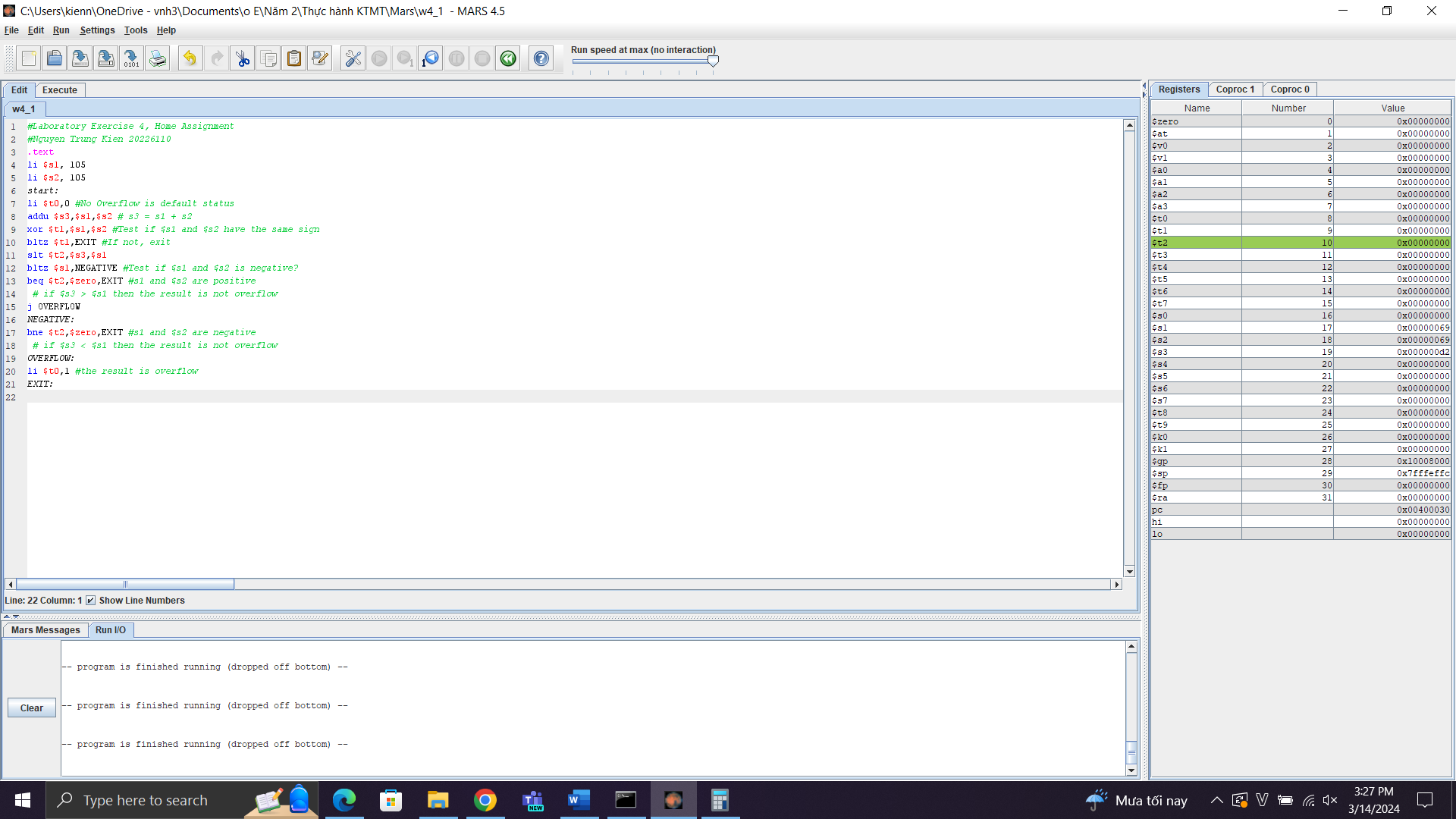
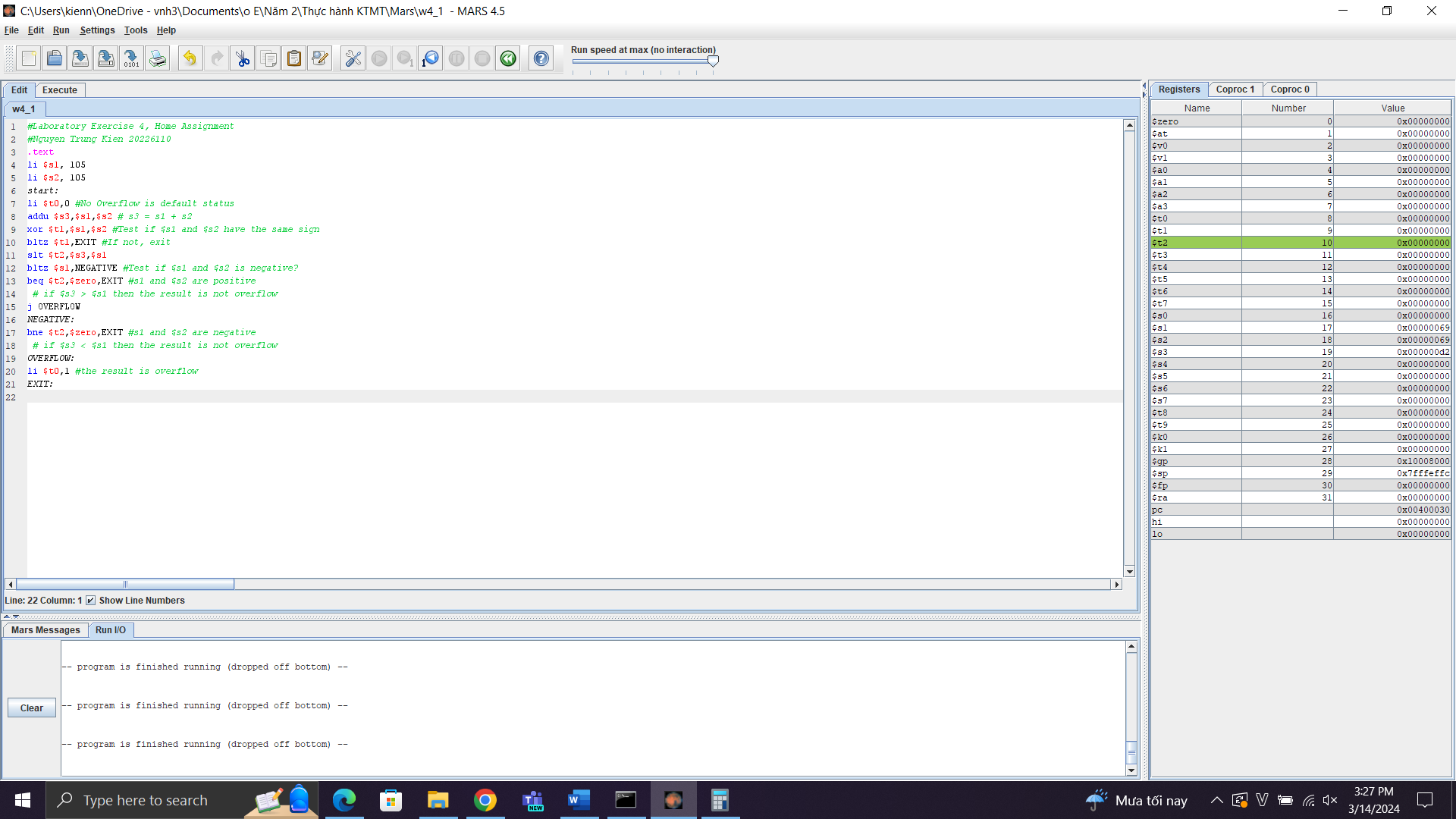
**Computer Architecture Lab Report Week 4**

