**Bài thực hành số 4**

**Lớp: 139365 – Học phần: Thực Hành Kiến Trúc Máy Tính   
Đào Minh Nhật – 20215107**

**Assignment 1:**

Code:Ảnh có chứa văn bản

Mô tả được tạo tự động

Giải thích:

Dòng 6: ban đầu gán thanh ghi t0 = 0 là trạng thái không tràn, ngược lại dòng 19 gán thanh ghi t0 = 1 là trạng thái tràn.

Dòng 7: tổng s3 = s1 + s2

Dòng 8: kết quả của phép xor giữa s1 và s2 đc gán vào thanh t1 dùng để xét xem s1 và s2 có cùng dấu hay không.

Dòng 9: Nếu t1 < 0 tức là s1 và s2 là 2 số trái dấu nhau thì sẽ nhảy tới exit, t0 = 0 trạng thái không tràn.

Dòng 10:Nếu s3 < s1 => t2 = 1, ngược lại t2 = 0

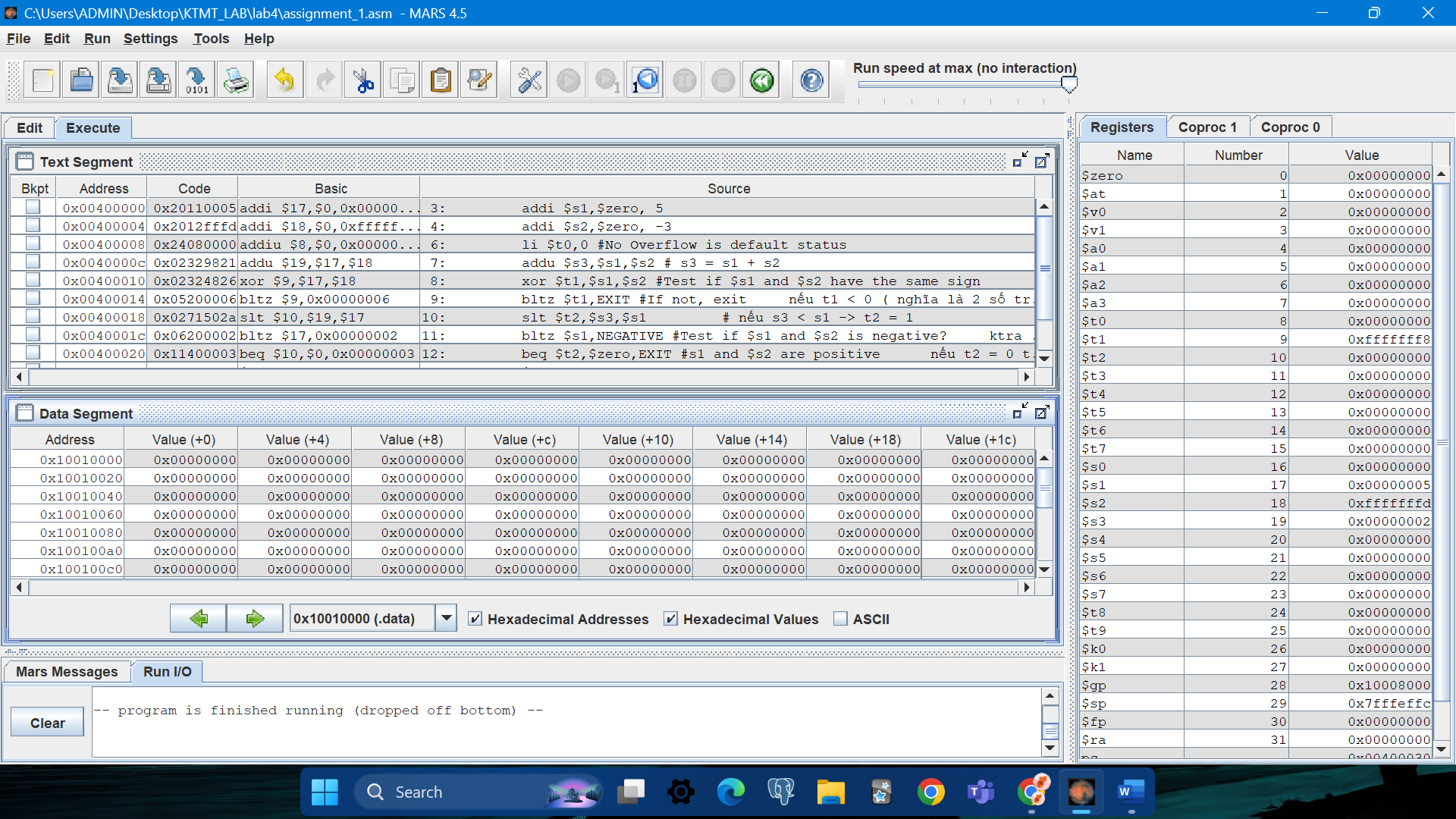
Dòng 11: Nếu s1 < 0 ( s1 là số âm ) thì nhảy tới NEGATIVE. Ngược lại thì thực hiện tiếp dòng 12.

