

Lab #2

9/12/19

Sung Choi

\_\_\_\_\_\_\_

915596703

Von Kaukeano

001 / 002

3613

## **ACTIVITIES:**

## **Activity 1**

## Read each code and write its own mathematical or logical expression (30 pts).

Run the following codes in the Atmel Studio environment. Examine the values of each register. Fill each section (it’s operation) with mathematical or logical expression. Also, attach the *lss* file screenshots after you run the code. (See the given example in Figure 1).

|  |  |  |
| --- | --- | --- |
| **Section** | **Assembly Code** | **Mathematical and Logical Expression** |
| Example | LDI R17, 0xA3  LDI R18, 4  ADD R17, R18  DEC R18  BRNE PC-2 | R17 = 0xA3  R18 = 4  R17 = R17 + R18  R18 = R18 – 1  If Z=0, then PC=PC-2, else PC=PC+1 (exit the loop) |
| *lss* screenshot |  | |

**<Figure 1. Example of *lss* file screenshot>**

|  |  |  |
| --- | --- | --- |
| **Section** | **Assembly Code** | **Mathematical Expression** |
| 1-1. | LDI R22, $14  LDI R21, $4  SUB R22, R21  BRNE PC-1 | R22 = $14  R21 = $4  R22 = R22 – R21  If Z = 0, then PC = PC – 1, else PC = PC + 1. Finish |
| *lss* screenshot |  | |
| 1-2. | LDI R16, $15  LDI R17, $22  ADC R16, R17  BRCC PC-1 | R16 = $15  R17 = $22  R16 = R16 + R17 w/ carry  If C = 1, then PC = PC -1, else PC+ 1. Finish |
| *lss* screenshot |  | |
| 1-3. | LDI R25, $15  LDI R26, $52  DEC R26  CP R25, R26  BRNE PC-2 | R25 = $15  R26 = $52  R26 = R26 – 1  R25 – R26, If R26 < R25 will create carry.  If C = 1, then PC = PC -2, else PC+ 1. Finish |
| *lss* screenshot |  | |

## **Activity 2**

## Read the error message and correct the error parts in the code (30 pts).

2-1. This assembly code loads the hex values to the general purposed registers, R16 and R17. Then, it swaps the contents of the registers. Answer the questions.

ldi r16, 560 ; load a value 560 in decimal to R16

ld r17, 0x81 ; load a value 0x81 in hex to R17

; The following section is for swapping the values of the register 16 and 17

mov r22, r16 ; move the value of R16 to R22

mov r16, r17 ; move the value of R17 to R16

mov r17, r22 ; move the value of R22 to R17

**Questions:**

Q. What is the syntax error in this code?

Answer:

LD should be LDI

560 is out of range. Too high for the amount of bytes it holds

Q. How to fix the error?

Answer:

Reduce to less than 255.

Change LD to LDI.

**Code and Result:**

Show the modified code and attach the screenshots of the register values.

1) Show your modified code. (Copy and Paste the code)

; Created: 9/12/2019 9:35:55 AM

; Author : Von Kaukeano

; Replace with your application code

start:

LDI R16, 255

LDI R17, 0x81

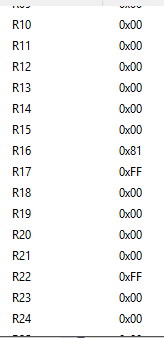
mov R22, R16

mov R16,R17

mov R17,R22

here: rjmp here

2) Show the final value of the register 16 and 17. (Screenshots)



2-2. This assembly code loads four hex values ($10, $42, $a8, and $11) to the address 0x0100, 0x101, 0x0102 and 0x0103 in the data memory.

Ldi $10, r0 ; load a hex value $10 to a GPR

sts 0x100,r0 ; store the value to the memory location 0x0100

ldi r17, $42 ; load a hex value $42 to a GPR

sts 0x101,r17 ; store the value to the memory location 0x0101

ldi r1, $a8 ; load a hex value $a8 to a GPR

sts 0x102,r1 ; store the value to the memory location 0x0102

ldi r18, $11 ; load a hex value $11 to a GPR

sts 0x103,r18 ; store the value to the memory location 0x0103

**Questions:**

Q. What is the syntax error in this code?

Answer:

The register number should be before the hex value.

Cannot load into R0-R15.

Q. How to fix the error?

Answer:

LDI R20,$10

LDI R16 $A8

**Code and Result:**

Show the modified code and attach the screenshots of the values in the memory locations.

1) Show your modified code. (Copy and paste the code)

2) Show the final values of the memory location 0x100, 0x101, 0x102, and 0x103 (Screenshot)

start:

LDI R20, $10

STS 0x100, R0

LDI R17, $42

STS 0x101, R17

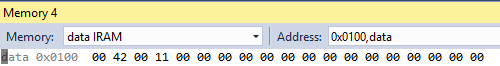
LDI R16,$A8

STS 0x102, R1

LDI R18, $11

STS 0x103,R18

here: rjmp here



2-3. This assembly code loads value $7 to the memory address 0x0200. Then, load the contents of 0x0200 to the memory location 0x220. The content of the location 0x220 is decremented continuously until the value hits zero by using the register R1.

ld r17, $7 ; load $7 to R17

sts 0x200, r17 ; store R17 value to 0x200

ldi r1, 0x200 ; load the content of ox200 to R1

sts 0x220, r1 ; store the value of R1 to the location 0x220

dec r1 ; decrement by 1

brne PC-3 ; return to the step with the current program counter – 3

**Questions:**

Q. What is the syntax error in this code?

Answer:

LD should be LDI

Cannot load into R1 and 0x200 is out of range

Q. How to fix the error?

Answer:

LDI R17, $7

LDI R16 0x20

**Code and Result:**

Show the modified code and attach the screenshots of the values in the memory locations.

