**BÁO CÁO THỰC HÀNH KIẾN TRÚC MÁY TÍNH (IT3280) TUẦN 3**

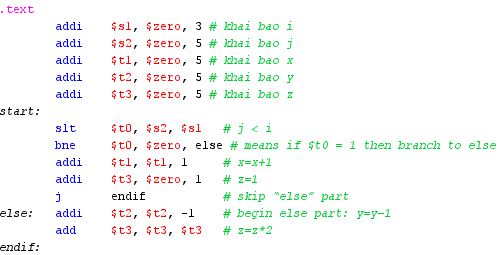
***Họ và tên: Phạm Đức Long***

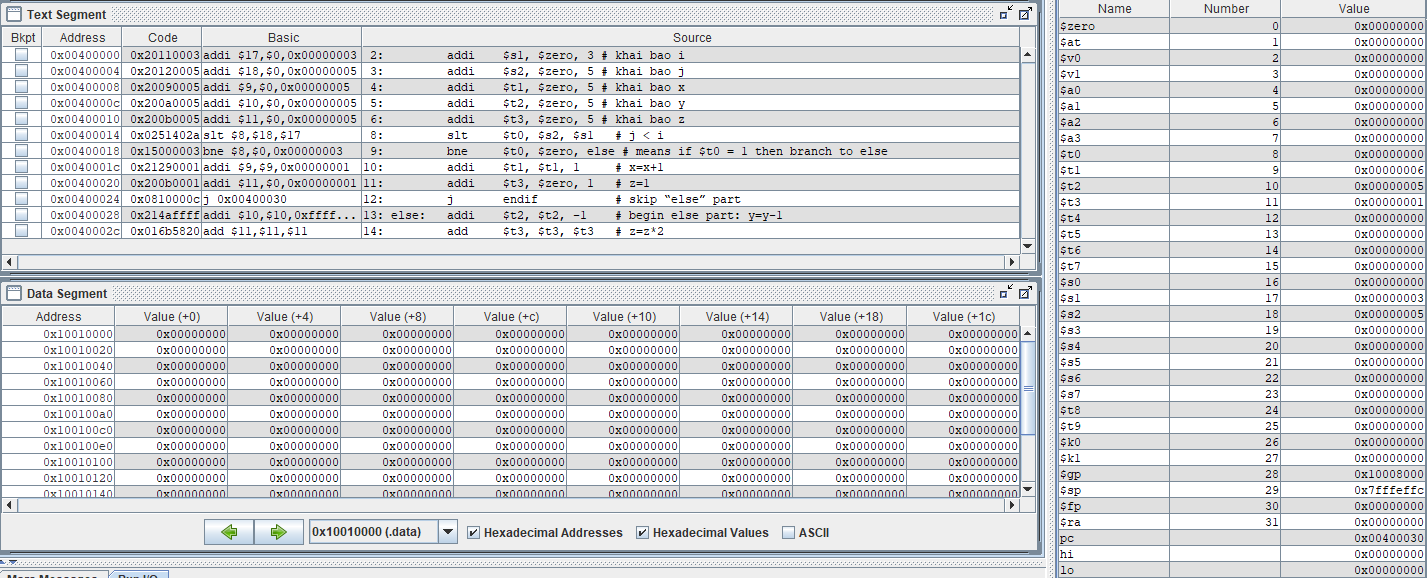
***MSSV: 20225737***

Assignment 1:

Ta xét hai trường hợp i = 3, j = 5 và ngược lại.

