**Laboratory Exercise 4**

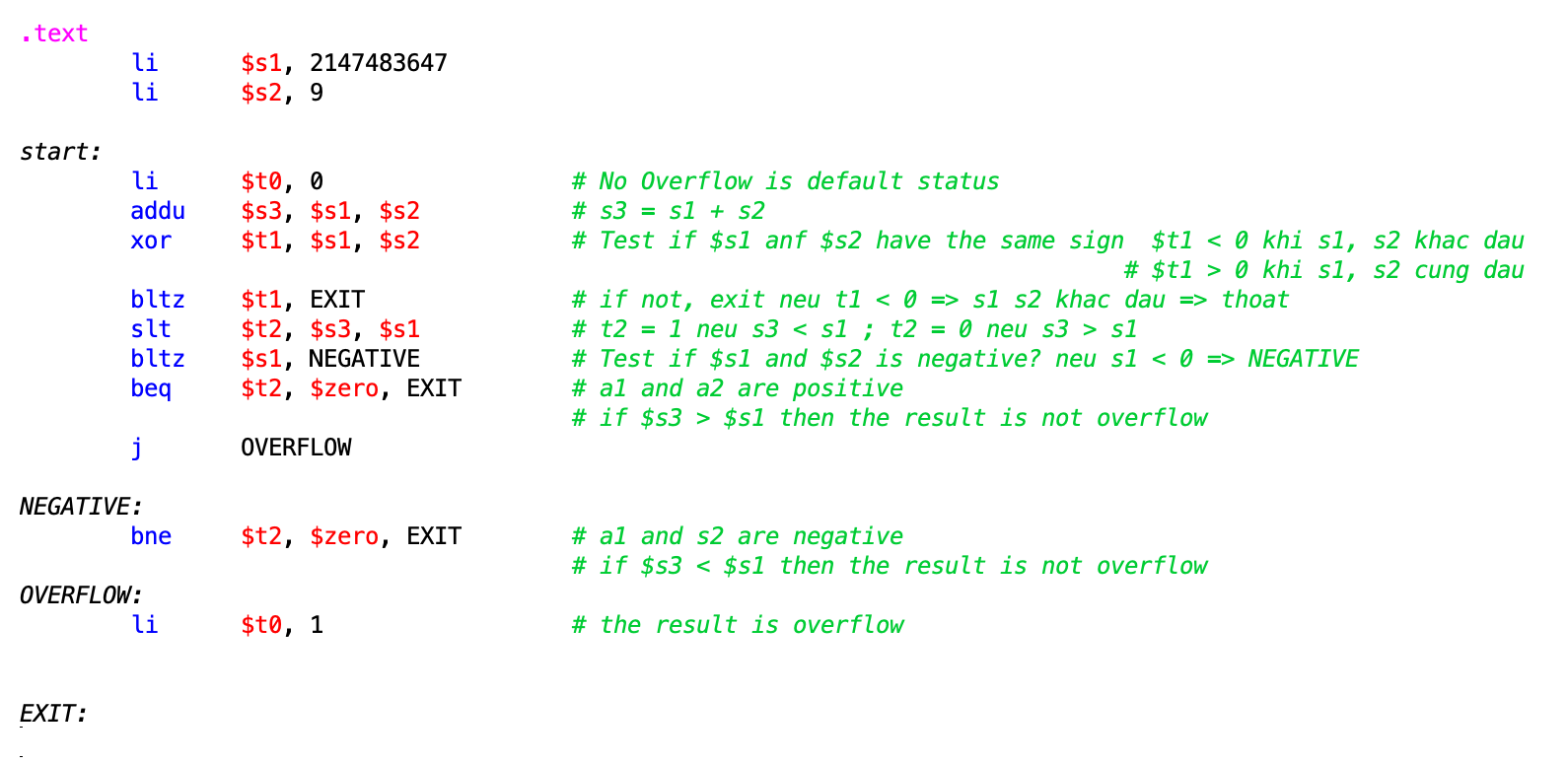
**Arithmetic and Logical operation**

Sinh viên : Hoàng Minh Nguyệt

MSSV: 20176839

Assignment 1

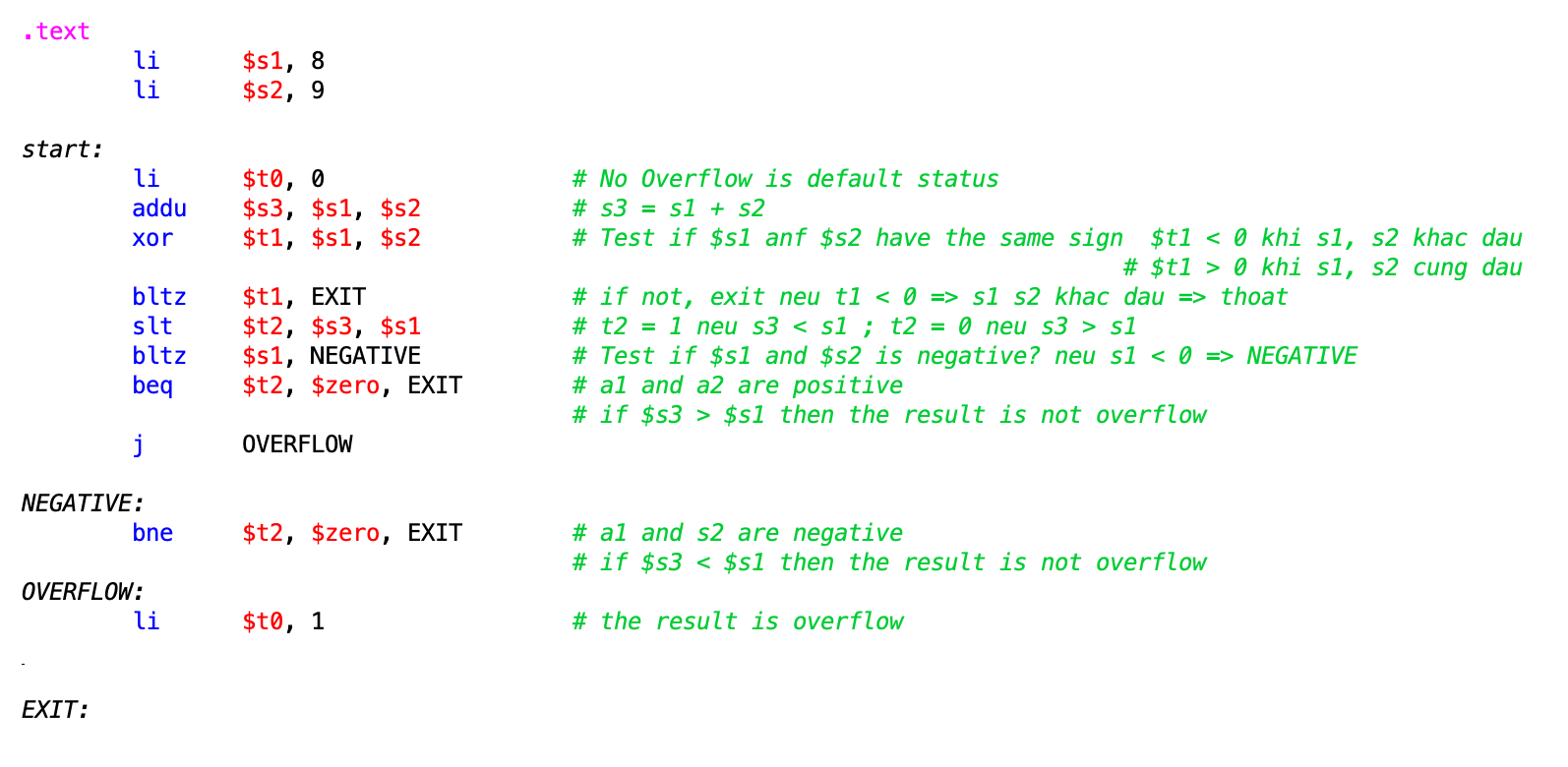
TH1: Khởi tạo chương trình gán s1 và s2 dương , trường hợp xảy ra Overflow



