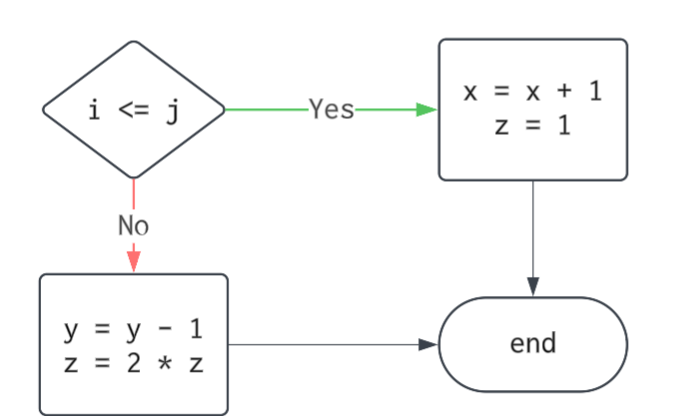
BÁO CÁO THỰC HÀNH KIẾN TRÚC MÁY TÍNH TUẦN 3

Họ và tên: Nguyễn Quý Đức MSSV: 20235682

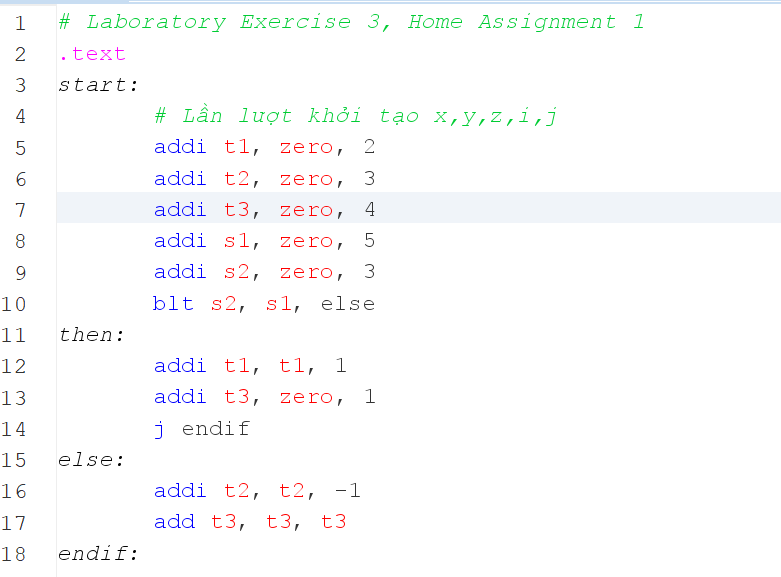
Mã HP: IT3280 Mã lớp: 156788

1. Assignment 1

- Lưu đồ:



- Nhập lệnh:



- Thực hiện từng dòng lệnh

+ Lần lượt khởi tạo giá trị x,y,z,i,j:



+ Do lựa chọn i=5 và j=3 nên (i<=j) return false, chương trình nhảy đến else và thực hiện các lệnh trong nhánh else, thay vì then.

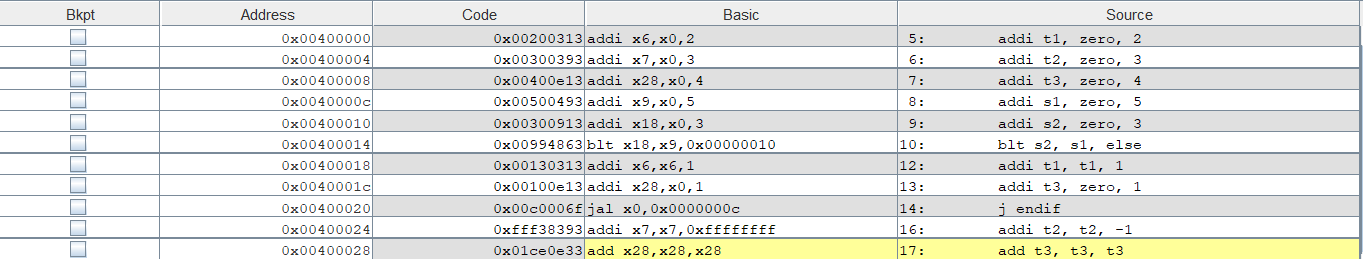


+ Thanh ghi pc luôn nhảy thêm 4 sau mỗi lệnh



(do i<=j return false nên chương trình nhảy đến nhảy đến nhãn else có địa chỉ 0x00400024)