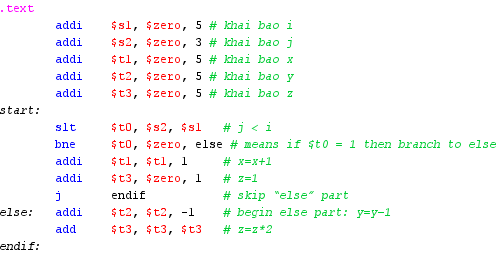
* TH1: i = 3 và j = 5 ( i <= j):

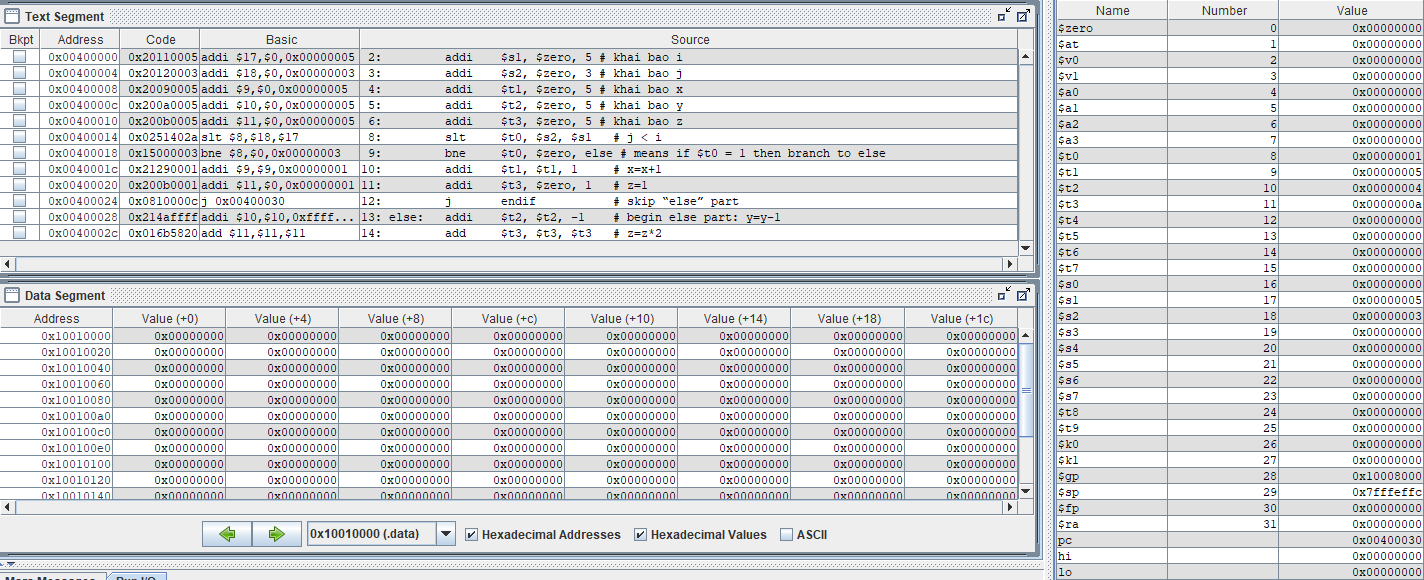




|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | $s1 | $s2 | $t0 | $t1 | $t2 | $t3 | pc |
| Ban đầu | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x40000000 |
| Khai báo i | 0x00000003 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x40000004 |
| Khai báo j | 0x00000003 | 0x00000005 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x40000008 |
| Khai báo x, y, z | 0x00000003 | 0x00000005 | 0x00000000 | 0x00000005 | 0x00000005 | 0x00000005 | 0x40000014 |
| Chạy lệnh slt | 0x00000003 | 0x00000005 | 0x00000000 | 0x00000005 | 0x00000005 | 0x00000005 | 0x40000018 |
| Chạy lệnh bne | 0x00000003 | 0x00000005 | 0x00000000 | 0x00000005 | 0x00000005 | 0x00000005 | 0x4000001c |
| Chạy lệnh addi thứ 1 của start | 0x00000003 | 0x00000005 | 0x00000000 | 0x00000006 | 0x00000005 | 0x00000005 | 0x40000020 |
| Chạy lệnh addi thứ 2 của start | 0x00000003 | 0x00000005 | 0x00000000 | 0x00000006 | 0x00000005 | 0x00000001 | 0x40000024 |

