**BÁO CÁO THỰC HÀNH KTMT TUẦN 2**

**Họ tên: Nguyễn Đức Đại Dương**

**MSSV: 20225616**

Assignment 1

A screenshot of a computer

Description automatically generated

-Sự thay đổi thanh ghi $s0: từ giá trị 0x00000000 chuyển thành 0x00003007 sau lệnh thứ nhất rồi lại trở lại giá trị ban đầu sau lệnh thứ 2

-Sự thay đổi của thanh ghi pc: tăng thêm 0x00000004 sau mỗi giá trị

-So sánh mã máy:

+ Lệnh 1:

Là lệnh I, opcode: 8 => 001000, rs: 0 =>00000, rt: 16=>10000, imm:0x3007=0011 0000 0000 0111

Lệnh máy là: 0010 0000 0001 0000 0011 0000 0000 0111

+Lệnh 2: Là lệnh R, opcode: 0 => 000000, rs: 0 =>00000, rt: 0=>00000, rd: 16=>10000, sh: 0=>00000, fn: 32=>100000

Lệnh máy là: 0000 0000 0000 0000 1000 0000 0010 0000

🡺Các lệnh đúng như tập lệnh đã quy định

-Nếu sửa lệnh thứ 2 thành addi $s0, $zero, 0x2110003d thì nó thành lệnh gán 32 bit, vì số 0x2110003d cần lưu trữ ở dạng 32 bit

Assignment 2

A screenshot of a computer

Description automatically generated

-Sự thay đổi của thanh ghi $s0: từ giá trị 0x00000000 chuyển thành 0x21100000 sau lệnh thứ nhất, rồi biến thành giá trị 0x2110300d sau lệnh thứ hai

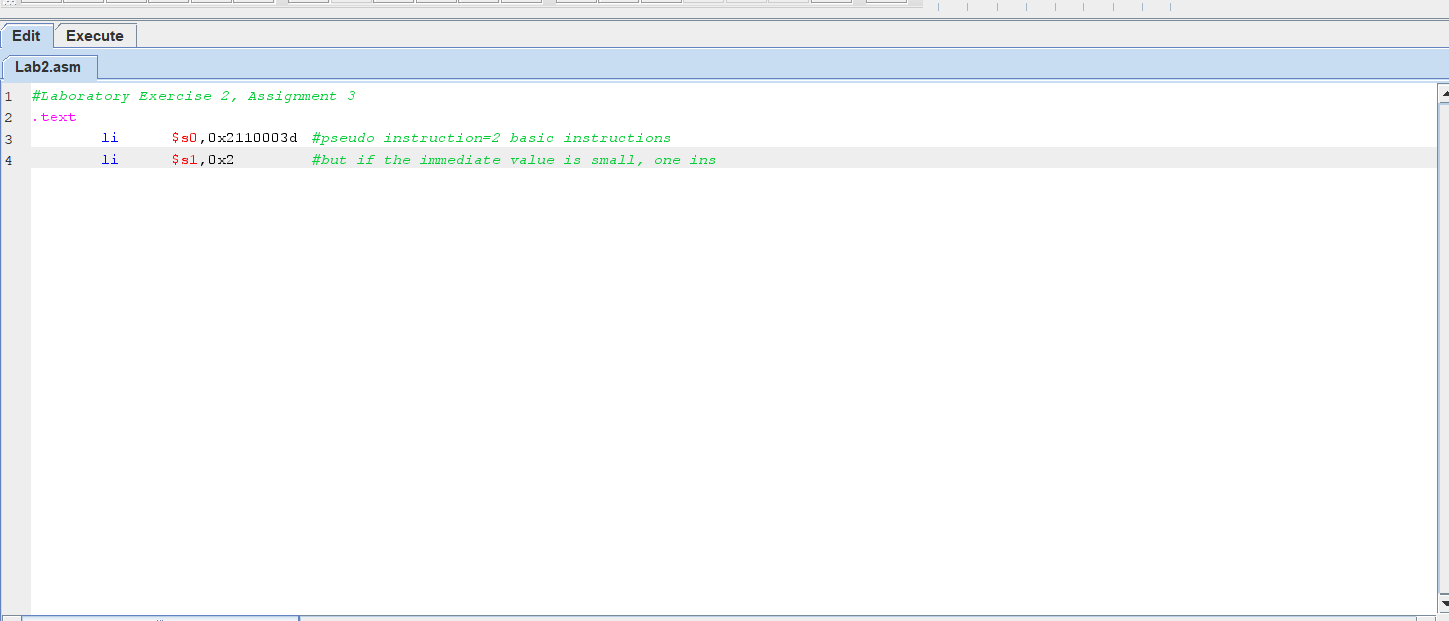
-Sự thay đổi của thanh ghi pc: tăng thêm một khoảng có giá trị là 0x00000004 sau mỗi giá trị