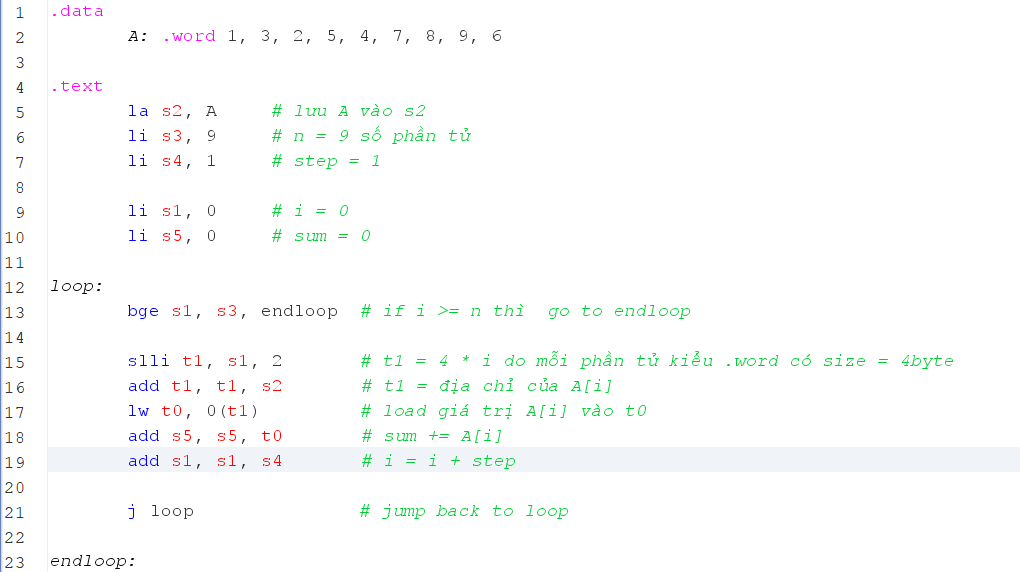


Tóm lại:

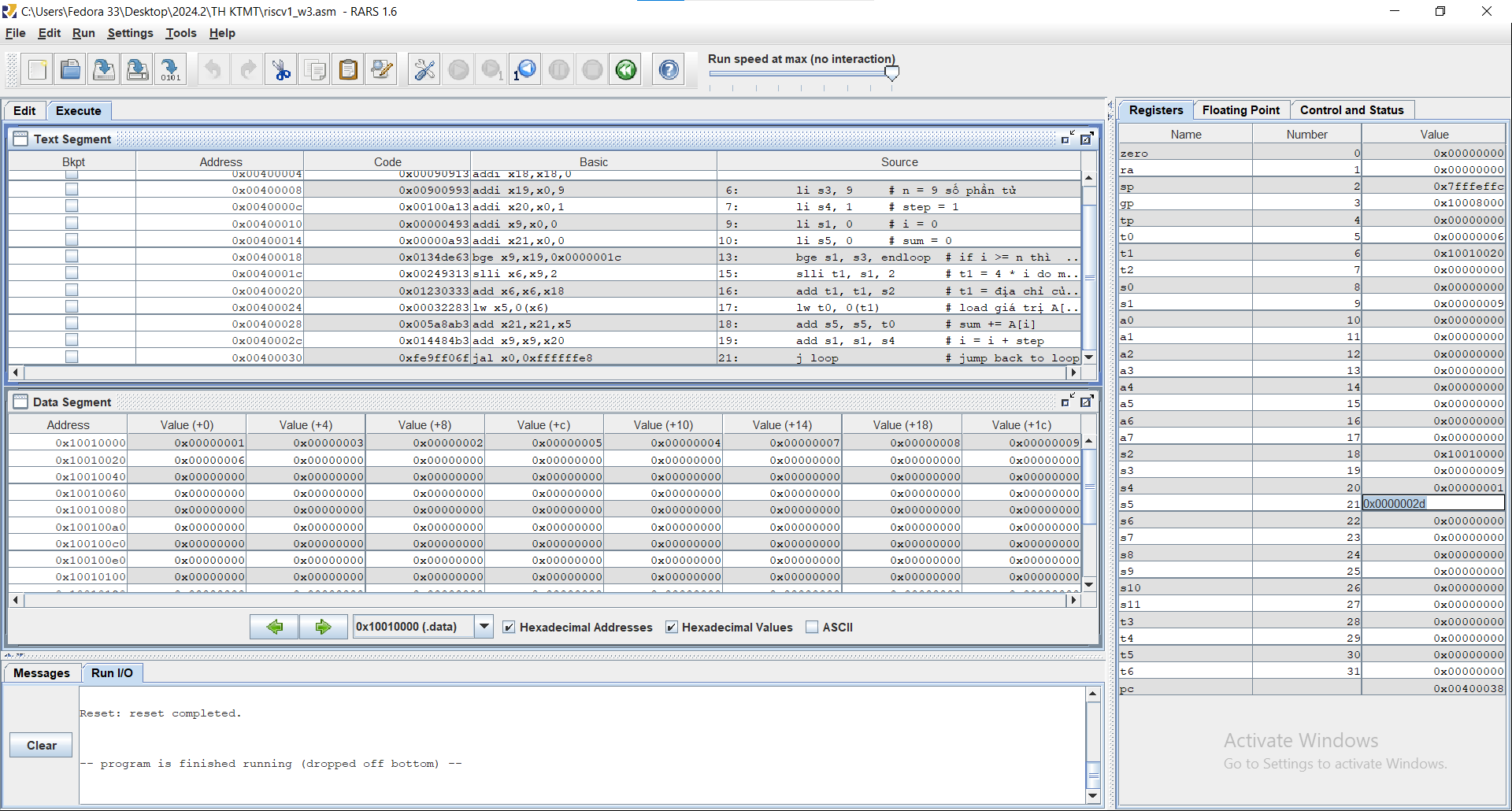
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Trạng thái | T1 | T2 | T3 | S1 | S2 |
| Ban đầu | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| Sau khi khởi tao | 0x00000002 | 0x00000003 | 0x00000004 | 0x00000005 | 0x00000003 |
| Sau lệnh blt | - | - | - | - | - |
| Sau lệnh endif | - | 0x00000002 | 0x00000008 | - | - |