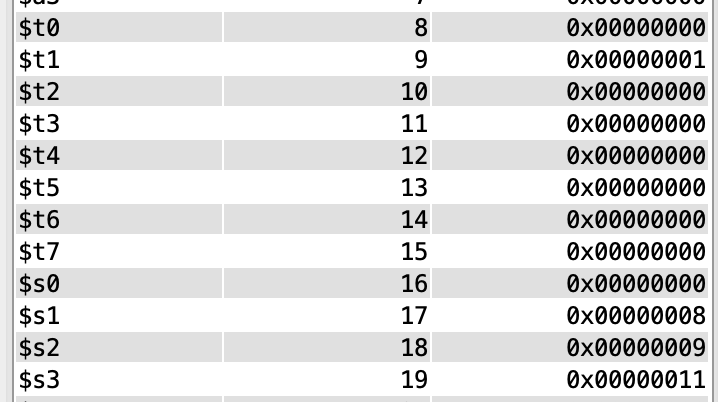
Kết quả hiển thị sau từng lệnh chạy  

Ta thấy, kết quả cuối cùng cho ra t0 = 1 => Overflow

TH2: Khởi tạo chương trình gán s1 và s2 dương , trường hợp không xảy ra Overflow

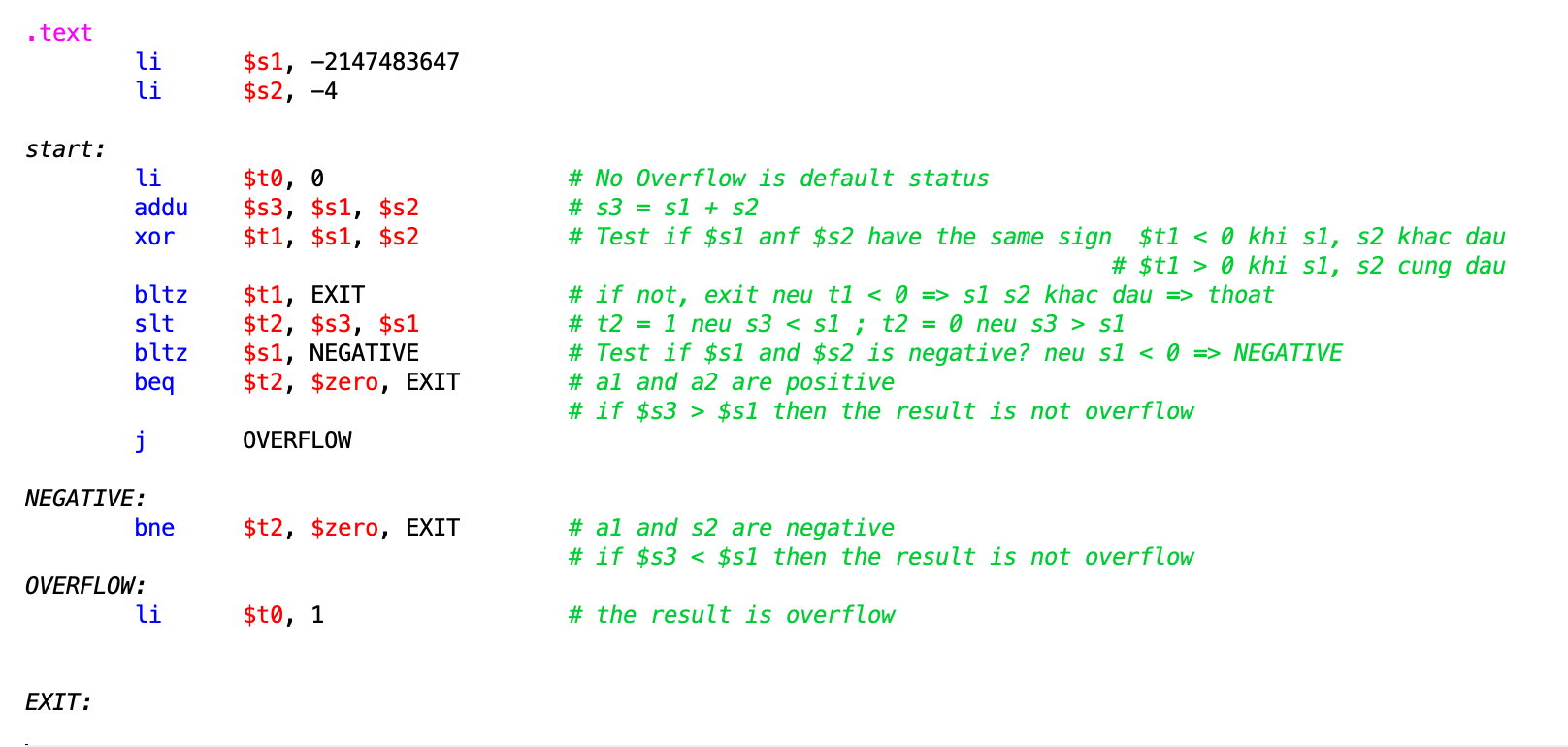


Kết quả chương trình

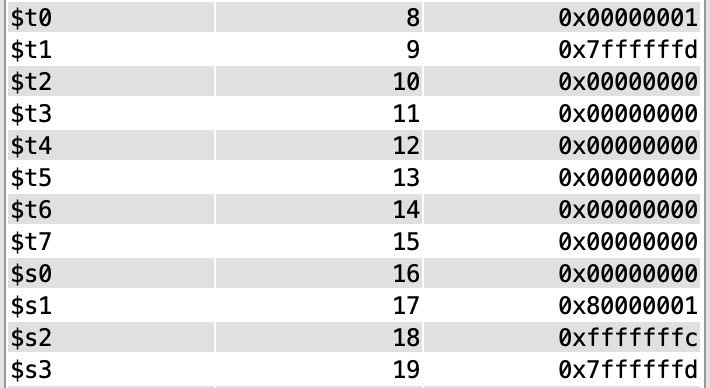


Sau khi chương trình chạy xong ta được giá trị t0= 0 => Không xảy xa overflow

