

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.

Finding the sum of numbers

Introduction :

- This task involves iterating through the registers and finding the sum
- It will always take n computations where n is the total number of values.

Code for Sum

```
.CSEG
```

```
LDI ZL, LOW(NUM<<1) ; Load the Z pointer with the array NUM  
LDI ZH, HIGH(NUM<<1) ; Load the Z pointer with the array NUM
```

```
; R16 will always store the sum of the array  
LPM R16, Z+ ; Load the first value of array
```

```
LDI R18, 0x09 ; Counter
```

```
LOOP :  
    LPM R17,Z+ ; Load the next value to R17  
    ADD R16,R17 ; Adding  
    DEC R18 ; decreasing the counter  
BRNE LOOP
```

```
NOP
```

```
NUM: .db 0x01,0x09,0x08,0x00,0x16,0x12,0x13,0x14,0x15,0x19
```

Usage of registers in flash memory in this code :

Registers	Usage
R18	Counter variable
R16 and R17	To store sum and temporary variable to store the loaded value
Z pointer	Store the array address to flash memory where the values are given.

Process

- Getting the values from the array and storing in a temporary variable.
- Adding the current sum to the loaded value
- updating the sum