Họ và tên : Lê Ngọc Anh Quân – 20176852

Bài 2 :

Source code :

.data

A: .word 7, -2, 5, 1, 5,6,7,3,6,8,8,59,5

Aend: .word

.text

main:

la $a0,A #$a0 = Address(A[0])

la $a1,Aend

addi $a1,$a1,-4 #$a1 = Address(A[n-1])

j sort #sort

after\_sort: li $v0, 10 #exit

syscall

end\_main:

sort:

beq $a0,$a1,done #single element list is sorted

j max #call the max procedure

after\_max:

lw $t0,0($a1) #load last element into $t0

sw $t0,0($v0) #copy last element to max location

sw $v1,0($a1) #copy max value to last element

addi $a1,$a1,-4 #decrement pointer to last element

j sort #repeat sort for smaller list

done:

j after\_sort

max:

addi $v0,$a0,0 #init max pointer to first element

lw $v1,0($v0) #init max value to first value

addi $t0,$a0,0 #init next pointer to first

loop:

beq $t0,$a1,ret #if next=last, return

addi $t0,$t0,4 #advance to next element

lw $t1,0($t0) #load next element into $t1

slt $t2,$t1,$v1 #(next)<(max) ?

bne $t2,$zero,loop #if (next)<(max), repeat

addi $v0,$t0,0 #next element is new max element

addi $v1,$t1,0 #next value is new max value

j loop #change completed; now repeat

ret:

j after\_max

Nhận xét :

main:

la $a0,A #$a0 = Address(A[0])

