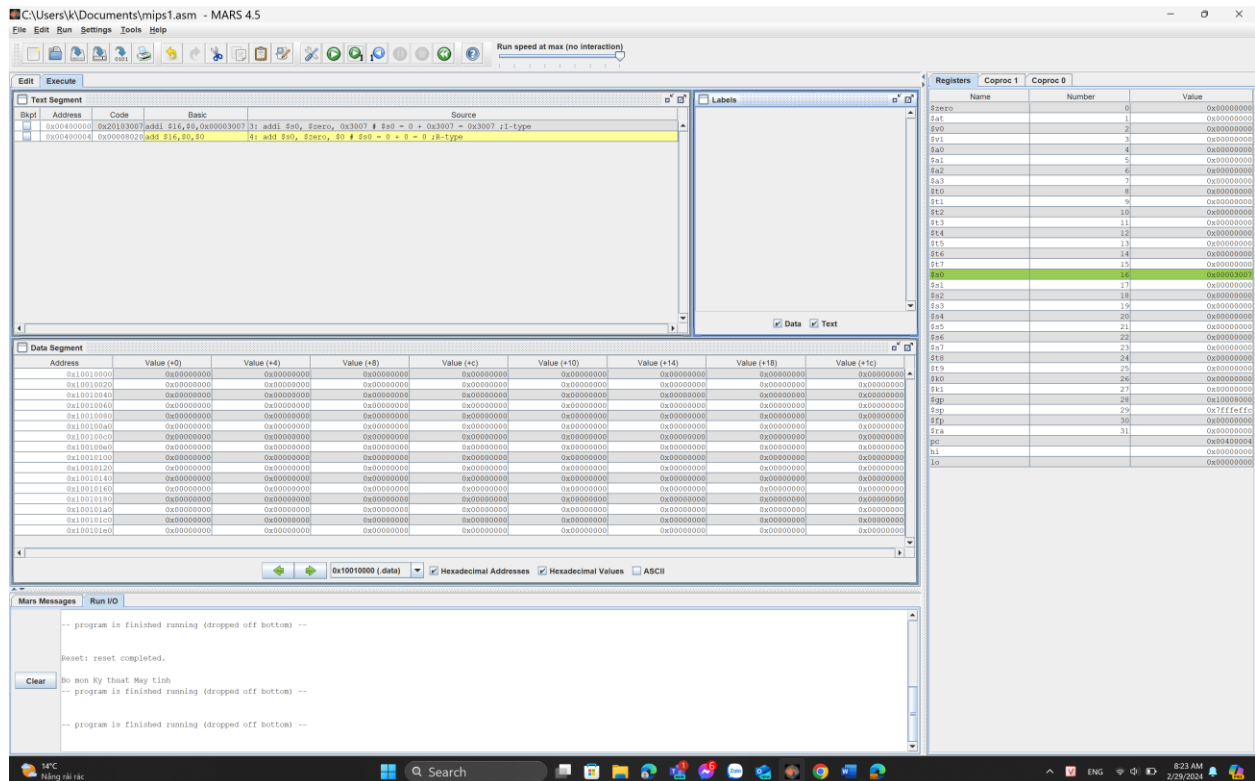


THỰC HÀNH KIẾN TRÚC MÁY TÍNH

LABORATORY EXERCISE 2

Họ tên: Vũ Quốc Bảo
MSSV: 20225694

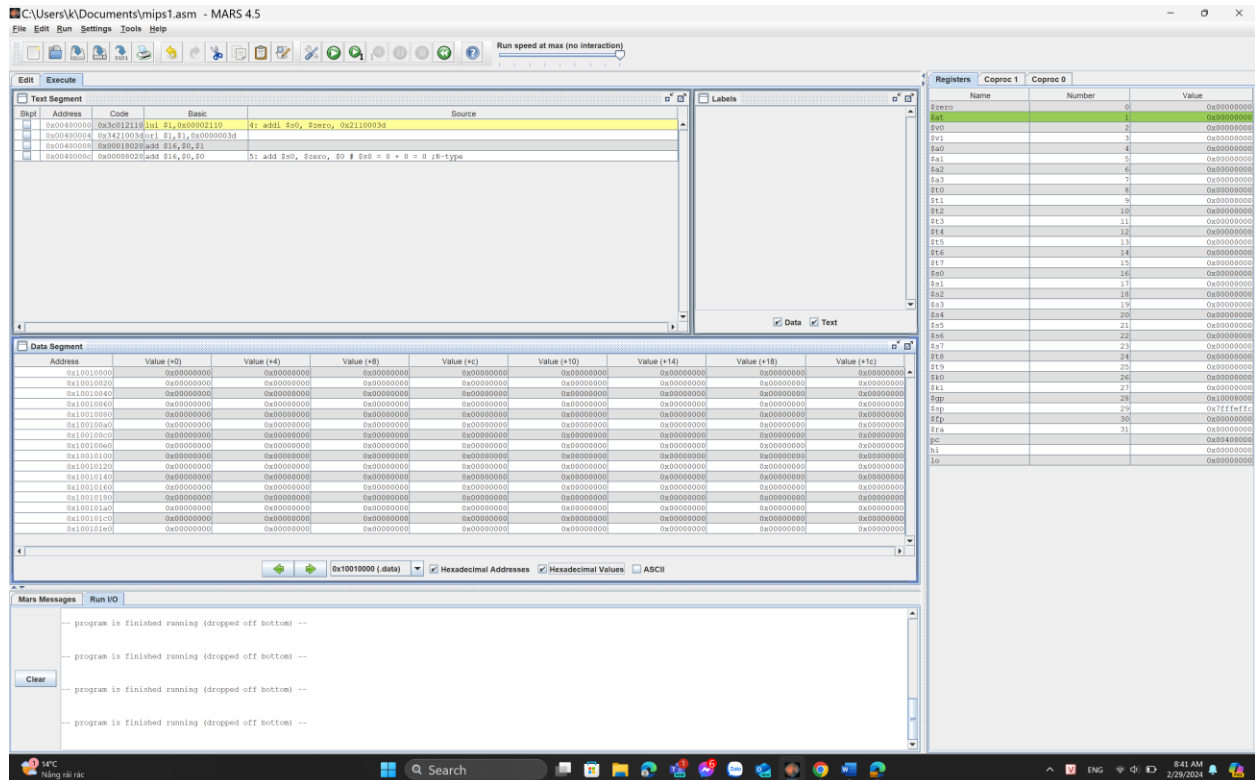
Assignment 1: Lệnh gán số 16-bit



- Quan sát cửa sổ Register, sự thay đổi giá trị của thanh ghi:
 - + \$s0: bắt đầu từ `0x00000000` -> `0x00003007` -> `0x00000000`.
 - + pc: bắt đầu từ `0x00400000` -> `0x00400004` -> `0x00400008`.

- So sánh mã máy:

Ta có khuôn dạng máy của lệnh `addi` là I, với `$s0 = 16`, `$zero = 0`; opcode của `addi` là 8. Nên ta tính được mã máy như sau: `001000 00000 10000 00110 00000 000111`. Từ đây ta có thể thấy chứng tỏ các lệnh đó đúng như tập lệnh đã qui định.