TH3: Khởi tạo chương trình gán s1 và s2 âm , trường hợp xảy ra Overflow

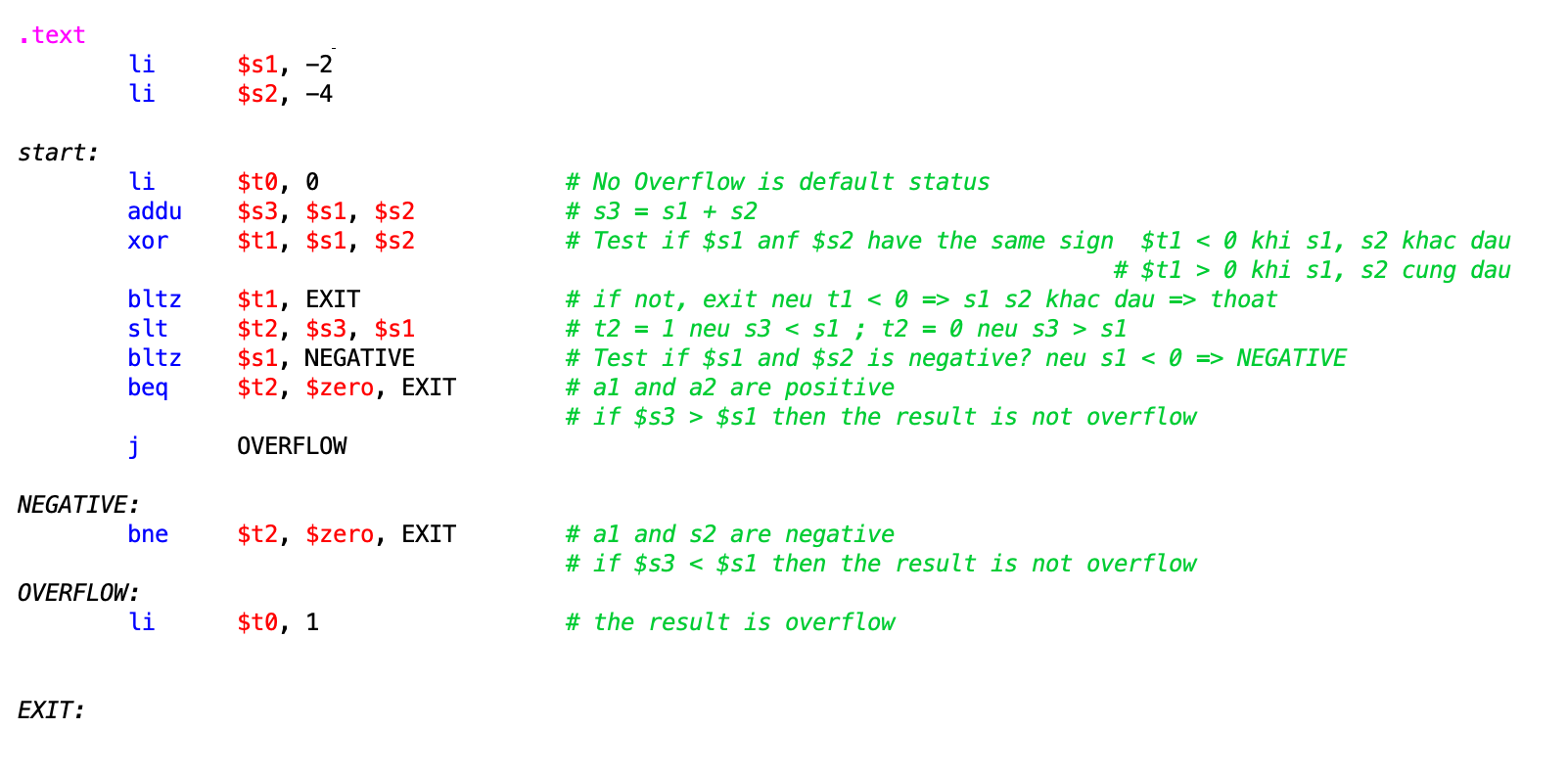


Kết quả chương trình

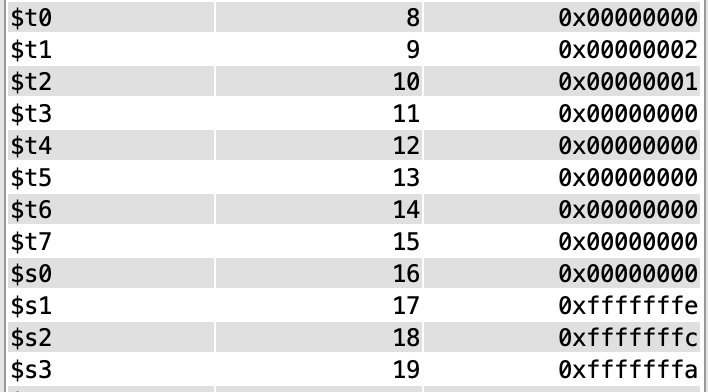


Sau khi chương trình chạy xong ta được giá trị t0= 1 => Xảy xa overflow

TH4: Khởi tạo chương trình gán s1 và s2 âm , trường hợp không xảy ra Overflow

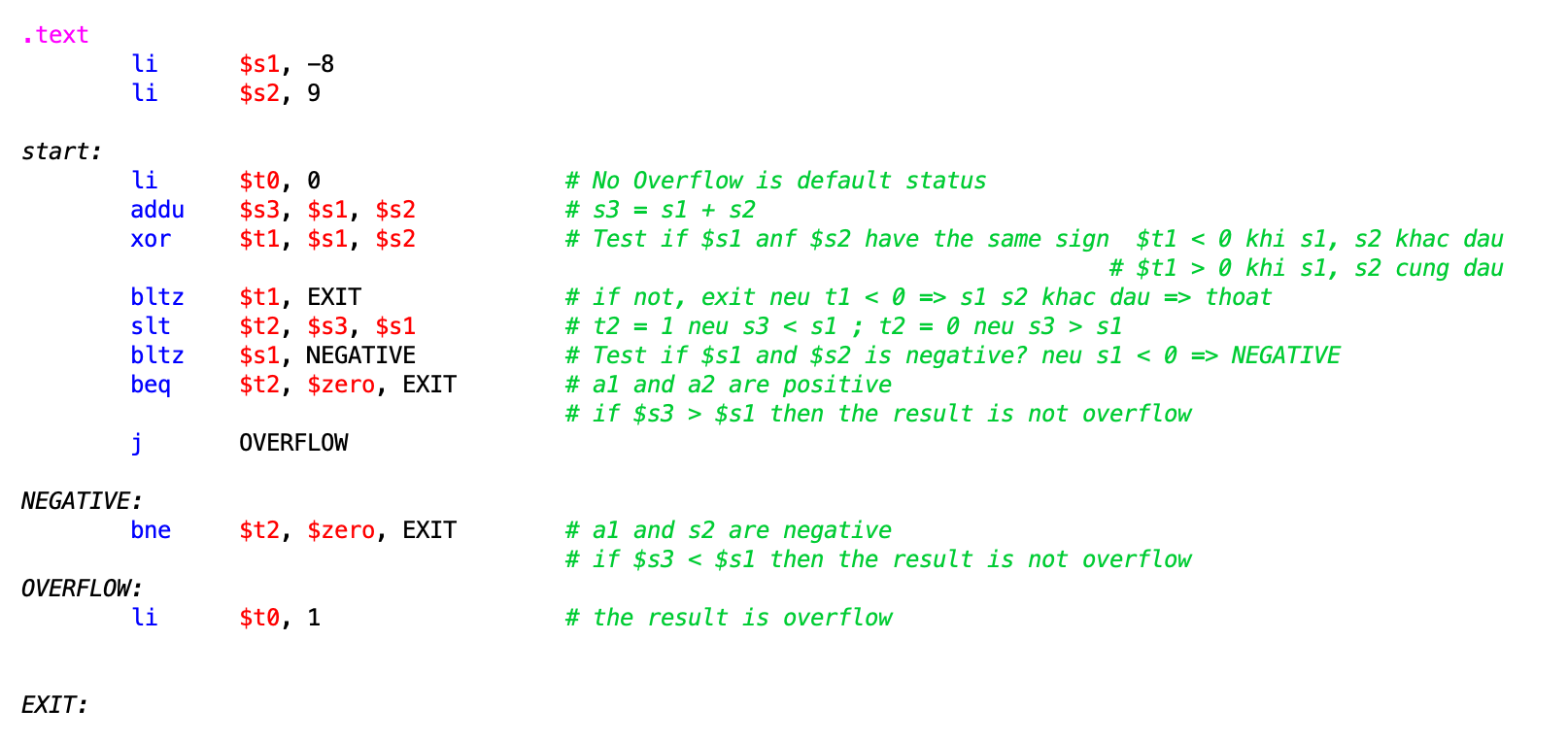


