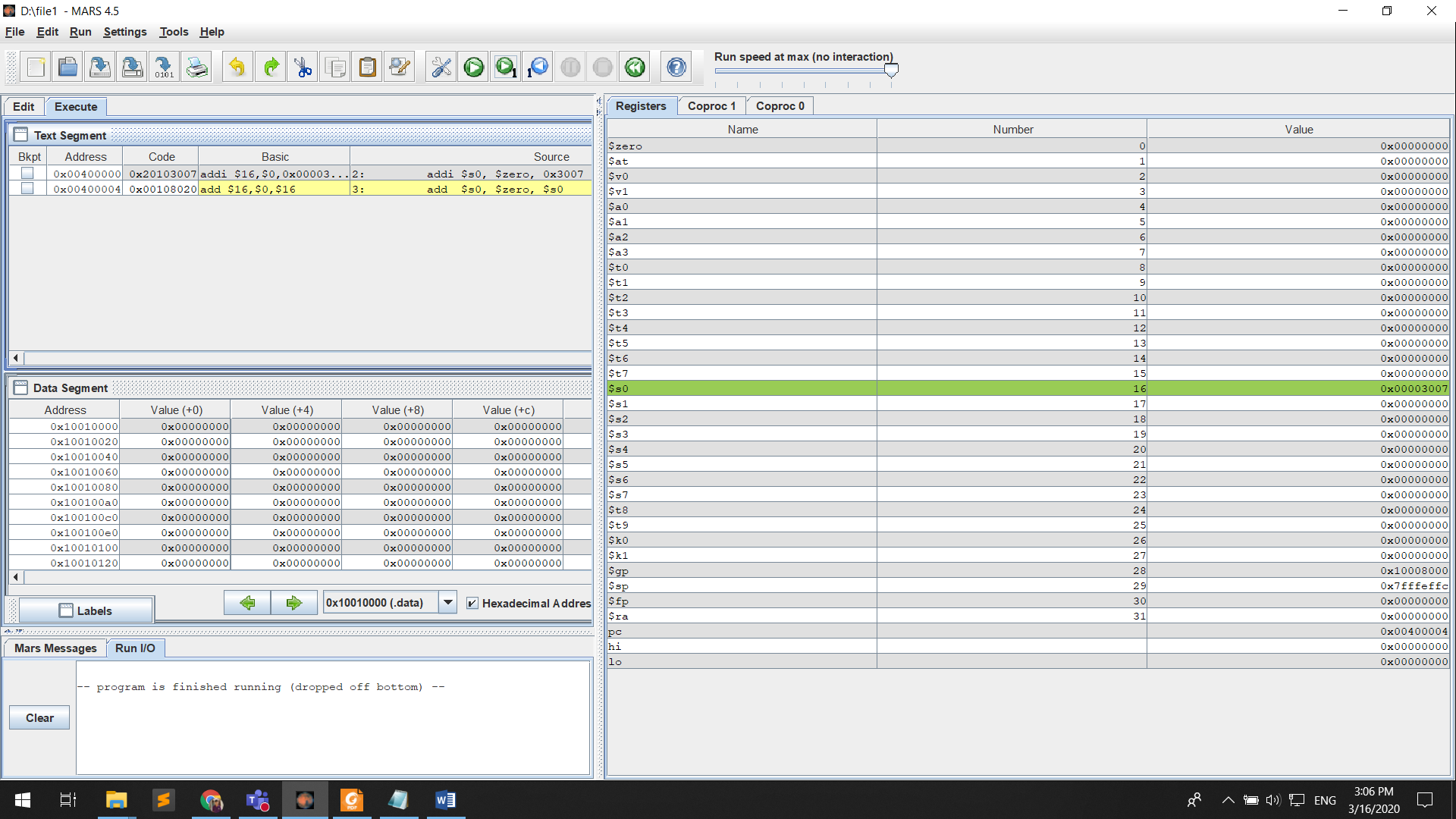
**Báo cáo thực hành kiến trúc máy tính tuần 2**

**Giảng viên: Lê Bá Vui**

**Sinh viên: Hoàng Đức Anh MSSV: 2017668**

**Assignment 1: lệnh gán số 16-bit**

**Màn hình chạy**



$s0 : 0x00000000=>0x00003007

Pc : 0x00400000=>0x00400004

Mã máy: addi : 0x20103007 (001000 00000 10000 0011000000000111)