* Việc kiểm thử là đúng khi ta thấy j >= i (với giá trị thanh ghi $s1 là 3 (0x00000003) và giá trị thanh ghi $s2 là 5 (0x00000005)) thì giá trị thanh ghi $t0 sẽ trả về 0 (địa chỉ 0x00000000) và giá trị thanh ghi này đúng với thanh ghi $zero nên trình biên dịch không thực hiện lệnh rẽ nhánh mà tiếp tục các lệnh addi, từ đó tính toán và gán giá trị cho thanh ghi $t1 là 6 (0x00000006) và gán giá trị thanh ghi $t3 là 1 (0x00000001).
* TH2: i = 5 và j = 3 ( i > j):

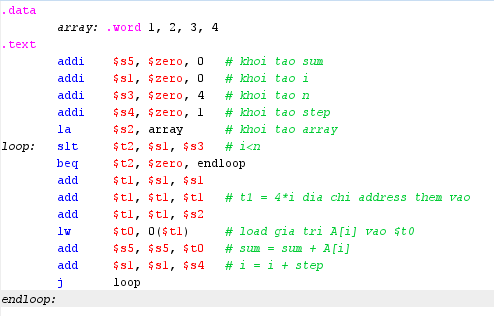




|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | $s1 | $s2 | $t0 | $t1 | $t2 | $t3 | pc |
| Ban đầu | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x40000000 |
| Khai báo i, j | 0x00000005 | 0x00000003 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x40000008 |
| Khai báo x, y, z | 0x00000005 | 0x00000003 | 0x00000000 | 0x00000005 | 0x00000005 | 0x00000005 | 0x40000014 |
| Chạy lệnh slt | 0x00000005 | 0x00000003 | 0x00000001 | 0x00000005 | 0x00000005 | 0x00000005 | 0x40000018 |
| Chạy lệnh bne | 0x00000005 | 0x00000003 | 0x00000001 | 0x00000005 | 0x00000005 | 0x00000005 | 0x40000028 |
| Chạy lệnh addi thứ 1 của else | 0x00000005 | 0x00000003 | 0x00000001 | 0x00000000 | 0x00000004 | 0x00000000 | 0x4000002c |
| Chạy lệnh add của else | 0x00000005 | 0x00000003 | 0x00000001 | 0x00000000 | 0x00000004 | 0x0000000a | 0x40000030 |

* Việc kiểm thử là đúng khi ta thấy j < i (với giá trị thanh ghi $s1 là 5 (0x00000005) và giá trị thanh ghi $s2 là 3 (0x00000003)) thì giá trị thanh ghi $t0 sẽ trả về 1 (địa chỉ 0x00000001) và giá trị thanh ghi này khác với thanh ghi $zero nên trình biên dịch đã thực hiện lệnh rẽ nhánh, từ đó tính toán và gán giá trị cho thanh ghi $t2 là 4 (địa chỉ là 0x00000004) và gắn giá trị thanh ghi $t3 bằng 2 lần giá trị chính nó là 10 (địa chỉ là 0x0000000a) .

Assignment 2:



* Đầu tiên ta khởi tạo mảng có 4 phần tử có các giá trị là 1, 2, 3, 4.
* Tiếp đến ta khởi tạo các giá trị i, n, step (bước nhảy), step = 4 => sẽ có 4 lần lặp => sau 4 vòng lặp ta sẽ thu được tổng các phần tử trong chuỗi chính là giá trị của thanh ghi $s5.
* Sự thay đổi của các thanh ghi:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | $t1 | $t2 | $s1 | $s5 |
| Ban đầu | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| Vòng loop 1 | 0x10010000 | 0x00000001 | 0x00000001 | 0x00000001 |
| Vòng loop 2 | 0x10010004 | 0x00000001 | 0x00000002 | 0x00000003 |
| Vòng loop 3 | 0x10010008 | 0x00000001 | 0x00000003 | 0x00000006 |
| Vòng loop 4 | 0x1001000c | 0x00000001 | 0x00000004 | 0x0000000a |
| Vòng ‘loop’ 5 | 0x1001000c | 0x00000000 | 0x00000004 | 0x0000000a |

* Nhận xét:
* Vòng lặp dừng lại khi $t2 = 0x00000000, tương ứng với điều kiện câu lệnh: beq $t2, $zero, endloop
* Kết quả giá trị sum = $s5 = 0x0000000a = 10 = 1 + 2 + 3 + 4, điều này đúng với tính toán khi ta tính tổng các phần tử của mảng.