Kết quả chương trình

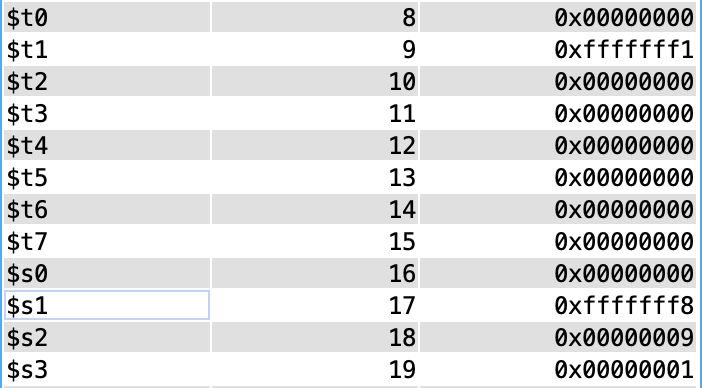


Sau khi chương trình chạy xong ta được giá trị t0= 0 => Không xảy xa overflow

TH5: Khởi tạo chương trình gán s1 và s2 trái dấu



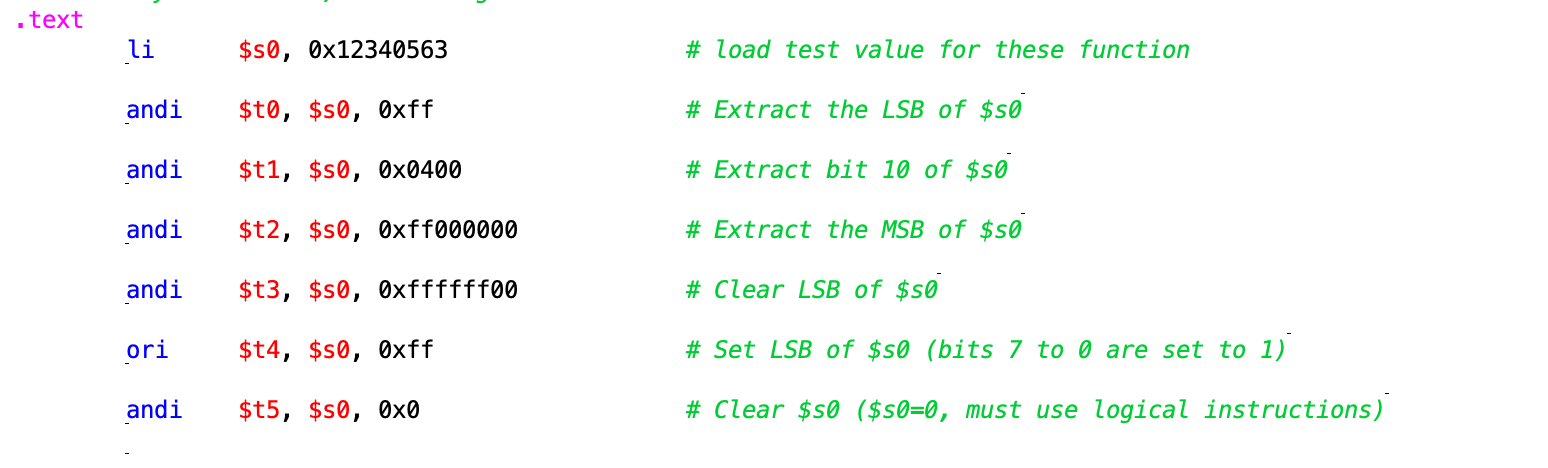
Kết quả chương trình



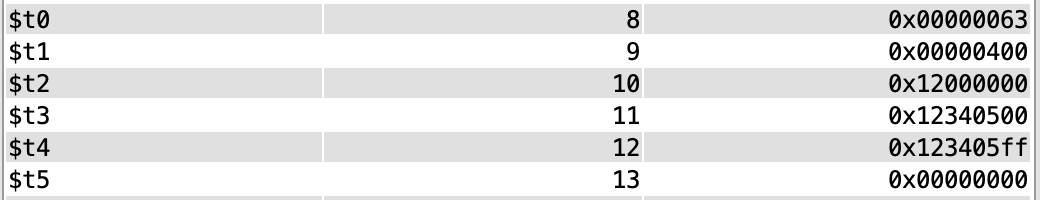
Sau khi chương trình chạy xong ta được giá trị t0= 0 => Không xảy xa overflow