=> op = 8

=> op/func = 8

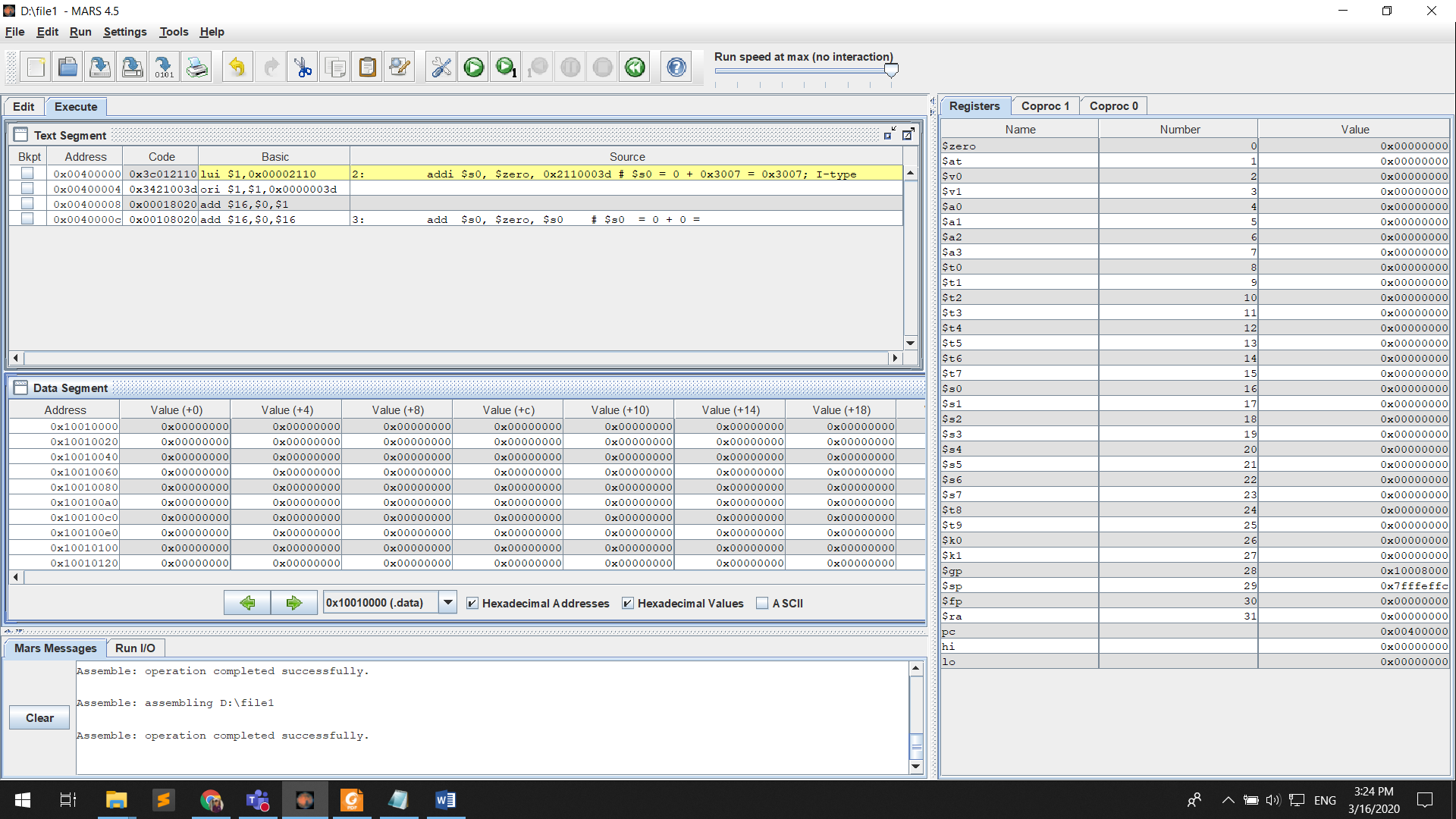
=>đúng với khuôn dạng lệnh I

mã máy: add :0x00108020 (000000 00000 10000 10000 00000 100000)