Do (i<=j) return false (i=5, j=3) nên ysau=3-1=2 và zsau =4\*2=8 (thỏa mãn với (x,y,z) ban đầu là (2,3,4))

1. Assignment 2



- Nhập lệnh: (ở trang trước)

- Trạng thái các thanh ghi:

- Giải thích mã:

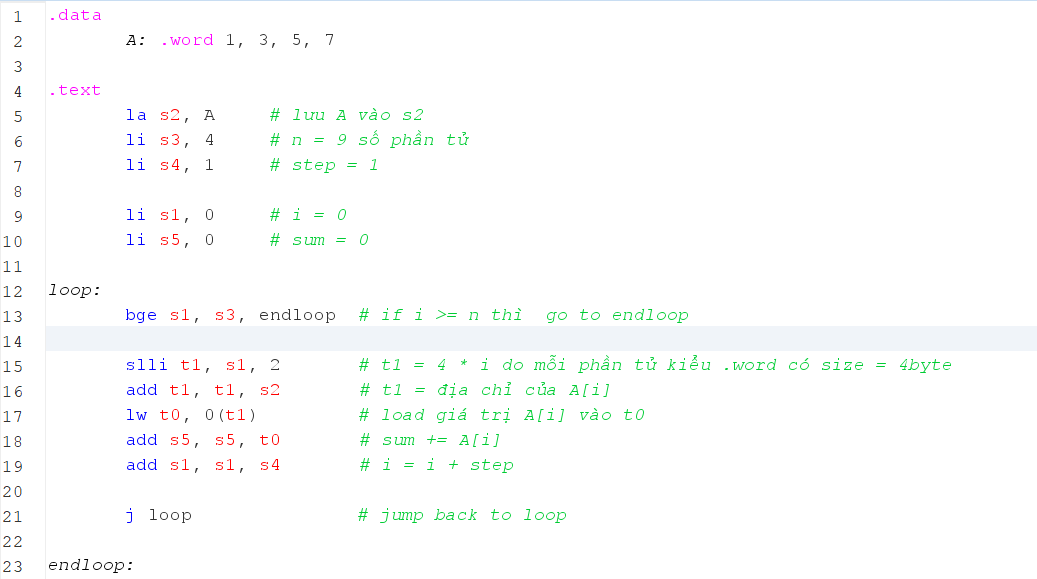
* Khởi tạo: s2 = địa chỉ A, s1 = 0, s3 = 9 (số phần tử), s4 = 1 (step), s5 = 0 (sum).
* Vòng lặp

+ Kiểm tra i >= n, nếu đúng thì kết thúc.  
+ Tính địa chỉ A[i] bằng slli t1, s1, 2, rồi lw t0, 0(t1).  
+ Cộng A[i] vào s5, tăng i.  
+ Kết thúc.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Trạng thái | S1 | T0 | T1 | S5 |
| Ban đầu | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| Loop 1 | 0x00000001 | 0x00000001 | 0x10010000 | 0x00000001 |
| Loop 2 | 0x00000002 | 0x00000003 | 0x10010004 | 0x00000004 |
| Loop 3 | 0x00000003 | 0x00000002 | 0x10010008 | 0x00000006 |
| Loop 4 | 0x00000004 | 0x00000005 | 0x1001000c | 0x0000000b |
| Loop 5 | 0x00000005 | 0x00000004 | 0x10010010 | 0x0000000f |
| Loop 6 | 0x00000006 | 0x00000007 | 0x10010014 | 0x00000016 |
| Loop 7 | 0x00000007 | 0x00000008 | 0x10010018 | 0x0000001e |
| Loop 8 | 0x00000008 | 0x00000009 | 0x1001001c | 0x00000027 |
| Loop 9 | 0x00000009 | 0x00000006 | 0x10010020 | 0x0000002d |
| Endloop | - | - | - |  |

Vòng lặp kết thúc khi s1 = 0x09 (>= s3), thu được tổng s5 = 0x2d (= 45 ở hệ thập phân) => thỏa mãn