Assignment 2

Chương trình

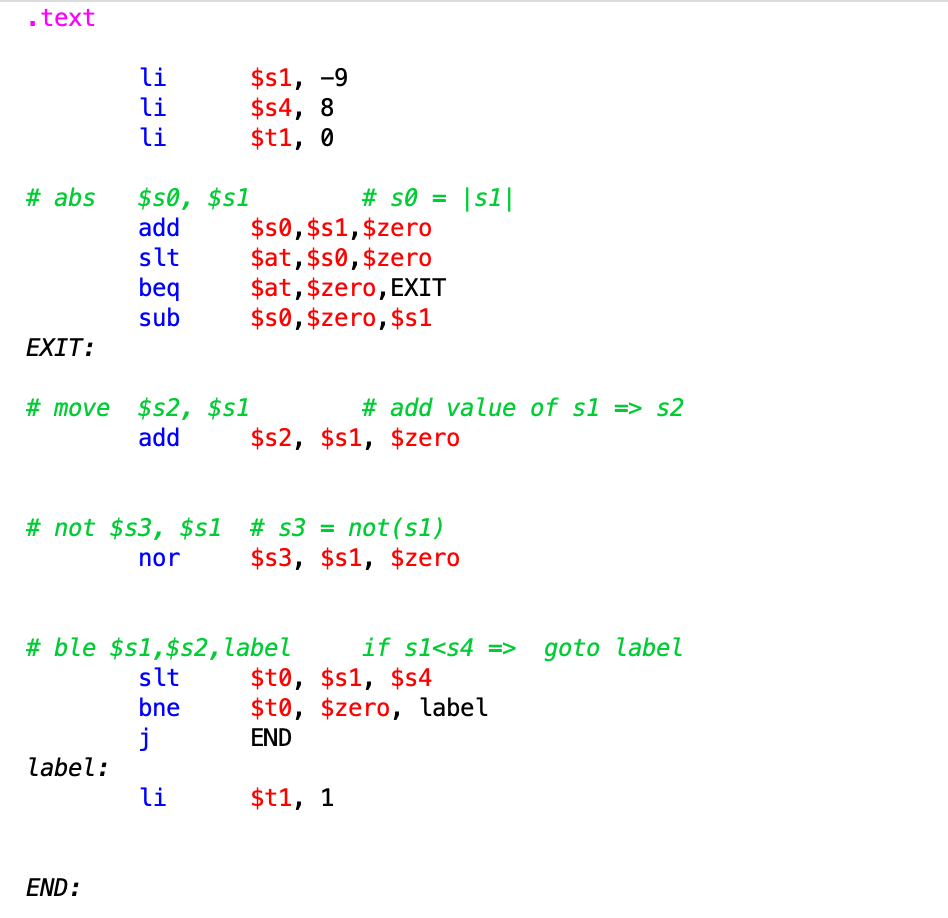


Kết quả chạy các giá trị từ t0 đến t5



Assignment 3

Chương trình



Kết quả chạy từng lệnh của chương trình



Ta thấy s0 = |s1| = 9

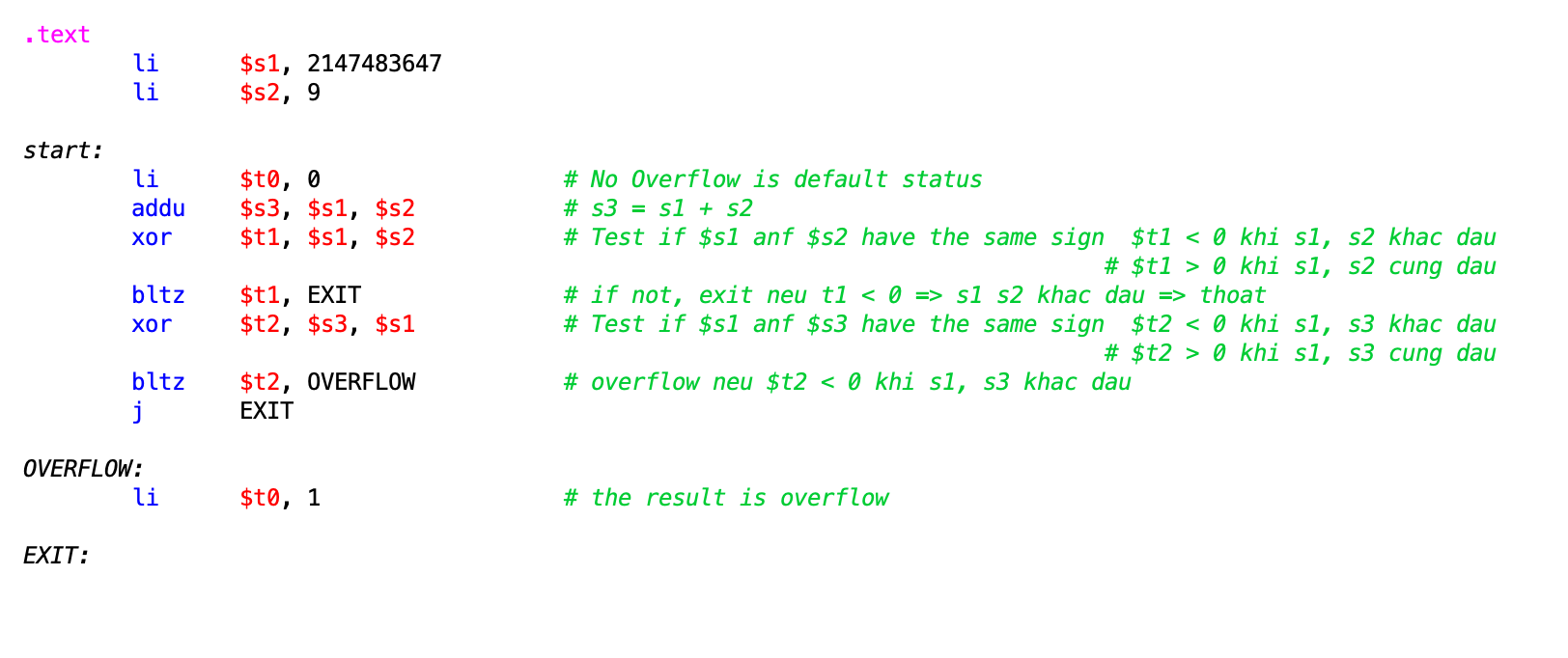
s2 lưu giá trị của s1 = -9

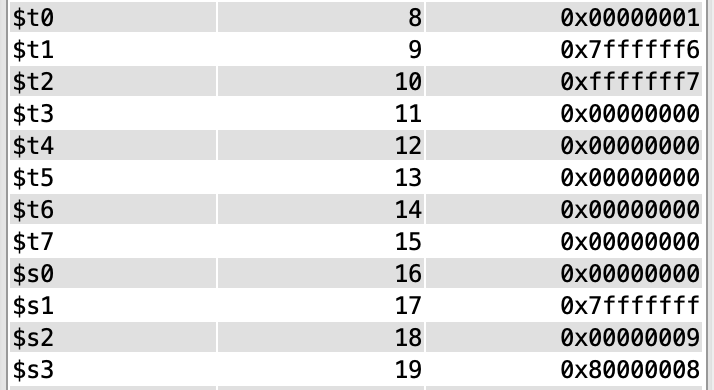
s3 = not (s1) = 8

lệnh so sánh s1 với s4, do s1 = -9 < s4 =8 => t1 được gán giá trị 1