* Op = 0; func = 32
* Op/func = 0/32
* đúng với khuôn dạng lệnh R

**Sửa lại lệnh:**

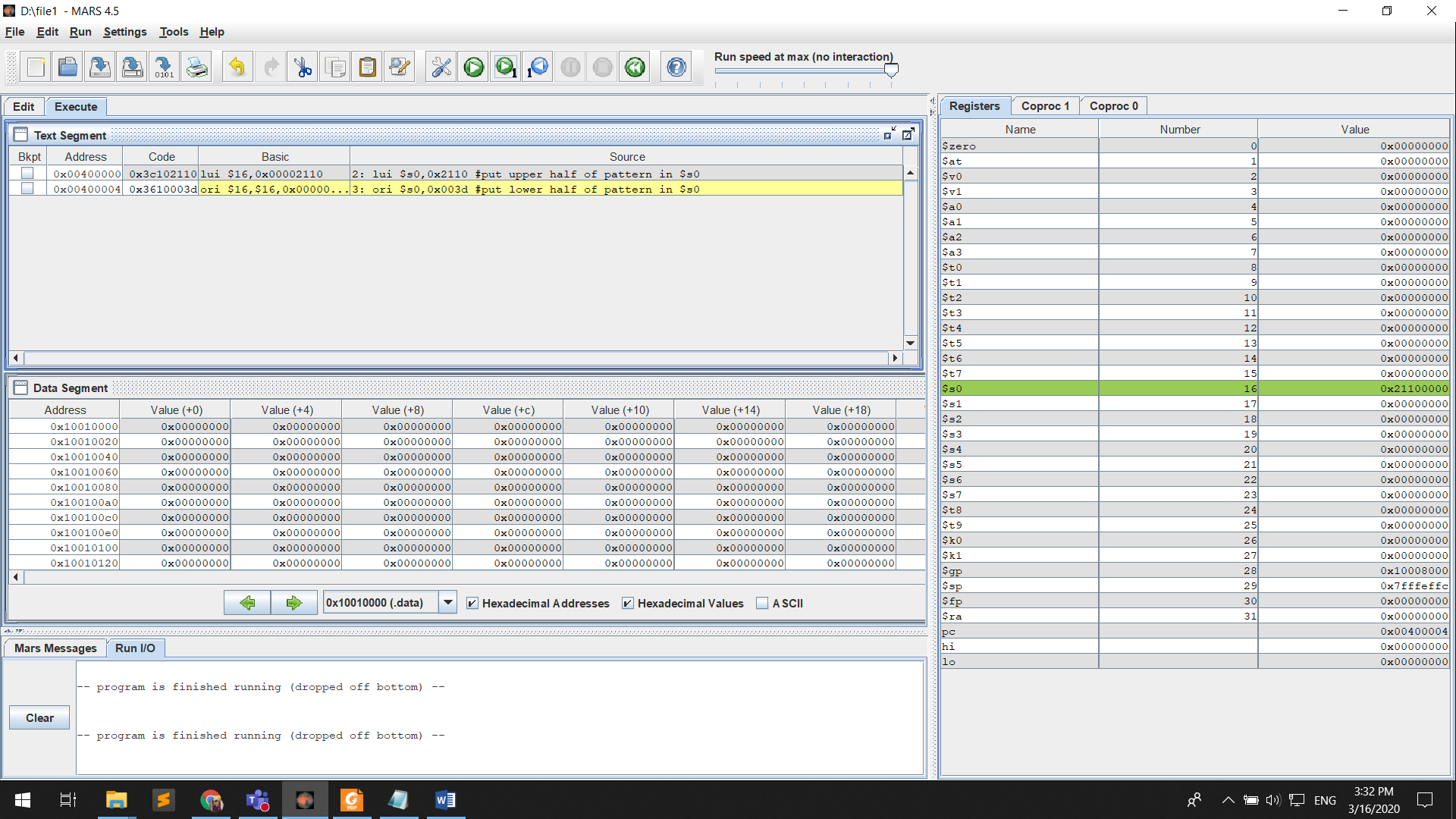
**Màn hình:**



0x2110003d được tách thành 2 số 0x00002110 và 0x0000003d lý do là vì công cụ mars ko làm việc với các số 32 bit mà nó tách thành 2 số 16 bit

