Aaron Rito
ECE 375
prelab 5

1. List and describe all of the addition, subtraction, and multiplication instructions (i.e. ADC, SUBI, FMUL, etc.) available in AVR's 8-bit instruction set.

5. ADC – Add with Carry.
6. ADD – Add without Carry
7. ADIW – Add Immediate to Word
8. AND – Logical AND
9. ANDI – Logical AND with Immediate
10. ASR – Arithmetic Shift Right
48. COM – One's Complement
53. DEC – Decrement
58. EOR – Exclusive OR
59. FMUL – Fractional Multiply Unsigned.
60. FMULS – Fractional Multiply Signed
61. FMULSU – Fractional Multiply Signed with Unsigned
65. INC – Increment
81. MUL – Multiply Unsigned
82. MULS – Multiply Signed
83. MULSU – Multiply Signed with Unsigned
84. NEG – Two's Complement
86. OR – Logical OR
87. ORI – Logical OR with Immediate
95. ROL – Rotate Left trough Carry
96. ROR – Rotate Right through Carry
97. SBC – Subtract with Carry
98. SBCI – Subtract Immediate with Carry SBI – Set Bit in I/O Register
102. SBIW – Subtract Immediate from Word
123. SUB – Subtract Without Carry
124. SUBI – Subtract Immediate

2.Write pseudocode for an 8-bit AVR function that will take two 16-bit numbers (from data memory addresses $0111:$0110 and $0121:$0120), add them together, and then store the 16-bit result (in data memory addresses $0101:$0100).

```
lpm $0111:$0110 onto r16, r17
lpm $0121:$0120 onto r18, r19
adc r16 , r18
if carry inc r17
add r17, r 19
ST 0100, r16
ST 0101, r17
```

3. Write pseudocode for an 8-bit AVR function that will take the 16-bit number in $0111:$0110, subtract it from the 16-bit number in $0121:$0120, and then store the 16-bit result into $0101:$0100.

```
lpm $0111:$0110 onto r16, r17
lpm $0121:$0120 onto r18, r19
sub r16 , r18
sbc r17, r 19
```

```
if carry LSR r16
ST 0100, r16
ST 0101, r17
```