- Thay bằng bộ giá trị khác: A[] = {1,3,5,7}

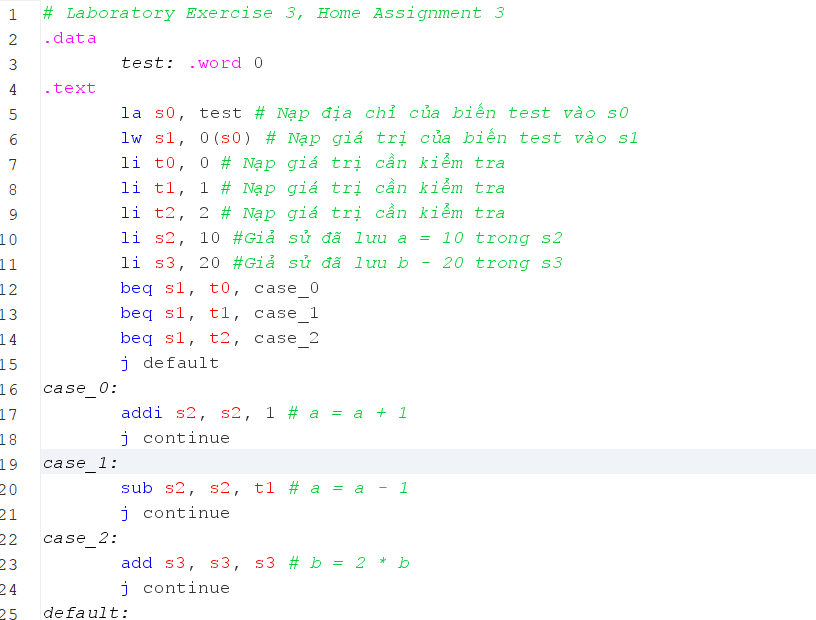




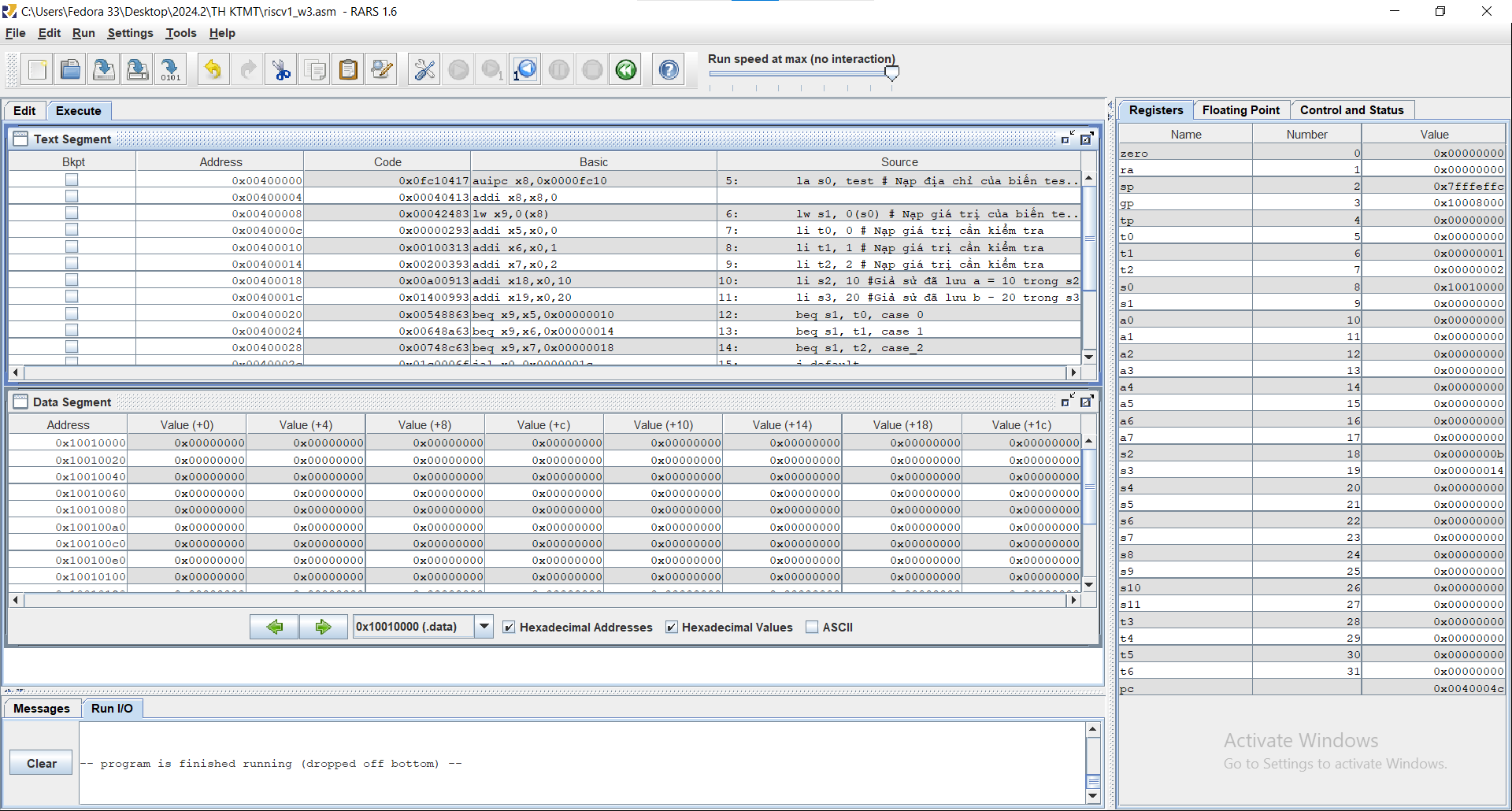
Thu được s5 = 0x10 (= 16 trong hệ thập phân) => thỏa mãn

1. Assignment 3

- Nhập lệnh: (giả sử a = 10 = 0x10 lưu trong s2 và b = 20 = 0x14 lưu trong s3)