A screenshot of a computer

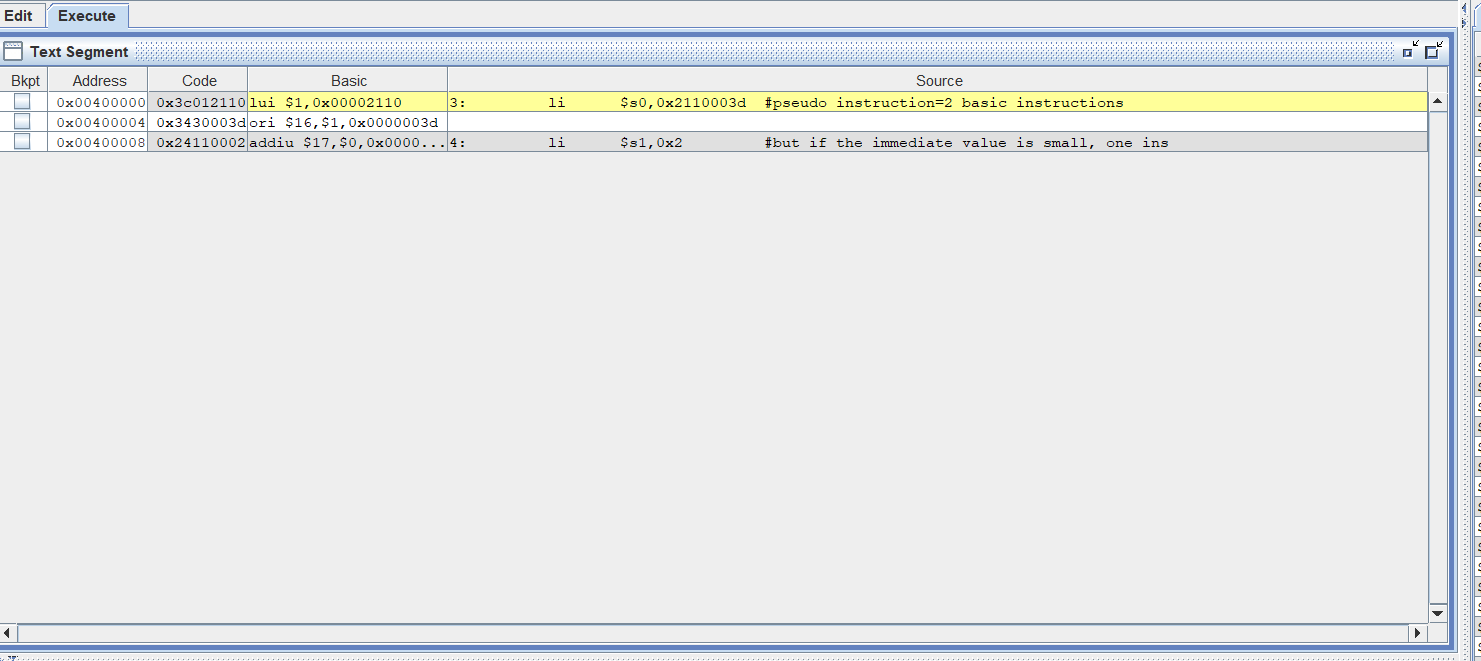
Description automatically generated

- Các byte đầu tiên ở vùng lệnh trùng với cột Code(Mã máy theo Hexa) trong cửa sổ Text Segment ở phần thực thi

Assignment 3



Kết quả thực thi:

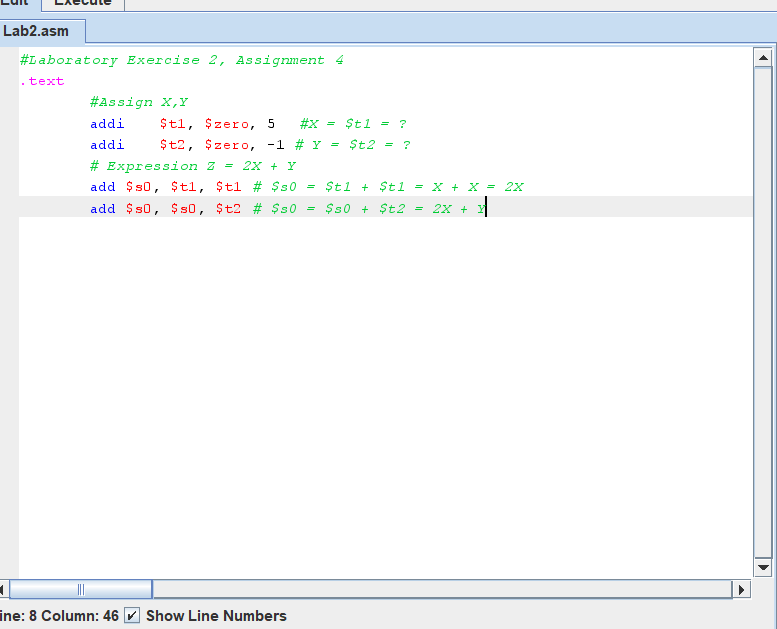


-Điều bất thường đó là lệnh li thứ nhất tự tách ra thành 2 lệnh đó là lệnh lui và lệnh ori, còn lệnh li thứ hai bị biến thành lệnh addiu

-Giải thích:

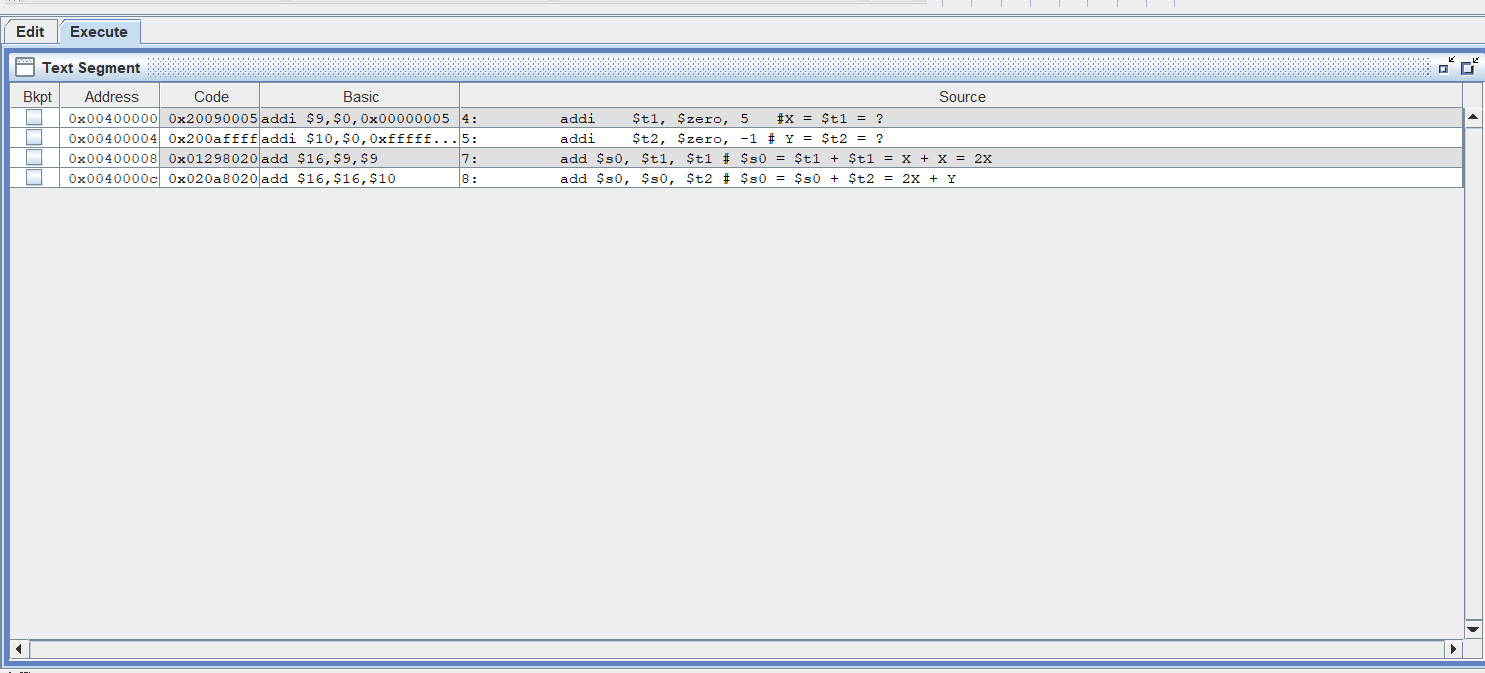
Lệnh li gán giá trị số nguyên bất kỳ, lệnh li thứ nhất thực hiện gán số 0x2110003d là số loại 32 bit nên tự động tách thành 2 lệnh như trên, còn lệnh li thứ hai thực hiện gán số 0x2 là loại số 16 bit không dấu nên chuyển thành lệnh addiu.