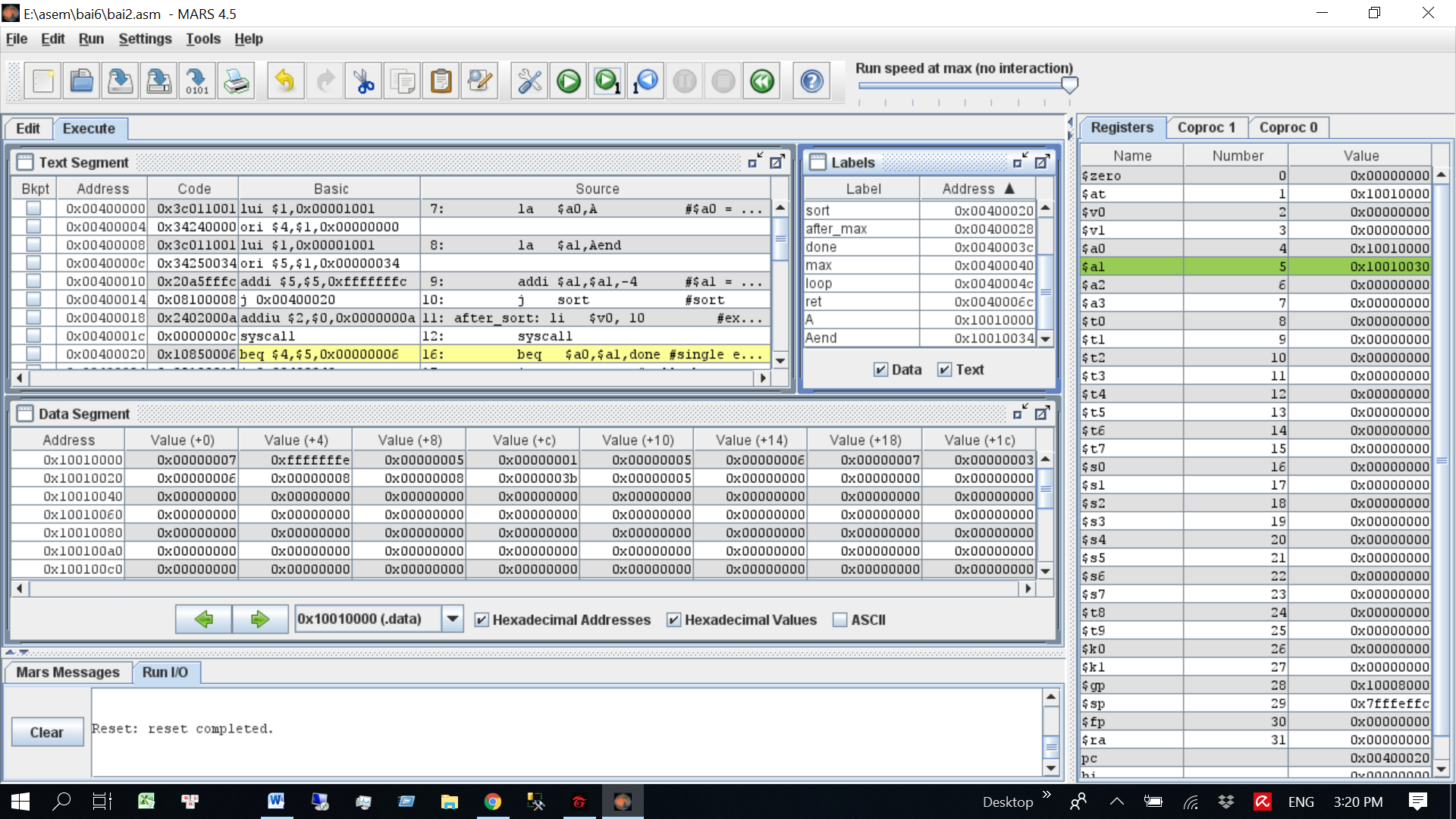
la $a1,Aend

addi $a1,$a1,-4 #$a1 = Address(A[n-1])

j sort

* Gán $a0 là địa chỉ của A[0].$a1 là địa chỉ của A[N-1] . Sau đó chạy sort

Kết quả :



Câu lệnh tiếp theo :

sort:

beq $a0,$a1,done #single element list is sorted