Assignment 3:

* TH1: text = 1:

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | $s1 | $s2 | $s3 | pc |
| Ban đầu | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| Sau khi khởi tạo | 0x00000001 | 0x00000000 | 0x00000000 | 0x00400018 |
| Sau lệnh beq case\_1 và thực hiện case\_1 | 0x00000001 | 0xffffffff | 0x00000000 | 0x00400034 |

* Do $s1 = $t1 = 1 nên lệnh hàm sẽ thực hiện hàm case\_1 và thanh ghi pc nhảy đến giá trị của case\_1 là 0x00400030 thực hiện lệnh trừ thanh ghi $s2 cho 1 ta được $s2 = -1 ( thỏa mãn).
* TH2: text = 0:

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

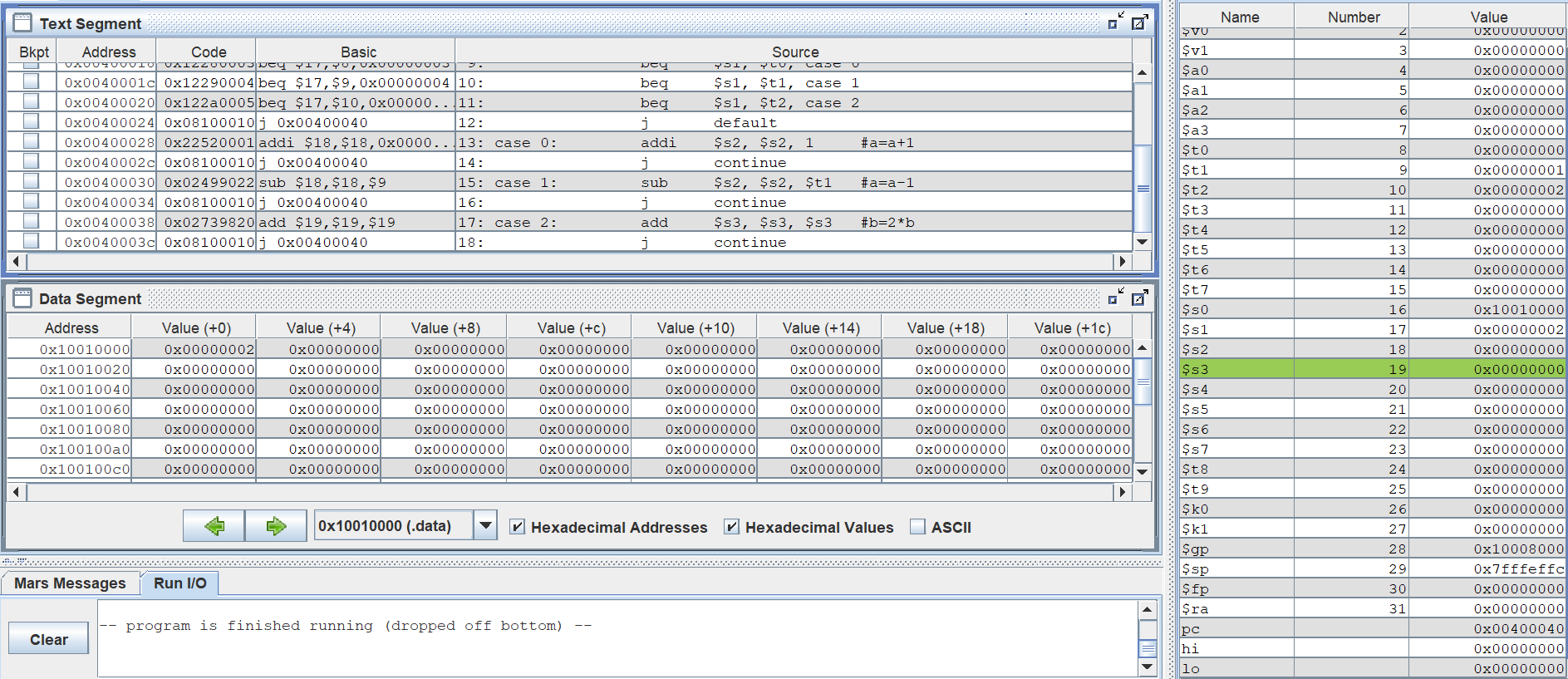
Description automatically generated

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | $s1 | $s2 | $s3 | pc |
| Ban đầu | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| Sau khi khởi tạo | 0x00000000 | 0x00000000 | 0x00000000 | 0x00400018 |
| Sau lệnh beq case\_0 và thực hiện case\_0 | 0x00000000 | 0x00000001 | 0x00000000 | 0x0040002c |