- Trạng thái các thanh ghi sau khi chạy toàn bộ lệnh

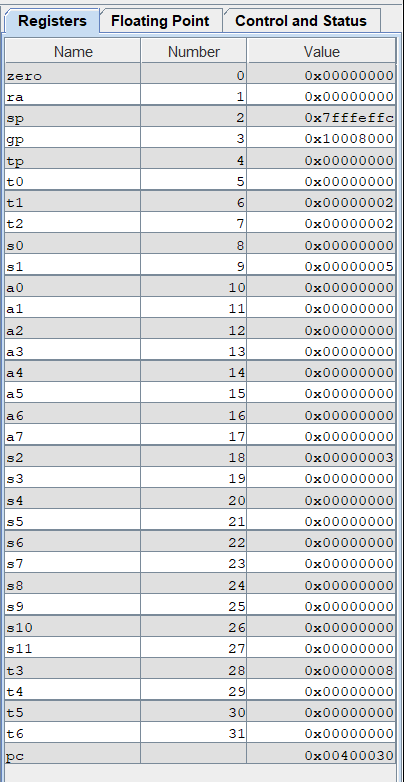
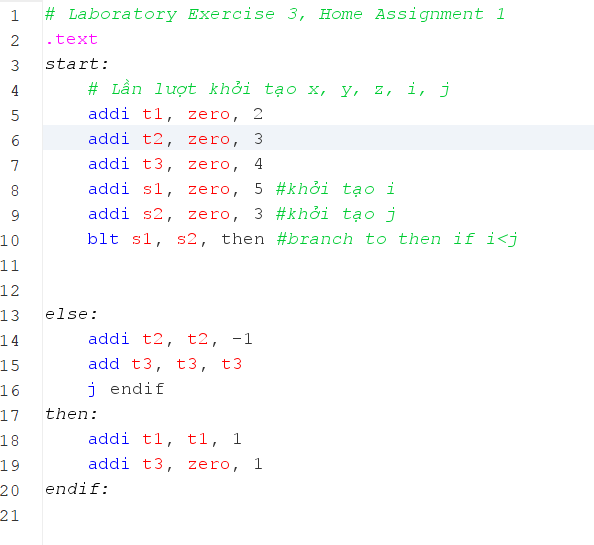


|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Trạng thái | S1 | S2 | S3 | Pc |
| Ban đầu | 0x00000000 | 0x00000000 | 0x00000000 | 0x00400000 |
| Khai báo | 0x00000000 | 0x0000000a | 0x00000014 | 0x00400020 |
| Sau case\_0 | - | 0x0000000b | 0x00000014 | 0x00400030 |
| Sau case\_1 | - | - | - | 0x00400038 |
| Sau case\_2 | - | - | - | 0x00400040 |
| Continue | - | - | - | 0x0040004c |

1. Assignment 4

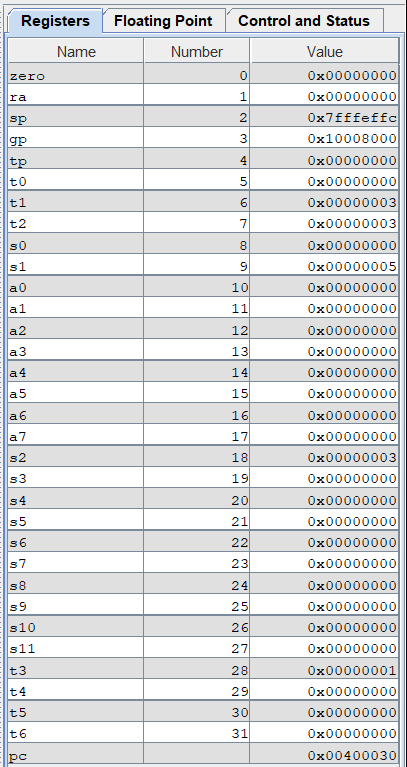
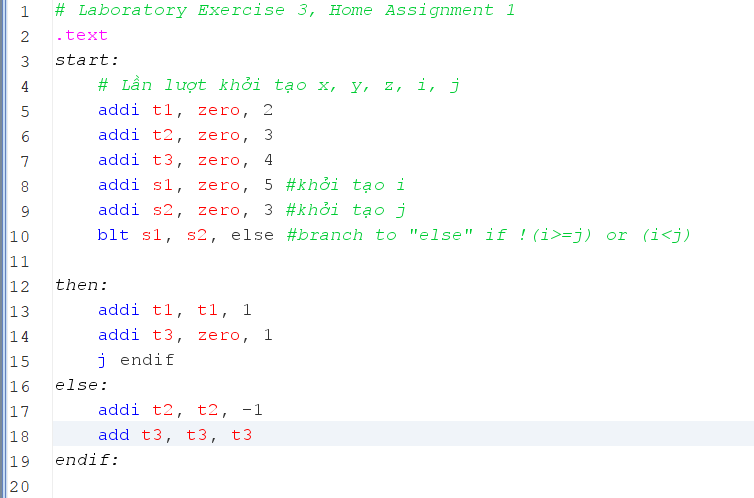
Khởi tạo bộ (x,y,z,i,j) = (2,3,4,5,3)