**Full name: Nguyen Trung Kien**

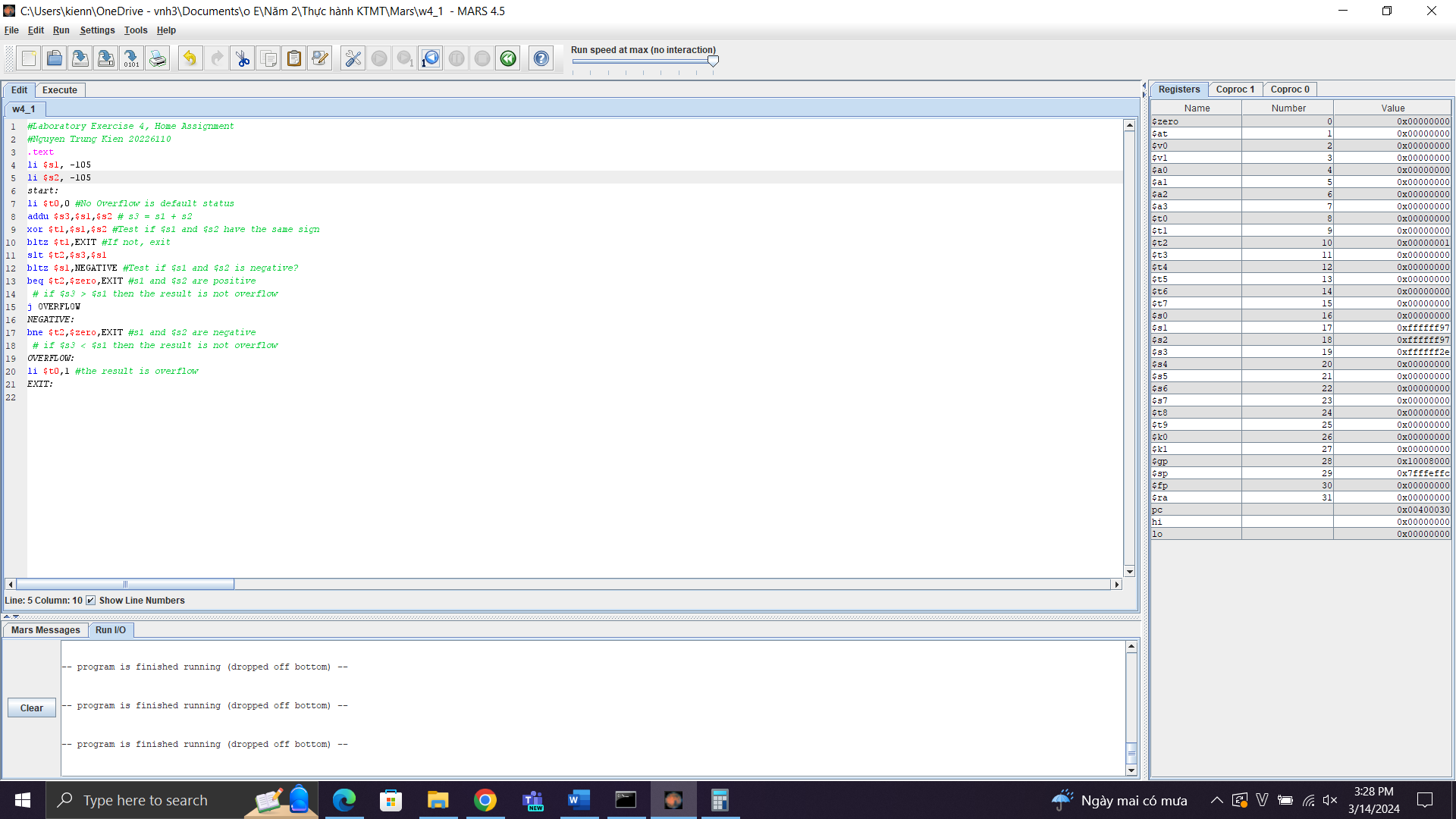
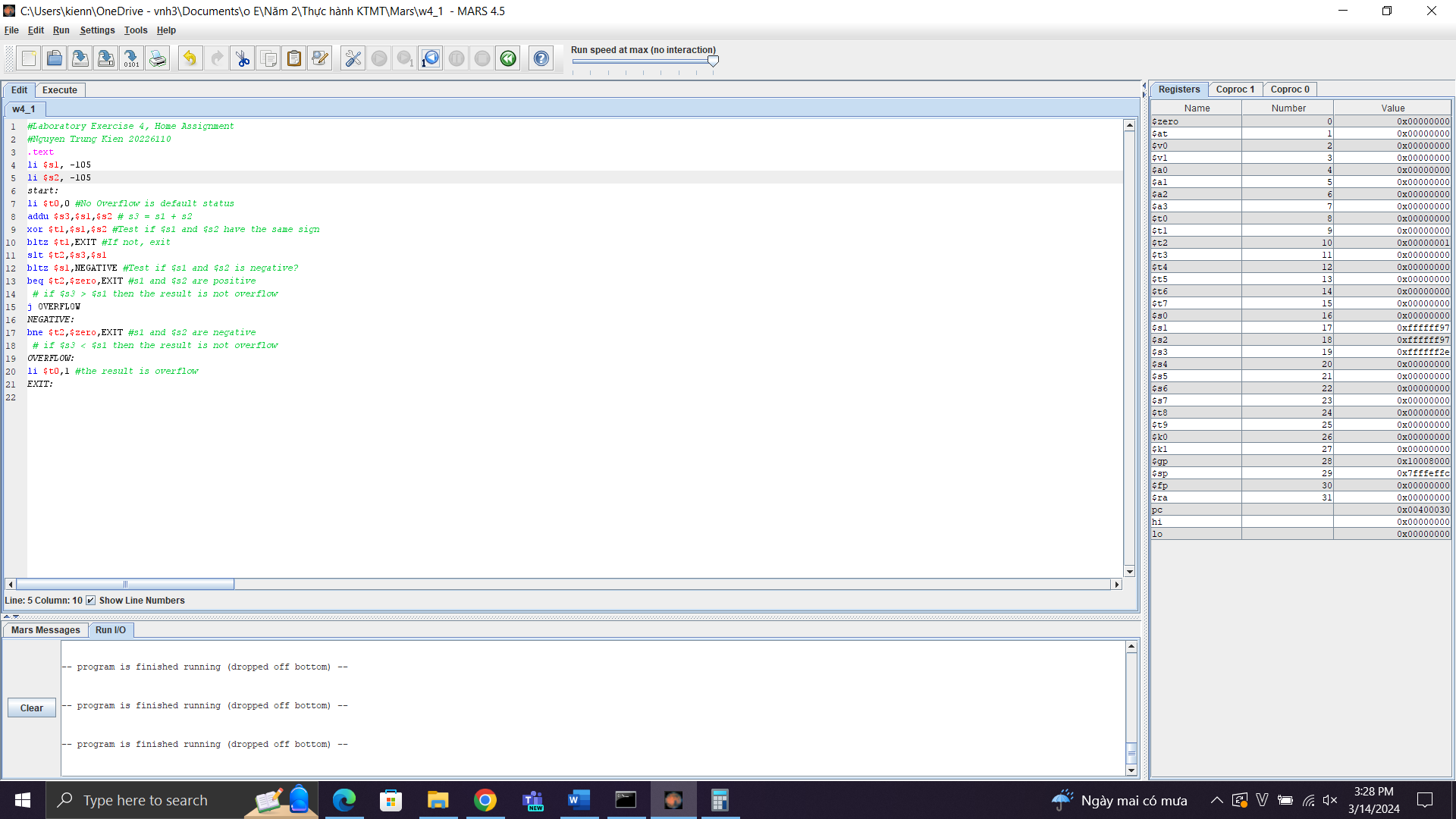
**Student ID: 20226110**

Assignment 1

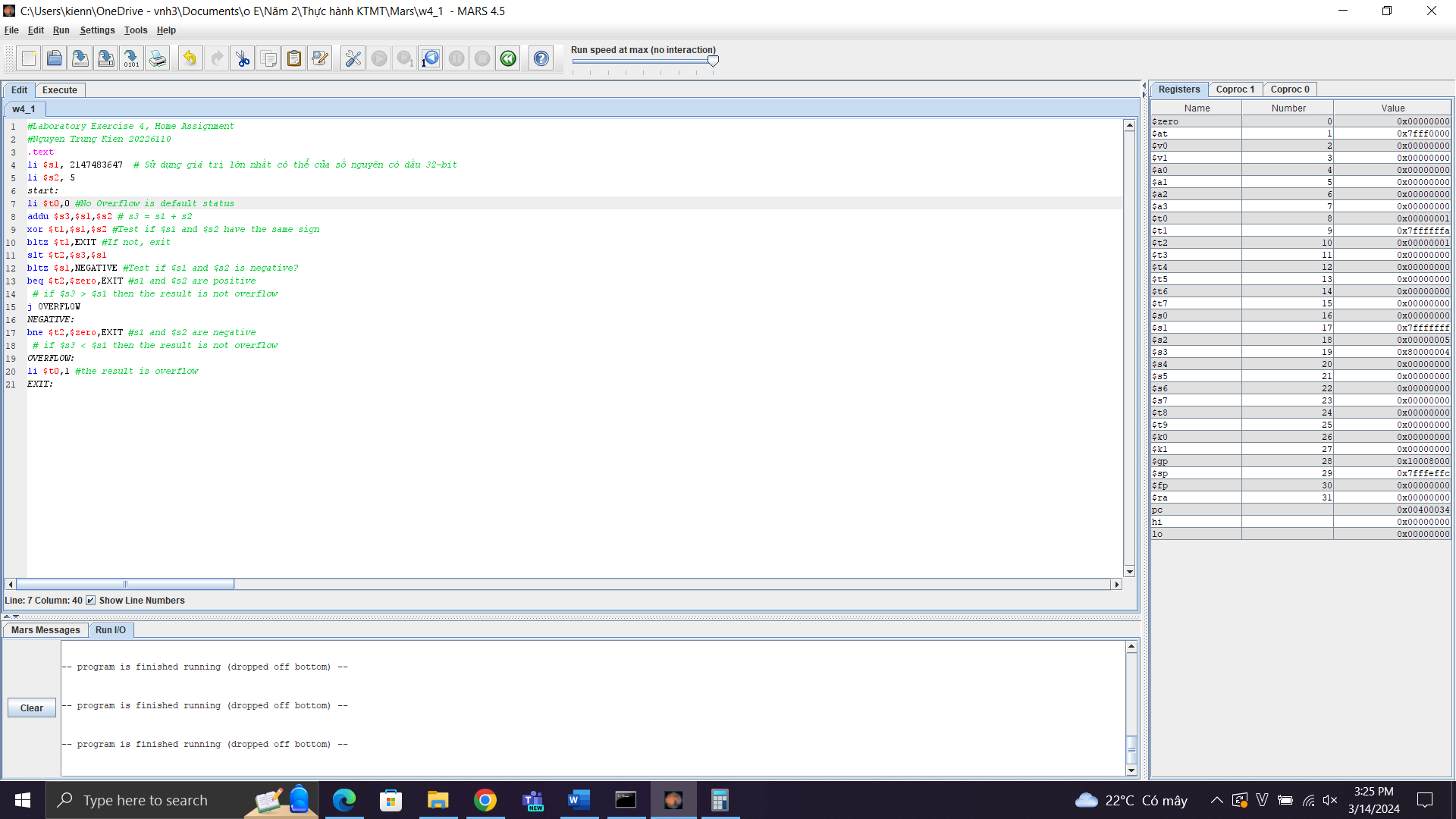
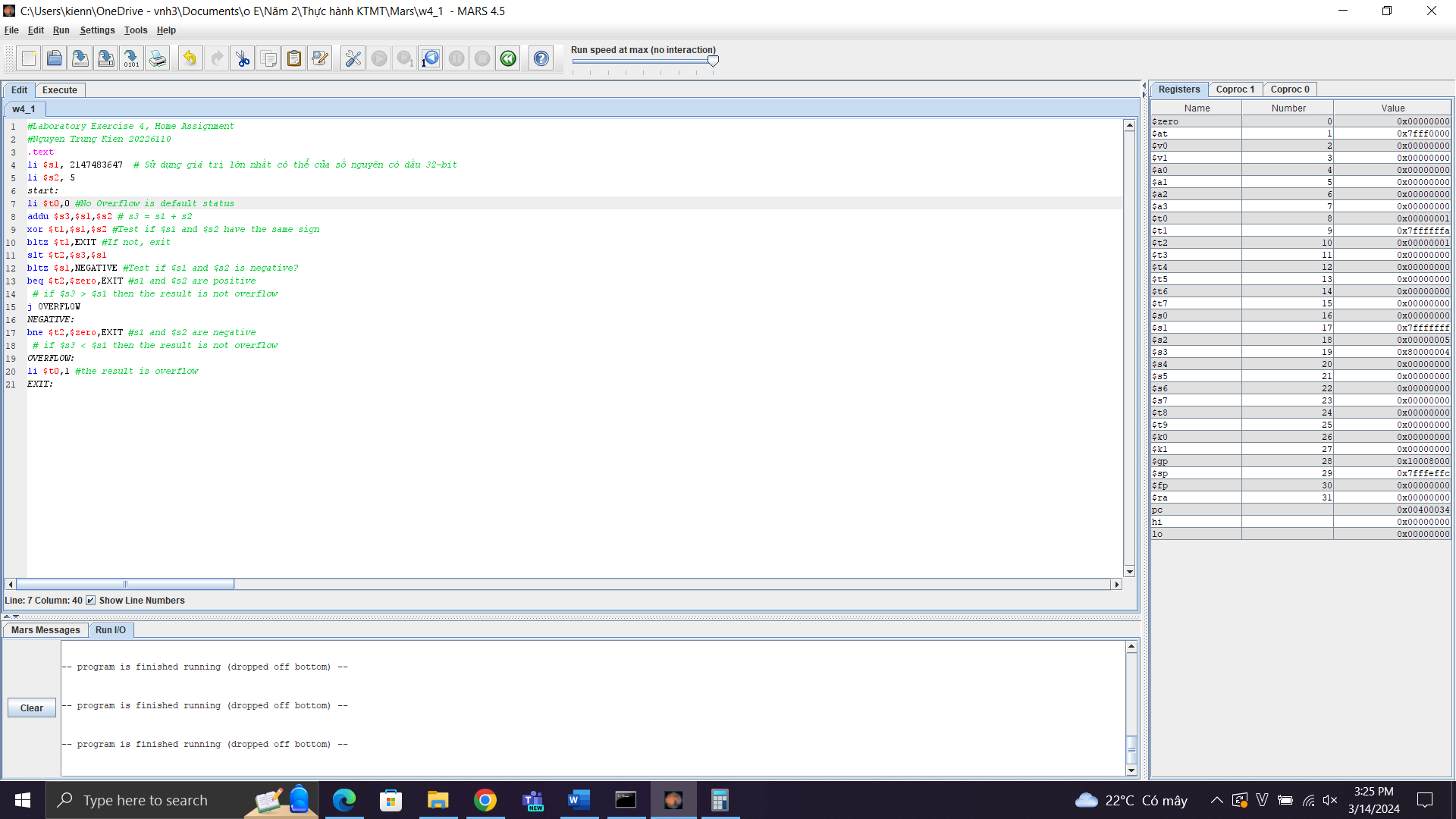
* TH1: Khởi tạo $s1 = $s2 = 105(0x00000069)