Assignment 4

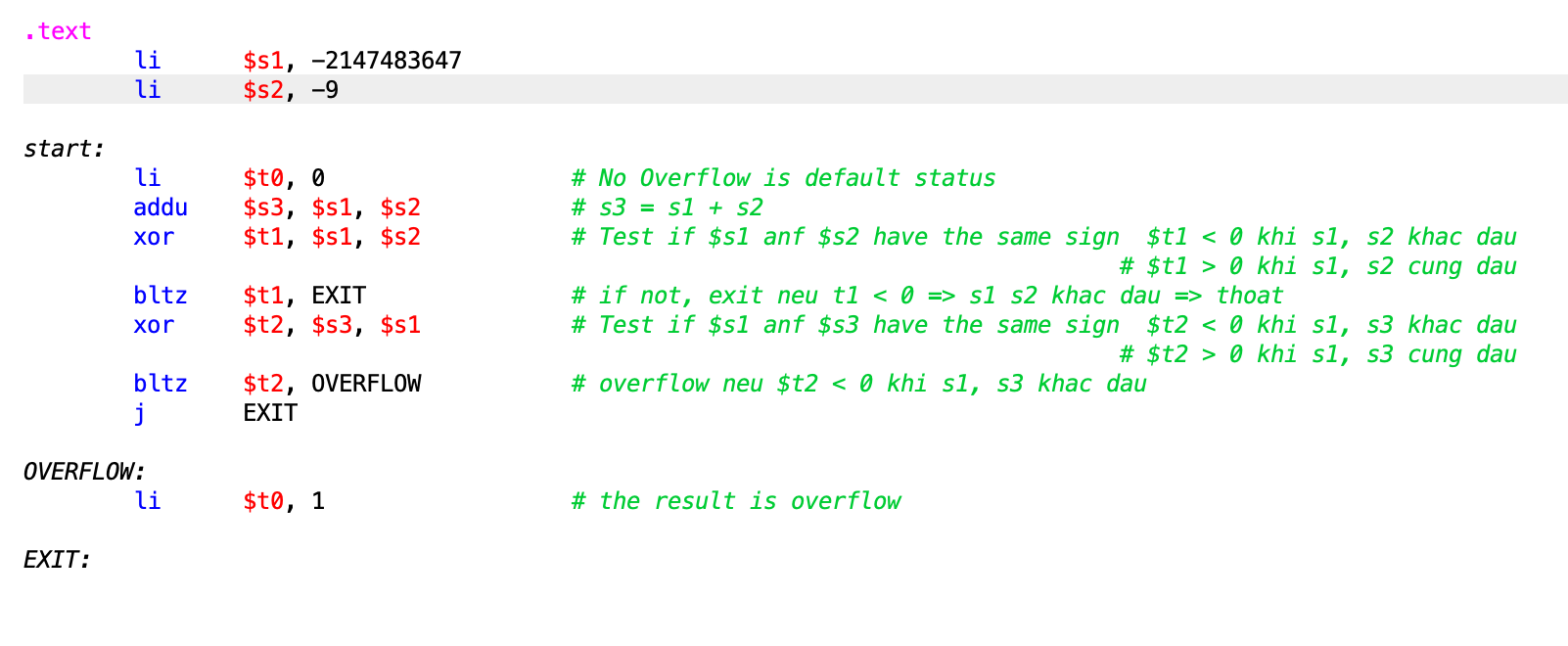
TH1: Khởi tạo chương trình gán s1 và s2 dương , trường hợp xảy ra OVERFLOW

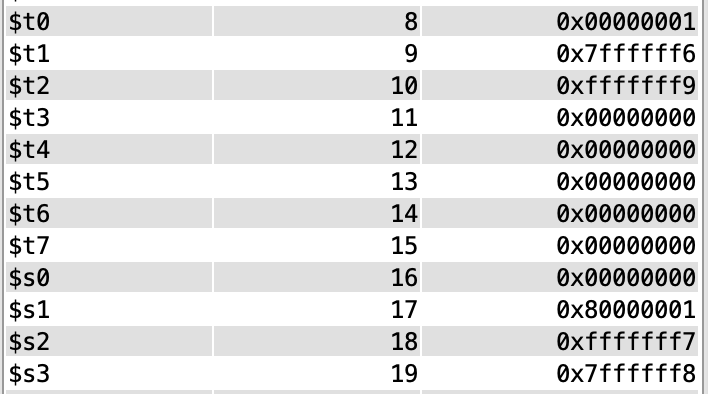
Kết quả chạy



Sau khi chương trình chạy xong ta được giá trị t0= 1 => Xảy xa OVERFLOW

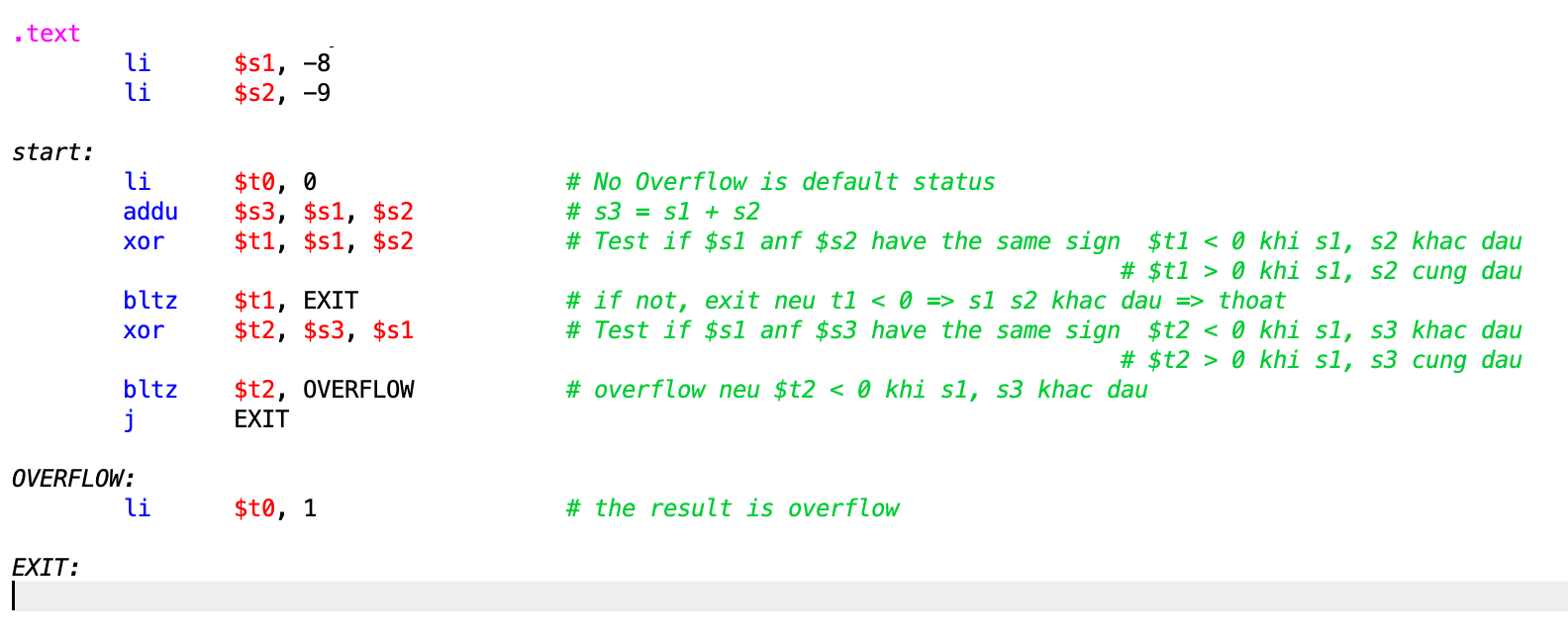
TH2: Khởi tạo chương trình gán s1 và s2 âm , trường hợp xảy ra OVERFLOW

