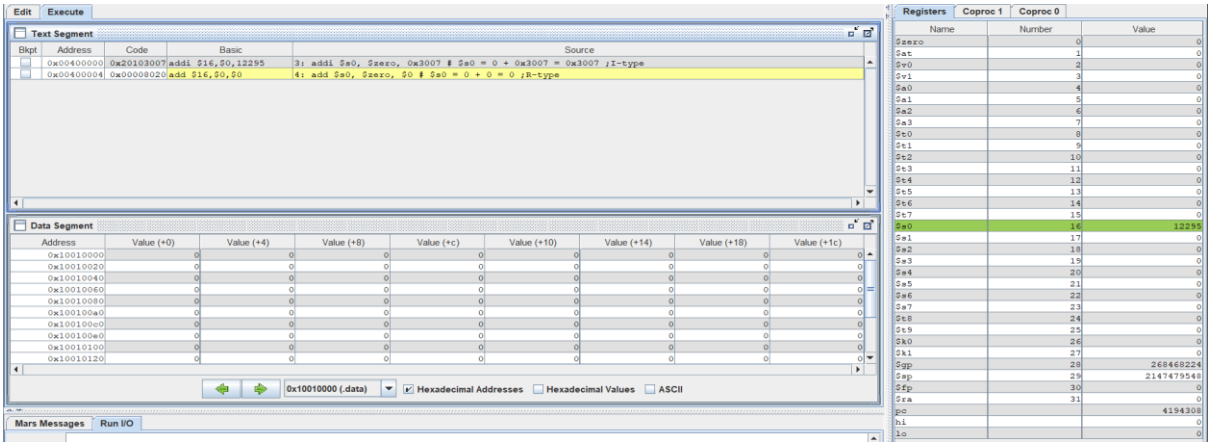
BÁO CÁO THỰC HÀNH TUẦN 1

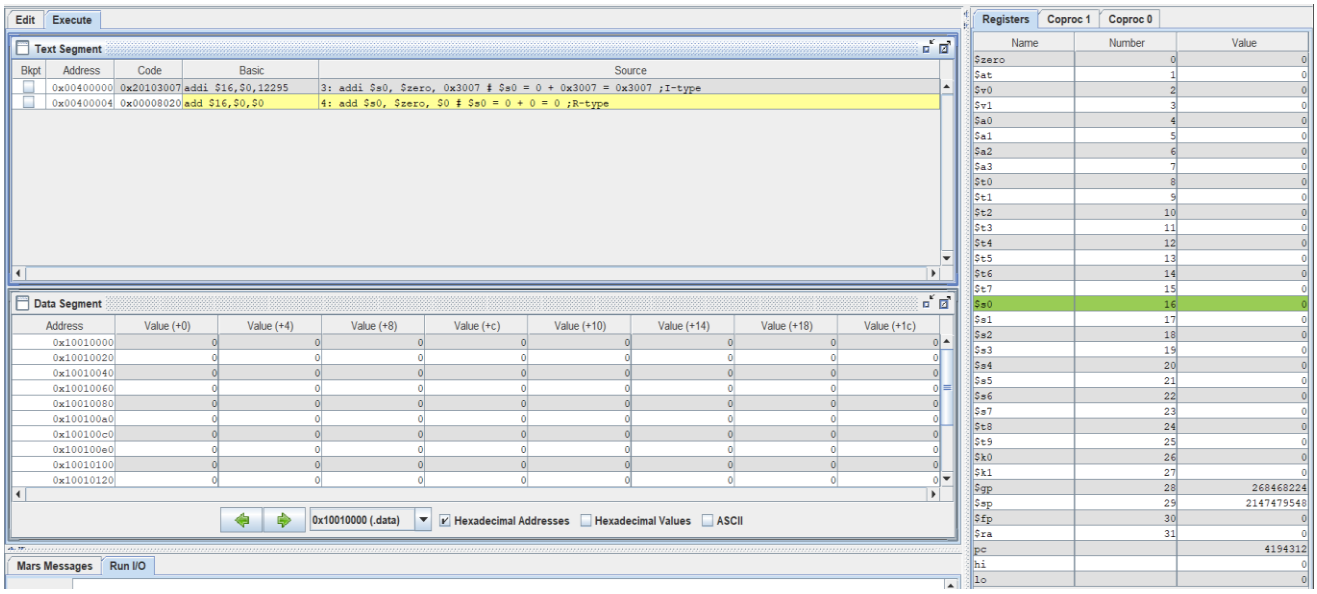
Assignment 1

- Lệnh addi $s0, $zero, 0x3007



⇨ $s0 = 12295 , $pc = 4194308

- Lệnh add $s0, $zero, $0



⇨ $s0 = 0 , $pc = 4194312

So sánh mã máy:

addi $s0, $zero, 0x3007

- Format : I
- Op 8 : 00 0000
- imm 0x3007 : 0011 0000 0000 0111
- rs $0 : 0 0000
- rt $16: 1 0000
  - ➔ Mã máy : 0010 0000 0001 0000 0011 0000 0000 0111
  - ➔ Mã rút gọn : 0x20103007
  - ➔ Trùng với khuôn dạng lệnh -> chương trình chạy đúng

add add $s0, $zero, $0

- Format : R
- Op 0 : 00 1000
- rs 0 : 0 0000
- rt 0: 0 0000
- rd 16 : 1 0000
- shamt: 0 0000
- function 32 : 10 0000
  - ➔ Mã máy : 0010 0000 0000 0000 1000 0000 0010 0000
  - ➔ Mã rút gọn : 0x20008020
  - ➔ Trùng với khuôn dạng lệnh -> chương trình chạy đúng

Sau khi sửa lệnh addi $s0, $zero, 0x2110003d và chạy thì được chia thành các lệnh sau:

- Lệnh thứ 1 : lui $1,0x00002110
- Lệnh thứ 2 : ori $1,$1,0x0000003d
- Lênh thứ 3 : add $16,$0,$1
- Lệnh 4 như cũ: add $16,$0,$0
  - ➔ Vì lệnh lui là 32 bit : addi $s0, $zero, 0x2110003d nên phải chia thanh 2 lệnh 16 bit là lui $s0,0x2110 và ori $s0,0x003d

Assignment 2

- Khi chưa chạy lệnh nào:

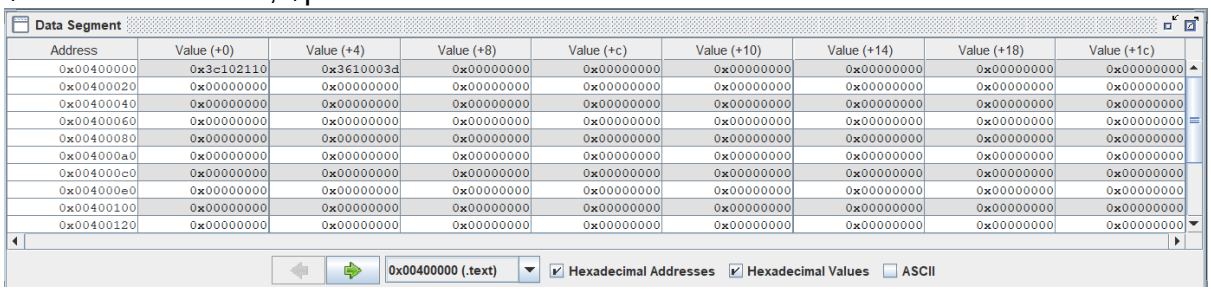
$s0 = 0x00000000 , $pc = 0x00400000

- Khi chạy lệnh lui $16,0x00002110

$s0 = 0x21100000 , $pc= 0x00400004

- Khi chạy lệnh ori $16,$16,0x0000003d

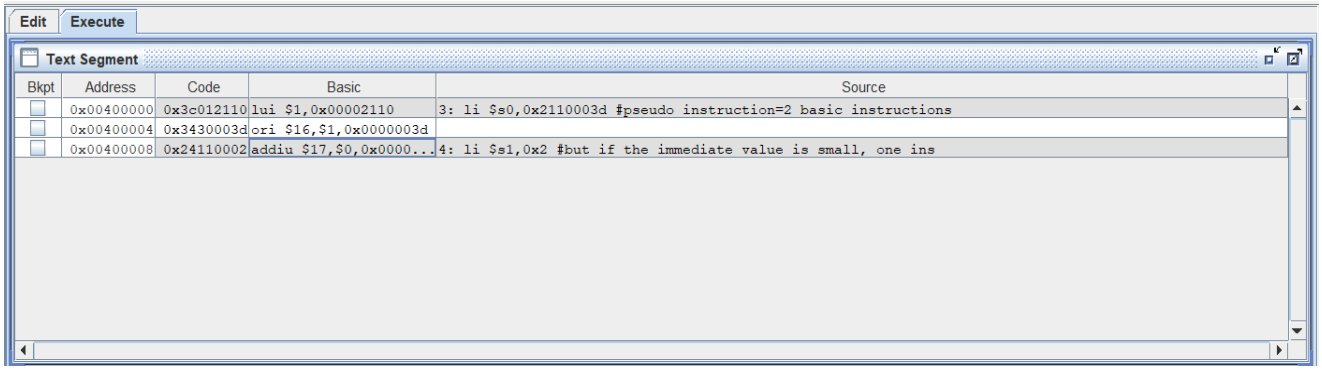
$s0 = 0x2110003d, $pc = 0x00400008



Lệnh lui $s0,0x2110 : Xuất hiện ở cột Value(+0) tại thanh ghi có địa chỉ 0x3c102110

Lệnh ori $s16,$s16, 0x003d : Xuất hiện ở cột Value(+4) tại thanh ghi có địa chỉ 0x3610003d

Assignment 3



- Điều bất thường của lệnh trên xuất hiện ở lệnh li $s1,0x2
  Vì lệnh li là 1 lệnh mở rộng nên chương trình phải đưa về lệnh cơ bản là addiu $16,$1,0x0000003d

Assignment 4

Sự thay đổi giá trị của các thanh ghi

- Lệnh 1: addi $9,$0,0x00000005
  Thanh ghi $t1 = 0x00000005 = 5

- Lệnh 2: addi $10,$0,0xffffffff
  Thanh ghi $t2 = 0xffffffff = -1

- Lệnh 3 : add $16,$9,$9
  Thanh ghi $s0 = 0x0000000a = 10

- Lệnh 4: add $16,$16,$10
  Thanh ghi $s0 = 0x00000009 = 9
  = 2$t1 + $t2
  ⇨ Kết quả chương trinh là đúng

Sự tương đồng giữa hợp ngữ và mã máy trong lệnh addi:

+ 16bits sau của phần mã máy giống với giá trị của lệnh addi thêm vào thanh ghi.

+ Vì kiểu lệnh I chỉ có thể thực hiện được với giá trị < 16bits

Kiểm nghiệm với khuôn mẫu kiểu lệnh I:

0x20090005: 0010 00-00 000-0 1001- 0000 0000 0000 0101

0x200affff: 1000 00-00 001-0 1011- 11111111111111

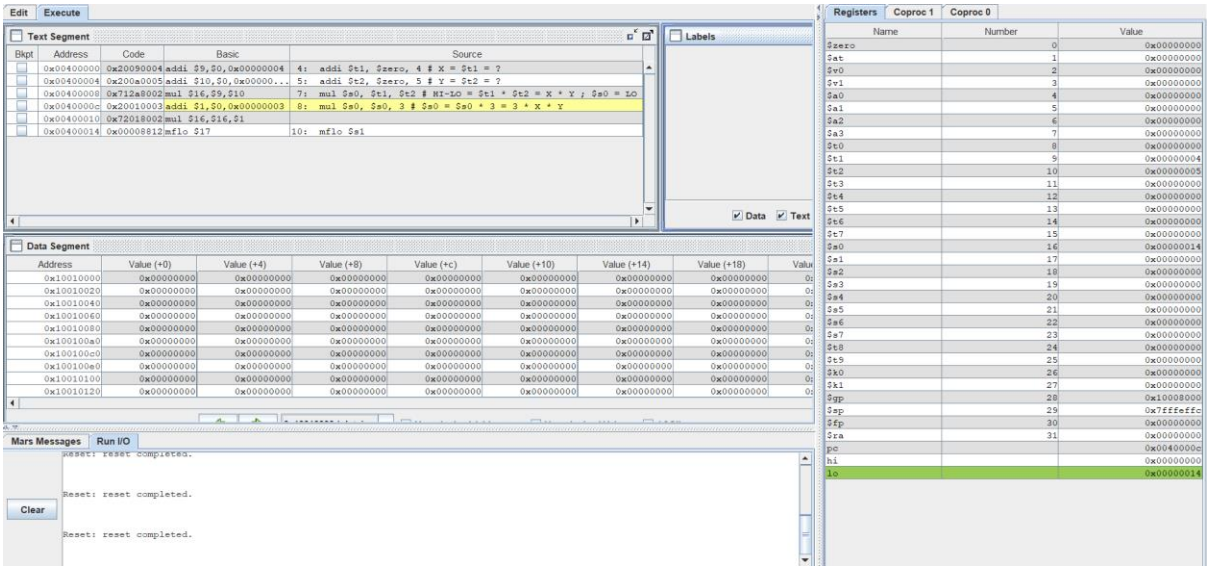
Kiểm nghiệm với khuôn mẫu kiểu lệnh R:

0x01298020: 0000 00-01 001-0 1001- 1000 0-000 00-10 0000

0x020a8020: 0000 00-10 000-0 1010- 1000 0-000 00-10 0000

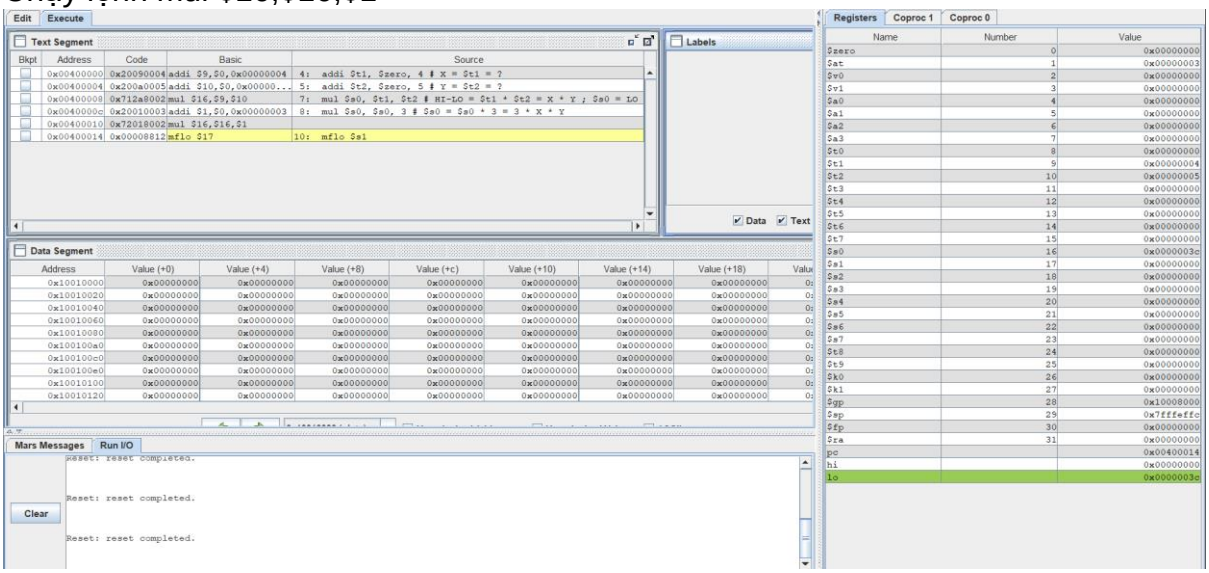
Assignment 5

- Chạy lệnh mul $16,$9,$10



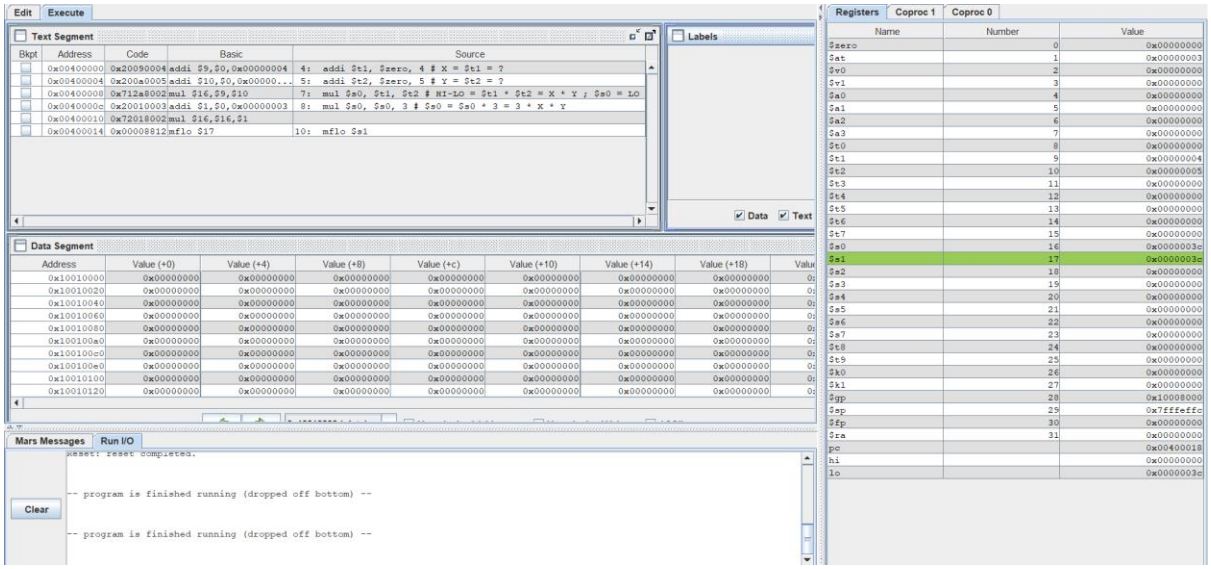
hi= 0x00000000, lo = 0x0000014 = 20

- Chạy lệnh mul $16,$16,$1



hi= 0x00000000, lo = 0x0000003c = 60

- Chạy lệnh mflo $17



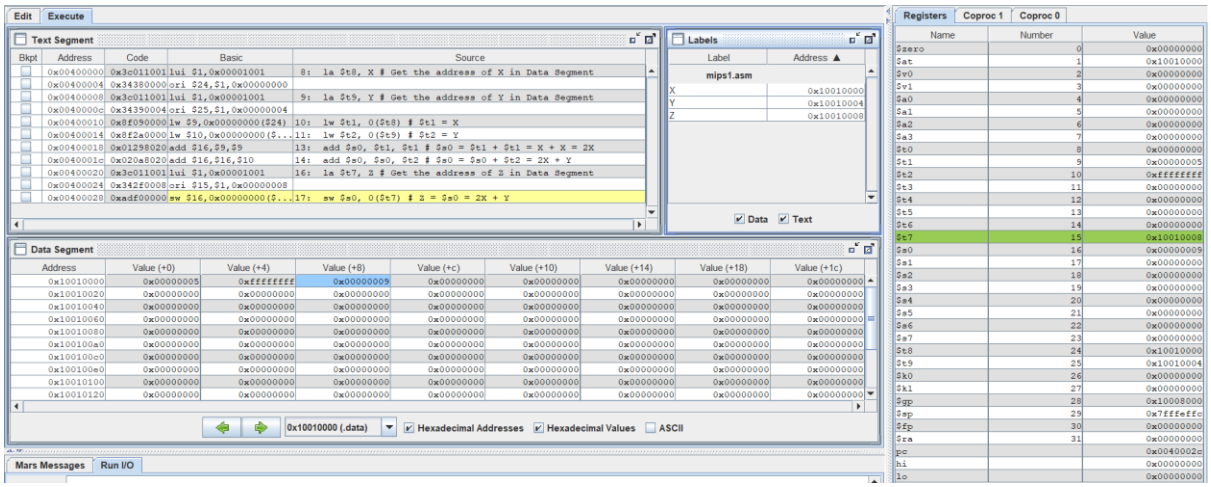
$s1 = 0x0000003c = 60

⇨ Kết quả chương trinh chạy đúng
Điều bất thường là khi sử dụng mul, ta không thể dùng hằng số mà cần phải sử dụng thêm lệnh addi$1,$0,0x00000003 để ghi giá trị 3 cho thanh ghi $at

Assignment 6

- Lệnh la được biên dịch thành 2 lệnh : lui $1,0x00001001 và ori $24,$1,0x00000000

- Ở cửa sổ Label, địa chỉ của X, Y, Z đung với các giá trị khởi tạo



Vai trò của lệnh lw:

Chép 1 word (4bytes) tại vị trí trong bộ nhớ ram vào thanh ghi

Vai trò của lệnh sw:

Lưu 1 word trong thanh ghi vào bộ nhớ RAM

Lệnh lb và sb:

+ lb: Chép 1 byte tại vị trí trong bộ nhớ RAM vào byte thấp của thanh ghi

+ sb: Lưu 1 byte thấp trong thanh ghi vào vị trí trong bộ nhớ RAM

-