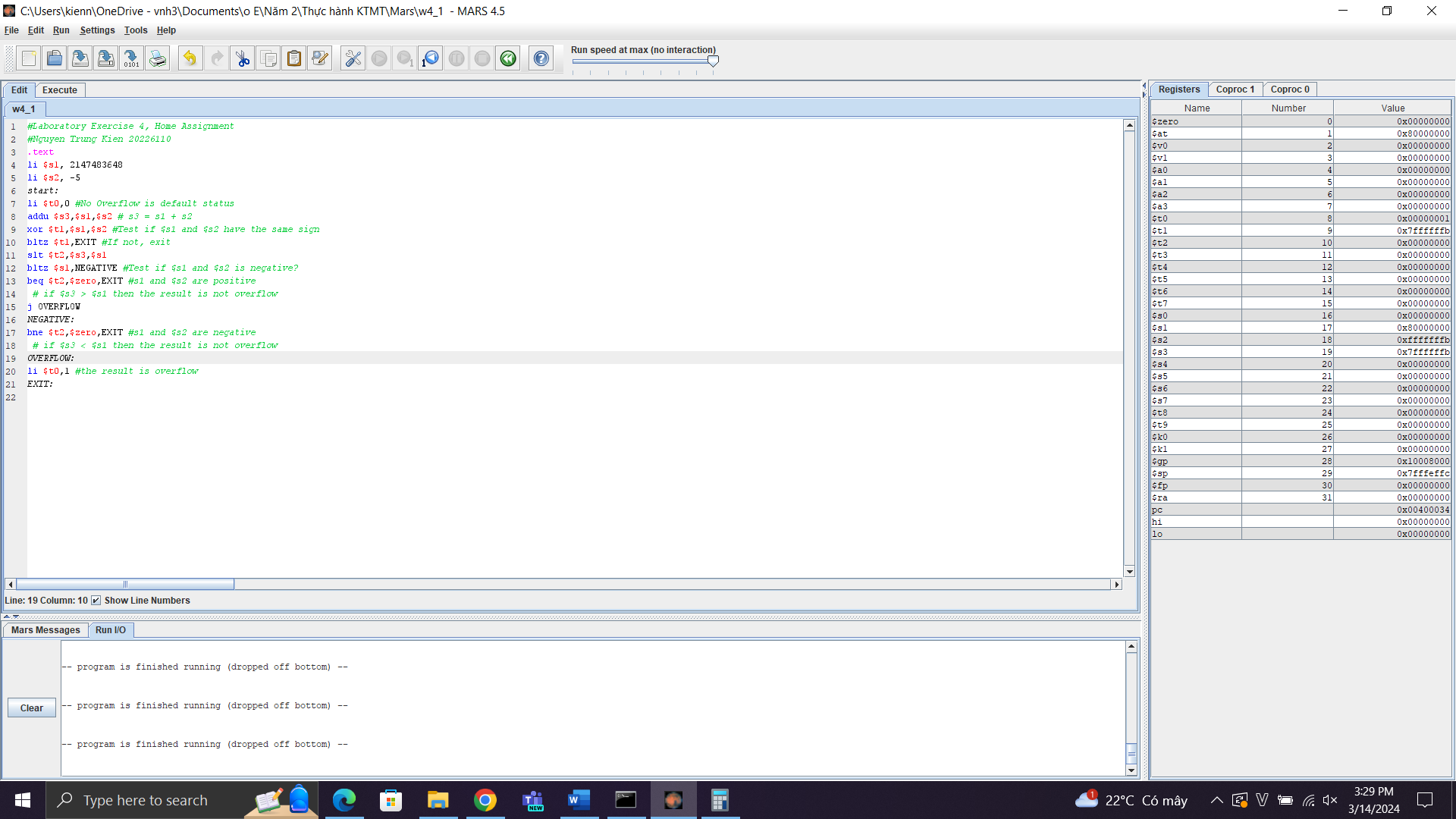
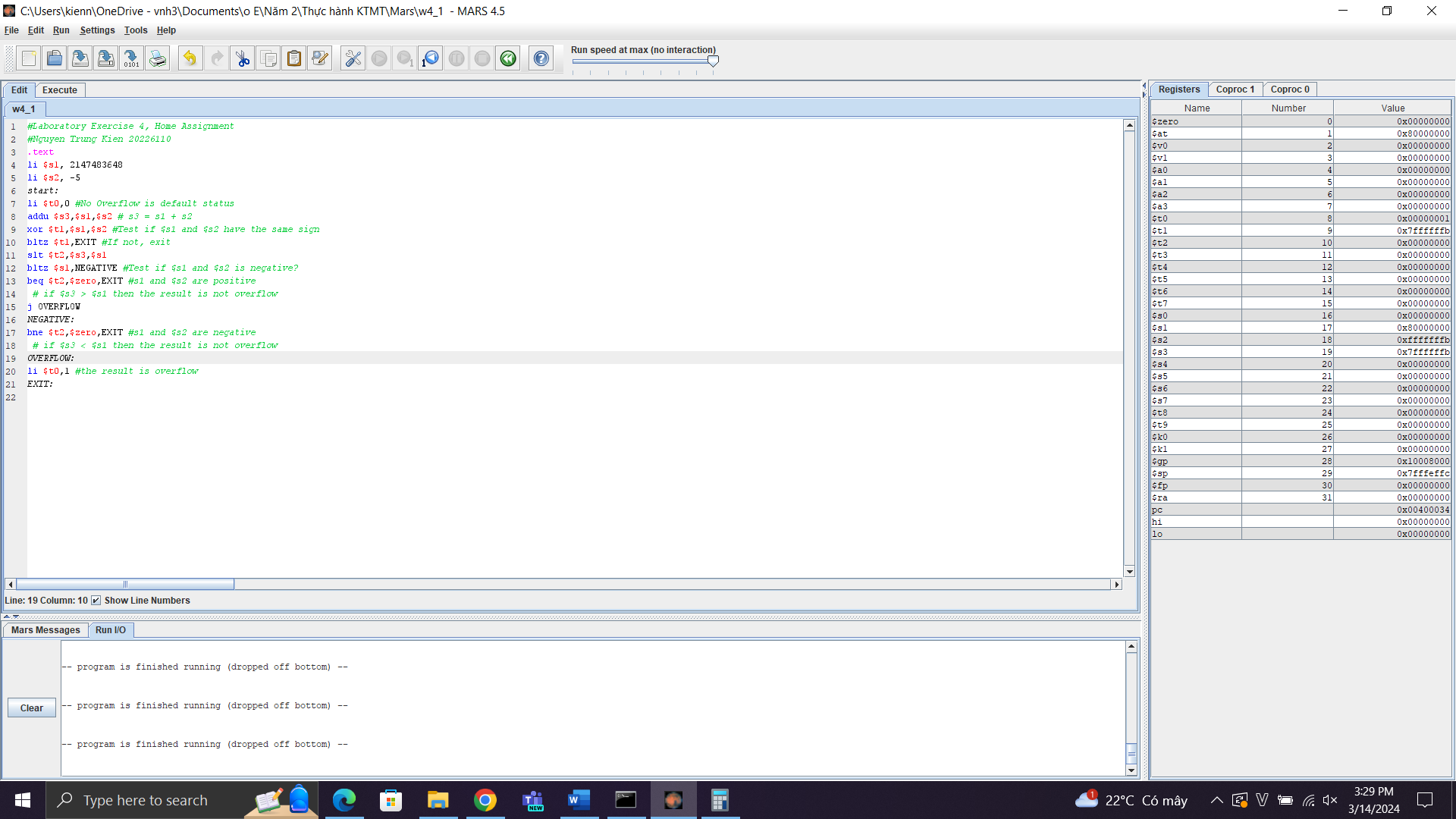
* Chương trình chạy đúng kết quả mong đợi: $t0 = 0 (không OVERFLOW)
* TH2: Khởi tạo $s1 = $s2 = -105(0xffffff97)