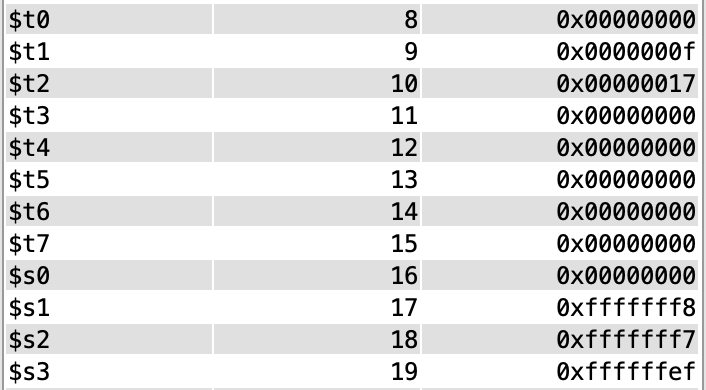
Kết quả chạy



Sau khi chương trình chạy xong ta được giá trị t0= 1 => Xảy xa OVERFLOW

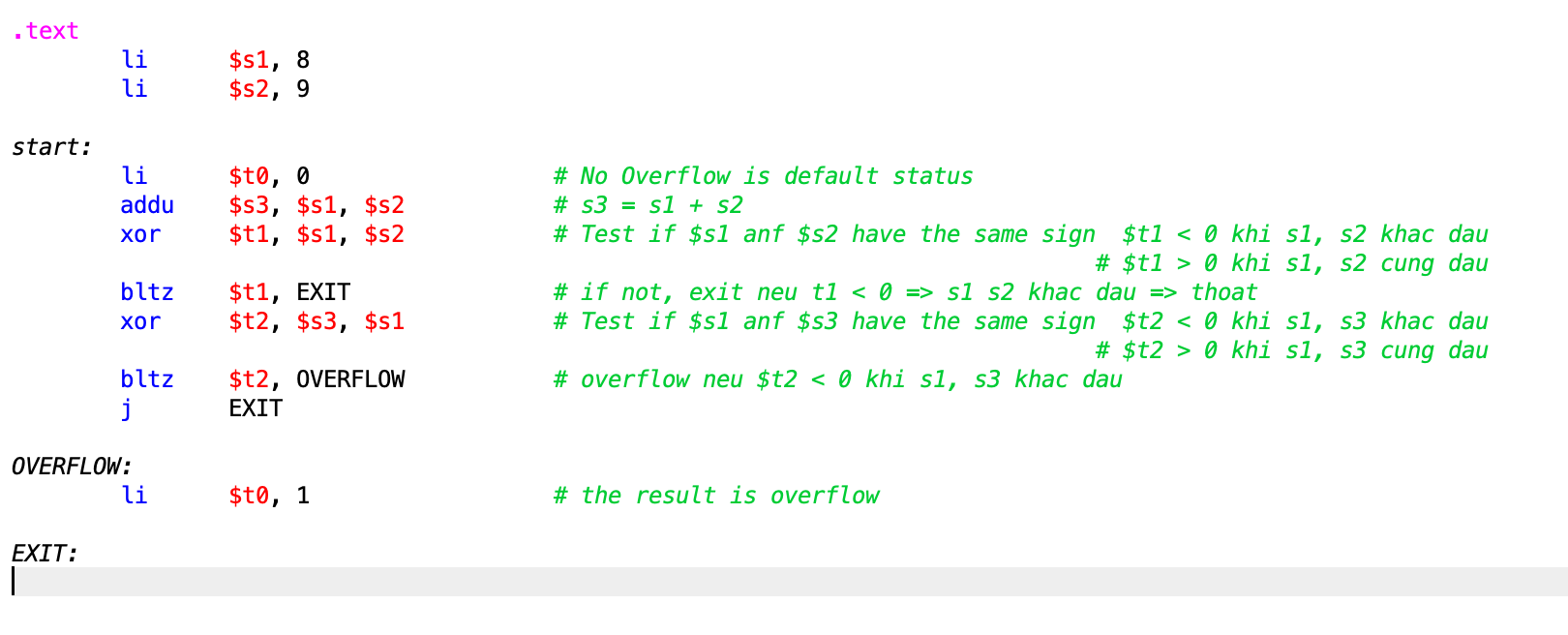
TH3: Khởi tạo chương trình gán s1 và s2 âm , trường hợp không xảy ra OVERFLOW