j max #call the max procedure

* So sánh $a0 và $a1 để biết được đã xét hết toàn bộ dãy chưa . nếu rồi nhảy đến done , nếu chưa nhảy đến max .

Câu lệnh tiếp theo :

max:

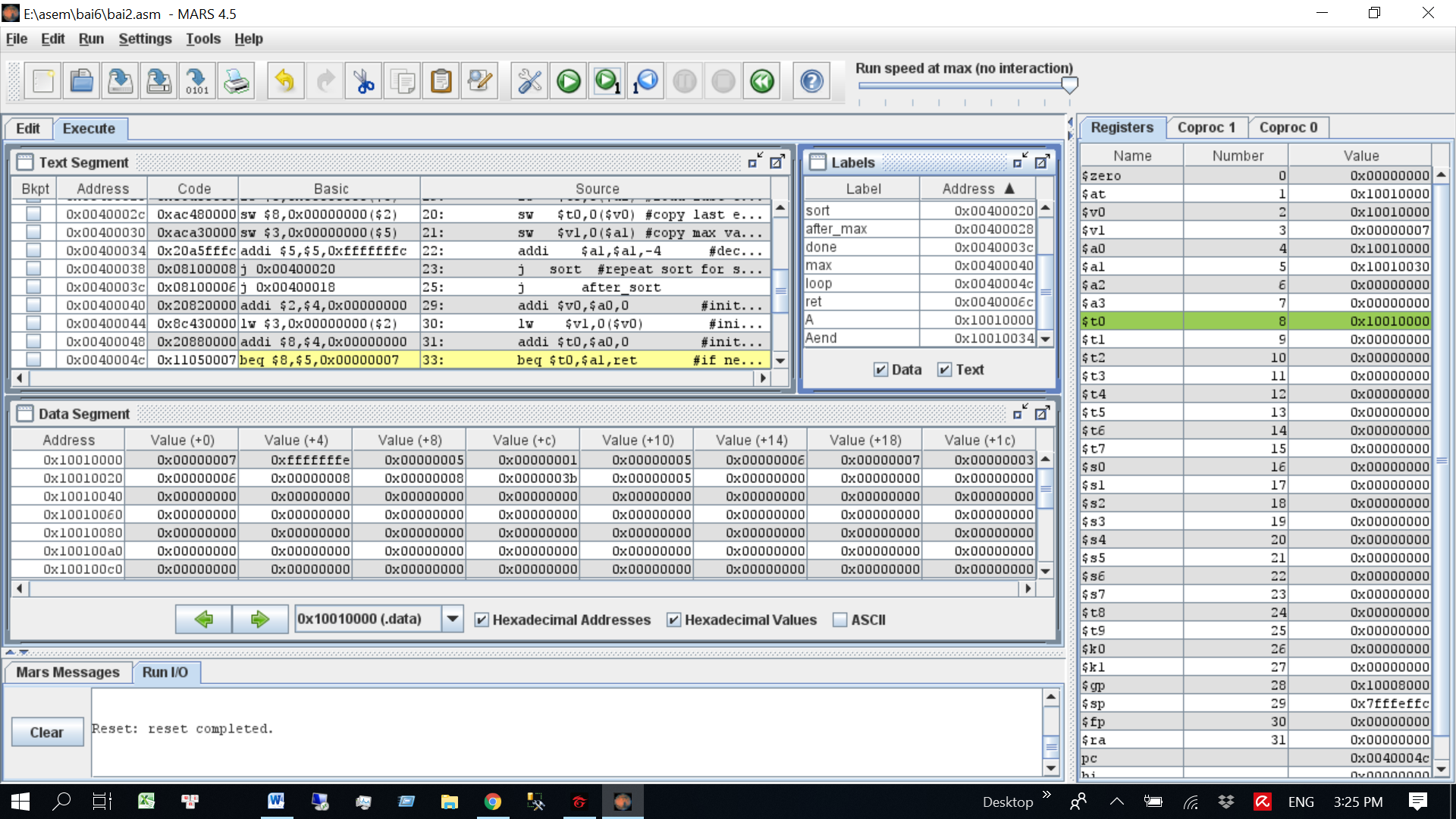
addi $v0,$a0,0 #init max pointer to first element