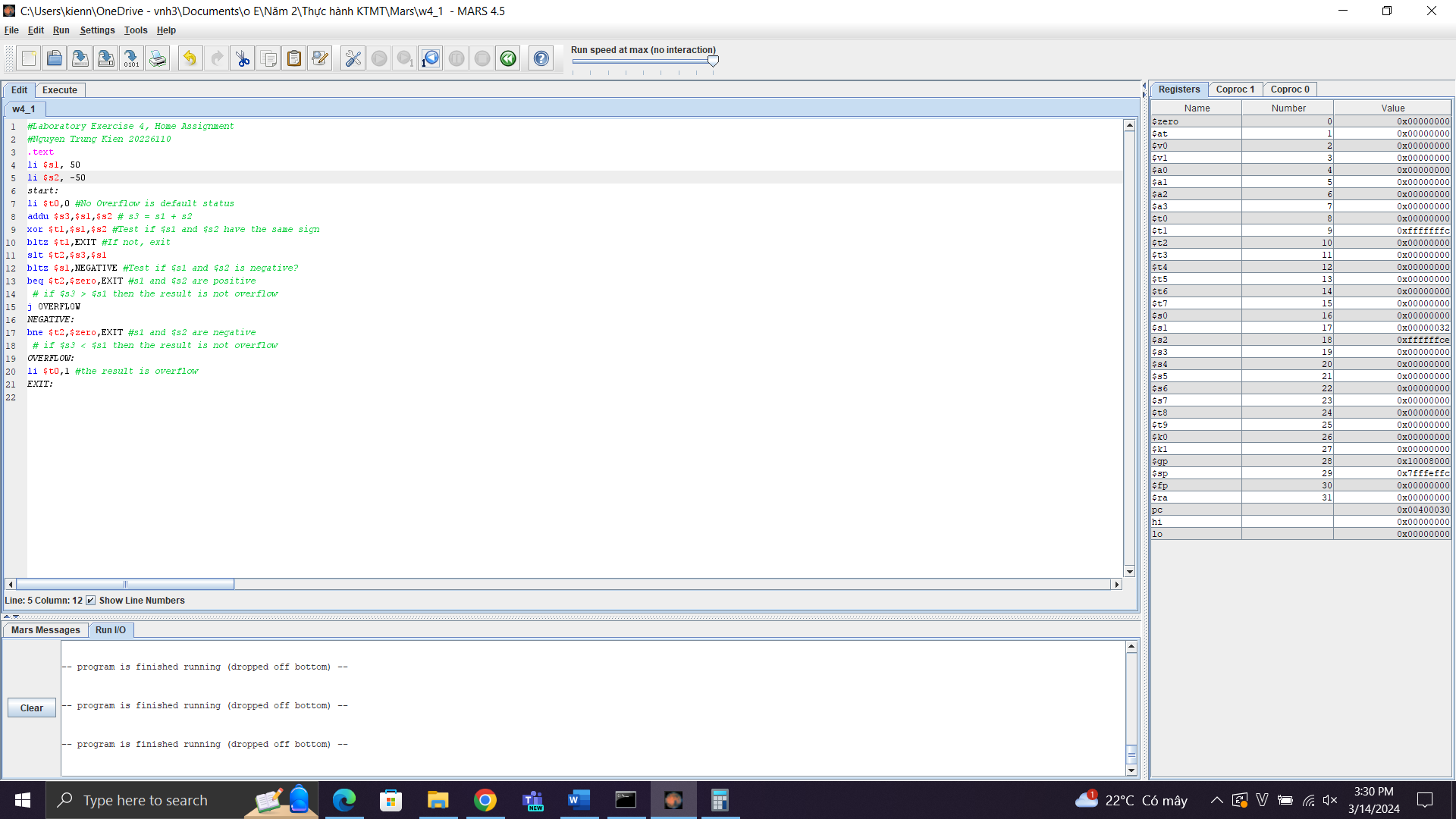
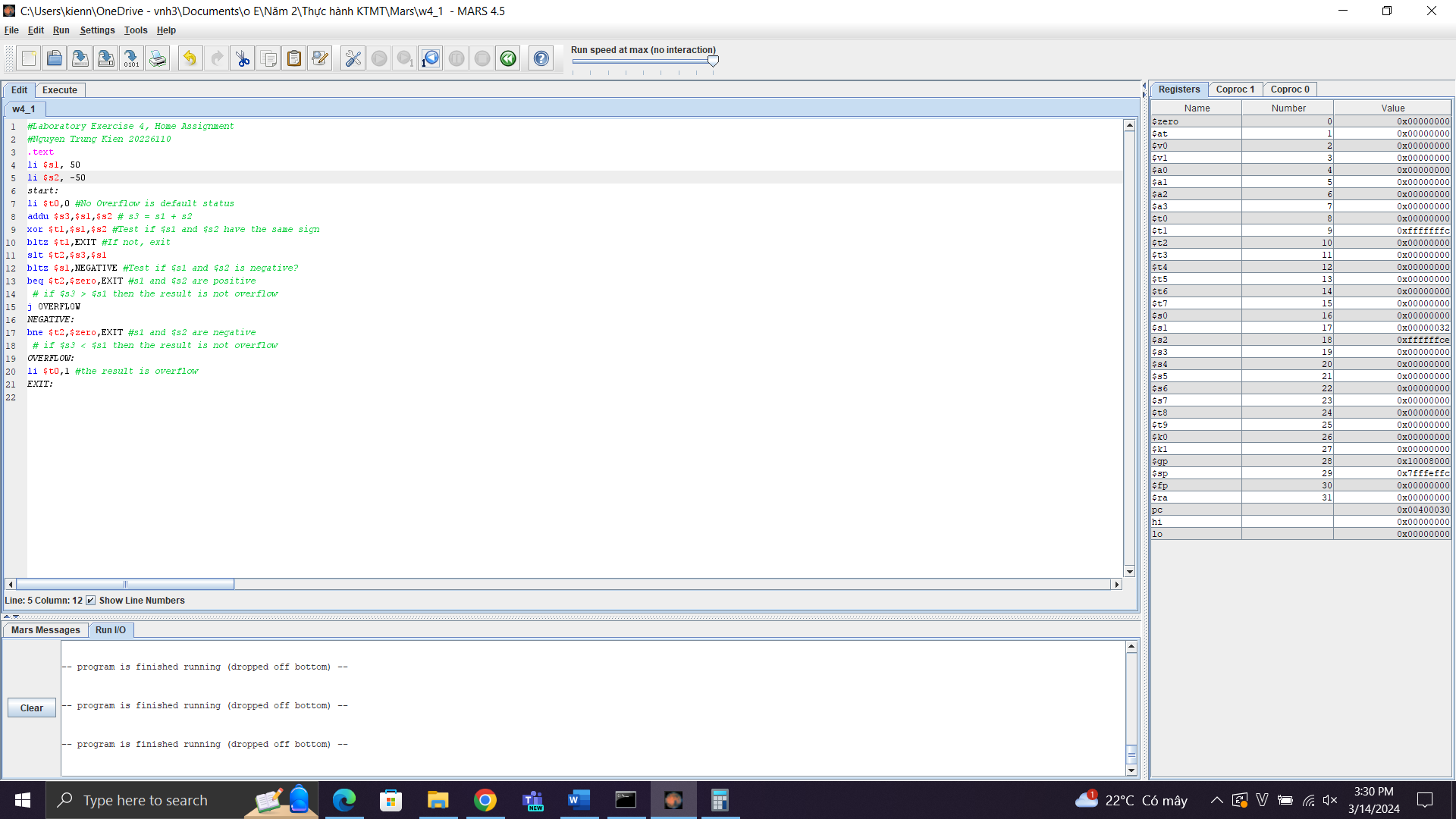
* Chương trình chạy đúng kết quả mong đợi: $t0 = 0 (không OVERFLOW)
* TH3: Khởi tạo $s1 = 2147483648(0x80000000), $s2 = 5(0x00000005)



* Chương trình chạy đúng kết quả mong đợi: $t0 = 1 (OVERFLOW)
* TH4: Khởi tạo $s1 = -2147483648(0x80000000), $s2 = -1(0xffffffff)