1. i < j



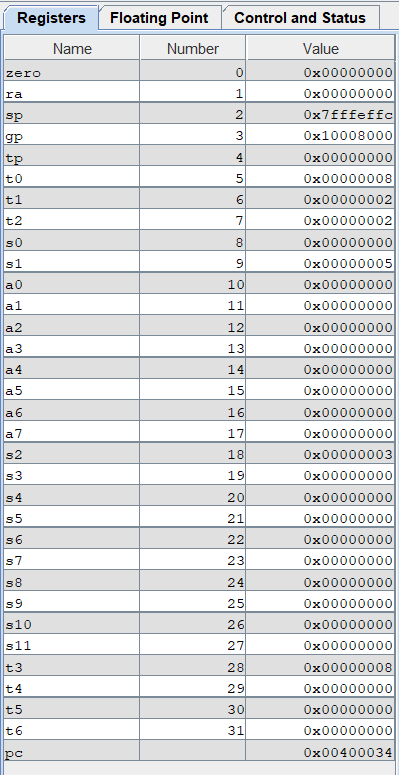
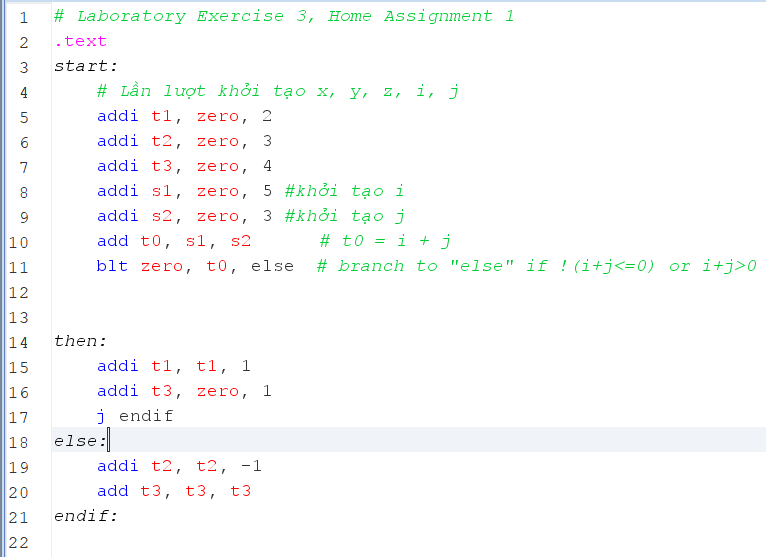
Sau khi thực hiện chương trình (x,y,z)=(2,2,8) do i<j false

1. i >= j



Sau khi thực hiện chương trình (x,y,z)=(3,3,1) do i>=j true

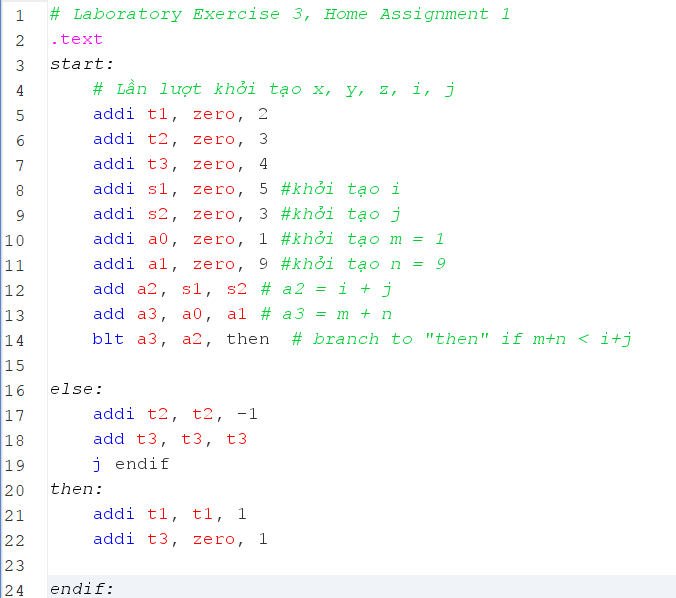
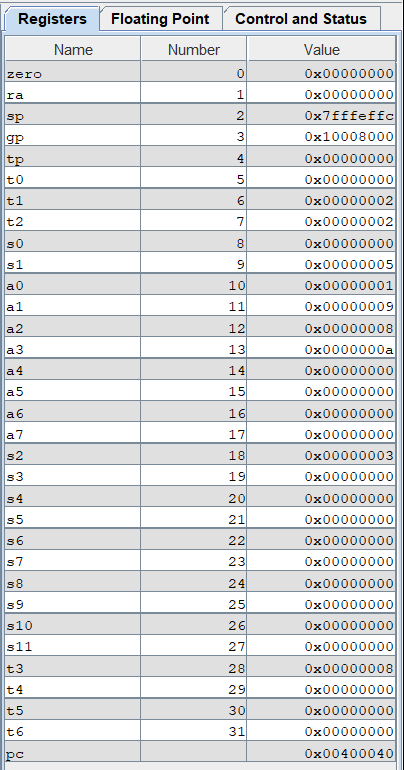
1. i + j <= 0