Assignment 4



-Có sự thay đổi của các thanh ghi: $t1 có giá trị trở thành 0x00000005, $t2 có giá trị trở thành 0xffffffff, $s0 thany đổi theo như kết quả Z khi thực hiện phép tính đã được giải thích như trong mã nguồn trên, còn thanh ghi pc cứ mỗi một lệnh tăng thêm 0x00000004.

Kết quả thực thi:



-Từ lệnh addj, điểm tương đồng với hợp ngữ và mã máy là: Sau khi chuyển các yếu tố trong lệnh loại I (addi là loại I) như opcode, rs, rt, imm thành giá trị nhị phân, và ghép lần lượt các giá trị nhị phân lần lượt: opcode-rs-rt-imm thì kết quả ra lệnh mã máy là số nhị phân 32 bit. Khi chuyển lệnh mã máy đó sang Hexa thì kết quả giống hệt như trong cột Code và dòng lệnh tương ứng như trên. Do đó nó là điểm tương đồng.

-Kiểm nghiệm các khuôn mẫu của 4 lệnh như bài 4:

addi $t1, $zero, 5

addi $s9, $0, 0x5

opcode: 8 => 001000

rs: $0 => 00000

rt: $9 => 01001

mm: 0x5 => 0000 0000 0000 0101

0010 0000 0000 1001 0000 0000 0000 0101

0x20090005