Assignment 2: lệnh gán số 32-bit

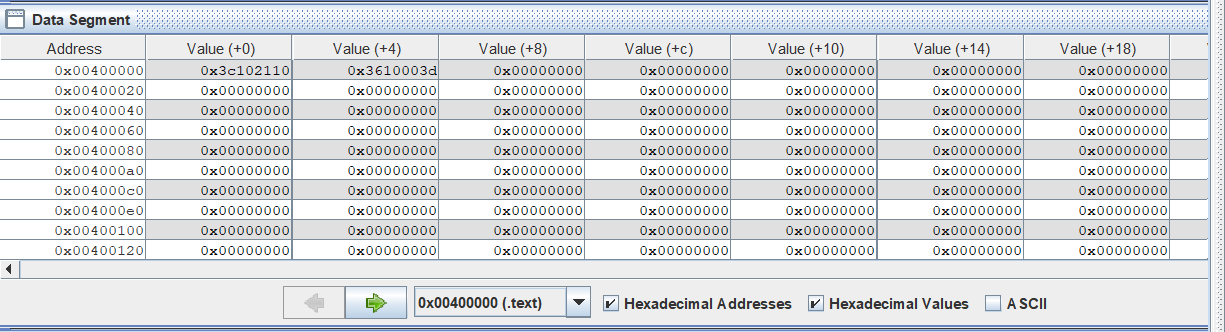
Màn hình chạy:



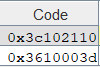
$s0 : 0x00000000=> 0x21100000

Pc : 0x00400000=> 0x00400004

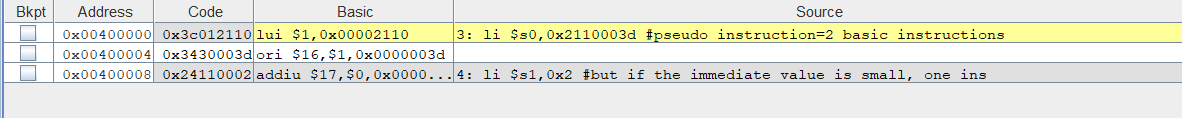
Cửa sổ Data Segment



Các byte đầu tiên của dòng lệnh trùng với cột code trong cửa sổ Text Segment



**Assignment 3: lệnh gán (giả lệnh)**lệnh li là giả lệnh giúp quá trình viết code nhanh chóng, đơn giản hơn



Vì giá trị 0x2110003d là giá trị 32 bit nên phải tách thành 2 phần 16 bit dùng hàm lui và ori để lấy địa chỉ