Sau khi thực hiện chương trình (x,y,z)=(2,2,8) do i+j<=0 false

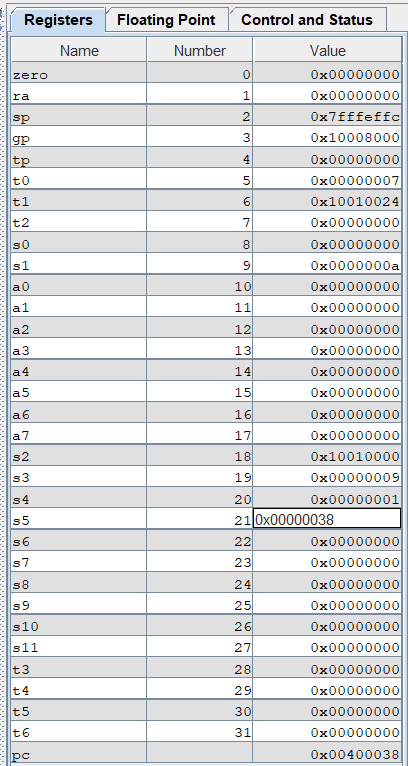
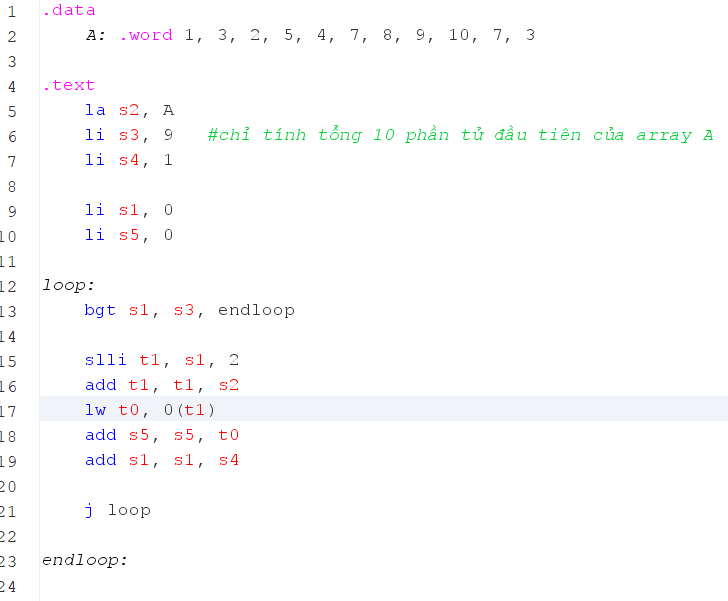
1. i + j > m + n

\*Khởi tạo (m,n)=(1,9)

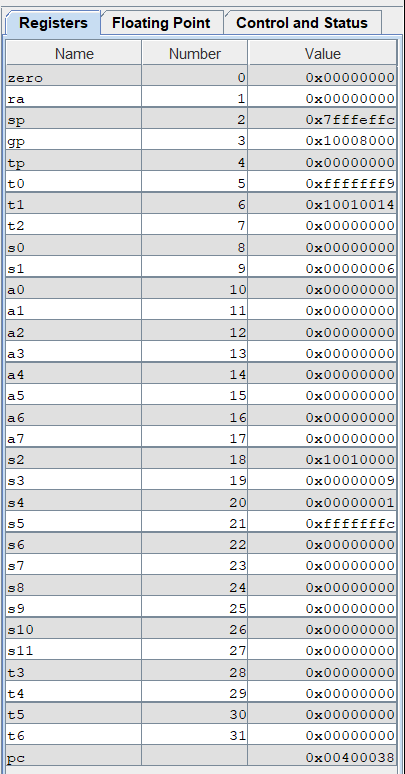
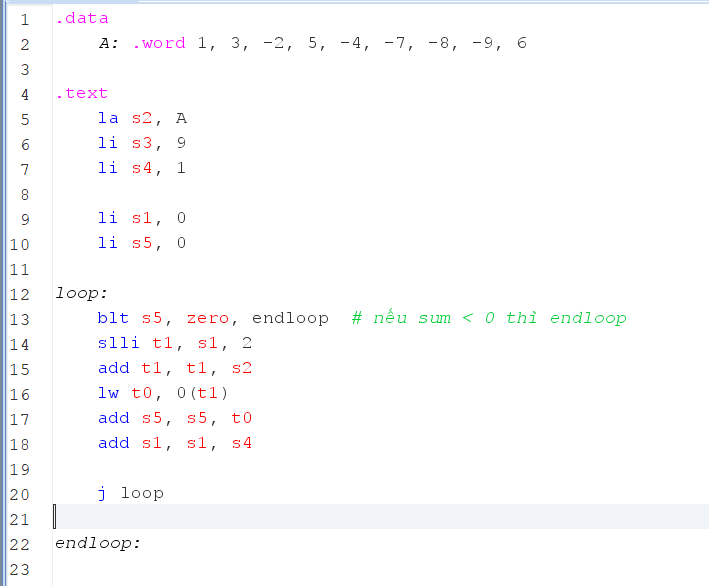
Sau khi thực hiện chương trình (x,y,z)=(2,2,8) do i+j>m+n false