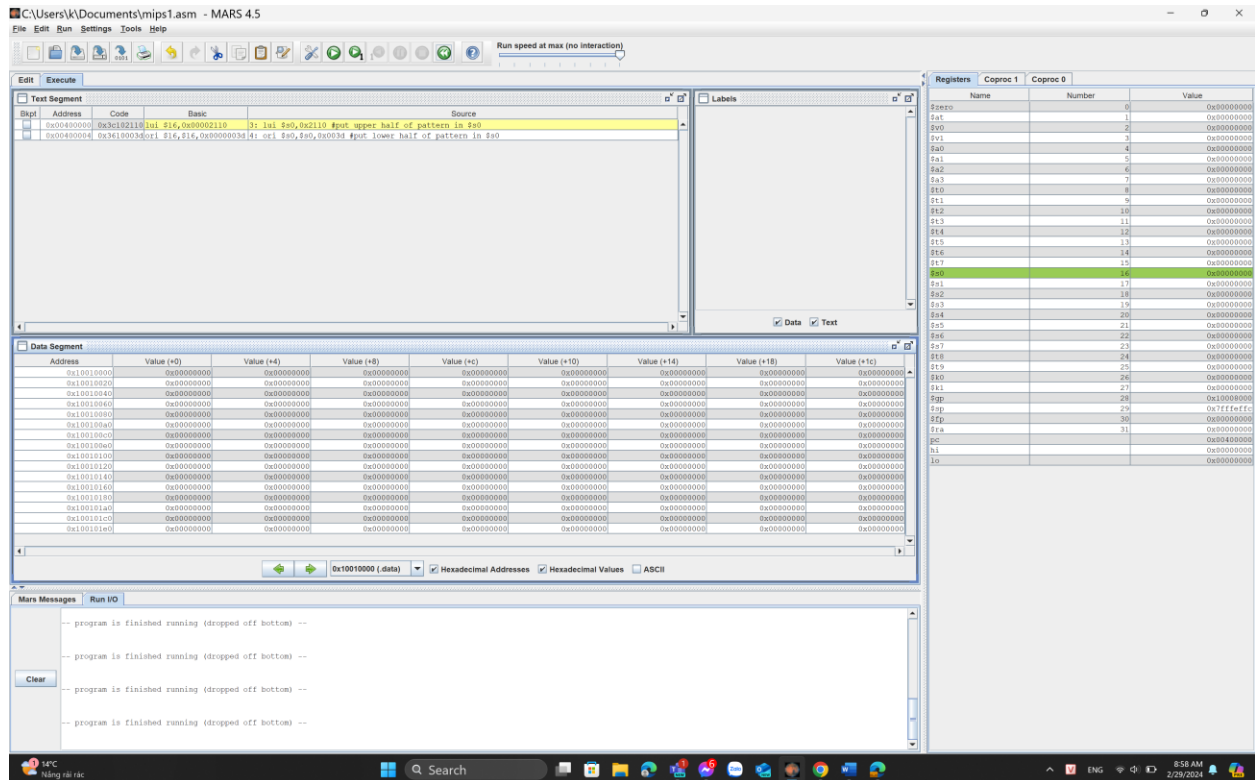


- Sau khi sửa lệnh lui:

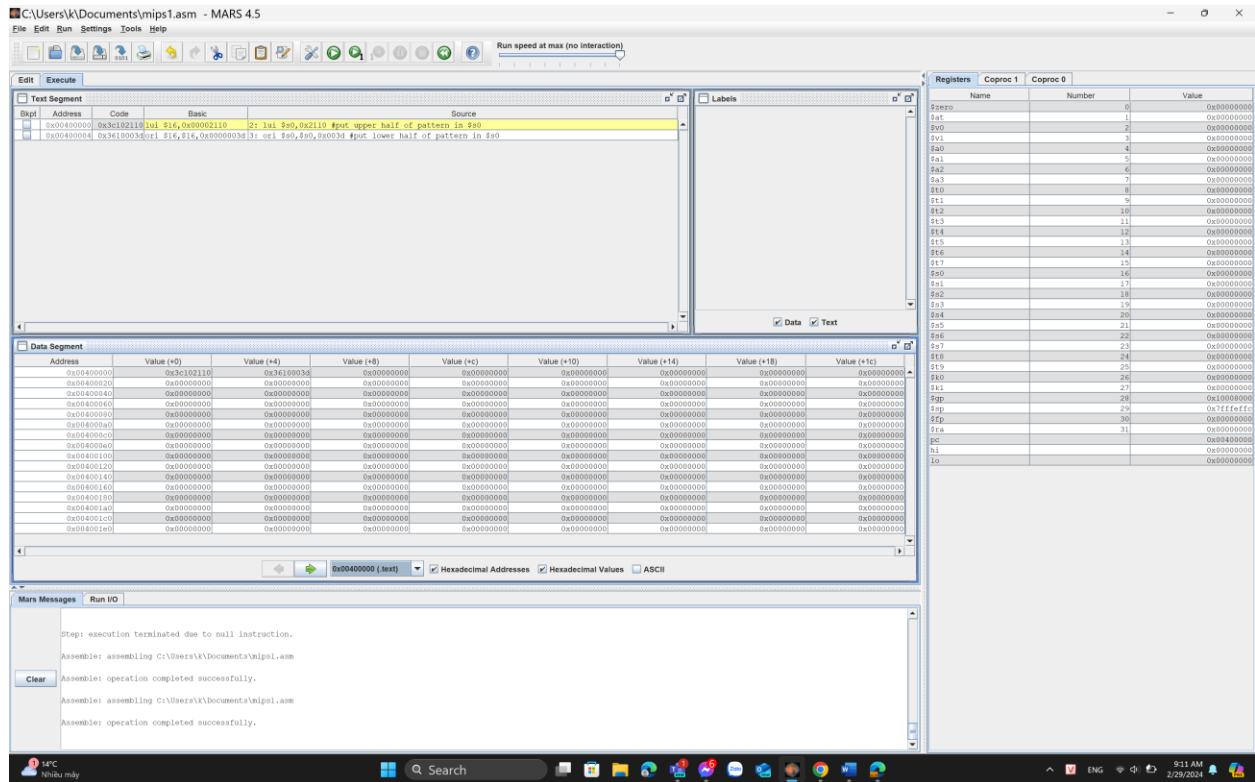
- + Thanh ghi \$at có giá trị thay đổi từ 0x00000000 -> 0x21100000 -> 0x2110003d.
- + Sau khi thanh ghi \$at có giá trị như trên thì thanh ghi \$s0 sẽ được cập nhật giá trị.

- Giải thích: vì giá trị 0x2110003d đã vượt quá 16-bit nên lệnh addi là giả lệnh được tạo nên từ 2 lệnh lui và ori, khi đó giá trị này sẽ được tách ra thành 0x21100000 rồi ghi vào thanh ghi \$at.