addi $t2, $zero, -1

addi $10, $0, 0xffffffff

opcode: 8 => 001000

rs: $0 => 00000

rt: $10 => 01010

imm 0xffffffff => 1111 1111 1111 1111

0010 0000 0000 1010 1111 1111 1111 1111

0x200affff

0x01298020

0000 0001 0010 1001 1000 0000 0010 0000

Opcode: 000000 => 0

rs: 01001 => $9

rt: 01001 => $9

rd: 10000 => $16

sh: 00000

fn: 100000

add $16, $9, $9

0x020a8020

0000 0010 0000 1010 1000 0000 0010 0000

Opcode: 000000 => 0

rs: 10000 => $16

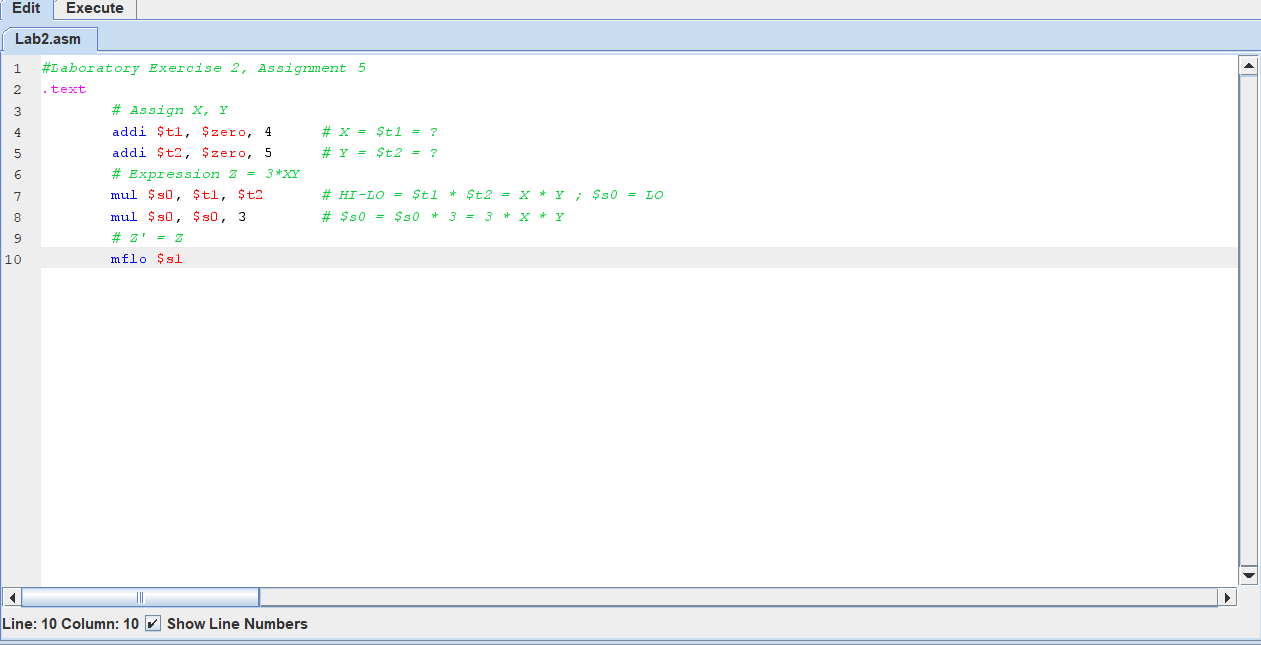
rt: 01010 => $10

rd: 10000 => $16

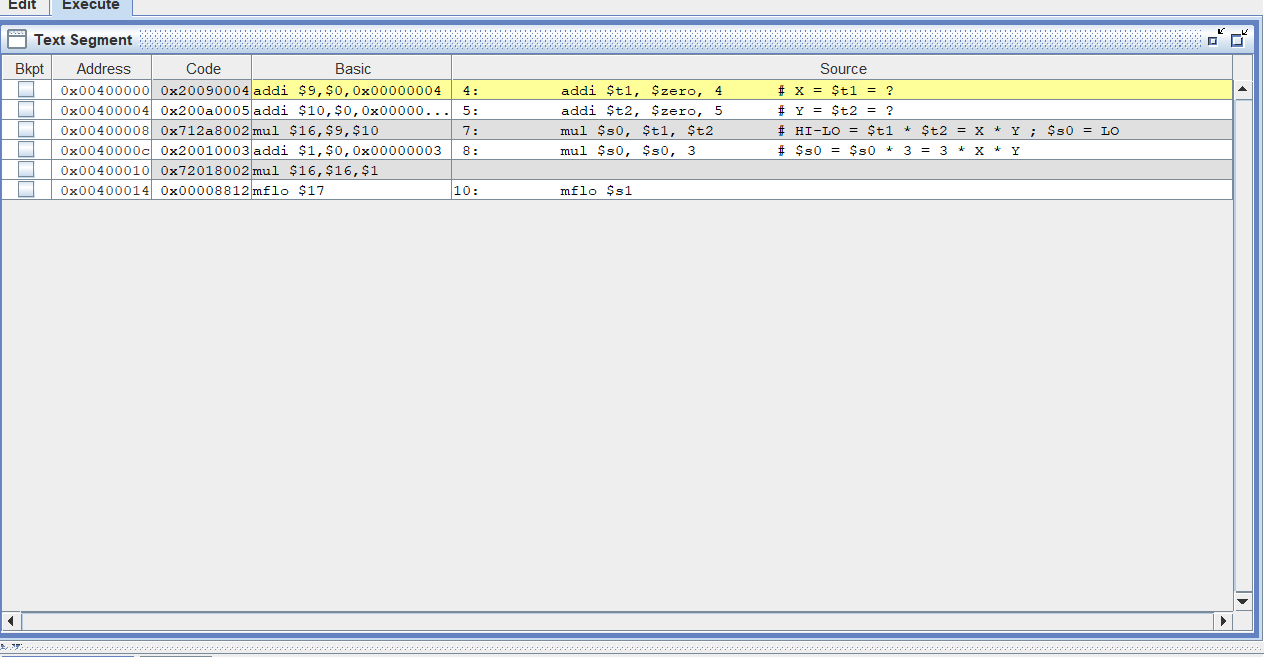
sh: 00000

fn: 100000 add $16, $16, $10

Assignment 5



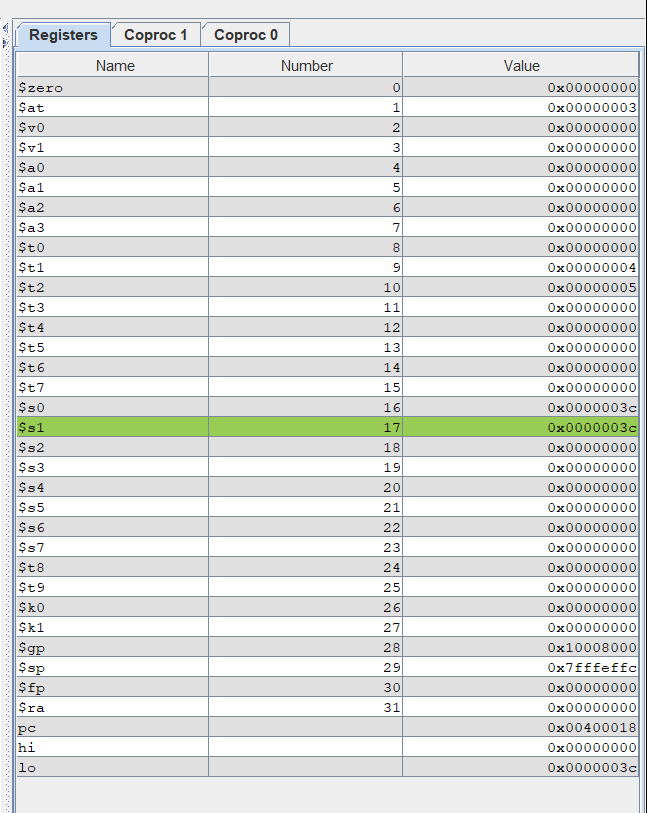
Kết quả thực thi



Điều bất thường : -Lệnh mul thứ hai sẽ bị tách thành lệnh addi và lệnh mul. Bởi vì lệnh mul không hỗ trợ việc nhân thanh ghi với một số, nên nó phải tách thành lệnh addi để lưu 1 số vào 1 thanh ghi và dùng lệnh mul sau khi tách để nhân 2 thanh ghi đó với nhau.

