

EE2016: Experiment 4

Group:19

Sparsh Gupta
EE23B117

Charan Srikanth
EE23B127

Kaushik Iyer
EE23B135

September 04, 2024

Abstract

This experiment was our first look into programming in AVR8 Assembly.

1 Objectives

The primary objectives of this experiment were to:

1. Getting familiar with the instruction set provided in AVR Assembly.
2. Understanding how to use Microchip Studio to build and debug the program.

2 Tasks

The following are the tasks given and our approach to solving it.

2.1 Task1 (Minimum and Maximum)

2.2 Pseudo Code

```
int* flash; // i.e Z

int i = 1;
int N = 9;

int max = *(flash++);
int min = max;

do { // loop
    int tmp = *(flash++);
    if (tmp > max){
        max = tmp;
    }

    if (min > tmp){
        min = tmp;
    }

    ++i;
} while(i <= N)
```

2.3 Assembly Code

```
.CSEG
LDI ZL,LOW(NUM<<1)
LDI ZH,HIGH(NUM<<1)

LDI R17,0x01 ;; i
LDI R18,0x09 ;; N
LPM R19, Z+ ;; max
MOV R20, R19 ;; min

loop: ;; Loop that goes through all the numbers in the flash
    LPM R21, Z+ ;; Load 1 from flash

    CP R19, R21
    BRCC skip_set_max ;; This will jump if R21 <= R19

    MOV R19, R21 ;; Set the maximum
    skip_set_max:

    CP R21, R20
    BRCC skip_set_min ;; This will jump if R20 <= R21

    MOV R20, R21 ;; Set the minimum
    skip_set_min:

    INC R17
    CP R18, R17
    BRCC loop ;; This will jump while R17 <= R18

NOP
NUM: .db 0xAA, 0xA0, 0xDF, 0xCF, 0x01, 0x02, 0x08, 0xA1, 0x71, 0x99
```

2.4 Task2 (Addition)

2.5 Pseudo Code

```
int* flash; // i.e Z

int i = 0;
int N = 9;

int sum = 0;

do {
    int tmp = *(flash++);
    sum += tmp;

    ++i;
} while(i <= N)
```

2.6 Assembly Code

```
.CSEG
LDI ZL,LOW(NUM<<1)
LDI ZH,HIGH(NUM<<1)

LDI R17,0x00 ;; i
LDI R18,0x09 ;; N
LDI R19,0x00 ;; sum

loop: ;; Loop that goes through all the numbers in the flash
    LPM R20, Z+ ;; Load 1 from flash

    ADD R19, R20;

    INC R17
    CP R18, R17
    BRCC loop ;; This will jump while R17 <= R18

NOP
NUM: .db 0xAA, 0xA0, 0xDF, 0xCF, 0x01, 0x02, 0x08, 0xA1, 0x71, 0x99
```

2.7 Task3 (Sorting)

2.7.1 Pseudo Code

```
int* flash; // i.e Z
int* data; // i.e X

int i = 0;
int j = 1;
int N = 4;

do { // fill_data_loop
    int tmp = *(flash++);
    *(data++) = tmp;
    ++i;
} while(i <= N)

i = 0;
do { // loop1 (repeats N + 1 times)
    reset_data_pointer();
    j = i + 1;
    do { // loop2 (repeats N - i times)
        int tmp1 = *(data++);
        int tmp2 = *data;

        if (tmp1 > tmp2){
            --data;
            *(data++) = tmp2;
            *data = tmp1;
        }

        ++j;
    } while(j <= N)
    ++i;
} while(i <= N)
```

2.7.2 Assembly Code

```
.CSEG
LDI ZL,LOW(NUM<<1)
LDI ZH,HIGH(NUM<<1)

LDI XL,0x60
LDI XH,0x00

LDI R17,0x00 ;; i
LDI R18,0x01 ;; j
LDI R19,0x04 ;; N

fill_data_loop: ;; Loop that populates the iram with data in flash
    LPM R16, Z+ ;; Load 1 from flash
    ST X+, R16 ;; Store into iram

    INC R17
    CP R19, R17
    BRCC fill_data_loop ;; This will jump while R17 <= R19

LDI R17,0x00 ;; Reset i to 0

loop1: ;; Loop 5 times
    ;; Reset pointer to data
    LDI XL,0x60
    LDI XH,0x00

    ;; Set j to i + 1
    MOV R18, R17
    INC R18
    loop2: ;; Loop 5 - j + 1 times
        ;; Load adjacent numbers from iRAM into R20 and R21
        LD R20,X+
        LD R21,X

        ;; Compare :)
        CP R21, R20
        BRCC skip_swap ;; Jumps if R20 <= R21

        ;; Swaps iRam[X-1] and iRam[X]
        DEC XL
        ST X+,R21
        ST X,R20

    skip_swap:
        INC R18
        CP R19, R18
        BRCC loop2

    INC R17
    CP R19, R17
    BRCC loop1

NOP
NUM: .db 0xAA, 0xA0, 0xDF, 0xCF, 0x01
```

3 Comments

1. Initially we had to experiment quite a bit to figure out how to implement loops.
2. Our next hurdle was figuring out how to write to the Data Memory. It took a little bit of searching online to get that sorted.