1. Show your modified code. (Copy and paste the code in the box)

start:

LDI R17, $7

STS 0x200, R17

LDI R16, 0x20

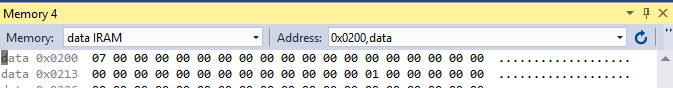
STS 0x220,R1

DEC R16

BRNE PC-3

here: rjmp here

2. Show the final values in the memory location 0x200, 0x0220 (Screenshots)



## **Activity 3**

## The assembly codes do not correctly perform each task given. Modify the codes based on the task given for each code (40 pts).

3-1. The assembly code is created to increment value of r16 from $8 to $F and to decrement value of r17 to zero (10 pts).

start:

ldi r16, $8

ldi r17, $7

inc r16

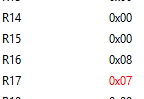
dec r17

brne PC-4

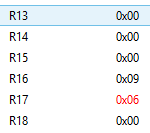
end: rjmp end

**Questions:**

Q. Show the **initial** value stored at R16 and R17 after you run the first two lines.. (Attach the screenshot)



What are the **final** values of r16 and r17 without correction? (Attach the screenshot)



Q. What are the **expected final values** of R16 and R17 based on the given task?

Answer:

R16 = 0F

R17 = 0

How to modify the code to do correct operation? Show the modified code and the final values of R16 and R17 after the correction. (Copy and paste the code and attach the screenshot of the R16 and R17 values)

**Code:** modified code (copy and paste)

start:

LDI R16, $8

LDI R17, $7

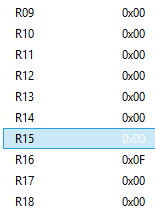
INC R16

DEC R17

BRNE PC-2

end: rjmp end

**Result:** R16 and R17 values (Screenshot)



3-2. The assembly code is programmed to accomplish two tasks based on the arithmetic operation for R16 and R17 (30 pts).

Task 1) R16 = R16 + (R20 – R18)

Task 2) R17 = R17 + R16. (use the R16 value from the Task 1)

start:

ldi r16, $19

ldi r17, $e5

ldi r18, 3

sub r18, r20

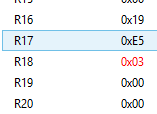
add r18, r16

add r16, r17

end: rjmp end

**Questions:**

Q. Show the initial value stored at R16, R17, R18, and R20 before you run the arithmetic operation. (Attach the screenshot)



What are the mathematical expressions of the R16, R17, R18, and R20 without correction? What are the final values for these registers after you run the code without correction?

Answer:

R18 = R18 – R20

R18 = R18 + R16

R16 = R16 + R17

R16 = 0xFE ; R17 = 0xE5 ; R18 = 0x1C ; R20 = 0x00

Q. What are the expected final values of R16 and R17 based on the given task?

Answer:

R16 = 16

R17 = FB

How to modify the code to do the correct operation? Show the modified code and the corrected values of R16 and R17. (Copy and paste the code and attach the screenshot of the R16 and R17 values)

**Code:** modified code (copy and pasted)

start:

LDI R16, $19

LDI R17, $e5

LDI R18, 3

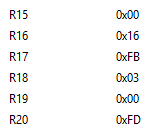
SUB R20, R18

ADD R16, R20

ADD R17, R16

end: rjmp end

**Result:** R16 and R17 values (Screenshot)



**ECE3613 Processor System Laboratory Rubric**

**Lab #: 2**

**Section: 001 / 002**

**Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Activity** | **Section** | **Task** | **Full Points** | **Earned Points** | **Comment** |
| 1 | 1.1 | Math Exp | 5 |  |  |
|  |  | lss file screenshot | 5 |  |  |
|  | 1.2 | Math Exp | 5 |  |  |
|  |  | lss file screenshot | 5 |  |  |
|  | 1.3 | Math Exp | 5 |  |  |
|  |  | lss file screenshot | 5 |  |  |
| Subtotal | | | 30 |  |  |
| 2 | 2.1 | Question | 5 |  |  |
|  |  | Code & Result | 5 |  |  |
|  | 2.2 | Question | 5 |  |  |
|  |  | Code & Result | 5 |  |  |
|  | 2.3 | Question | 5 |  |  |
|  |  | Code & Result | 5 |  |  |
| Subtotal | | | 30 |  |  |
| 3 | 3.1 | Question | 5 |  |  |
|  |  | Code & Result | 5 |  |  |
|  | 3.2 | Question | 15 |  |  |
|  |  | Code & Result | 15 |  |  |
| Subtotal | | | 40 |  |  |
| Total | | | 100 |  |  |