lw $v1,0($v0) #init max value to first value

addi $t0,$a0,0 #init next pointer to first

* Gán $v0 là con trỏ đến số lớn nhất , và khi bắt đầu gán nó là số đầu tiên . Lấy $v1 để load giá trị lớn nhất . gán con trỏ tiếp theo đến số đầu tiên .

Kết quả $v1 có giá trị 7 , $v0 chỉ địa chỉ số đầu tiên của mảng. $t0 chỉ địa chỉ số đầu tiên



Câu lệnh tiếp theo :

loop:

beq $t0,$a1,ret #if next=last, return

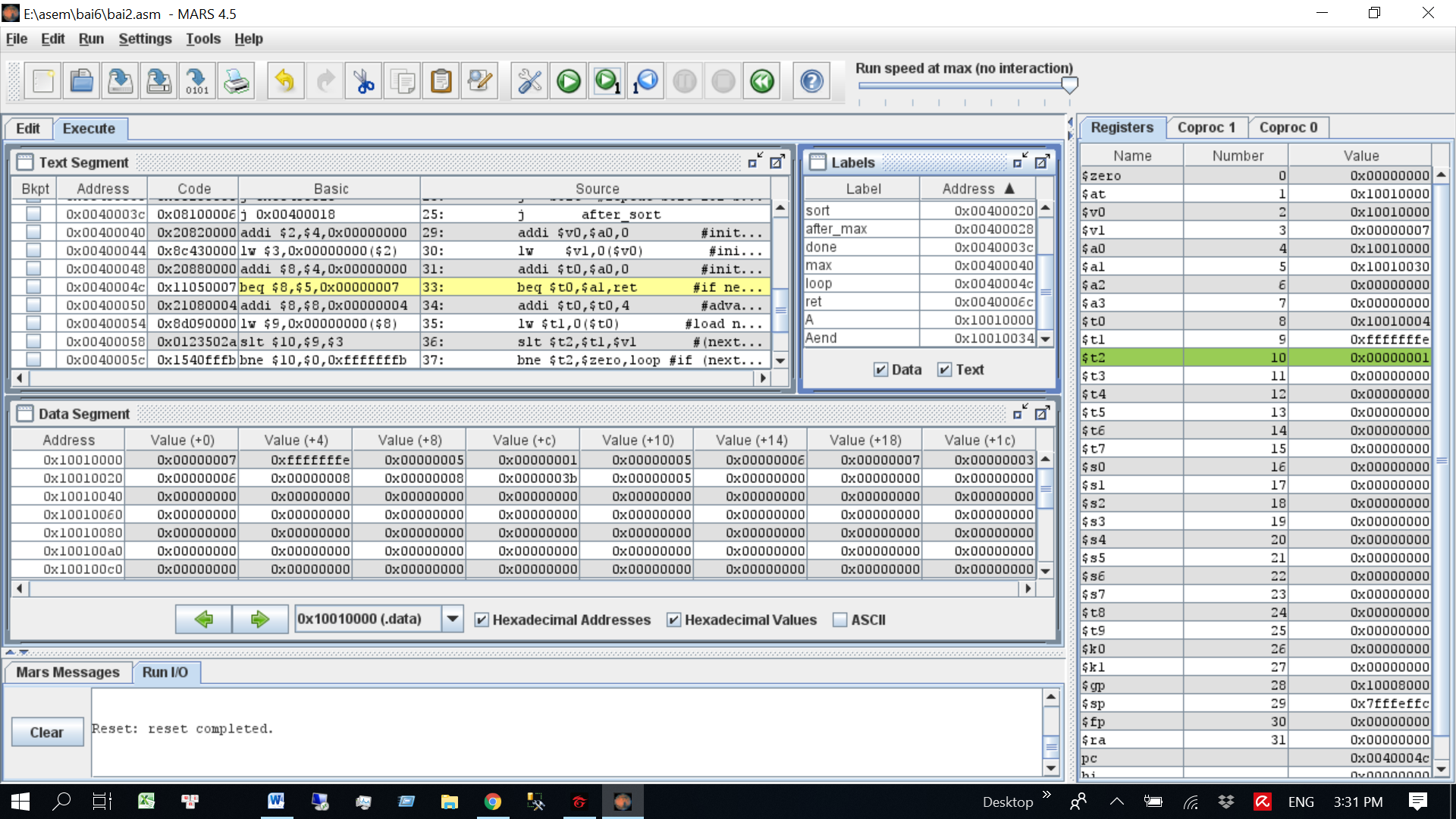
addi $t0,$t0,4 #advance to next element