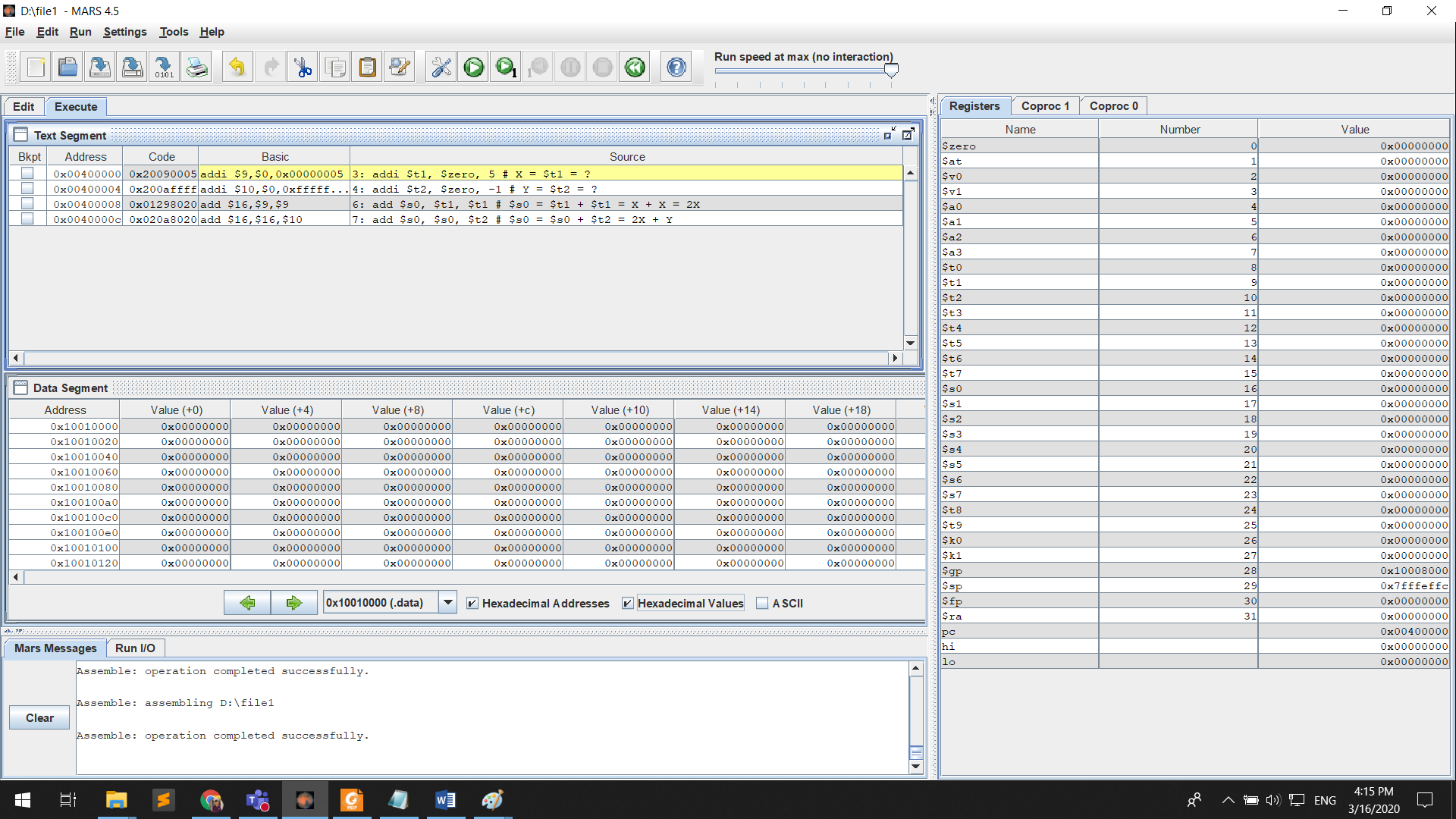
Lui $1, 0x00002110 gán địa chỉ 0x00002110 vào $1

Ori $16, $1, 0x0000003d tiến hành so sánh và gán nửa dưới vào $16

Lệnh addiu $17, $, 0x00000002 gán giá trị vào $17

Assignment 4: tính biểu thức 2x + y = ?

Màn hình kết quả



Sự thay đổi của các thanh ghi

1: $t1 0x00000005

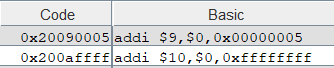
2: $t2 0xffffffff

3: $s0 0x0000000a

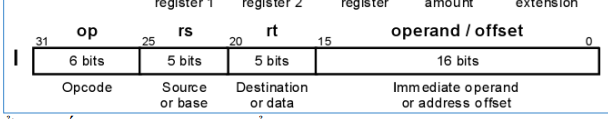
4: $s0 0x00000009

**Kết quả chính xác**

Điểm tương đồng giữa hợp ngữ và mã máy: 5 bit cuối của mã máy chính là đầu vào của hợp ngữ