Assignment 2: lệnh gán số 32-bit

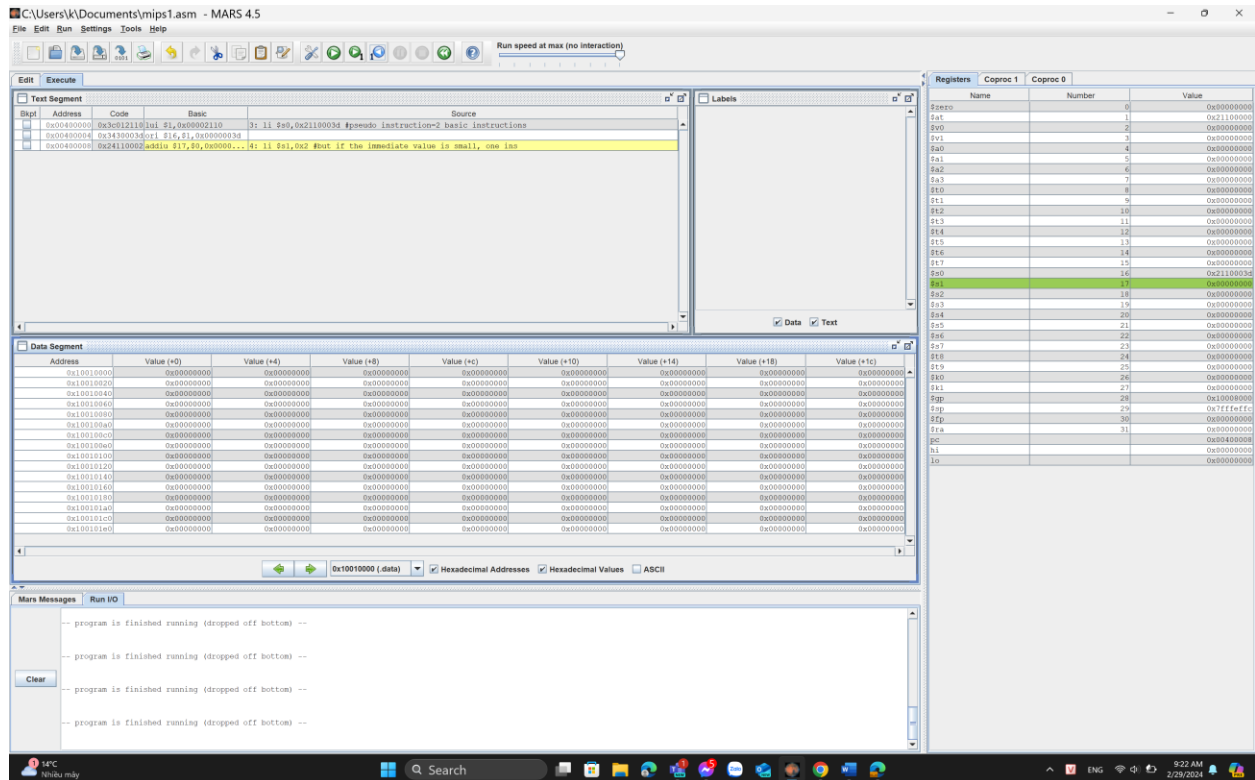


- Quan sát cửa sổ Register, sự thay đổi giá trị của thanh ghi:
 - + \$s0: bắt đầu từ `0x00000000` -> `0x21100000` -> `0x2110003d`.
 - + \$pc: bắt đầu từ `0x00400000` -> `0x00400004` -> `0x00400008`.



- Quan sát các byte trong vùng lệnh .text ở cửa sổ Data Segment, ta thấy các byte đầu tiên của vùng lệnh trùng với cột Code của cửa sổ Text Segment.