lw $t1,0($t0) #load next element into $t1

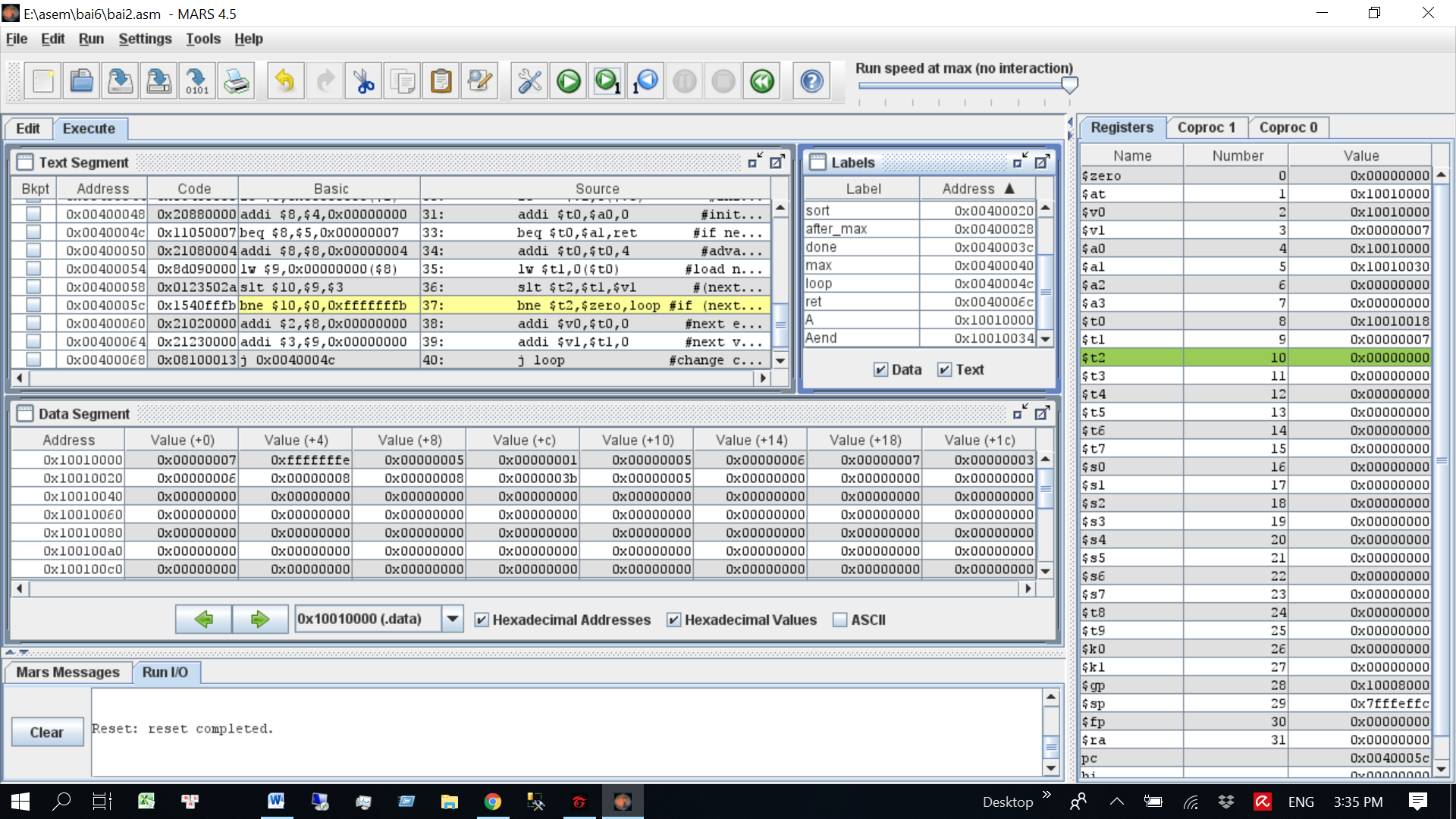
slt $t2,$t1,$v1 #(next)<(max) ?

bne $t2,$zero,loop #if (next)<(max), repeat

* Nếu $t0 = $a1 thì chạy ret . gán $t0 =$t0 + 4 để chỉ phần tử tiếp theo . sau đó gán $t1 là giá trị của phần tử A[1]. Nếu $t1 < max= $v1 thì vòng lặp
* Kết quả:$t0=0x10010004 , $t1=0xfffffffe = -2 < max = $v1 = 0x…7

 Câu lệnh tiếp theo :

Chạy vòng lặp đến khi tìm được số >= 7 .



Ở đây $t1=7=$v1 nên chạy tiếp chứ k chạy đến loop

Câu lệnh tiếp theo :