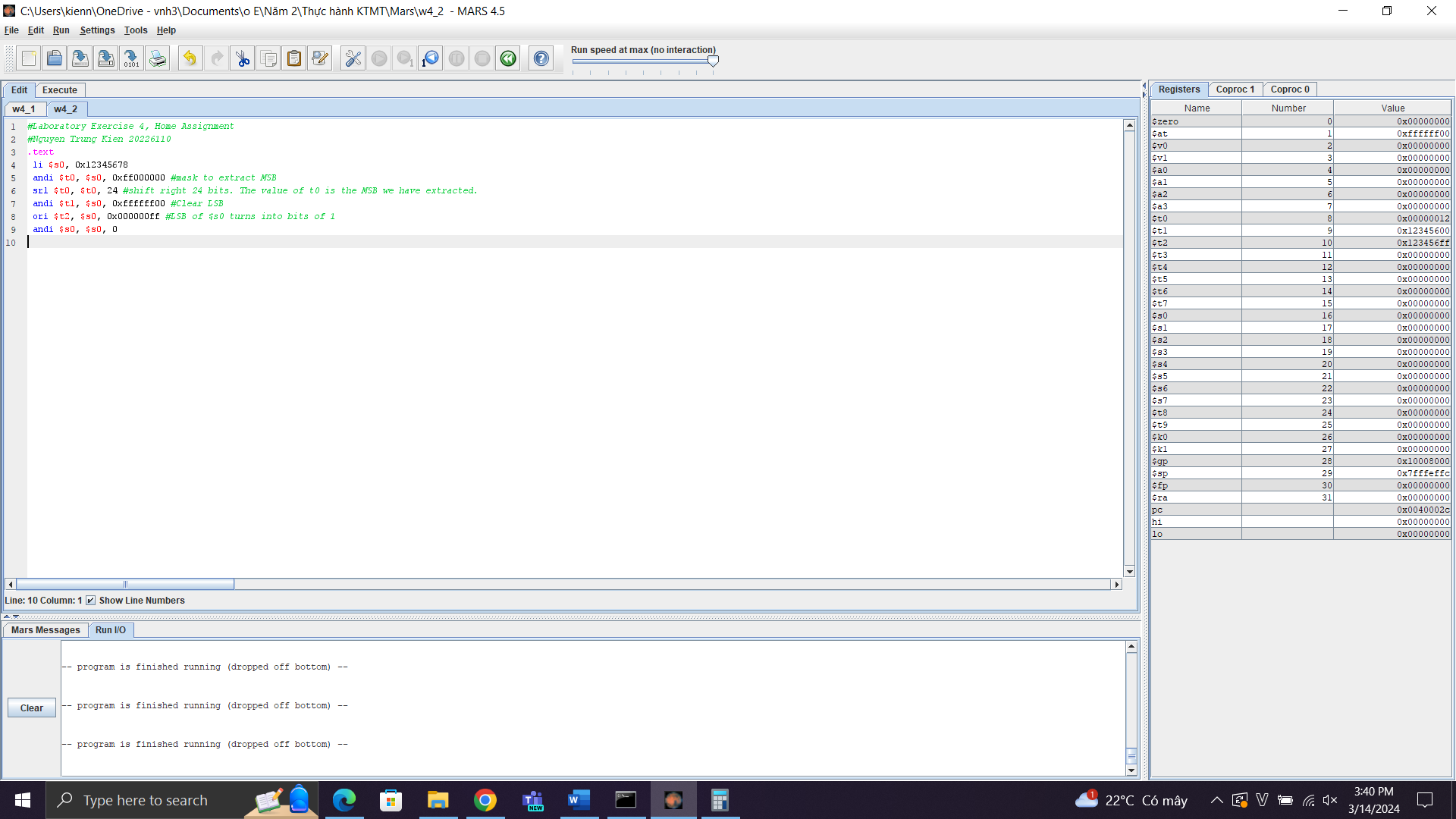
 

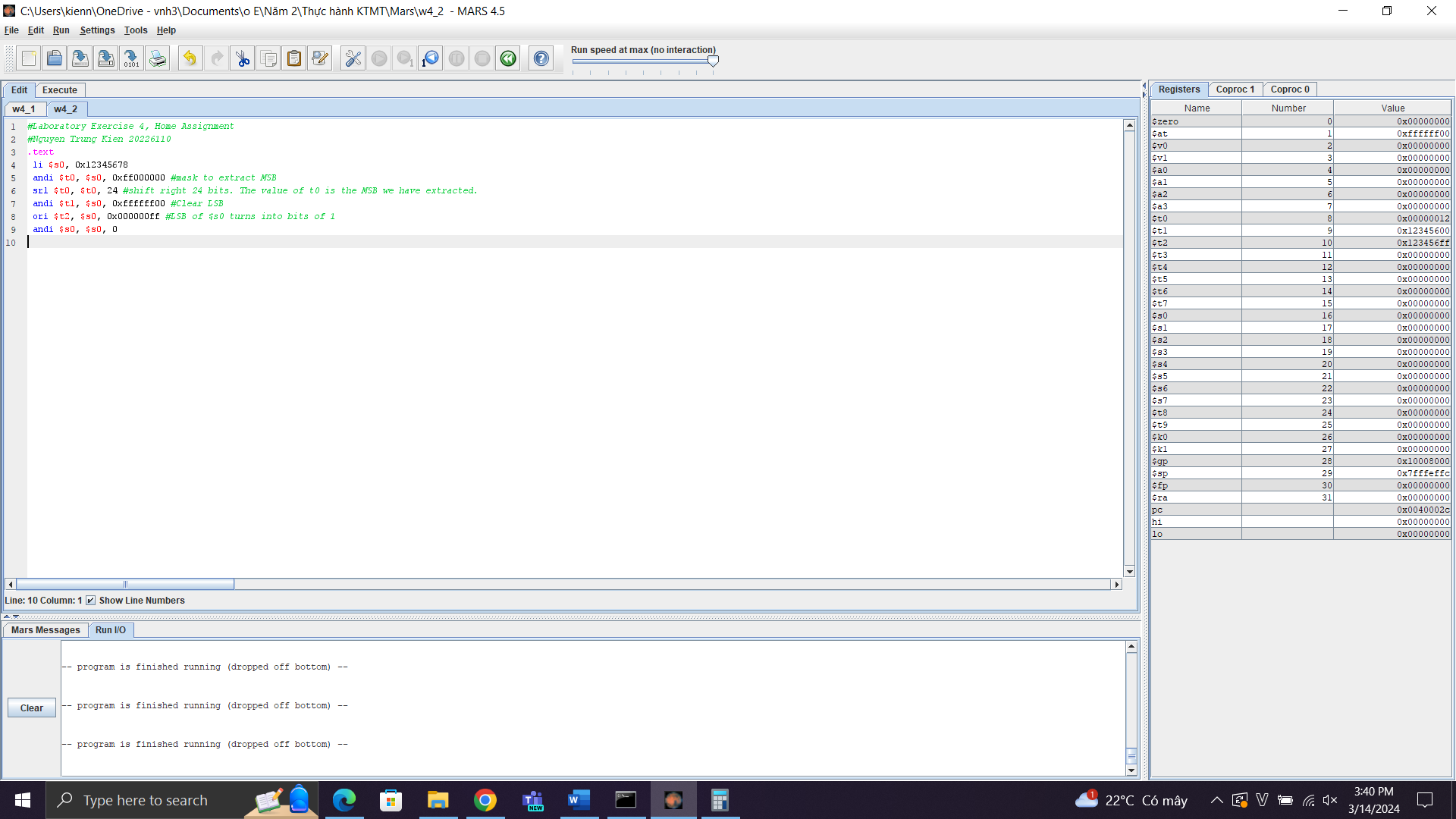
* Chương trình chạy đúng kết quả mong đợi: $t0 = 1 (OVERFLOW)
* TH5: Khởi tạo $s1 =100(0x00000064), $s2 = -100(0xffffff9c)

* Chương trình chạy đúng kết quả mong đợi: $t0 = 0 (không OVERFLOW)

Assignment 2





1. “li $s0, 0x12345678” - Lệnh này tải giá trị “0x12345678” vào thanh ghi $s0.

2. “andi $t0, $s0, 0xff000000” - Lệnh này thực hiện phép AND nhị phân giữa giá trị trong “$s0” và giá trị “0xff000000”, và lưu kết quả vào thanh ghi $t0. Thao tác này cô lập 8 bit cao nhất của $s0.

3. “srl $t0, $t0, 24” - Lệnh này dịch phải nội dung của $t0 24 bit. Xét đến thao tác trước, điều này có nghĩa là 8 bit cao nhất của $s0 giờ đây nằm ở 8 bit thấp nhất của $t0. Ta đã trích xuất thành công giá trị MSB ra thanh ghi t0.

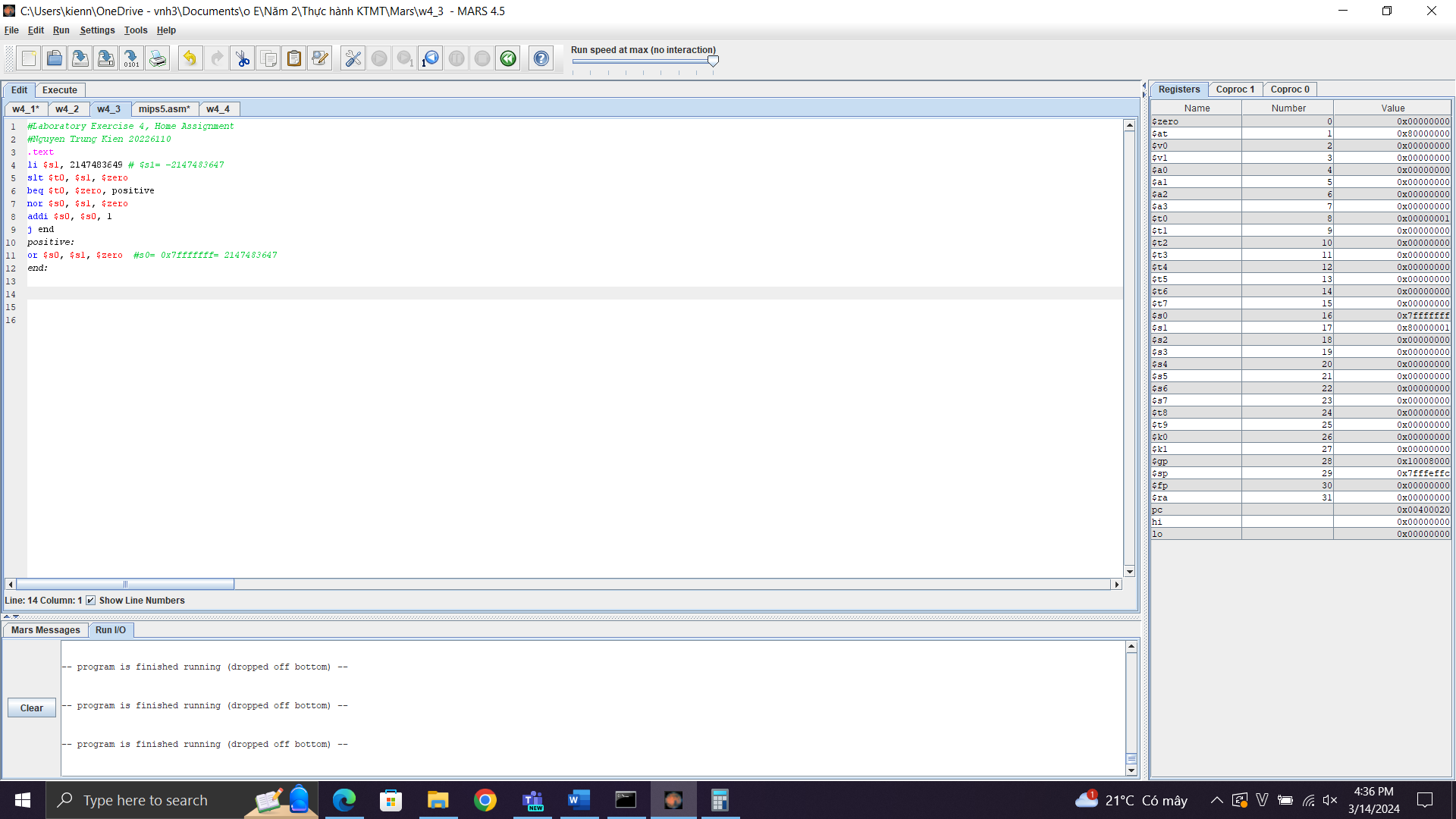
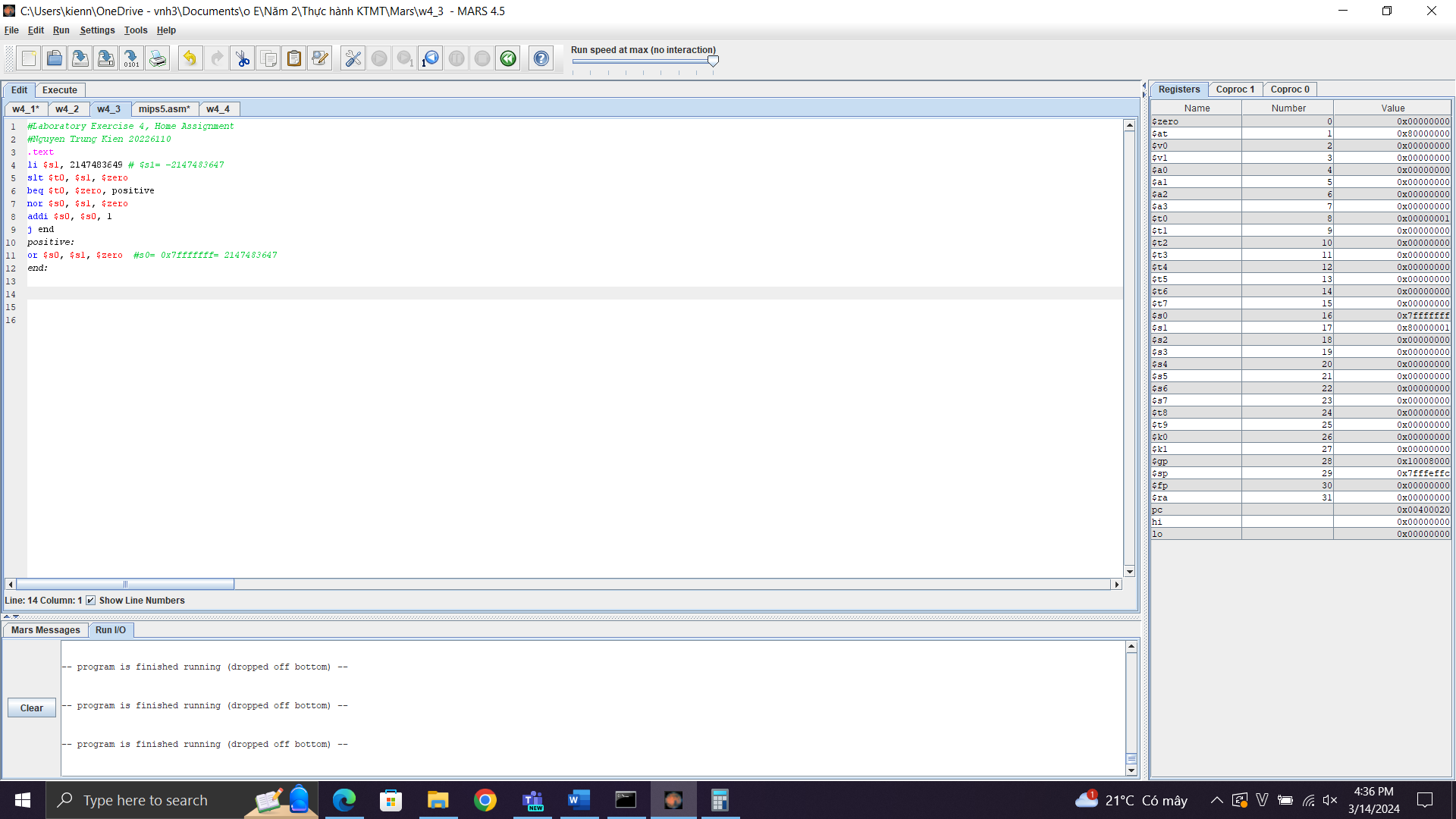
4. “andi $t1, $s0, 0xffffff00” - Thực hiện phép AND nhị phân giữa $s0 và “0xffffff00”, lưu kết quả vào $t1. Thao tác này xóa 8 bit thấp nhất của $s0, giữ nguyên các bit khác không đổi.

5. “ori $t2, $s0, 0x000000ff” - Lệnh này thực hiện phép OR nhị phân giữa $s0 và 0x000000ff, lưu kết quả vào $t2. Điều này đặt 8 bit thấp nhất của $t2 thành 1, trong khi các bit khác giữ nguyên như trong $s0.

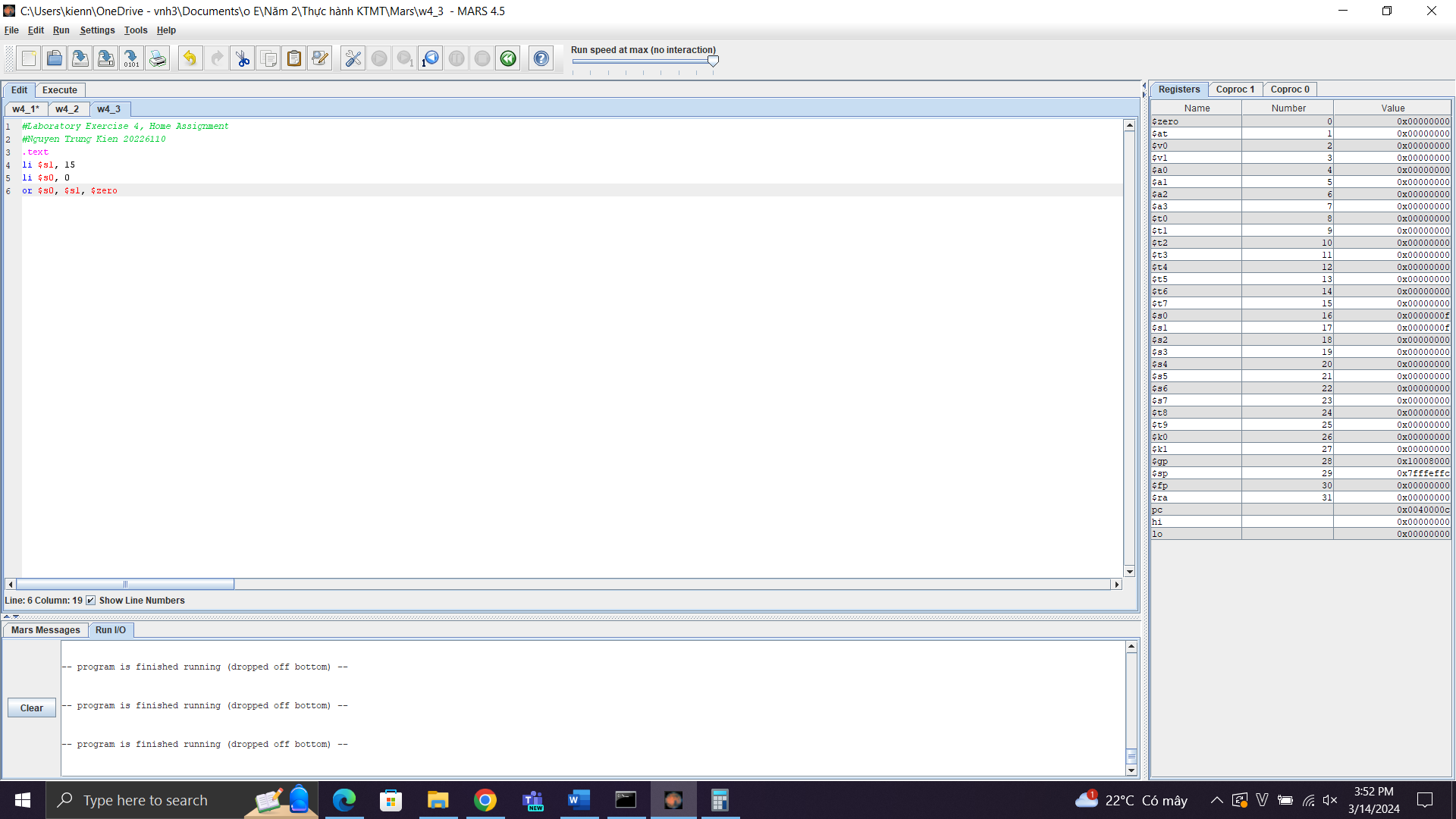
6. “andi $s0, $s0, 0” - Lệnh này thực hiện phép AND nhị phân giữa $s0 và 0, làm sạch tất cả các bit trong $s0, đặt $s0 thành 0.

Assignment 3

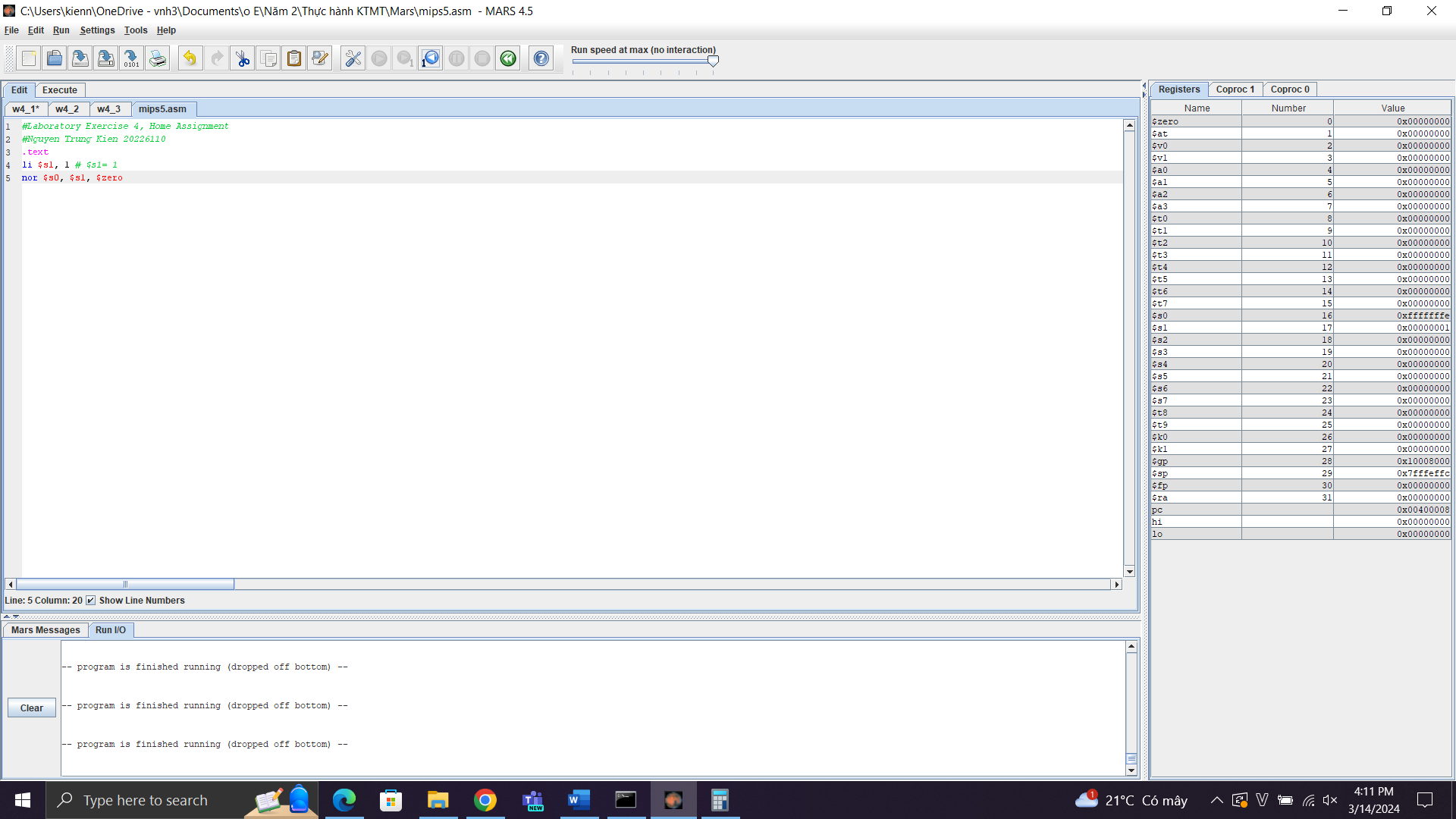
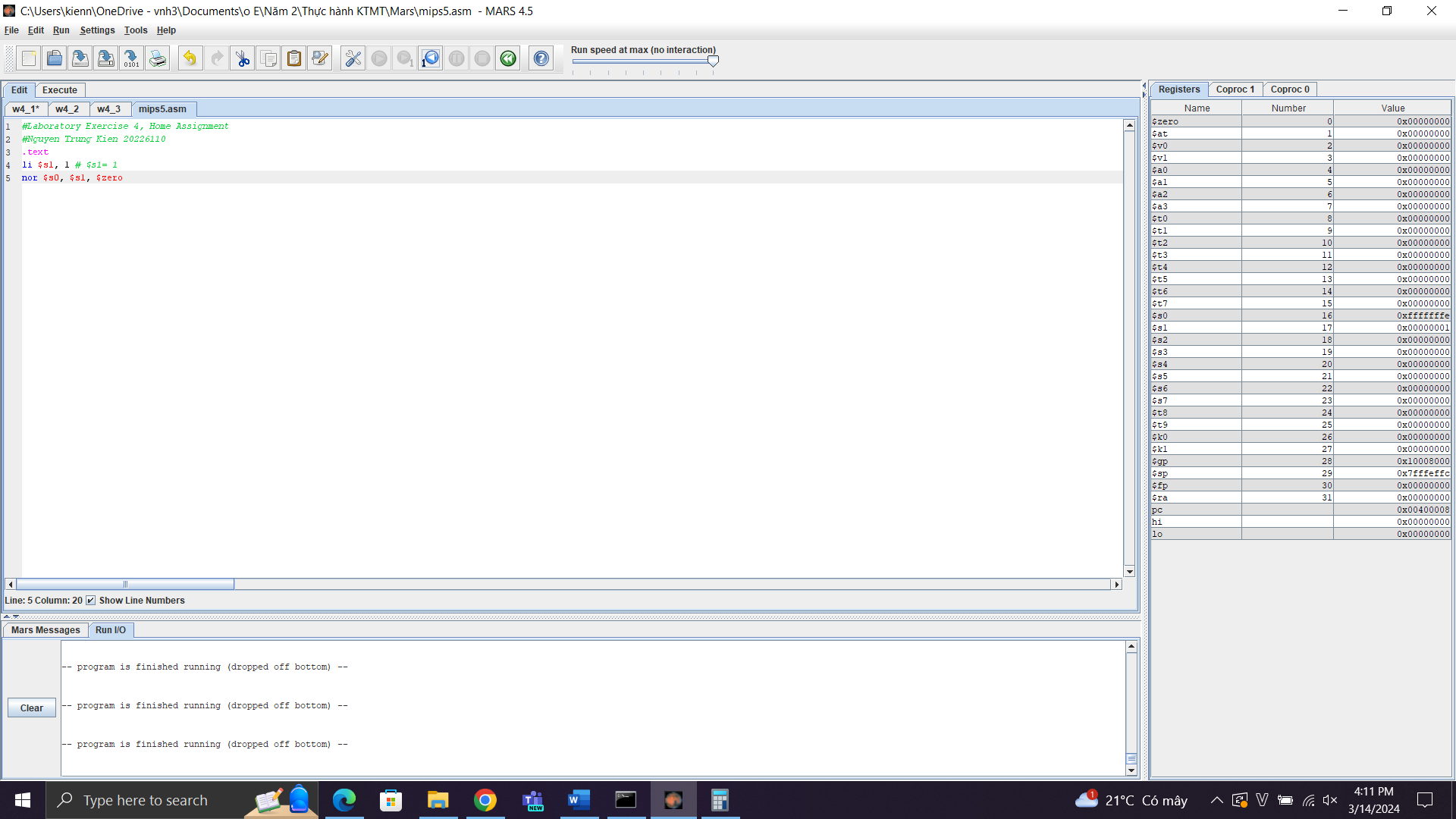
1. abs $s0, $s1