* Do $s1 = $t0 = 0 nên lệnh hàm sẽ thực hiện hàm case\_0 và thanh ghi pc nhảy đến giá trị của thẻ case\_0 là 0x00400028 thực hiện lệnh cộng thanh ghi $s2 cho 1 ta được $s2 = 1 ( thỏa mãn).
* TH3: text = 2:

A screenshot of a computer program

Description automatically generated



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | $s1 | $s2 | $s3 | pc |
| Ban đầu | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| Sau khi khởi tạo | 0x00000002 | 0x00000000 | 0x00000000 | 0x00400018 |
| Sau lệnh beq case\_2 và thực hiện case\_2 | 0x00000002 | 0x00000000 | 0x00000000 | 0x0040003c |

🡪 Như vậy do $s1 = $t2 = 2 nên lệnh hàm sẽ thực hiện hàm case\_2 và thanh ghi pc nhảy đến giá trị của thẻ case\_0 là 0x00400038 thực hiện lệnh cộng thanh ghi $s3 với thanh ghi $s3 ta được $s3 = 0 ( thỏa mãn).

Assignment 4:

1. i < j:

* Code:

.text

addi $s1, $zero, 3 # khai bao i

addi $s2, $zero, 5 # khai bao j

addi $t1, $zero, 5 # khai bao x

addi $t2, $zero, 5 # khai bao y

addi $t3, $zero, 5 # khai bao z

la $t4, m

lw $s4, 0($t4) # khai bao m

la $t5, n

lw $s5, 0($t5) # khai bao n

start:

slt $t0, $s1, $s2 # i<j

bne $t0, $zero, else

addi $t1, $t1, 1 # x=x+1

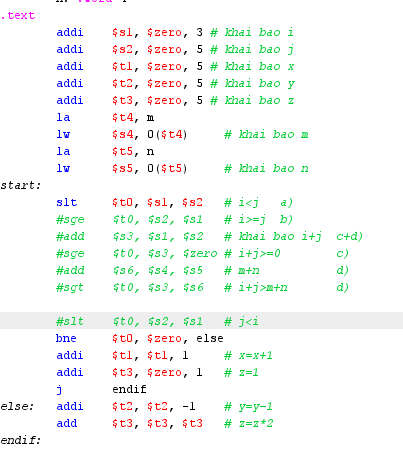
addi $t3, $zero, 1 # z=1

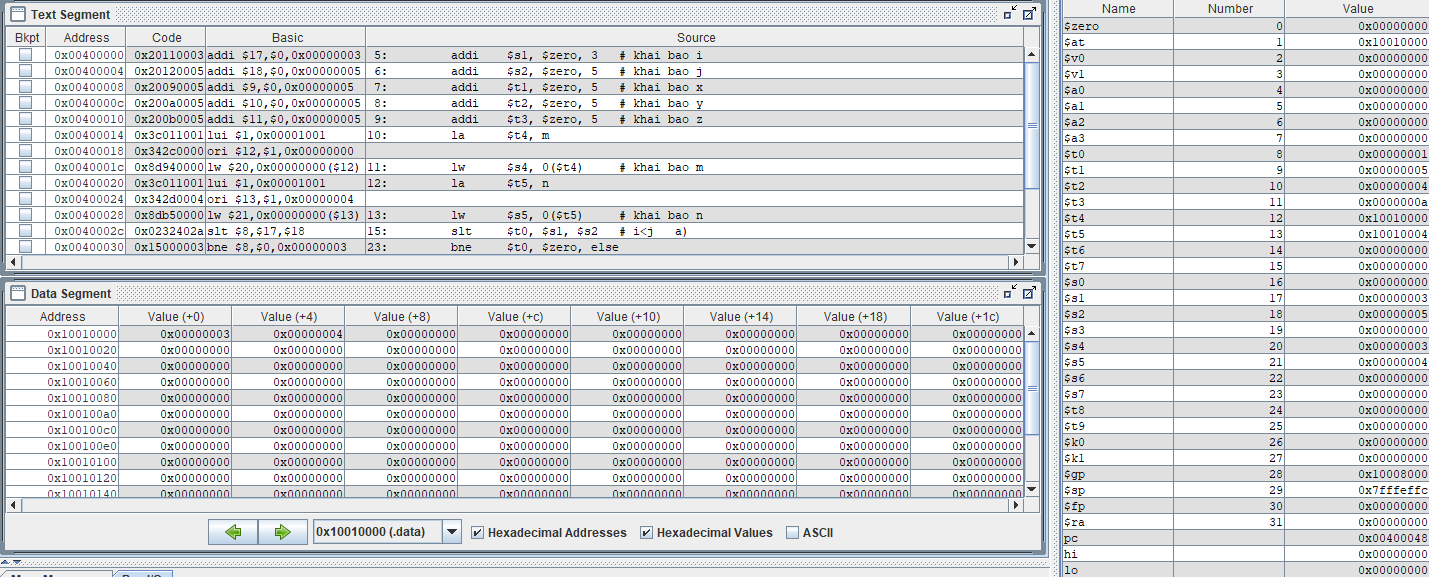
j endif

else: addi $t2, $t2, -1 # y=y-1

add $t3, $t3, $t3 # z=z\*2

endif:





1. i >= j:

* Code:

.text

addi $s1, $zero, 3 # khai bao i

addi $s2, $zero, 5 # khai bao j

addi $t1, $zero, 5 # khai bao x

addi $t2, $zero, 5 # khai bao y

addi $t3, $zero, 5 # khai bao z

la $t4, m

lw $s4, 0($t4) # khai bao m

la $t5, n

lw $s5, 0($t5) # khai bao n

start:

sge $t0, $s1, $s2 # i>=j b)

