Electrical and Computer Engineering
EE3612Microprocessor Systems Laboratory

Lab 3: Introduction to Assembly Language Programming PART 2

Fall 2019

OBJECTIVES:

- To learn how to program assembly code using AVR assembly instructions
- To learn how to read and use flags in AVR assembly program
- To learn how to manipulate the sequence of the code using branching instructions and labels

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: Load numbers to GPRs and memory locations
- Activity 2: Basic arithmetic operations and flags
- Activity 3: Program sequence control using branching instructions

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 – 9/24/2019 2:59pm, section 02 – 9/26/2019 8:59am

ACTIVITIES:

Activity 1

Write assembly codes to perform the given instructions (40 pts).

1.1 Load the number 57 in decimal directly to the three GPRs. You select three GPRs and load the number 57 with three different number format (20 pts).

Requirements:

1) Load the decimal number 57 in the first GPR

2) Load the hex format of the number 57 in the second GPR

3) Load the binary format of the number 57 in the third GPR.

- Your code must show three different formats of the number 57 in decimal in three different registers.
- Your code must stop at the end of the code with the infinite loop using a label
  (ex. here: rjmp here)

1.2 Load the number $9A to the data memory location 0x0100 and load the number $B2 to the data memory location 0x0150. Then, swap the values of the memory locations (20 pts).

Requirements:

1) Load the number $9A in 0x0100

2) Load the number $B2 in 0x0150

3) Swap the numbers, so the final values of the locations must be $B2 in 0x0100 and $9A in 0x0150.

- Your code must show the loading the values in the memory locations and the process of swapping the contents of the location 0x0100 and 0x0150.
- Your code must stop at the end of the code with the infinite loop using a label
  (ex. here: rjmp here)

Activity 2

Write assembly codes to perform the basic arithmetic operations and observe the flags related the arithmetic operations (30 pts).

2.1 Add the numbers, $41 and $E8 and store the sum in the memory location 0x0110. Find the flags that change and list them (10 pts).

Requirements:

1) Load $41 and $E8 in GPRs

2) Add the numbers

3) Store the sum in the memory location 0x0110

4) Find the flags that change and list them

- Your code must show the operation and the flags that change when the arithmetic operation is completed.
- Your code must stop at the end of the code with the infinite loop using a label
  (ex. here: rjmp here)

2.2 Store the number $59 to the location 0x0200 and store the 2's complement of $59 to the location 0x0201. Fine the flags that change when you perform the arithmetic operation and list them (10 pts).

Requirements:

1) Load $59 in a GPR and store the number in the memory location 0x0200

2) Find the 2's complement of $59 and store in the memory location 0x0201

3) Find the flags that change and list them

- Your code must show the operation and the flags that change when the arithmetic operation is completed.
- Your code must stop at the end of the code with the infinite loop using a label
  (ex. here: rjmp here)

2.3 Subtract $8 from $10 twice and find the flags that change when you perform the arithmetic operation and list them (10 pts).

Requirements:

1) Load $8 and $10 in GPRs

2) Subtract $8 from $10, then find the flags that change

3) Subtract $8 from the result of 2), then find the flags that change

- Your code must show the operation and the flags that change when the arithmetic operation is completed.

- Your code must stop at the end of the code with the infinite loop using a label
  (ex. here: rjmp here)

Activity 3

Write assembly codes to perform the iterations required to complete the task using the AVR branching instructions (30 pts).

Store $6 in the memory location 0x0100. Decrease the value by 1 and store each value in the next location. Complete the task with two branch instructions and labels. **The branching instructions must use different flags**. Explain the branching instructions you select.

You write two separate codes with each branch instruction.

Requirements:

1) Load $5 in a GPR and store the value in the memory location 0x0100

2) Decrement the value by 1 each loop and store the decreased values in the memory locations (see the following order)

0x0100 = $5

0x0101 = $4

0x0102 = $3

0x0103 = $2

0x0104 = $1

0x0105 = $0

3) Select one AVR branching instruction and write an assembly program to accomplish the task.

4) Select another AVR branching instruction that uses a different flag from the previous branching instruction in 3) and write another assembly program to accomplish the task.

- You must explain the selected branching instructions and their flags.
- You must also show the values in the specified memory locations at the end of the operation.
- Your code must stop at the end of the code with the infinite loop using a label
  (ex. here: rjmp here)