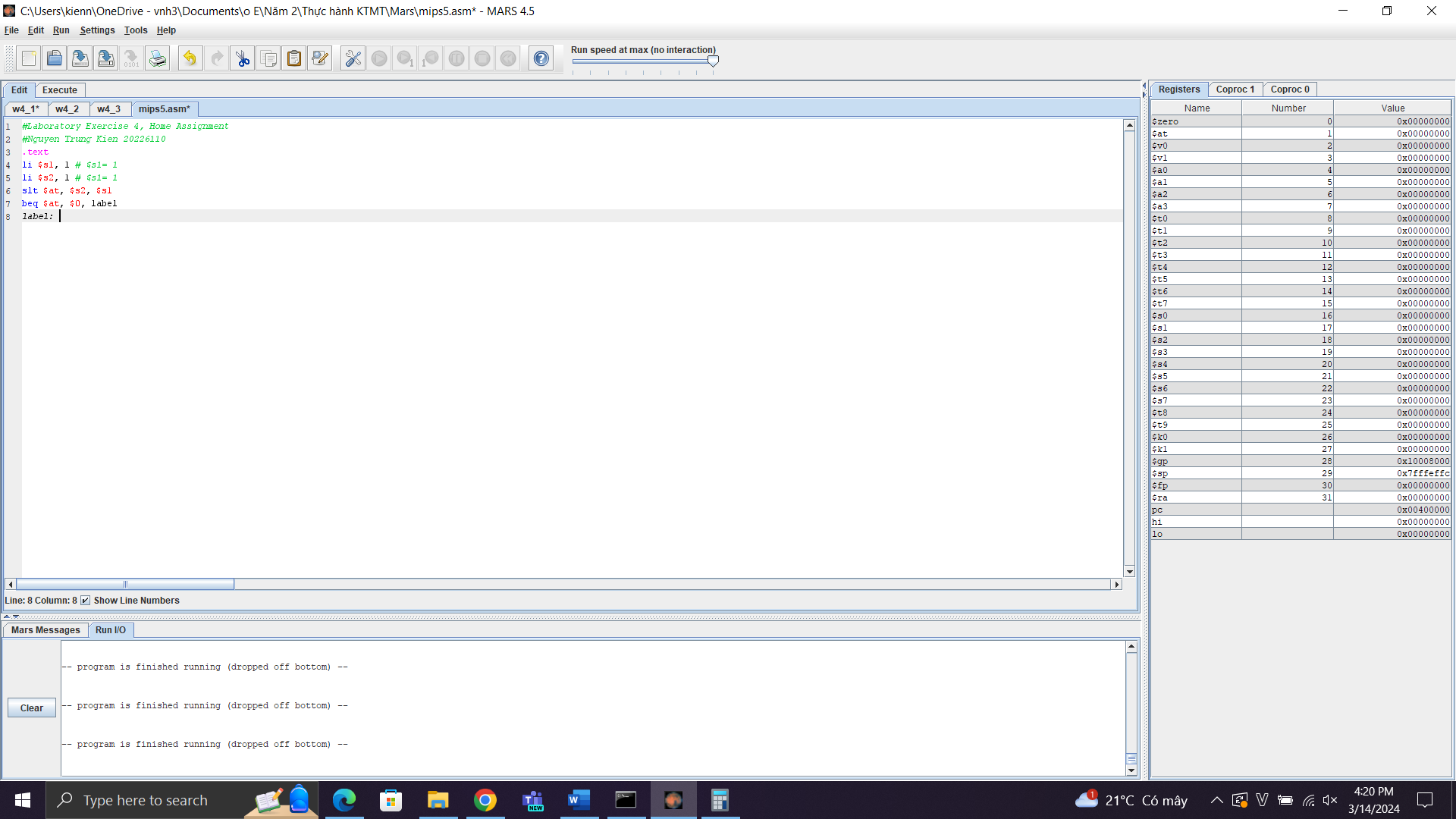
1. move $s0, $s1



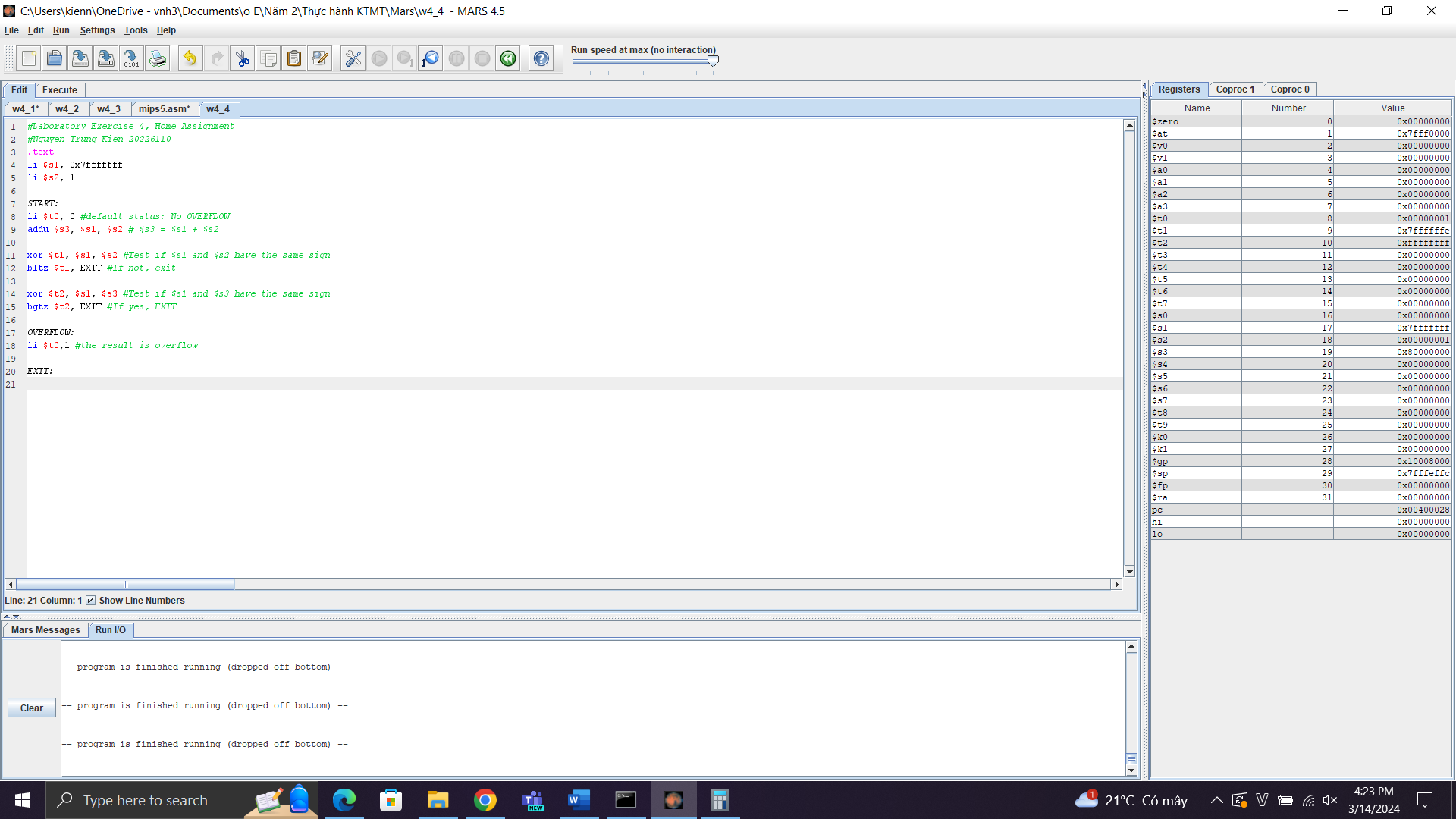
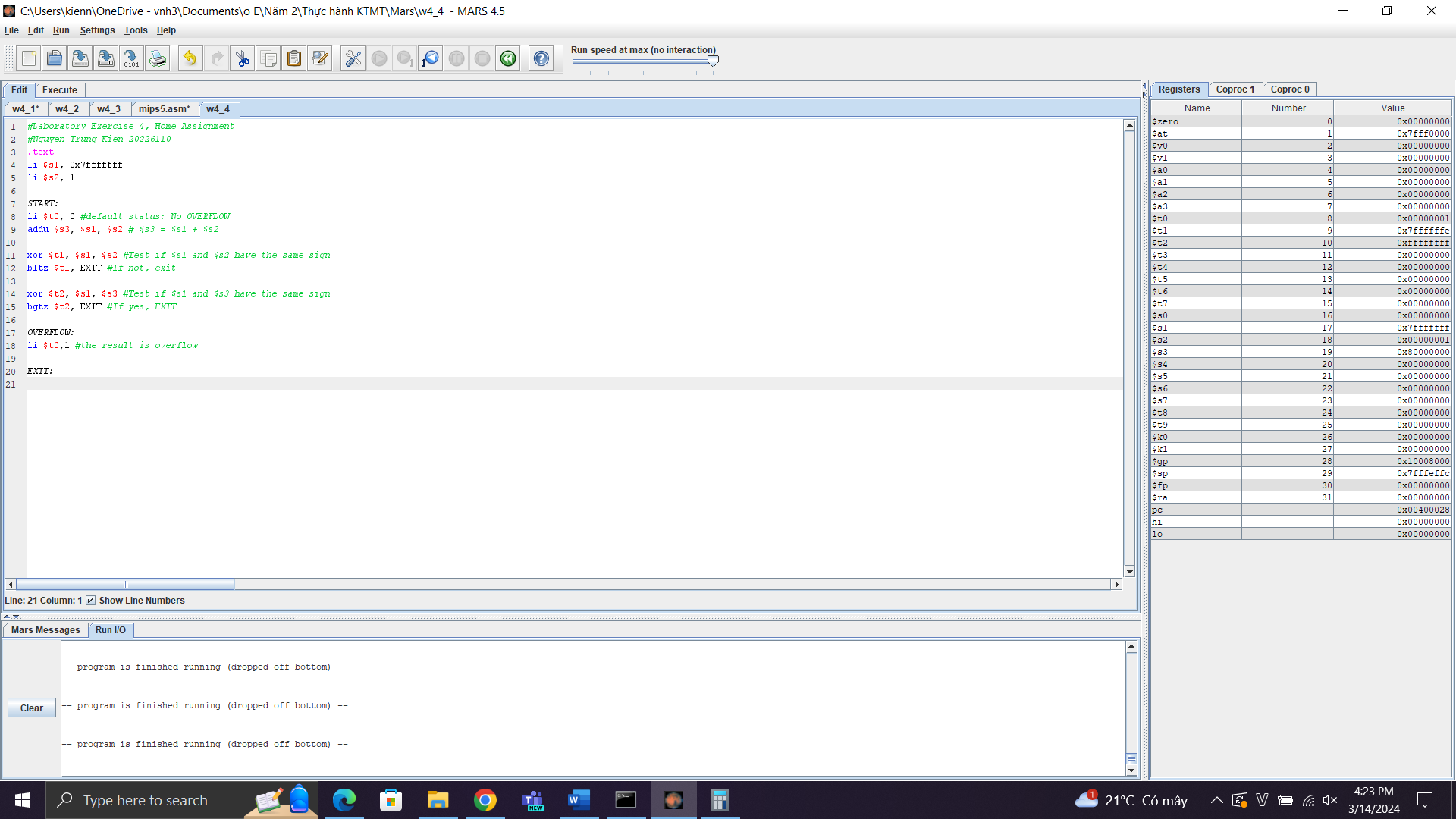
1. not $s0, $s1