bne $t0, $zero, else

addi $t1, $t1, 1 # x=x+1

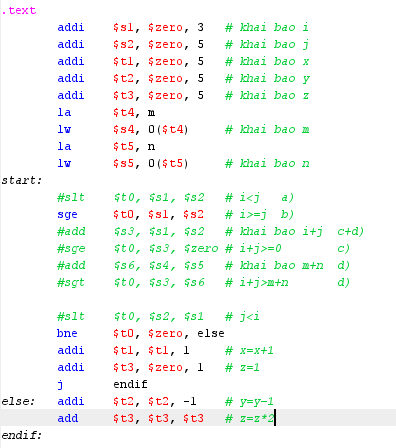
addi $t3, $zero, 1 # z=1

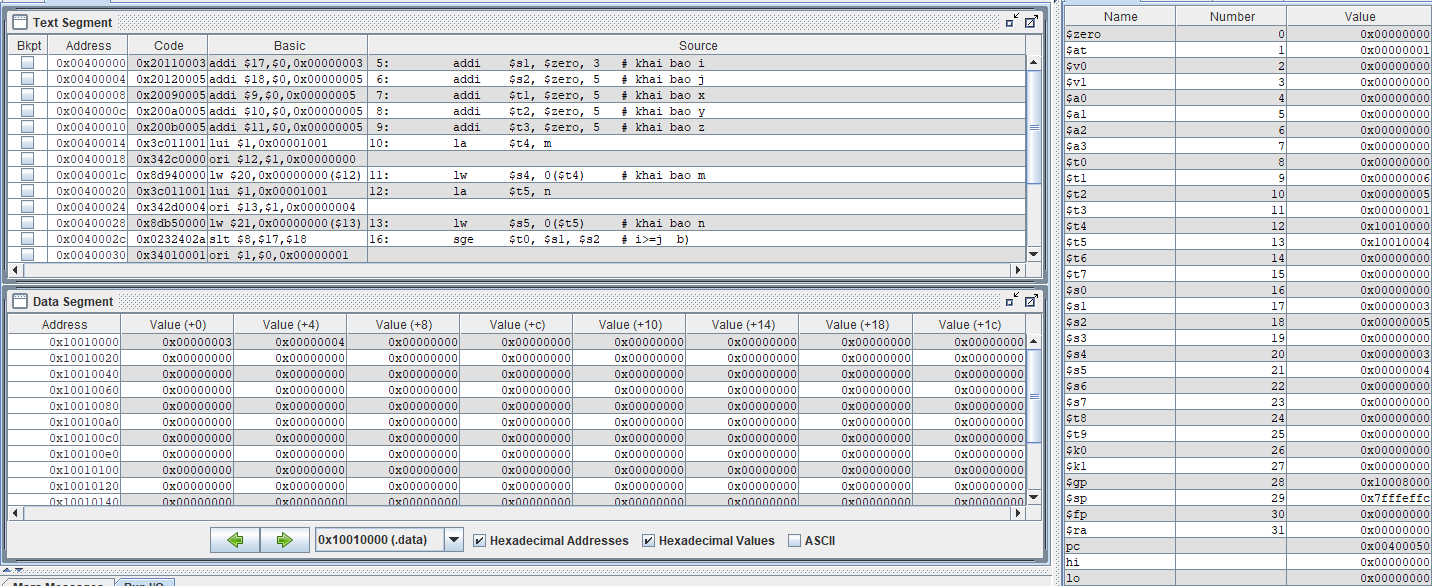
j endif

else: addi $t2, $t2, -1 # y=y-1

add $t3, $t3, $t3 # z=z\*2

endif:





1. i + j <= 0:

* Code:

.text

addi $s1, $zero, 3 # khai bao i

addi $s2, $zero, 5 # khai bao j

addi $t1, $zero, 5 # khai bao x

addi $t2, $zero, 5 # khai bao y

addi $t3, $zero, 5 # khai bao z

la $t4, m

lw $s4, 0($t4) # khai bao m

la $t5, n

lw $s5, 0($t5) # khai bao n

start:

add $s3, $s1, $s2 # khai bao i+j c)

sge $t0, $s3, $zero # i+j>=0 c)

