbitrary order and write the final results to data memory
- In this report , we have included the code of the tasks and our experience with the assembly language.

ATmega-8 and Microchip studio :

- Atmega-8 is an 8-bit RISC single-chip microcontroller developed by Atmel.
- The number 8 in its name represents that it can operate 8 bits at a time while processing the information i.e in a way it represents the capacity of the microcontroller.
- Some features of AVR microcontroller are
 - I/O ports.
 - Internal instructions flash memory
 - SRAM upto 16KB
 - Timers
- Flash memory is used to store the programs whatever we have written in the microchip studio.
- Each instruction will occupy the size of 2 bytes/16 bits in flash memory except for the instructions like **STS** , **JMP** which will occupy 4 bytes in the memory.
- For example the following code ,

```
LDI R16,0x01
```

will occupy 2 bytes in the memory.

- Flash memory also has 32 registers (from R0 to R31) with three pointers ,
 - Z pointer : R30 and R31
 - Y pointer : R28 and R27
 - X pointer : R25 and R26
- These registers are used to hold memory in addition to having SRAM whose address starts from 0x60.
- We will see the instructions to implement the logic in the following sections.

Bonus Question

Introduction :

- This task involves taking two 4-bit numbers from PORT-D and displaying their sum using LEDs connected to PORT-C.

Code

```
[style=CStyle] include <avr/io.h> include <stdint.h> include <util/delay.h> int main(void) DDRC = 0x1F; //output port DDRD = 0x00; //input port while(1) unsigned char data = PIND; //receiving data // we get the 4-bit numbers as one 8-bit number // eg: number1 - 0101 1110; number2 PORTC = (data<4)+(data>4); delay_ms(10);
```

Process

- Getting two 4-bit numbers from PORT-C.
- Adding the two numbers.
- Displaying the result using LEDs.

4-bit Addition

Introduction :

- This task involves adding two 4-bit numbers and displaying the result using LEDs connected to PORT-C

Code

```
#include <avr/io.h>
#include <stdint.h>
#include <util/delay.h>
int main(void)
{
    DDRC = 0x1F; //output port
    unsigned char a = 15;
    unsigned char b = 200;

    while(1)
    {
        PORTC = a + b;
        _delay_ms(10);
    }
}
```

Process

- Adding the two numbers.
- Displaying the result using LEDs.