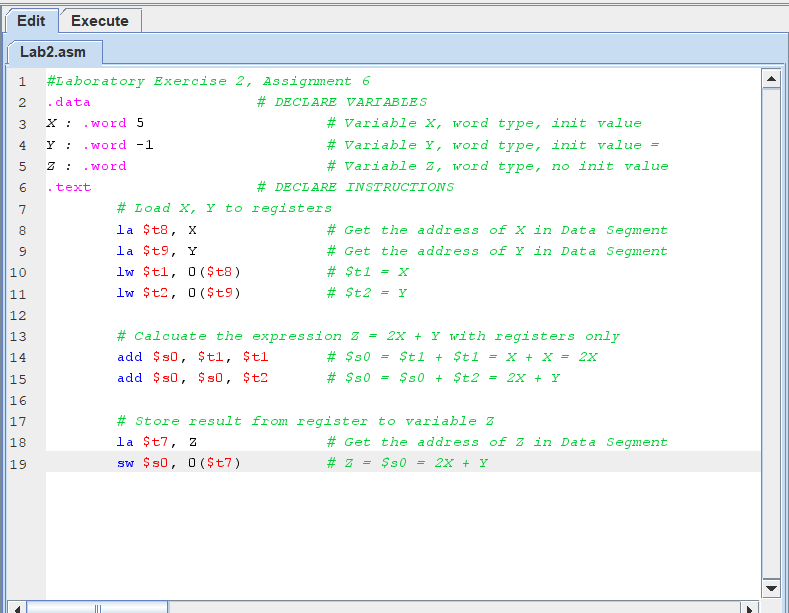
-Sự thay đổi các thanh ghi: thanh ghi $t1, $t2 dùng để gán giá trị 4 và 5, thanh ghi lo thay đổi thành 0x0000003c, thanh ghi hi không thay đổi, các thanh ghi thay đổi như trên chú thích của mã nguồn trên. Còn thanh ghi at lưu giá trị tạm thời 0x00000003 khi thực hiện phép nhân với số 3.

-Kết quả các thanh ghi sau khi kết thúc chương trình:



* Đúng như lý thuyết

Assignment 6



-Lệnh la được biên dịch thành 2 lệnh lui và ori, có tác dụng gán 1 số 32 bit vào thanh ghi

-Khi biên dịch lệnh la thành mã máy, các biến X, Y, Z với hằng số là bằng nhau.

-Khi thực thi chương trình:

A screenshot of a computer

Description automatically generated

-Các thanh ghi bị thay đổi và có thay đổi khi chạy chương trình là: $t1, $t2, $t7, $t8, $t9, $s0, at (ghi vào bộ nhớ tạm) và pc (luôn tăng 0x00000004 sau mỗi lệnh)

-Lệnh lw: Lấy địa chỉ của biến kiểu word và lưu vào 1 thanh ghi

-Lệnh sw: Lấy địa chỉ của biến kiểu word lưu vào bộ nhớ