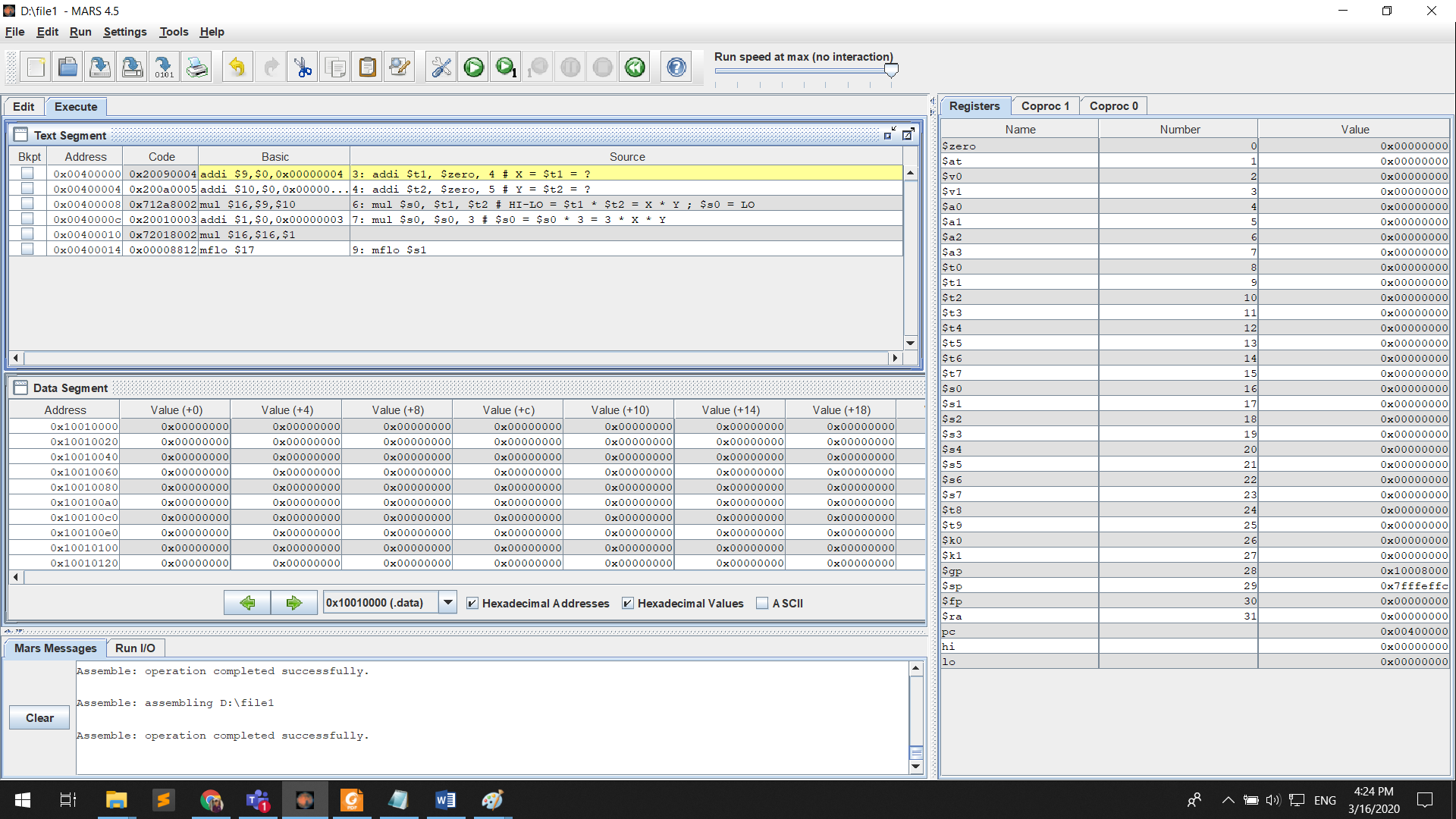
Lệnh addi : $rt = $rs + imm



* phù hợp

**Assignment 5: phép nhân**

Màn hình chạy



Điều bất thường và giải thích:

lệnh mul đầu tiên là lệnh mul basic nên nó ko đổi

lệnh mul thứ 2 không phải là lệnh basic nên nó biến đổi thành 2 lệnh

lệnh mul tiếp theo thực hiện phép mul trên 3 thanh ghi

**Sự biến đổi khi debug từng lệnh khi debug**

1: $t1 = 0x00000004 // gán giá trị 4 vào $t1

2: $t2 = 0x00000005 // gán giá trị 5 vào $t2

3: lo = 0x00000014 //

4: $at = 0x00000003 // gán giá trị 3 vào $at

5: lo = 0x0000003c

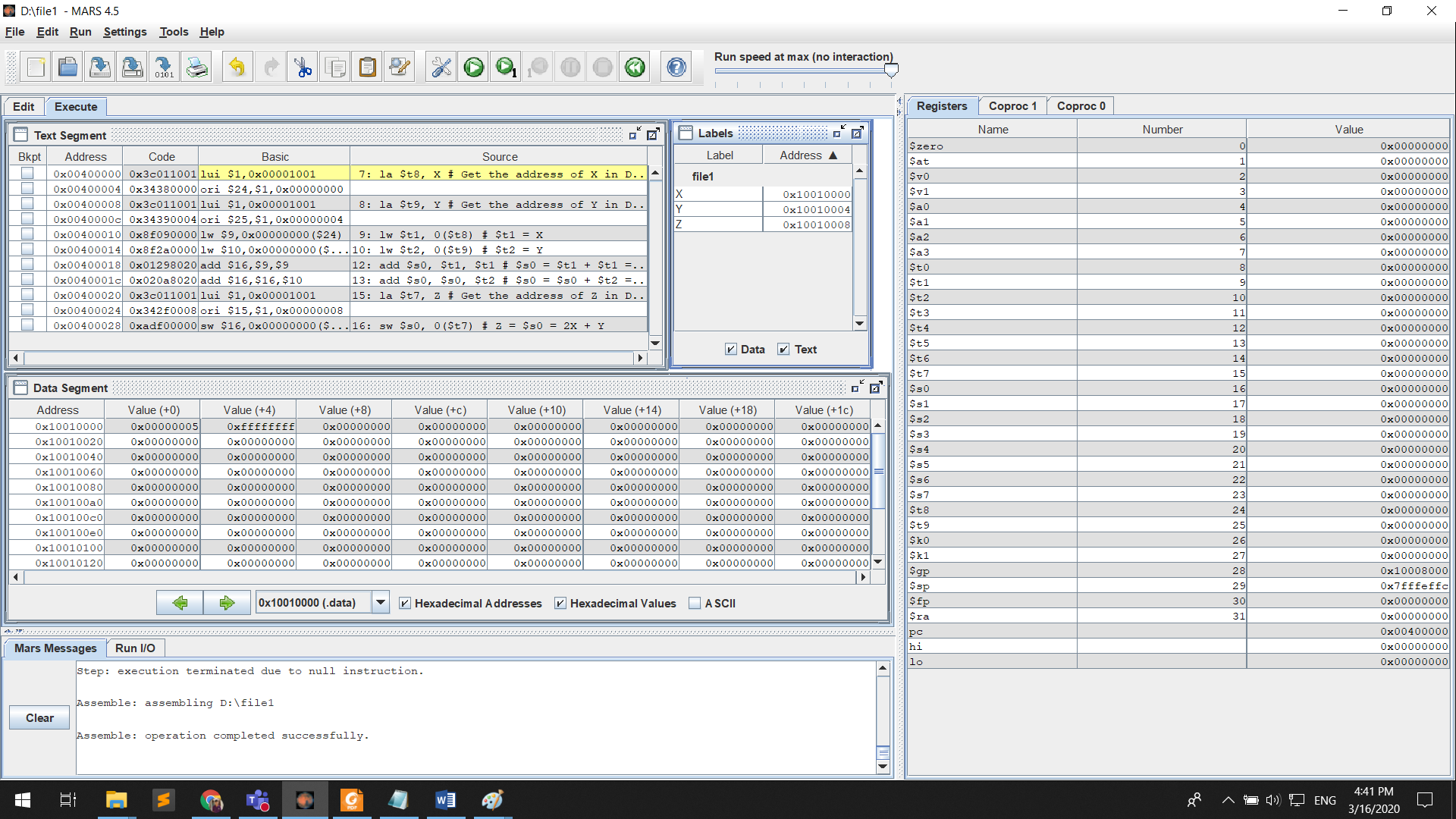
6: $s1 = 0x0000003c

* **kết quả chính xác**

Thanh ghi HI không có sự biến đổi gì

**Assignment 6: tạo biến và truy cập biến**

**Màn hình chạy**



Lệnh “la” là lệnh gán địa chỉ. Lệnh la được biên dịch bằng 2 lệnh “lui” và “ori”

16 bit cuối của địa chỉ X, Y, Z giống với 16 bit cuối sau khi biên dịch lệnh “La” thành mã máy

Sự biến đổi khi dùng công cụ Debug

1: $at = 0x10010000

2: $t8 = 0x10010000

3: $at = 0x10010000

4: $t9 = 0x10010004

5: $t1 = 0x00000005

6: $t2 = 0xffffffff

7: $s0 = 0x0000000a

8: $s0 = 0x00000009

9: $at = 0x10010000

10 $t7 = 0x10010008

Lệnh lw: lấy giá trị

Lệnh sw: lưu giá trị