Kết quả chạy

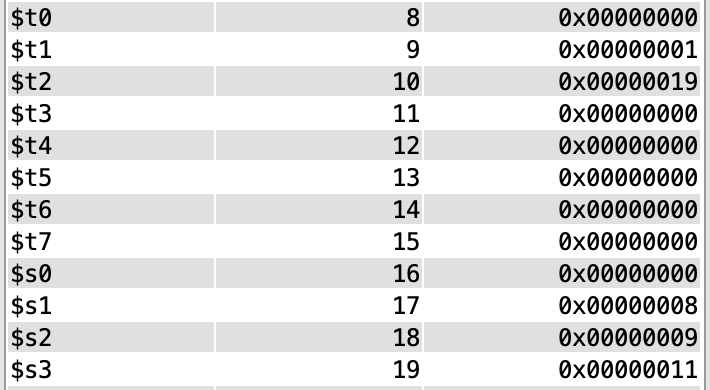


Sau khi chương trình chạy xong ta được giá trị t0= 0 => Không xảy xa OVERFLOW

TH4: Khởi tạo chương trình gán s1 và s2 dương , trường hợp không xảy ra OVERFLOW

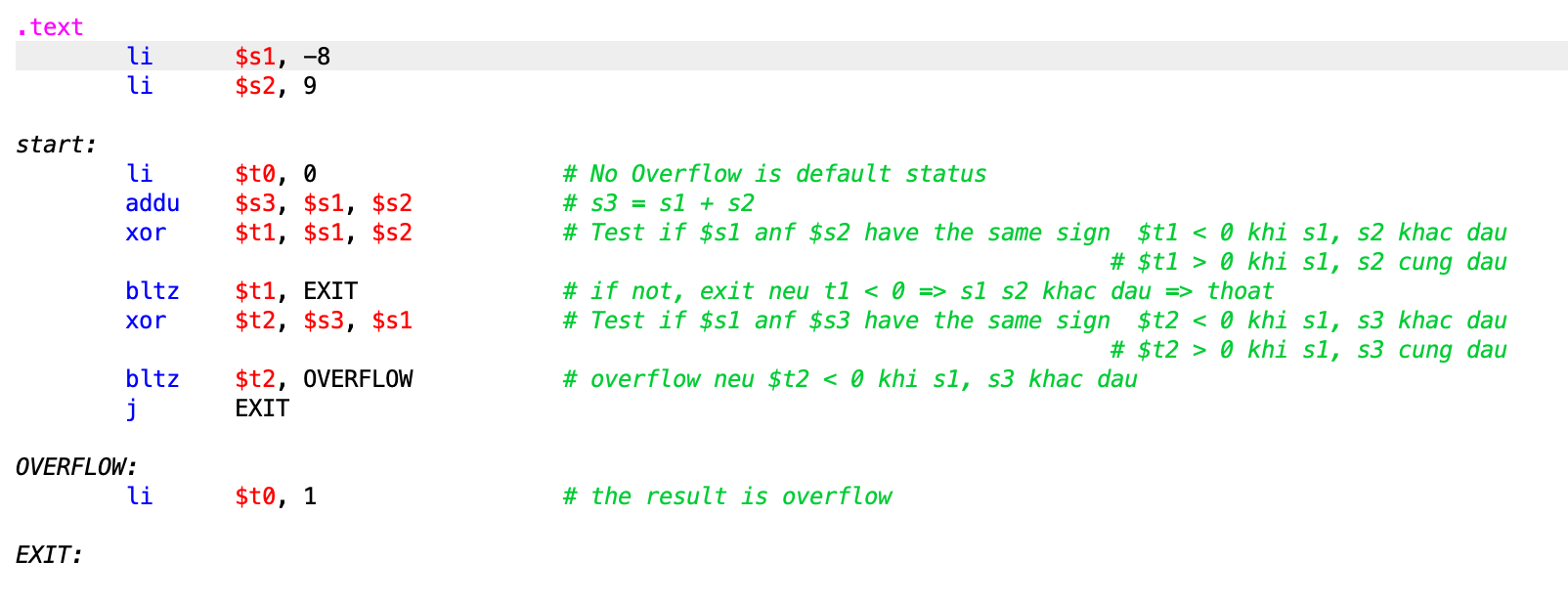


Kết quả chạy

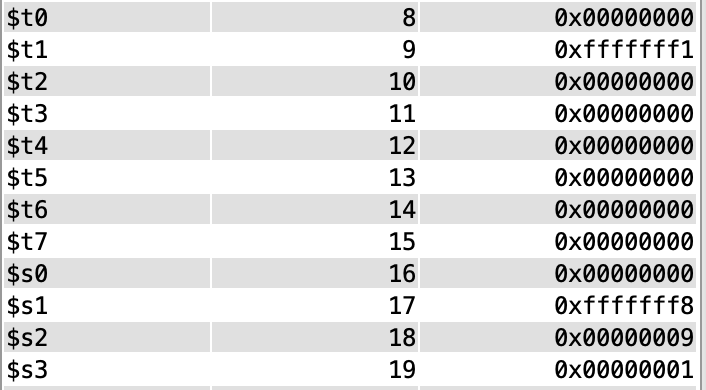


Sau khi chương trình chạy xong ta được giá trị t0= 0 => Không xảy xa OVERFLOW

TH5: Khởi tạo chương trình gán s1 và s2 trái dấu



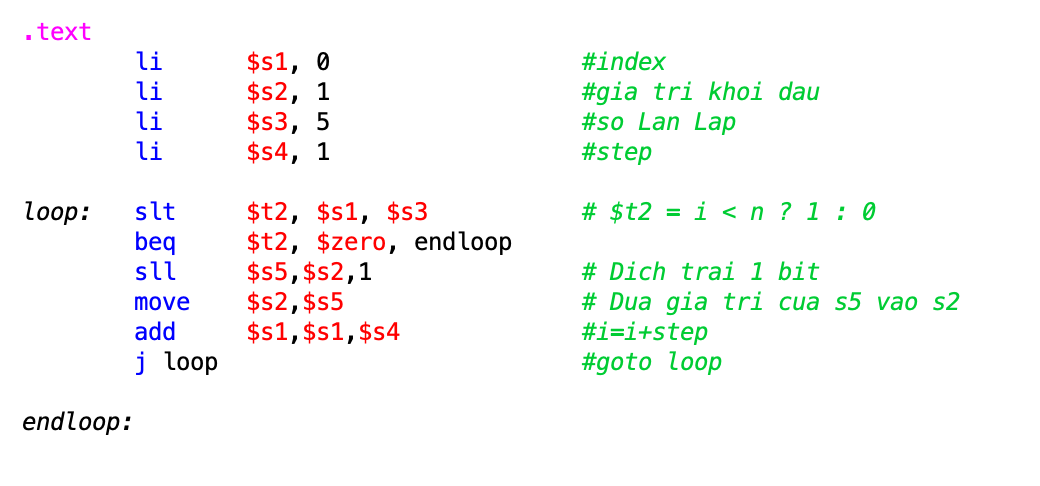
Kết quả chạy



Sau khi chương trình chạy xong ta được giá trị t0= 0 => Không xảy xa OVERFLOW

Assignment 5

Chương trình



Sự thay đổi của thanh ghi s5 sau mỗi lần lặp



Mỗi lần dịch trái 1 bit t có kết quả

2^1 =2

2^2 =4

2^3 =8

2^4 =10(16) = 16(10)

2^5 =20(16) = 32(10)

* Chương trình chạy đúng

**Conclusions**

* SLL Shift Left Logical: [sll $t1,$t2,5 ] Dịch sang trái giá trị của thanh ghi $t2 theo số bit của Immediate rồi lưu vào thanh ghi $t1
* SLLV Shift Left Logical Variable:[ sllv $t1,$t2,$t3 ] Dịch sang trái giá trị của thanh ghi $t2 theo số bit (giá trị của 5 bit thấp trong thanh ghi $t3 )rồi lưu vào thanh ghi $t1
* SRL Shift Left Logical: [sll $t1,$t2,5 ] Dịch sang phải giá trị của thanh ghi $t2 theo số bit của Immediate rồi lưu vào thanh ghi $t1
* SRLV Shift Left Logical Variable:[ sllv $t1,$t2,$t3 ] Dịch sang phải giá trị của thanh ghi $t2 theo số bit (giá trị của 5 bit thấp trong thanh ghi $t3 )rồi lưu vào thanh ghi $t1