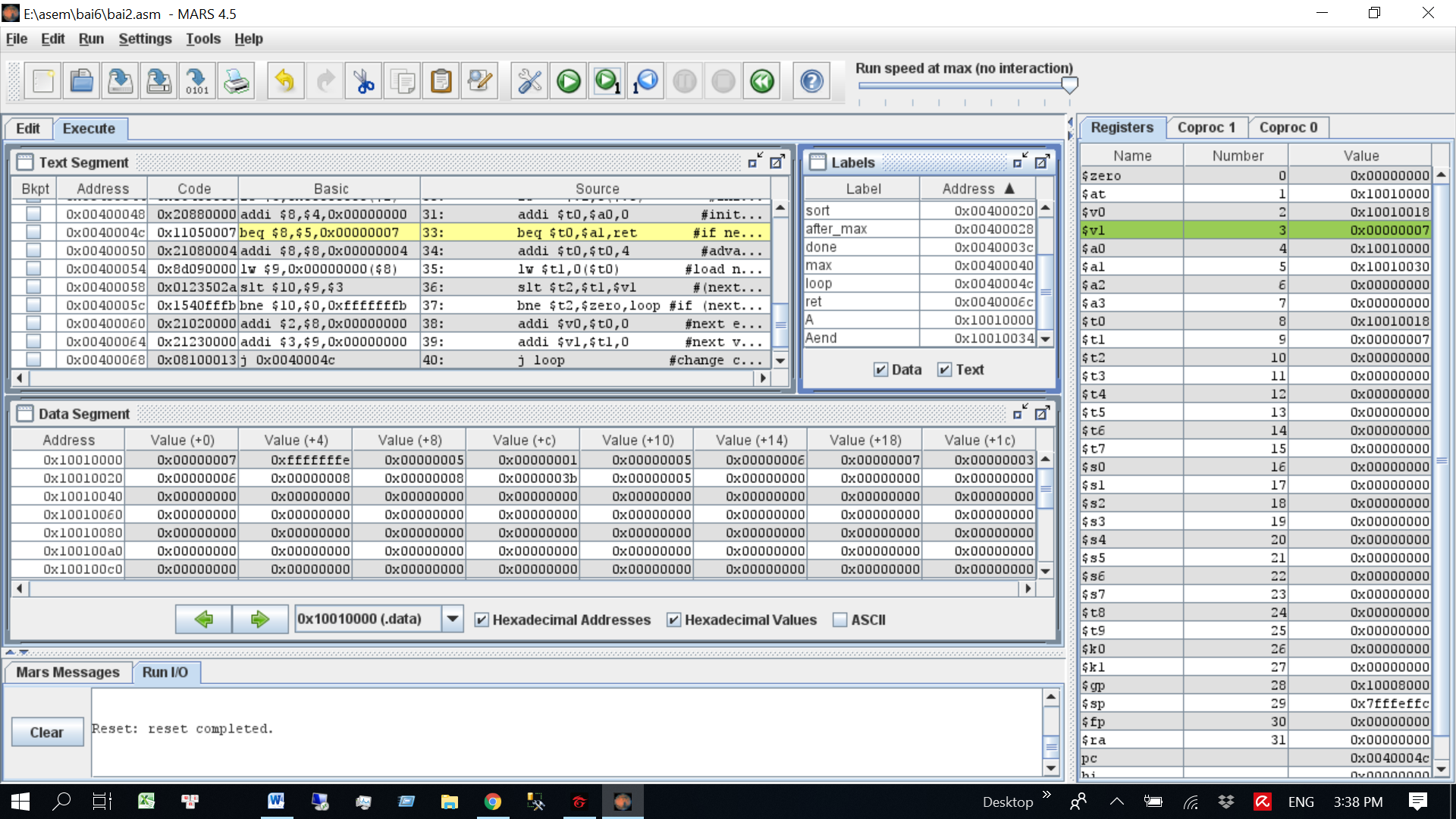
addi $v0,$t0,0 #next element is new max element

