**ECEGR 2220: Microprocessor Design**

**Spring 2018**

**LAB 2 REPORT**

**Name: Don-Thuan Le**

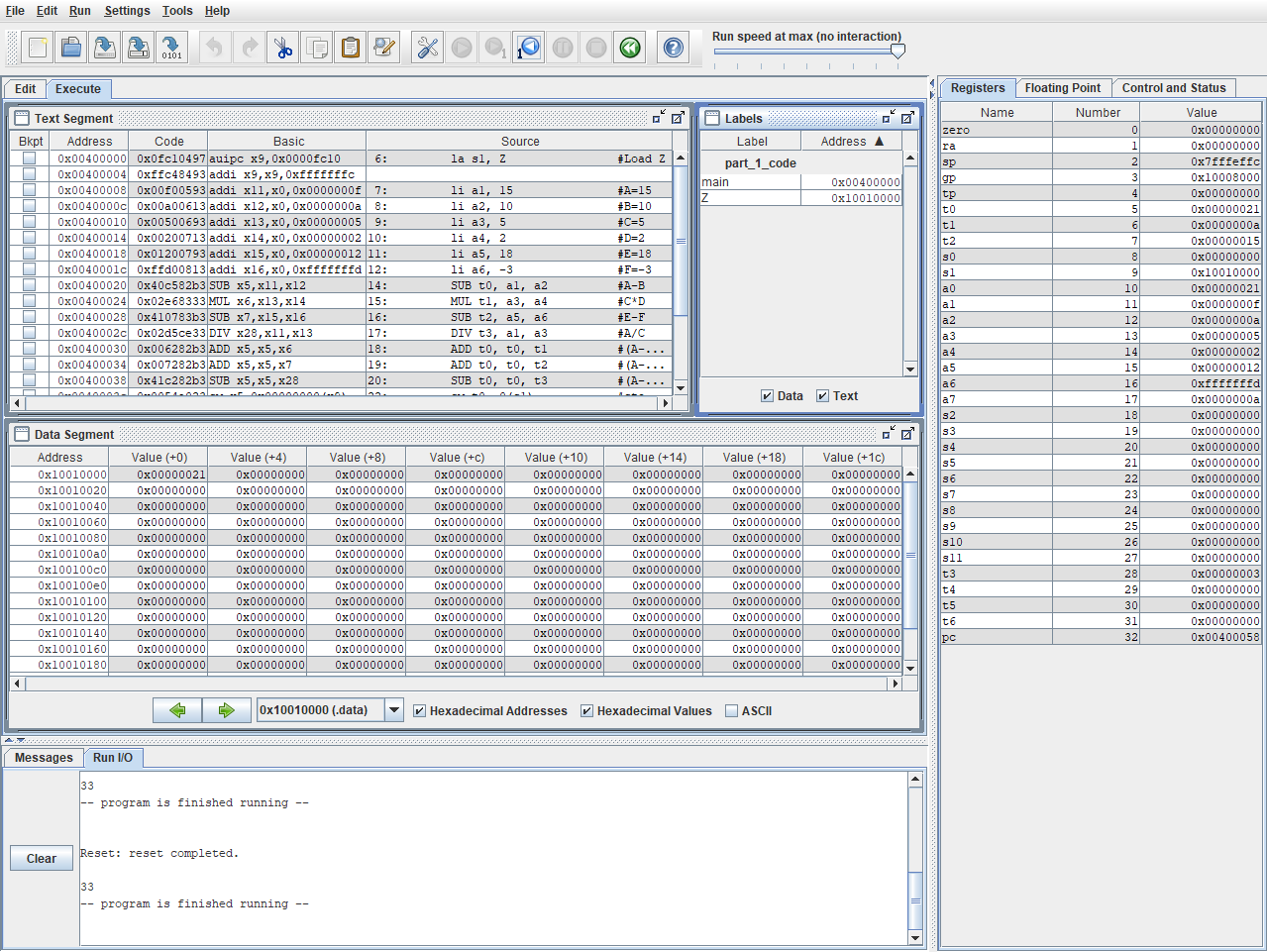
**Performed by:**

**Don-Thuan Le - Thanh Nguyen – Lauren Molina**

**Date of Report: 04/28/2018**

**SEATTLE UNIVERSITY**

**Department of Electrical and Computer Engineering**

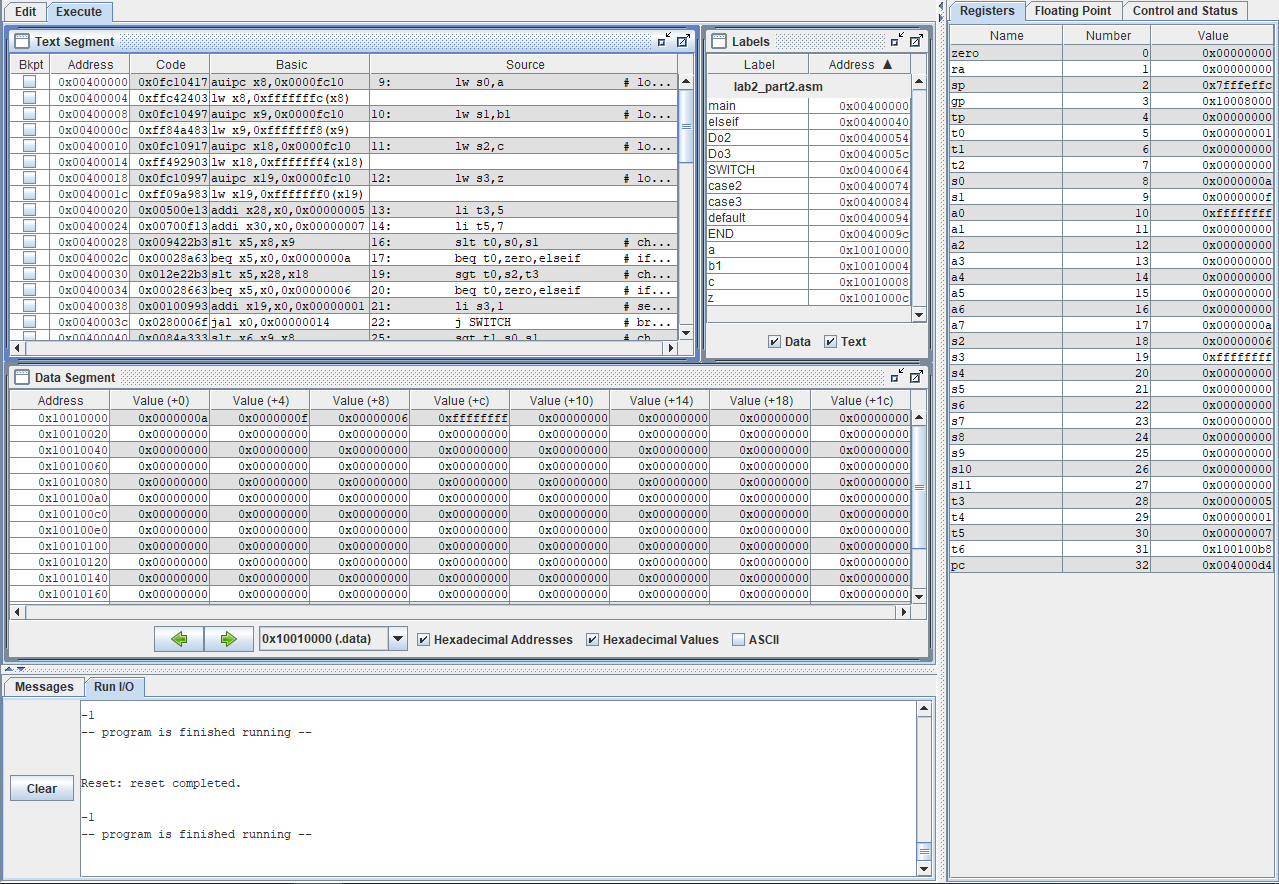
**PART 1: ARITHMETIC**

**Figure 1.