1. ble $s1, $s2, label

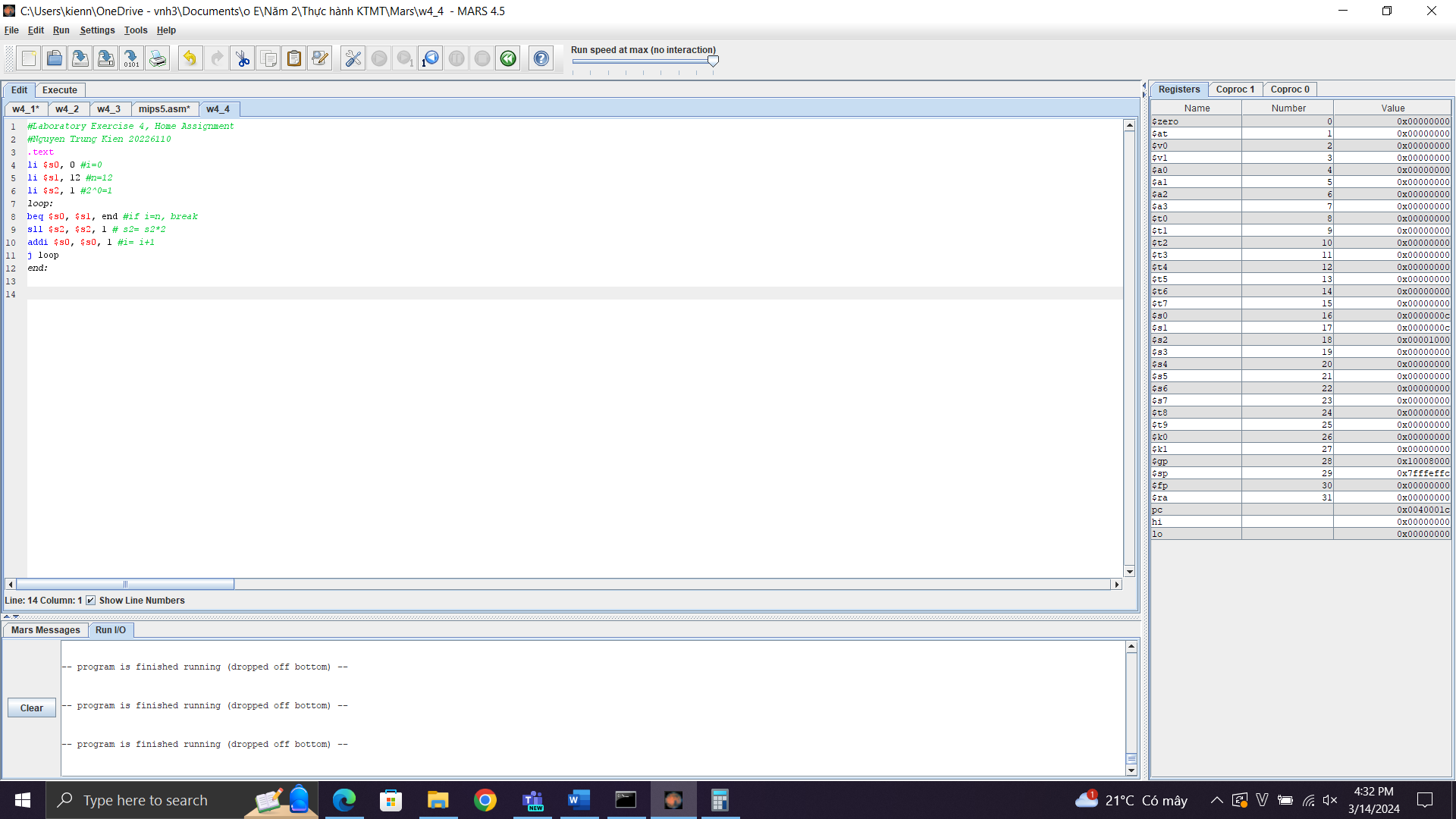


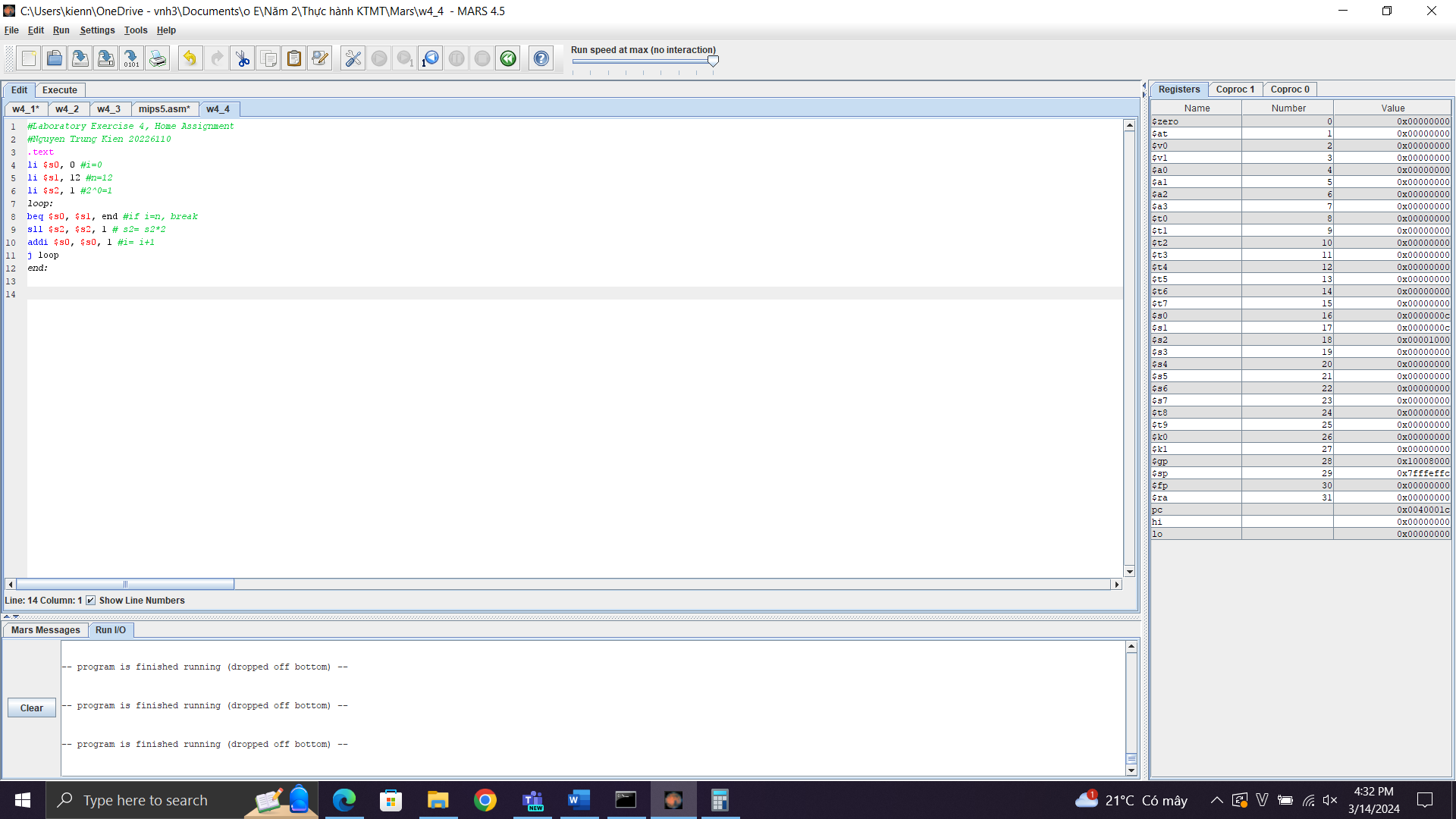
Assignment 4

Assignment 5

Xét trường hợp n = 12($s0), ta sẽ tính 2^n($s1) với kết quả mong đợi là 4096(0x00001000)





* Kết quả ở thanh ghi s2 trùng với kỳ vọng