Dòng 12: Nếu t2 = 0 tức là s3 > s1 (s1, s3 dương) => Nhảy tới exit, trạng thái không tràn.

Dòng 14: Lúc này s3 và s1 dương và s3 < s1 => nhảy tới overflow, thực hiện dòng 19 => trạng thái tràn.

Dòng 16: Nếu t2 != 0 tức là s3 < s1 ( s3, s1 âm) => exit, trạng thái không tràn. Ngược lại là trạng thái s3 > s1( s3, s1 âm) => trạng thái tràn.

Kết quả:



**Home Assignment 2:**

Code:

Ảnh có chứa văn bản

Mô tả được tạo tự động

Giải thích:

Dòng 3: load giá trị 0x0563 vào thanh ghi $s0.

Dòng 4: trích xuất byte cuối cùng của s0 vì 0xff có tất cả các bit bằng 1 trên byte thấp nhất và các bit bằng 0 trên các byte khác.

Dòng 5: trích xuất bit thứ 10 của s0 vì vì 0x0400 có bit thứ 10 bằng 1 và các bit còn lại bằng 0.

Kết quả:

Ảnh có chứa văn bản, ảnh chụp màn hình, máy tính

Mô tả được tạo tự động

**Assignment 2:**

Code:

Ảnh có chứa văn bản

Mô tả được tạo tự động

Kết quả:

Ảnh có chứa văn bản, ảnh chụp màn hình, máy tính

Mô tả được tạo tự độngẢnh có chứa văn bản, ảnh chụp màn hình, máy tính

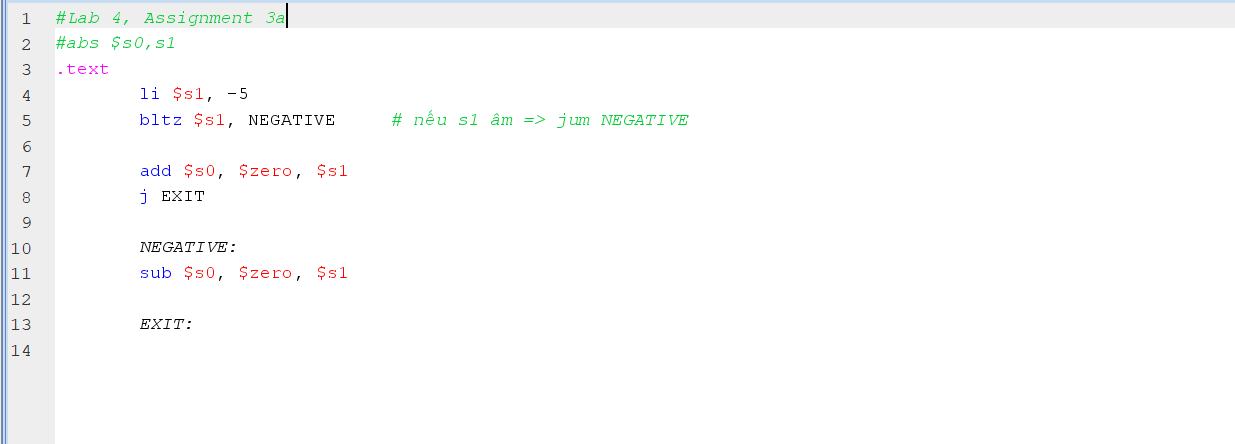
Mô tả được tạo tự độngẢnh có chứa văn bản, ảnh chụp màn hình, máy tính