** The screenshot of RARS after program for part 1 runs showing values in registers and in data memory.

* The screen shot above is the results of running the rars simulation program using the code for part 1 of the lab.
* For part 1, **A, B, C, D, E, F** are stored in registers **a1, a2, a3, a4, a5, a6**, respectively.
* From the labels tab, you can see that **Z** is saved in memory. The address of **Z** is loaded to the register **s1**. Z has the address of **0x10010000.**
* From the data segment, when the program finishes executing, the value of Z was **21 in hexadecimal (**or **33 in decimal)**, which is the correct sum of the given numbers.
* The value of Z is also printed out in decimal in Run I/O window.

**PART 2:**

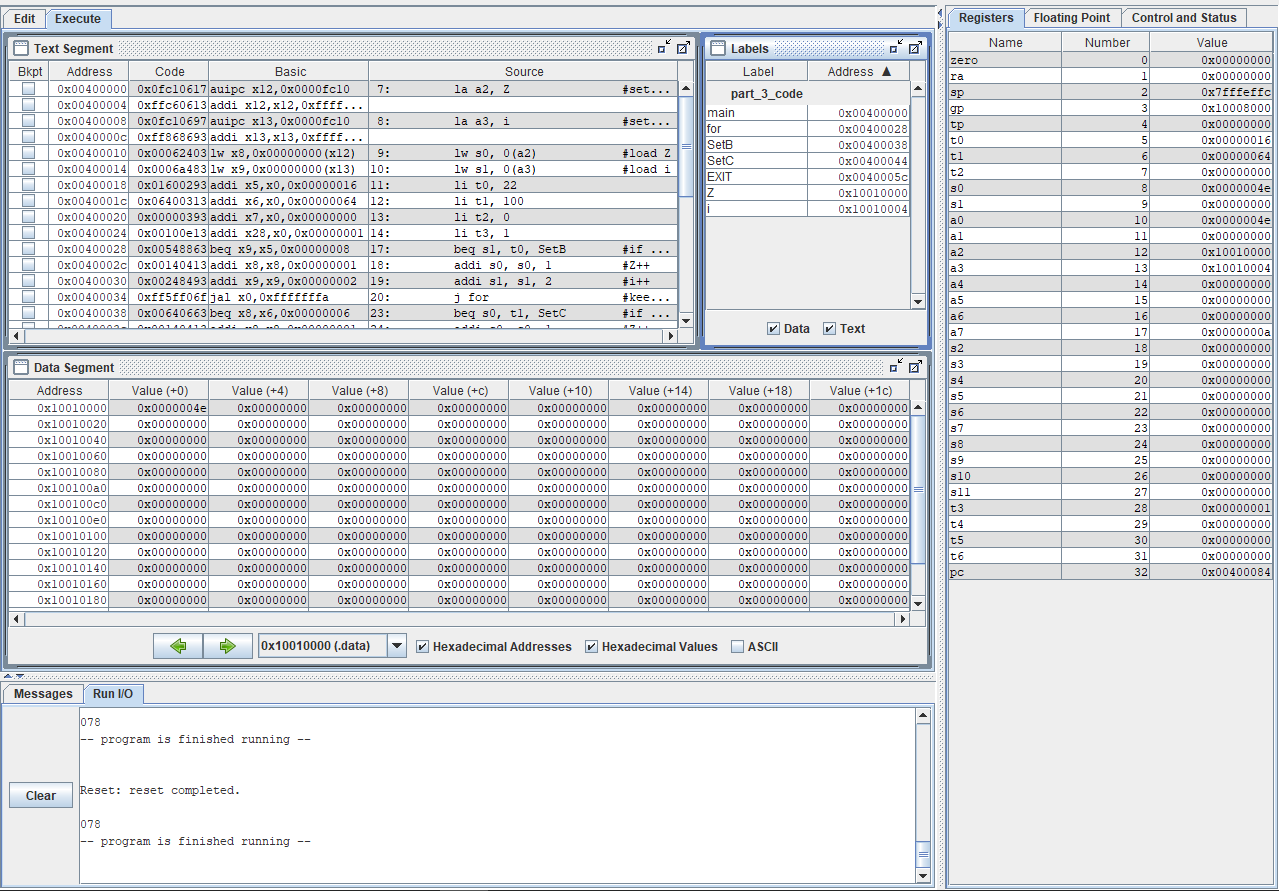
**Figure 2.** The screenshot of RARS after program for part 2 runs showing values in registers and in data memory.



