**ECE-3226-50:** Lab #1

Introduction to AVR Studio

Robert Campbell

**Objective:**

To introduce and gain a familiarity with AVR Studio 7 and assembly programming.

**Equipment:**

AVR Studio 7

**Procedure:**

1. A simple program to demonstrate the clear, decrement, negation, and compliment instructions.
2. jmp start  
   start: clr r18  
    dec r18  
    neg r18  
    clc  
    com r18  
   end: rjmp end



Question 1-1) What is the value of the C bit in the SREG register after execution of :

a. the DEC R18 instruction? 0

b. the NEG R18 instruction? 1

c. the COM R18 instruction? 1

Question 1-2) Explain how the C bit in the SREG register is updated by the three instructions mentioned in Question 1-1. (Use the instruction set document to answer this question.)

1. C does not change
2. C sets to 1 if there is an implied borrow, otherwise it is 0
3. C is set to 1
4. A simple program to demonstrate the load, increment, and add instructions.
5. jmp start  
   start: clr r18  
    clc  
    ldi r18 , 0xFF  
    inc R18  
    clr r18  
    clc  
    ldi r18 , 0xFF  
    Ldi r17 , 1  
    add r18 , r17  
    end: rjmp end
6. 

Question 2-1) What is the value of the C bit in the SREG register after execution of :

a. the INC R18 instruction? 0

b. the Add R18 , R17 instruction? 1

Question 2-2) Explain how the C bit in the SREG register is updated by the two instructions mentioned in Question 2-1. (Use the instruction set document to answer this question.)

1. The C bit does not update with the INC instruction
2. C bit is set to 1 if there is a carry out from the MSB
3. A simple program to demonstrate the logical and, compare, and subtraction instructions.
4. jmp start  
   start: ldi r17, 0xA7  
    andi r17, 0xC5  
    ldi r17, 0x5F  
    cpi r17, 0xFF  
    subi r17, 0xFF  
    end: rjmp end
5. 

Question 3-1) Explain the similarities and differences between the CPI and SUBI instructions.

The CPI instruction subtracts the value in the register from the constant, but only updates the SREG and doesn’t store the calculated value. SUBI stores the value in the provided register.

1. A simple program that demonstrates the add and subtract instructions.
2. jmp start  
   start: ldi r16, 0x7B  
    ldi r17, 0x49  
    add r17, R16  
    ldi r18, 0xF3  
    ldi r19, 0x63  
    add r19, R18  
    ldi r20, 0x45  
    ldi r21, 0xE3  
    sub r21, R20  
    end: rjmp end
3. 

Question 4-1) Assuming signed operands, are the results of R17, R19 and R21 valid signed results after the execution of the program? Give explanations for your answer.

R17: After the addition, the overflow bit was set to 1, so the result is an invalid number.

R19: One of the operands was larger than +127, so it is an invalid input for 8-bit signed numbers. The result is also greater than +127, so it is also an invalid 8-bit signed number.

R21: One of the operands is greater than +127, so it is an invalid 8-bit signed number. The result, -158, is an invalid 8-bit signed number. However, it is valid because the magnitude 158 is stored in the register with the SREG sign bit indicating that the value is negative.

Question 4-2) Assuming unsigned operands, are the results of R17, R19 and R21 valid unsigned results after the execution of the program? Give explanations for your answer.

R17: The operation overflows, as indicated by the overflow bit.

R19: The result would be an invalid 8-bit number (greater than 255), but the carry bit accounts for the difference.

R21: It is an invalid unsigned result as the result is less than 0.

**Discussion/Conclusion:**

This lab gave a basic introduction to AVR Studio 7 and a basic understanding of assembly programming. There were no problems in doing the lab.