Mô tả được tạo tự động

**Assignment 3:**

*a. abs $s0,s1*

Code:

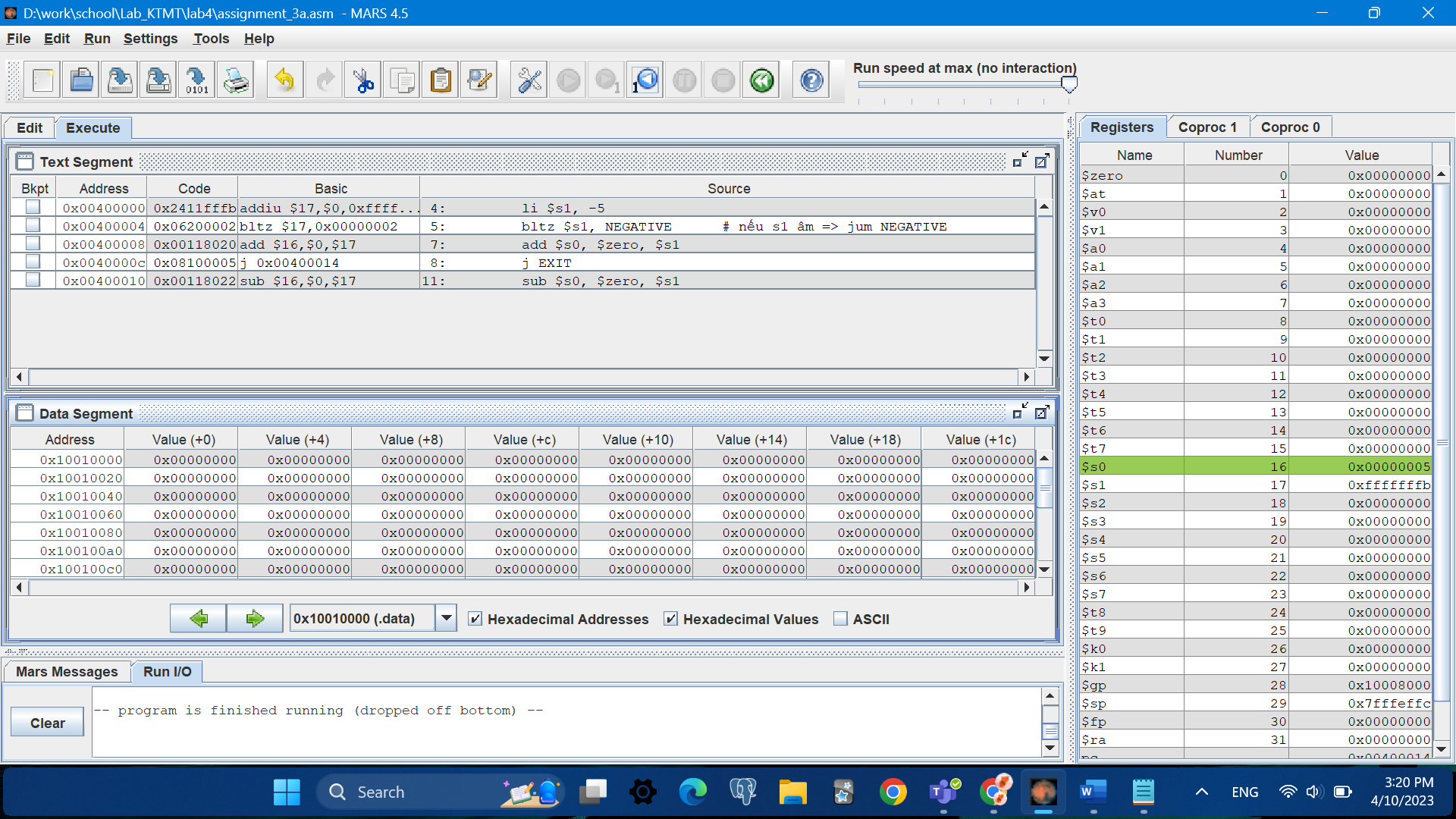


Giải thích:

Dòng 5: Nếu s1 < 0 thì sẽ rẽ nhánh tới NEGATIVE để thực hiện dòng 11 lấy số đối của s1 ghi vào thanh s0. Ngược lại thì thực hiện tiếp dòng 7.

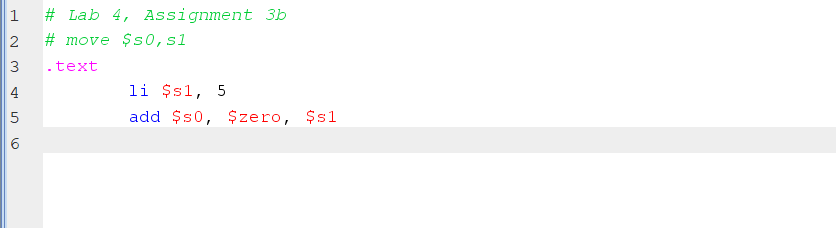
Dòng 7: Thực hiện phép cộng 0 với s1 rồi ghi vào thanh s0.

Kết quả:



*b. move $s0,s1*

Code:



Giải thích:

Dòng 5: thực hiện phép cộng với 0 để gán giá trị của thanh ghi s1 vào thanh ghi s0

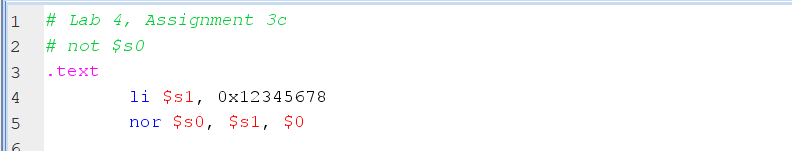
Kết quả:

Ảnh có chứa văn bản

Mô tả được tạo tự động

*c. not $s0*

Code:

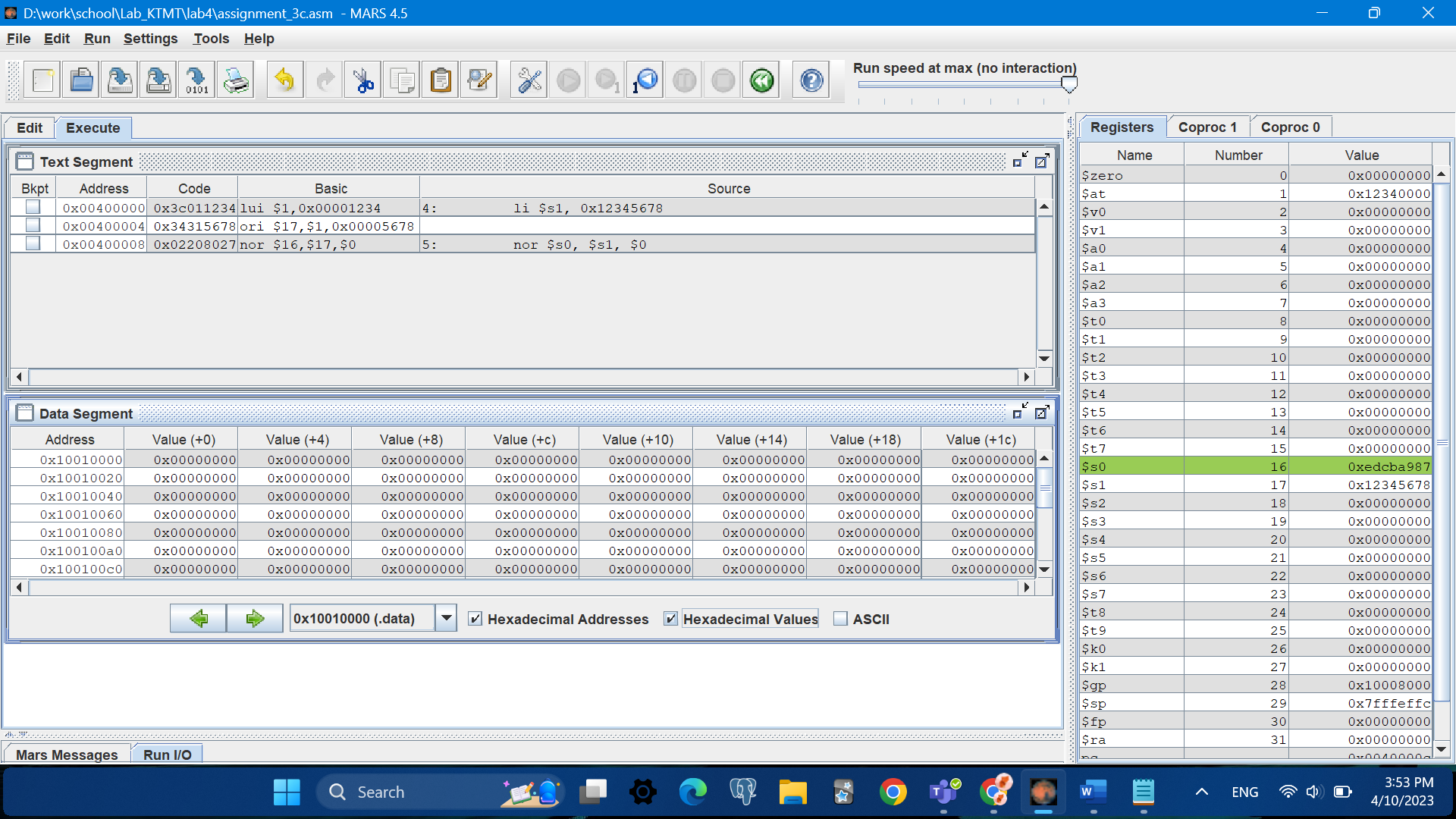


Giải thích:

Dòng 5: Để đảo bit ta dùng lệnh nor giữa s1 với 0

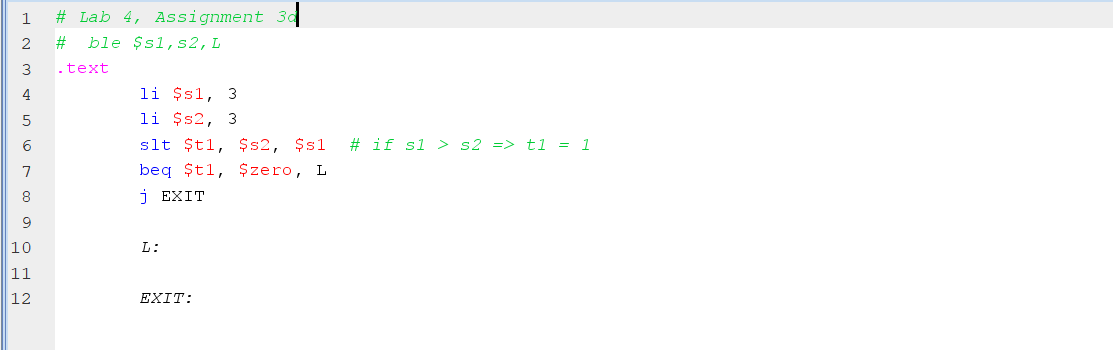
Vì 1 nor 0 = 0 ; 0 nor 0 = 1

Kết quả:



*d. ble $s1,s2,L*

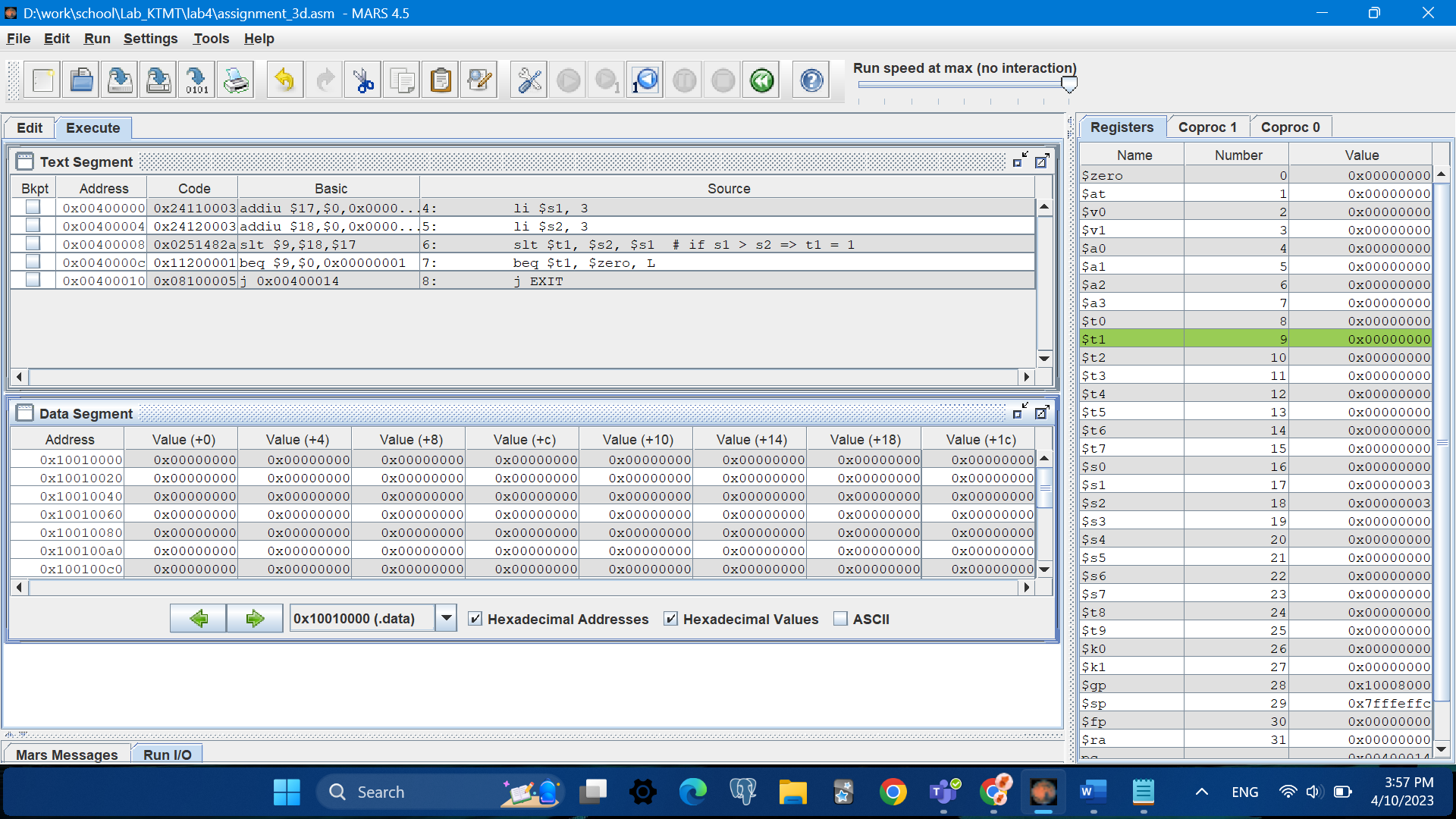
Code:



Giải thích:

Dòng 6: Nếu s2 < s1 thì t1 = 1

Dòng 7: Nếu t1 = 0 tức là s1 >= s2 thì rẽ nhánh tới L, ngược lại thực hiện tiếp dòng 8.

