ECE-3226/CSCI-3451

Lab #2: Introduction to Assembly Programming & Using Basic Operations

Contents:

- Overview
- Pre-lab
- To Be Submitted
- Lab Assignment

Overview

Purpose: The purpose of this lab is to teach you to begin constructing your own assembly programs. In particular, you will learn:

- how to construct your own assembly programs (using existing programs as guides)
- the function of some basic instructions, including:
  - arithmetic instructions (`add`, `sub`, `inc`, `dec`, etc.)
  - shift and rotate instructions (`lsr`, `asr`, etc.)
  - a jump instruction (`rjmp`)
- how to perform arithmetic on multi-byte numbers (i.e. 16-bit and 32-bit numbers)

Pre_lab

Please turn in the following parts at the beginning of the lab hour.

Part1a , Part1b , Part2b , Part2d , Part3b

Lab Assignment

- **Part 1:**

The following program contains a loop that counts up by 1:

```
        clr  r10
 Loop:
         inc  r10
         rjmp Loop
```

a. **Program:** Create a similar program that counts up by the even numbers (i.e. 0, 2, 4, 6, 8, ...)

Question 1_1: How would you modify the program you created to count up by the odd numbers (i.e. 1, 3, 5, 7, 9, ...)

b. **Program:** Create a program that starts at 255 and counts down by 4s (i.e. 255, 251, 247, 243, 239, ...)

- **Part 2:**

a. Consider the following program, execute one instruction at a time and list content of the R20 register.

```
ldi  r20, 0xBA
lsr  r20
lsr  r20
lsr  r20
lsr  r20
lsr  r20
```

Question 2_1: What is the decimal value in register r20 after executing each instruction? List the hex and decimal values in a table in your report.

Question 2_2: Considering the result after executing each LSR instruction, what mathematical operation is being performed by the LSR instruction?

Question 2_3: What mathematical operation is the above program (in its entirety) performing?

b. Program: Write a program that divides the decimal number 103 by 8.

c. Consider the following similar program, execute one instruction at a time and list content of the R20 register.

```
ldi  r20, 0xBA
 asr  r20
 asr  r20
 asr  r20
 asr  r20
 asr  r20
```

Question 2_4: What are the equivalent decimal values for both signed and unsigned representations of content of register r20 after executing each instruction? List the hex and both signed and unsigned decimal values in a table in your report.

Question 2_5: Does there seem to be any corresponding mathematical operation being performed by the ASR for unsigned numbers? How about for signed numbers?

Question 2_6: What complementary instruction exists in the instruction set that performs a multiply by 2?

d. Program: Write a program that multiplies the decimal value of 9 by 16.

Question 2_7: Does this instruction work for multiplying both signed and unsigned numbers by 2? If not, which representation does it work for?

e. Program: There is another way to create a program that multiplies the decimal value of 9 by 16, which requires only 3 instructions (including the initial LDI instruction). Give that program.

- **Part 3:**

  a. Consider the following program:

```
.equ NumA = 8600
.equ NumB = 6600
```

Then copy the rest of the assembly code into the normal code section of your assembly program:

```
ldi  r16, LOW(NumA)  ; put the lowest 8 bits of NumA in r16
ldi  r17, HIGH(NumA) ;put the highest 8 bits of NumA in r17
ldi  r20, LOW(NumB)  ;put the lowest 8 bits of NumB in r20
ldi  r21, HIGH(NumB) ;put the highest 8 bits of NumB in r21
add  r16, r20
adc  r17, r21
```

Question 3_1: What do the initial two lines you placed in the 'Declarations' section of your assembly program do? What is the `.equ` directive for?

Question 3_2: What results are stored in registers r16 and r17 after the program executes? What does this program do?

Question 3_3: Would the results be different in the instruction "`adc r17, r21`" was replaced with the instruction "`add r17, r21`"? How so? Why is the "`adc`" instruction needed?

Question 3_4: How would you modify the program to <u>subtract</u> NumB from NumA? Try out your modification. Does it perform as expected? Provide the result.

b. Program: Write a program that adds the value 150 million to the value 1.25 billion. The `LDI` instructions for NumA are given for you below:

```
ldi r16, LOW(NumA) ;put the lowest 8 bits of NumA in r16
ldi r17, BYTE2(NumA) ;put the next 8 bits of NumA in r17
ldi r18, BYTE3(NumA) ;put the next 8 bits of NumA in r18
ldi r19, BYTE4(NumA) ;put the highest 8 bits of NumA in
                      ; r19
```

Question 3_5: Find (and report) a pair of numbers for NumA and NumB that result in a carry of 1 between exactly two of the four pairs of byte additions between NumA and NumB. Show the numbers in hex.

Last modified: Monday 8/30/18