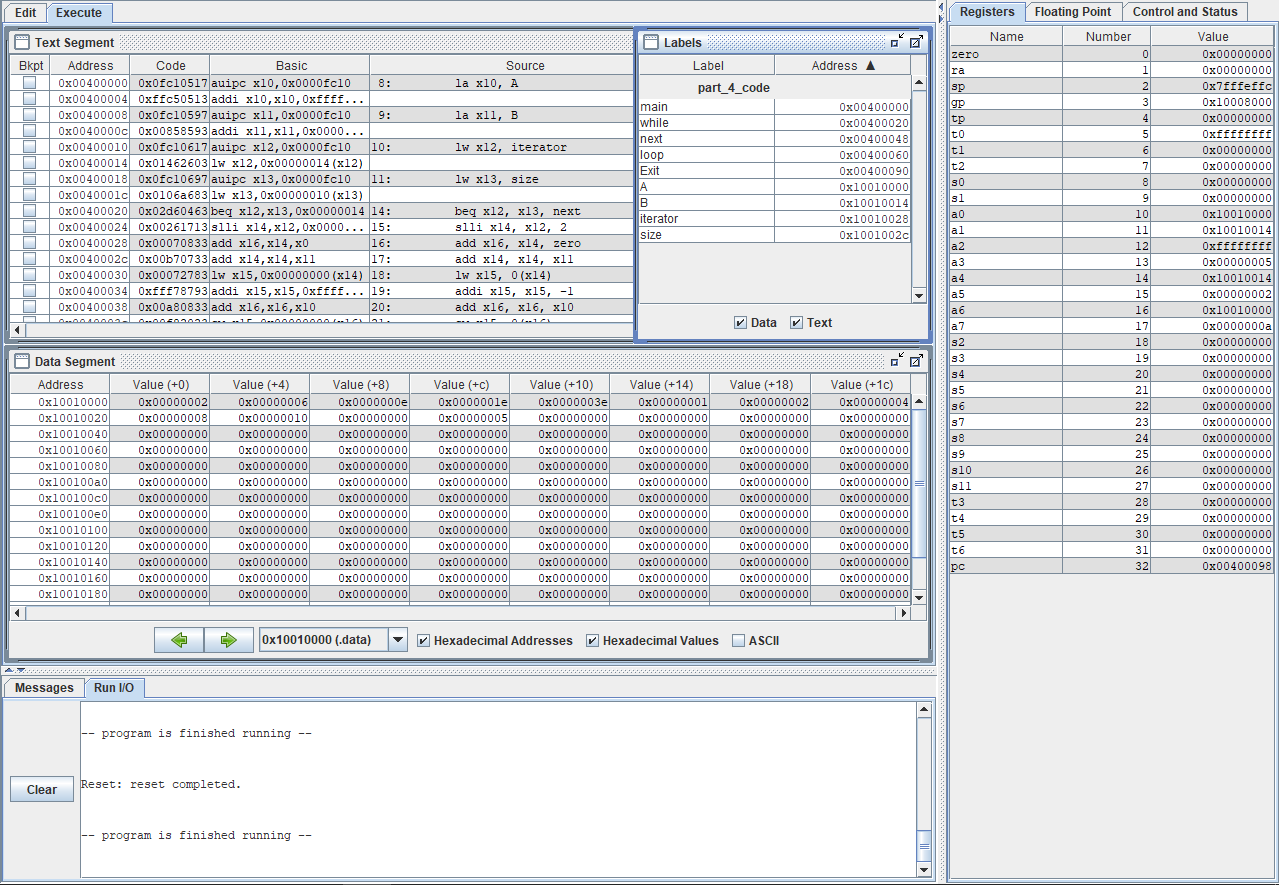
* The screen shot above is the results of running the rars simulation program using the code for part 2 of the lab. We used “b1” instead of b since “b” is a reserved word.
* From the labels tab, you can see that **A, B, C, and Z** are all saved in memory.
* In the labels tab you can see **A, B, C and Z** are all saved in memory. The addresses are:
  + **0x10010000 for A**
  + **0x10010004 for B**
  + **0x10010008 for C**
  + **0x1001000c for Z**
* Their values can also be seen in the data segment with Z giving the correct results.
* The value of Z is also printed out in decimal in Run I/O window.