1. Assignment 5
2. i < n



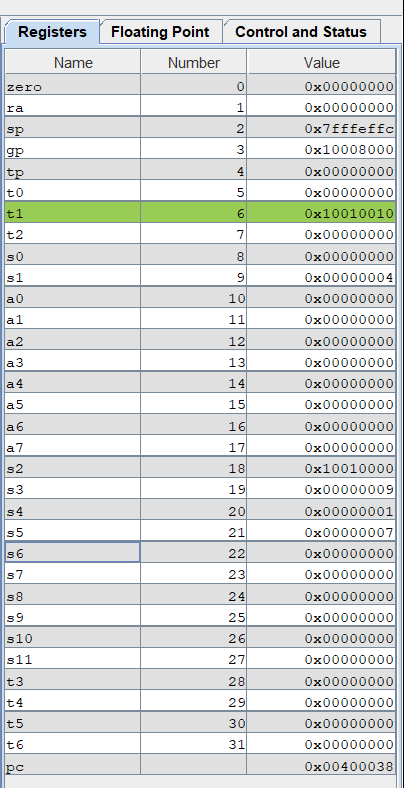
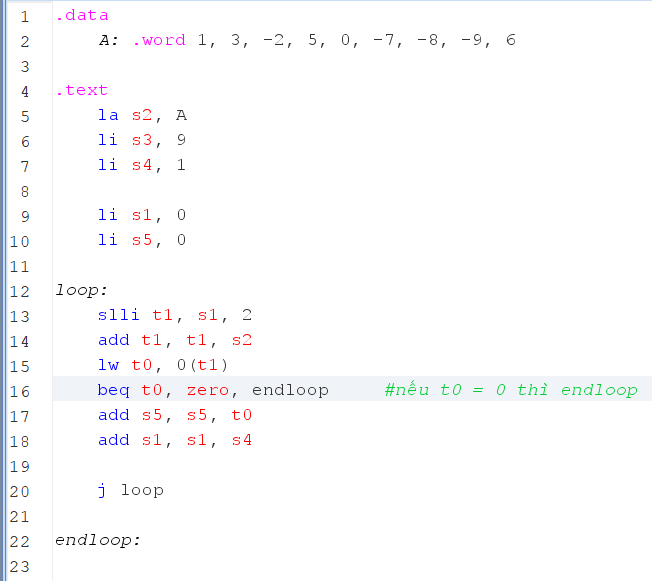
Tổng 10 phần tử đầu của dãy là 56 = 0x38

1. sum < 0



Tổng đến i=6 thì dừng lại do 1 + 3 - 2 + 5 – 4 - 7 = -4 < 0

1. A[i] == 0



Tổng đến A[5] thì dừng lại do A[5]==0,   
thu được sum = 1 + 3 – 2 + 5 = 7

1. Assignment

.data

A: .word -10, 25, -40, 30, -50, 15, 5

n: .word 7

max\_abs: .word 0

.text

la a0, A

lw a1, n

li a2, 0 # index i = 0

li a3, 0 # max\_abs = 0

loop:

bge a2, a1, endloop # If i >= n, exit loop

slli t0, a2, 2 # t0 = i \* 4 (each .word is 4 bytes)

add t0, a0, t0 # t0 = &A[i]

lw t1, 0(t0) # t1 = A[i]

blt t1, zero, rev # if A[i] < 0, take -A[i]

j comp

rev:

sub t1, zero, t1 # abs(A[i])

comp:

bge t1, a3, upd\_max # if abs(A[i]) >= max\_abs, upd

j next

upd\_max:

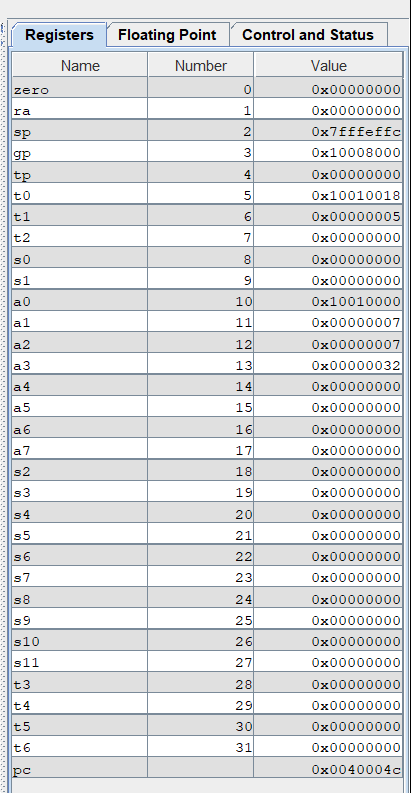
mv a3, t1 # max\_abs = abs(A[i])

next:

addi a2, a2, 1 # i++

j loop

endloop:



Giải thích:

- Khởi tạo dữ liệu: A[], n, max\_abs = 0

- Vòng lặp:

+ Duyệt các phần tử của A

+ Nếu A[i] âm, đổi dấu

+ So sánh với max\_abs, nếu lớn hơn, update value max\_abs = abs(A[i])

+ Tăng index i và lặp

- Endloop: thu được max\_abs thỏa mãn.

\*Trong ví dụ trên, giá trị max\_abs được lưu vào a3 có value = 0x32 = 50 => thỏa mãn