Kết quả: 

**Assignment 4:**

Code:

Ảnh có chứa văn bản

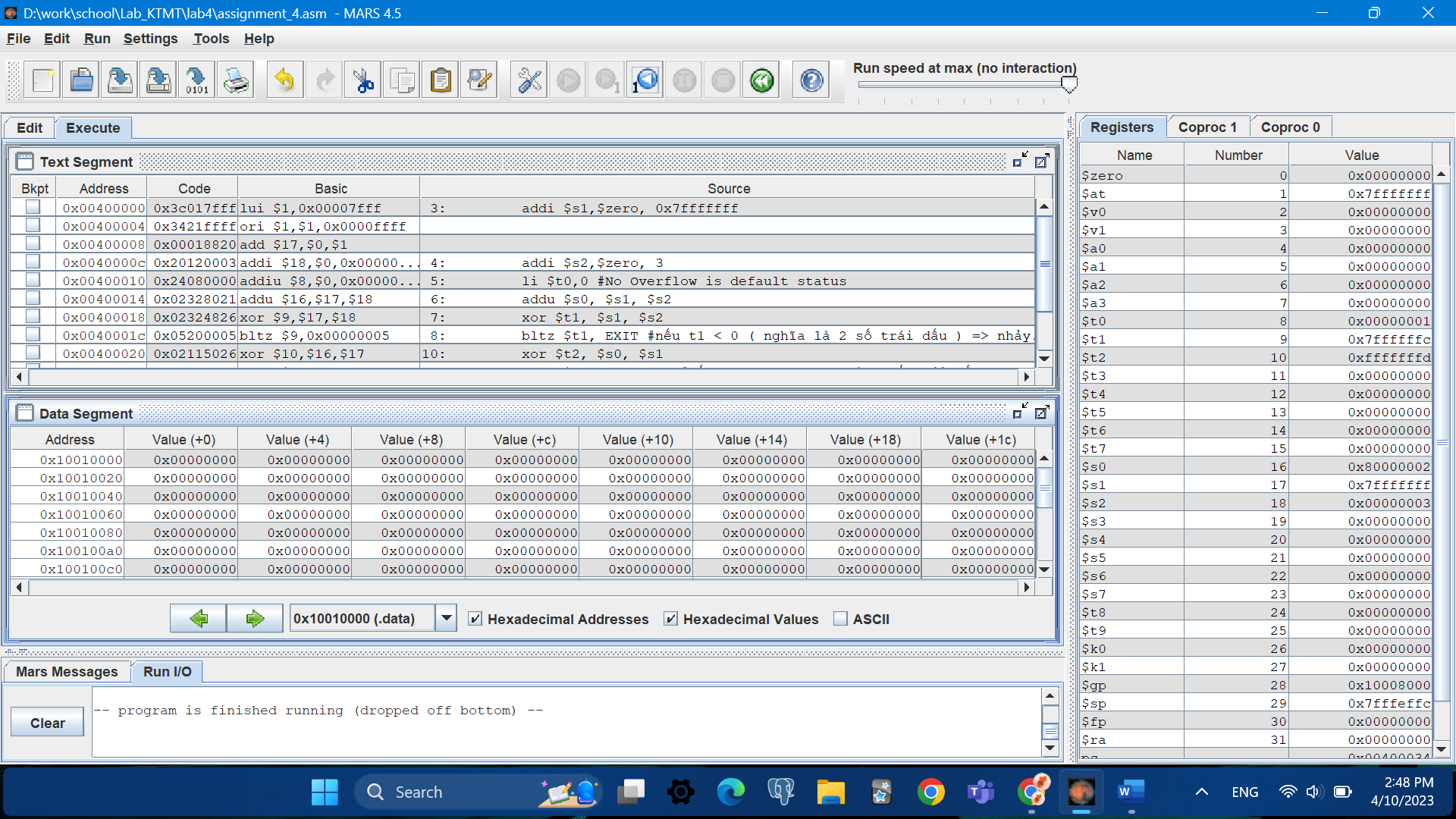
Mô tả được tạo tự động

Giải thích:

Dòng 8: so sánh kết quả xor giữa s1 và s2, nếu t1 < 0 nghĩa là 2 số trái dấu, nhảy tới exit, trạng thái không tràn. Ngược lại thực hiện tiếp dòng số 10.