addi $v1,$t1,0 #next value is new max value

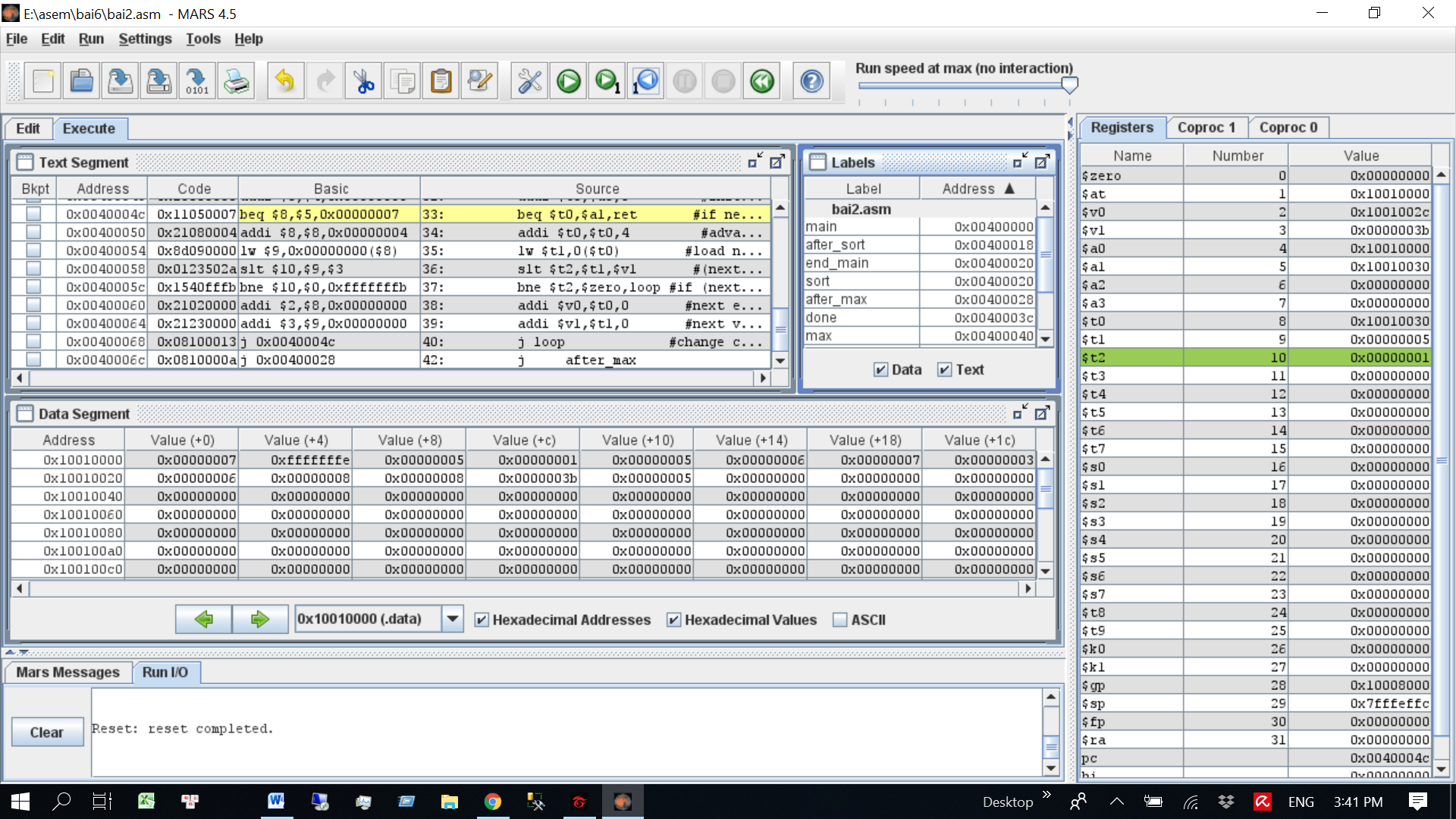
j loop #change completed; now repeat

* Gán $v0 bằng $t0 =0x11010018 để chỉ đến phần tử có giá trị lớn nhất cho đếm hiện tại tìm được trong dãy .$v1 = giá trị lớn nhất cho tới hiện tại = 0x…7.Sau đó chạy vòng lặp

Kết quả :



Lại tiếp tục chạy vòng lặp cho đến khi duyệt hết dãy , ta có kết quả sau :



Ở đây , $t0 = $a1 tức là nó đã duyệt hết tất cả phần tử , ta chạy lệnh ret .

Câu lệnh tiếp theo:

ret:

j after\_max

after\_max:

lw $t0,0($a1) #load last element into $t0

sw $t0,0($v0) #copy last element to max location

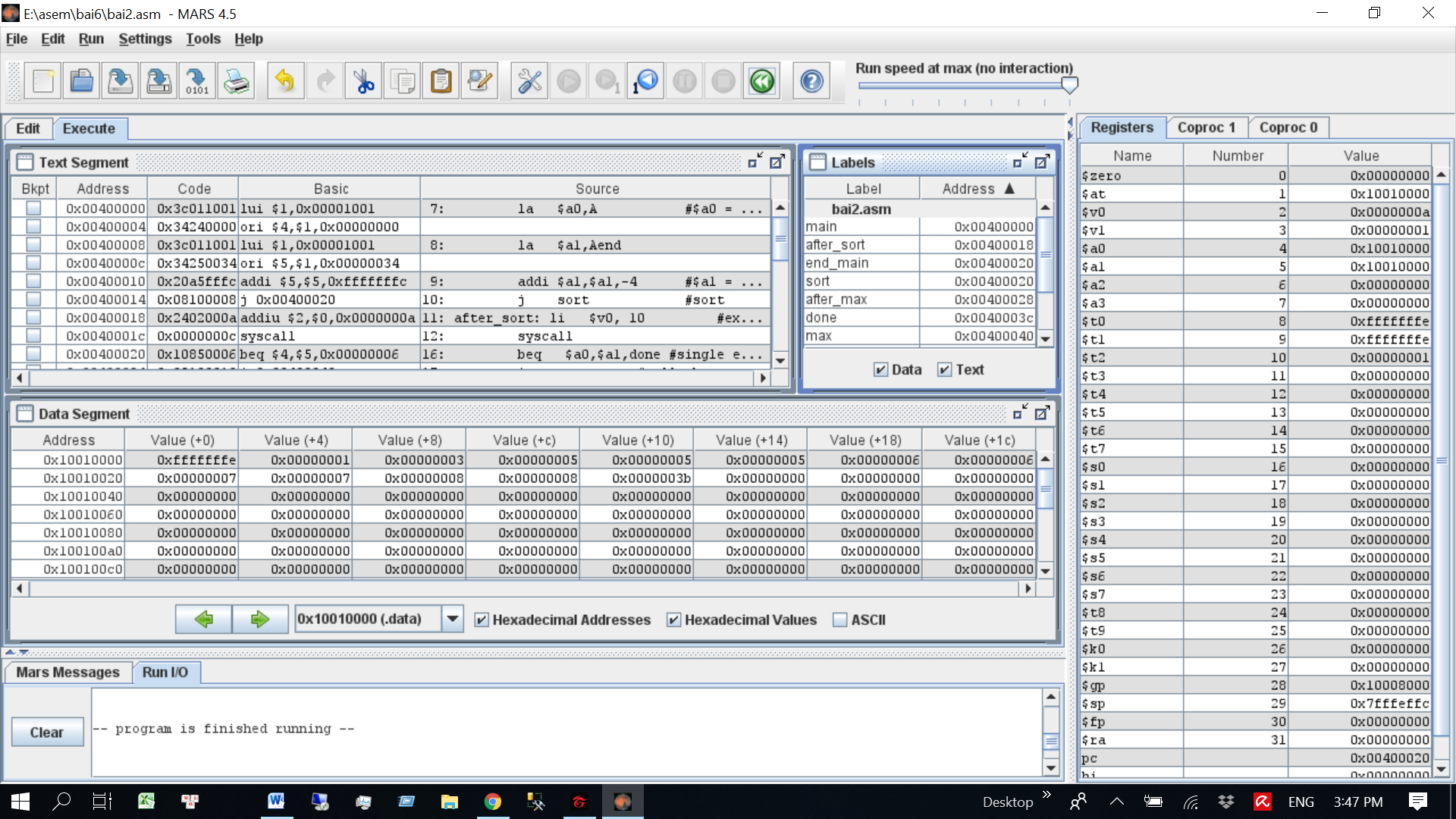
sw $v1,0($a1) #copy max value to last element

addi $a1,$a1,-4 #decrement pointer to last element

j sort #repeat sort for smaller list

* Chạy lệnh after\_max ta thực hiện swap phần tử lớn nhất là $v1 vô phần tử cuối cùng . sau đó thực hiện lại toàn bộ quá trình trên với mảng có n-1 phần tử trừ phần tử cuối cùng .

Kết quả :



Nhận thấy $a0 = $a1 tức thực hiện gán lui phần tử cuối bằng max cho đến số đầu tiên = > dãy đã được sắp xếp .