ECE 375 Lab 3

Introduction to AVR Simulation with Atmel Studio

**Lab Time: Tuesday 8-10**

Zachary DeVita

# Study Questions

1) What is the initial value of DDRB?

00000000

2) What is the initial value of PORTB?

00000000

3) Based on the initial values of DDRB and PORTB, what is Port B’s default I/O configuration?

When DDRB is set to zero then it must be in input mode. PORTB values set the configuration of the port. Zeros for PORTB indicate that it is in “pull-down” mode, ones indicate “pull-up” mode. When in pull-down mode, input from PINB is received when a bit is changed to a one.

*4) What 16-bit address (in hexadecimal) is the stack pointer initialized to?*

0x10FF

*5) What are the contents of register r0 after it is initialized?*

0xFF

*6) How many times did the code inside of LOOP end up running?*

4 times

*7) Which instruction would you modify if you wanted to change the number of times that the loop runs?*

I would change the second argument to ‘ldi’ (load indirect) from $04 to the hex value representation of the desired number of iterations. For instance, if I wanted 11 iterations I would replace $04 with $0b

*8) What are the contents of register r1 after it is initialized?*

0xAA

*9) What are the contents of register r2 after it is initialized?*

0x0F

*10) What are the contents of register r3 after it is initialized?*

0x0F

*11) What is the value of the stack pointer when the program execution is inside the FUNCTION subroutine?*

0x10FD

*12) What is the final result of FUNCTION? (What are the hexadecimal contents of memory locations $0105:$0104)?*

$0104: 0E $0105: BA

# Challenge

1) What type of operation does the FUNCTION subroutine perform on its two 16-bit inputs? How can you tell? Give a detailed description of the operation being performed by the FUNCTION subroutine.

8-bit pieces of data are taken from memory and loaded into registers X, Y & Z. Two of these 8-bit pieces of data are stored in each of the aforementioned registers; one in the low bit and one in the high bit. The data can then be used to represent 16-bit numbers. Register X is then loaded into register A and post-incremented. Register Y is then loaded into register B and post-incremented. A and B are then added together and the result is stored in the memory location pointed at by register Z. Z is then post-incremented. X is then loaded into register A, and Y to register B. A and B are then added together, but this time the carry is included in the addition. The result is stored in the memory pointed at by Z, and then Z is post-incremented. The next branch statement looks for a carry flag, and if there isn’t one found then the function exits, else the XH value is moved to the Z register.

*2) Currently, the two 16-bit inputs used in the sample code cause the “brcc EXIT” branch to be taken. Come up with two 16-bit values that would cause the branch NOT to be taken, therefore causing the “st Z, XH” instruction to be executed before the subroutine returns.*

A = FFFF, B = FFFF

*3. What is the purpose of the conditionally-executed instruction “st Z, XH”?*

It incorporates the carry if needed into the result; if there is a carry.