**PART 3: LOOPS**

**Figure 3.** The screenshot of RARS after program for part 3 runs showing values in registers and in data memory.



* The screen shot above is the results of running the rars simulation program using the code for part 3 of the lab.
* For part 3, **Z** and **i** are integer words which are saved in memory. Their addresses are loaded into registers **a2**, **a3** respectively and their values are loaded into registers **s0**, **s1** respectively.
* From the labels tab you can see that Z and I are both saved in memory.
* From Registers tab the labels tab, the addresses are:
  + **0x10010000** for **Z**
  + **0x10010004** for **i**
* Looking up these address in the data segment, we can see:
  + The value of **Z** is shown as **4e** in hexadecimal, which is **78** in decimal. This value of **78** is also printed out in Run I/O tab. This is correct number that Z should end on before the program finishes its execution.
  + The value of i is shown as **0** in hexadecimal, which is **0** in decimal. This value of **0** is also printed out in Run I/O tab (first digit of “078”). This is correct number that i should end on before the program finishes its execution.

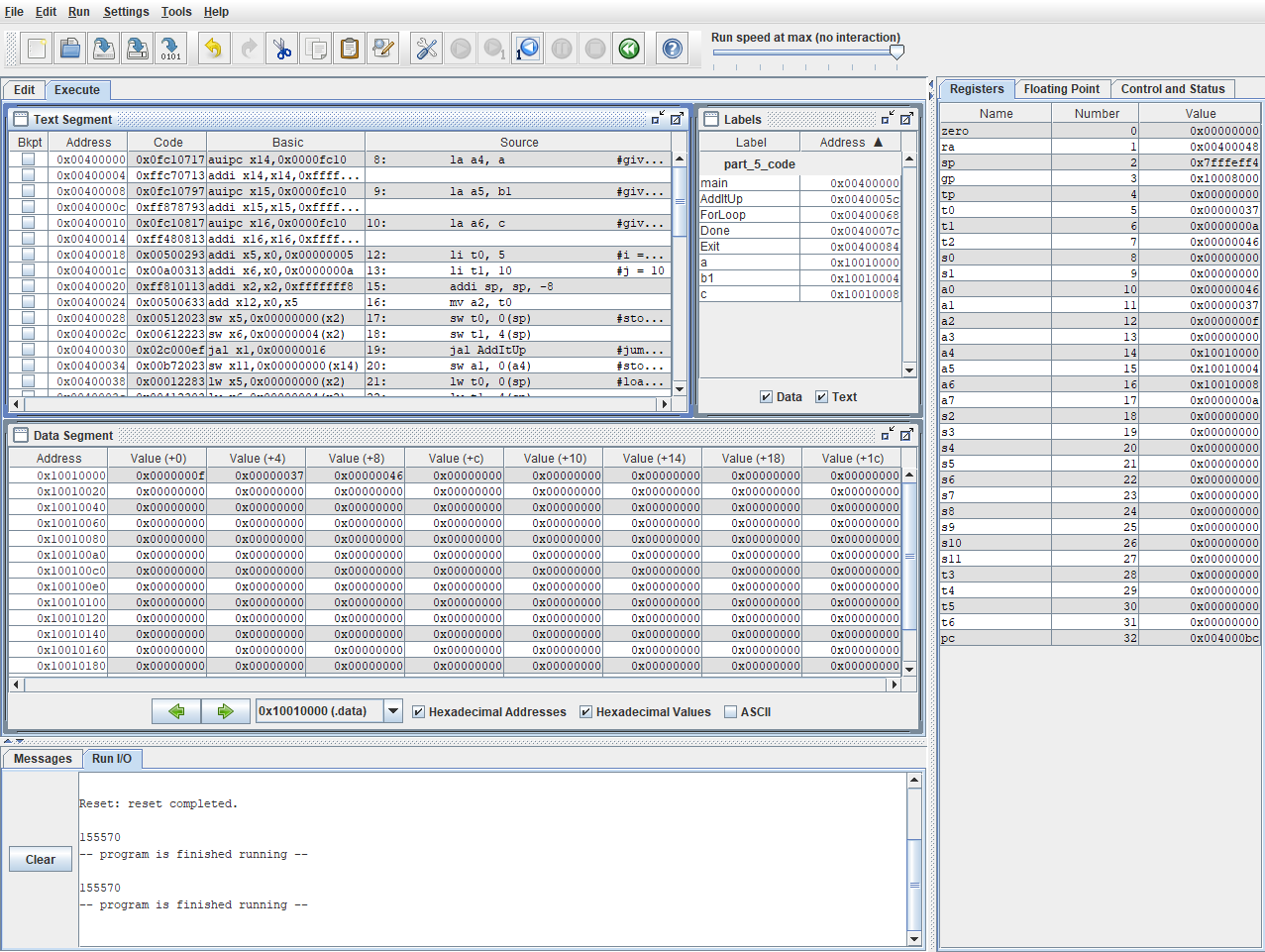
**PART 4: ARRAYS**

**Figure 4.** The screenshot of RARS after program for part 4 runs showing values in registers and in data memory.

* In part 4, array A are B are stored in memory. Their addresses are loaded into **x10, x11** respectively. Variable **i** is loaded into **x12**.
* The screen shot above is the results of running the rars simulation program using the code for part 4 of the lab.
* In the label tab you can see that the integer arrays A and B are both saved in memory.
* In the labels tab you can see **A, B, and i** are all saved in memory. The addresses are:
  + 0x10010000 for A
  + 0x100100014 for B
  + 0x100100028 for i
* The data segment shows that the values of B stay the same while the values in array A change as they are supposed to.

**PART 5: FUNCTION CALL**

**Figure 5.** The screenshot of RARS after program for part 5 runs showing values in registers and in data memory.



* In part 5, variables **a, b and c** are integer words in memory and are loaded into **a4, a5, a6** respectively. We used “b1” instead of b since “b” is a reserved word.
* The screenshot above is the results of running the rars simulation program using the code for part 5 of the lab.
* From the text segment you can see the stack was used.
* In the labels tab you can see **a, b, and c** are all saved in memory. The addresses are:
  + 0x10010000 for a
  + 0x10010004 for b
  + 0x10010008 for c
* Looking up these address in the data segment, we can see:
  + The value of **a** is shown as **f** in hexadecimal, which is **15** in decimal. This value of **15** is also printed out in Run I/O tab (first two digits of “155570”). This is correct number that **a** should end on before the program finishes its execution.
  + The value of **b** is shown as **37** in hexadecimal, which is **55** in decimal. This value of **55** is also printed out in Run I/O tab (middle two digits of “155570”). This is correct number that **b** should end on before the program finishes its execution.
  + The value of **c** is shown as **46** in hexadecimal, which is **70** in decimal. This value of **70** is also printed out in Run I/O tab (last two digits of “155570”). This is correct number that i should end on before the program finishes its execution.

**GitHub Link:**

<https://github.com/ledonthuan>