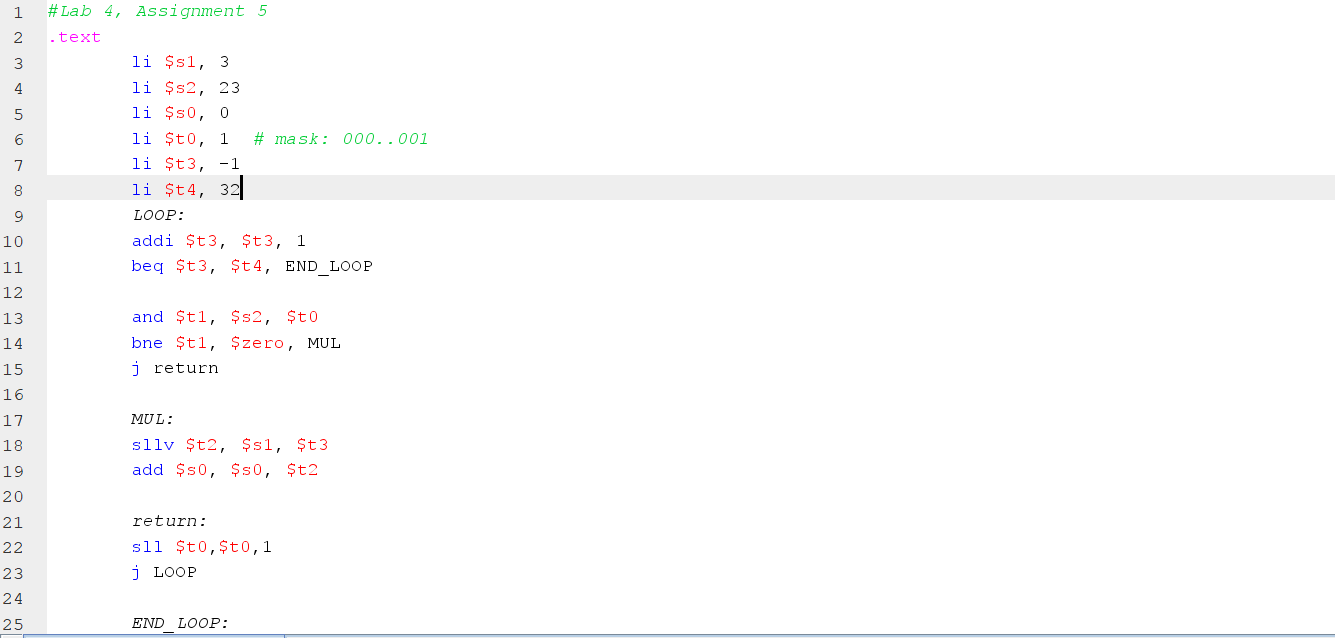
Dòng 11: t2 = s0 xor s1. Nếu t2 < 0 nghĩa là s0 trái dấu với s1 tức là xảy ra tràn => nhảy tới overflow. Ngược lại thì thực hiện dòng 12 jum tới exit, kết thúc với trạng thái không tràn.

Kết quả:



Assignment 5:

Code:



Giải thích:

S0 = s1 + s2  
 Dòng 11: Kiểm tra xem đã duyệt được hết 32 bit chưa, nếu đã duyệt được hết thì rẽ nhánh đến END\_LOOP để kết thúc vòng lặp.

Dòng 14: Nếu bit trên thanh s2 bằng 1 tức là kết quả and ở dòng 13 bằng 1 thì thực hiện nhân còn không thì thực hiện tiếp dòng 15 để quay lại vòng lặp.

Dòng 18: thực hiện dịch trái s1 theo giá trị bởi thanh ghi t3 là vị trí mà bit của mask bằng 1.

Kết quả:

