Electrical and Computer Engineering
EE3612Microprocessor Systems Laboratory

Lab 2: Introduction to Assembly Language Programming

Fall 2019

OBJECTIVES:

- To understand how to use AVR assembly instructions
- To understand how to detect the error in AVR assembly code
- To understand how to manipulate the sequence of the code using loop and program counter value

REFERENCES:

Mazidi and Naimi, "The AVR Microcontroller and Embedded Systems," , 2nd Ed. Chapters 1 and 2.

MATERIALS:

Atmel Studio 7

MAP OF THIS LAB:

- Activity 1: Assembly code and mathematical or logical expression
- Activity 2: Debugging and correction of the assembly code error
- Activity 3: Code modification to run the correct operation for the given task

LAB REPORT INSTRUCTIONS:

This lab consists of three activities. Use the given report frame to write your report and submit to Canvas assignment. **No full report is required.**

Submission Type: short assignment report in doc and pdf (use the given frame)
Due: section 01 – 9/17/2019 2:59pm, section 02 – 9/19/2019 8:59am

ACTIVITIES:

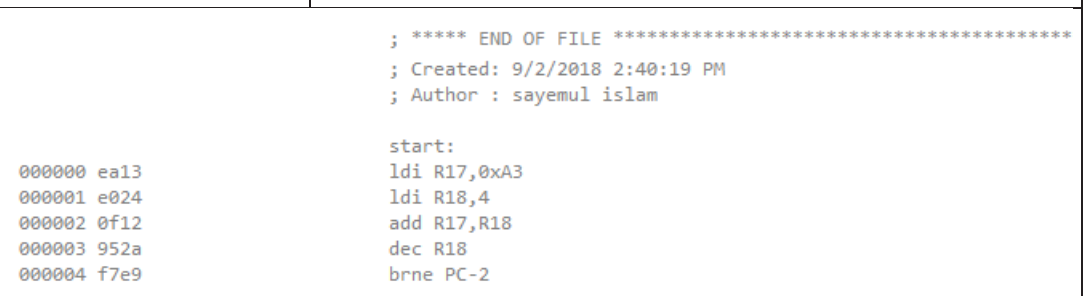
Activity 1

Read each code and write its own mathematical or logical expression (30 pts).

Run the following codes in the Atmel Studio environment. Examine the values of each register. Fill each section (it's operation) with mathematical or logical expression. Also, attach the *lss* file screenshots after you run the code. (See the given example in Figure 1).

Section	Assembly Code	Mathematical and Logical Expression

Example	LDI R17, 0xA3
	LDI R18, 4
	ADD R17, R18
	DEC R18
	BRNE PC-2
		R17 = 0xA3
		R18 = 4
		R17 = R17 + R18
		R18 = R18 – 1
		If Z=0, then PC=PC-2, else PC=PC+1 (next line or exit)

lss screenshot		

<Figure 1. Example of *lss* file screenshot>

Section	Assembly Code	Mathematical Expression

1-1.	LDI R22, $14
	LDI R21, $4
	SUB R22, R21
	BRNE PC-1	

lss screenshot		

1-2.	LDI R16, $15
	LDI R17, $22
	ADC R16, R17
	BRCC PC-1	

lss screenshot		

1-3.	LDI R25, $15
	LDI R26, $52
	DEC R26
	CP R25, R26
	BRNE PC-2	

lss screenshot		

Activity 2

Read the error message and correct the error parts in the code (30 pts).

2-1. This assembly code loads the hex values to the general purposed registers, R16 and R17. Then, it swaps the contents of the registers. Answer the questions.

```
ldi r16, 560          ; load a value 560 in decimal to R16
ld r17, 0x81          ; load a value 0x81 in hex to R17

; The following section is for swapping the values of the register 16 and 17
mov r22, r16          ; move the value of R16 to R22
mov r16, r17          ; move the value of R17 to R16
mov r17, r22          ; move the value of R22 to R17
```

Questions:

Q. What is the syntax error in this code?

Q. How to fix the error?

Screenshots: Attach the screenshots of the modified code and register values.

1) Show your modified code.

2) Show the final value of the register 16 and 17.

2-2. This assembly code loads four hex values ($10, $42, $a8, and $11) to the address 0x0100, 0x101, 0x0102 and 0x0103 in the data memory.

```
Ldi $10, r0          ; load a hex value $10 to a GPR
sts 0x100,r0         ; store the value to the memory location 0x0100
ldi r17, $42         ; load a hex value $42 to a GPR
sts 0x101,r17        ; store the value to the memory location 0x0101
ldi r1, $a8          ; load a hex value $a8 to a GPR
sts 0x102,r1         ; store the value to the memory location 0x0102
ldi r18, $11         ; load a hex value $11 to a GPR
sts 0x103,r18        ; store the value to the memory location 0x0103
```

Questions:

Q. What is the syntax error in this code?

Q. How to fix the error?

Screenshots: Attach the screenshots of the modified code and register values.

1) Show your modified code.

2) Show the final values of the memory location 0x100, 0x101, 0x102, and 0x103

2-3. This assembly code loads value $7 to the memory address 0x0200. Then, load the contents of 0x0200 to the memory location 0x220. The content of the location 0x220 is decremented continuously until the value hits zero by using the register R1.

```
ld r17, $7           ; load $7 to R17
sts 0x200, r17       ; store R17 value to 0x200
ldi r1, 0x200        ; load the content of 0x200 to R1
sts 0x220, r1        ; store the value of R1 to the location 0x220
dec r1               ; decrement R1 value by 1
brne PC-3            ; return to the step with the current program counter – 3
```

Questions:

Q. What is the syntax error in this code?

Q. How to fix the error?

Screenshots: Attach the screenshots of the modified code and the register/memory values.

1. Show your modified code.
2. Show the final values in the memory location 0x200, 0x0220

Activity 3

The assembly codes do not correctly perform each task given. Modify the codes based on the task given for each code (40 pts).

3-1. The assembly code is created to increment value of r16 from $8 to $F and to decrement value of r17 to zero (10 pts).

```
start:
        ldi r16, $8
        ldi r17, $7
        inc r16
        dec r17
        brne PC-4
end:    rjmp end
```

Questions:

Q. Show the initial value stored at r16 and r17 after you run the first two lines. What are the final values of r16 and r17 without correction?

Q. What are the expected final values based on the given task? How to modify the code to do correct operation? Show the modified code and the final values of R16 and R17 after the correction.

3-2. The assembly code is programmed to accomplish two tasks based on the arithmetic operation for R16 and R17 (30 pts).

Task 1) R16 = R16 + (R20 – R18)
Task 2) R17 = R17 + R16

```
start:
        ldi r16, $19
        ldi r17, $e5
        ldi r18, 3
        sub r18, r20
        add r18, r16
        add r16, r17
end:    rjmp end
```

Questions:

Q. Show the initial value stored at R16, R17, R18, and R20 before you run the arithmetic operation. What are the mathematical expressions of the R16, R17, R18, and R20 without correction? What are the final values for these registers after you run the code without correction?

Q. What are the expected final values of R16 and R17 based on the given task? How to modify the code to do the correct operation? Show the modified code and the corrected values of R16 and R17.