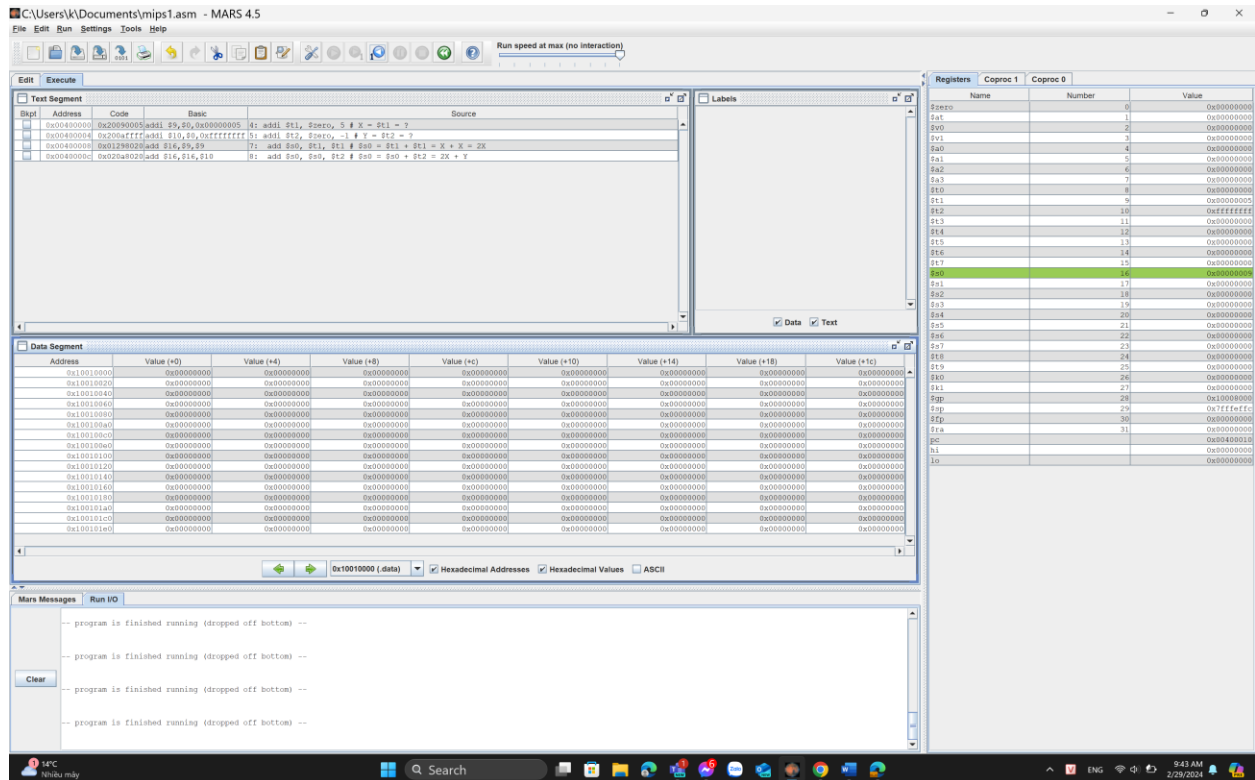
Assignment 3: lệnh gán (giả lệnh)



- Giải thích:

Ở phép toán đầu, do giá trị vượt quá số bit có thể ghi nên thanh ghi \$at sẽ được sử dụng như một biến tạm để lưu giá trị. Có thể hiểu giả lệnh li thay thế cho 2 lệnh lui và ori, khi đó thanh ghi \$at sẽ được cập nhật giá trị đầu vào là 0x21100000 theo lệnh lui. Ngay sau đó, giả lệnh li sẽ lập tức tải phần còn lại của giá trị là 0x0000003d và cập nhật trực tiếp vào bộ nhớ của \$s0. Còn đối với thanh ghi \$s1, do giá trị đầu vào nhỏ nên lệnh giả li sẽ tương đương với addi và ghi vào trực tiếp thanh ghi luôn mà không cần thông qua bất kì thanh ghi tạm thời nào khác.

Assignment 4: Tính biểu thức $2x + y = ?$



- Quan sát cửa sổ Register:

+ lệnh đầu tiên \$t1 được cập nhật: 0x00000000 -> 0x00000005.

+ lệnh thứ 2 \$t2 được cập nhật: 0x00000000 -> 0xffffffff.

+ lệnh thứ 3 \$s0 được cập nhật 2 lần, lần đầu cập nhật \$s0 = \$s1 + \$s1, sau đó cập nhật \$s0 = \$s0 + \$s2: 0x00000000 -> 0x0000000a -> 0x00000009.

+ pc: 0x00400000 -> 0x00400004 -> 0x00400008 -> 0x0040000c -> 0x00400010.

Sau khi chương trình kết thúc, ta thấy kết quả của thanh ghi \$s0 đúng với biểu thức $2x + y$ mà ta cần tính.

- addi \$t1, \$zero, 5: 001000 00000 01001 0000000000000101

addi \$zero \$t1 5

→ đúng với khuôn mẫu của kiểu lệnh I.

- add \$s0, \$t1, \$t1: 000000 01001 01001 10000 00000 010100

→ đúng với khuôn mẫu của kiểu lệnh R.