bne $t0, $zero, else

addi $t1, $t1, 1 # x=x+1

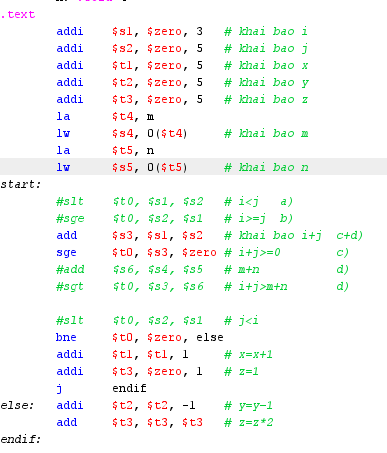
addi $t3, $zero, 1 # z=1

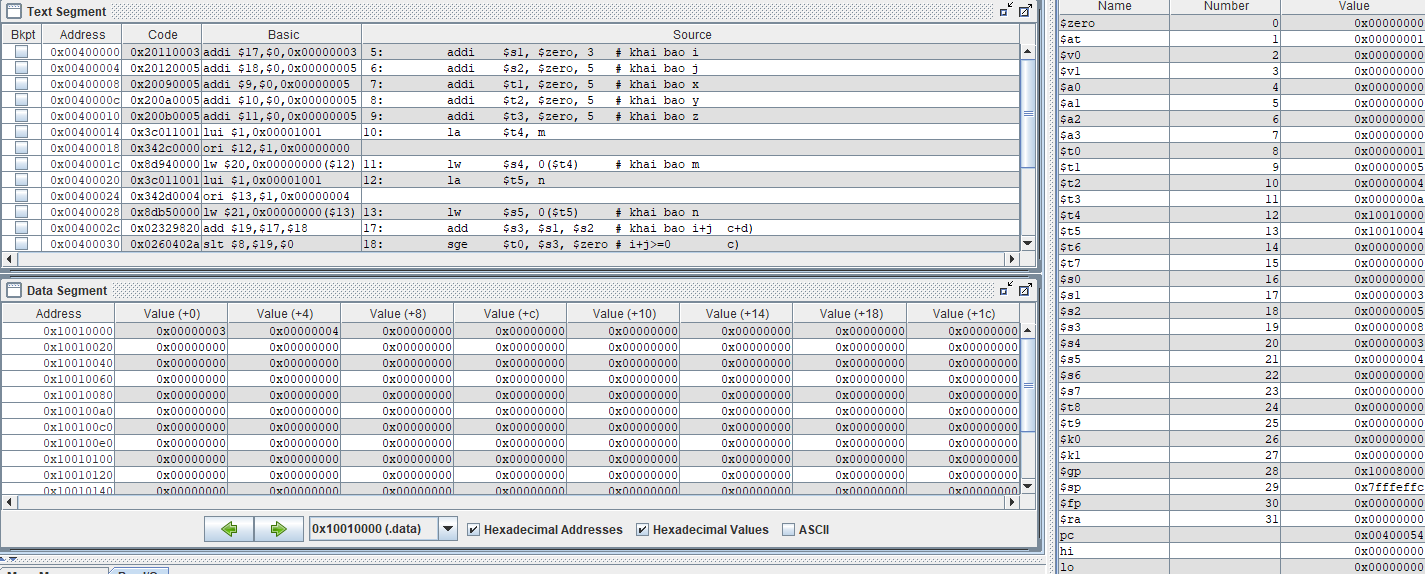
j endif

else: addi $t2, $t2, -1 # y=y-1

add $t3, $t3, $t3 # z=z\*2

endif:





1. i + j > m + n:

* Code:

.data

m: .word 3

n: .word 4

.text

addi $s1, $zero, 3 # khai bao i

addi $s2, $zero, 5 # khai bao j

addi $t1, $zero, 5 # khai bao x

addi $t2, $zero, 5 # khai bao y

addi $t3, $zero, 5 # khai bao z

la $t4, m

lw $s4, 0($t4) # khai bao m

la $t5, n

lw $s5, 0($t5) # khai bao n

start:

add $s3, $s1, $s2 # khai bao i+j c+d)

add $s6, $s4, $s5 # khai bao m+n d)

sgt $t0, $s3, $s6 # i+j > m+n d)

#slt $t0, $s2, $s1 # j<i

bne $t0, $zero, else

addi $t1, $t1, 1 # x=x+1

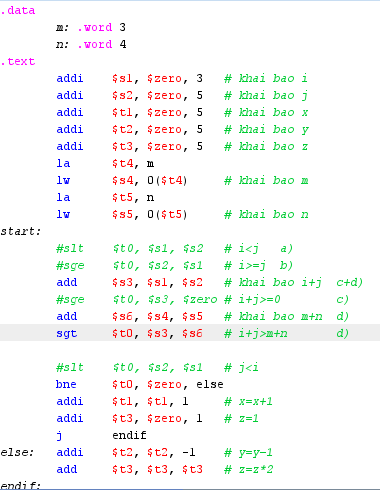
addi $t3, $zero, 1 # z=1

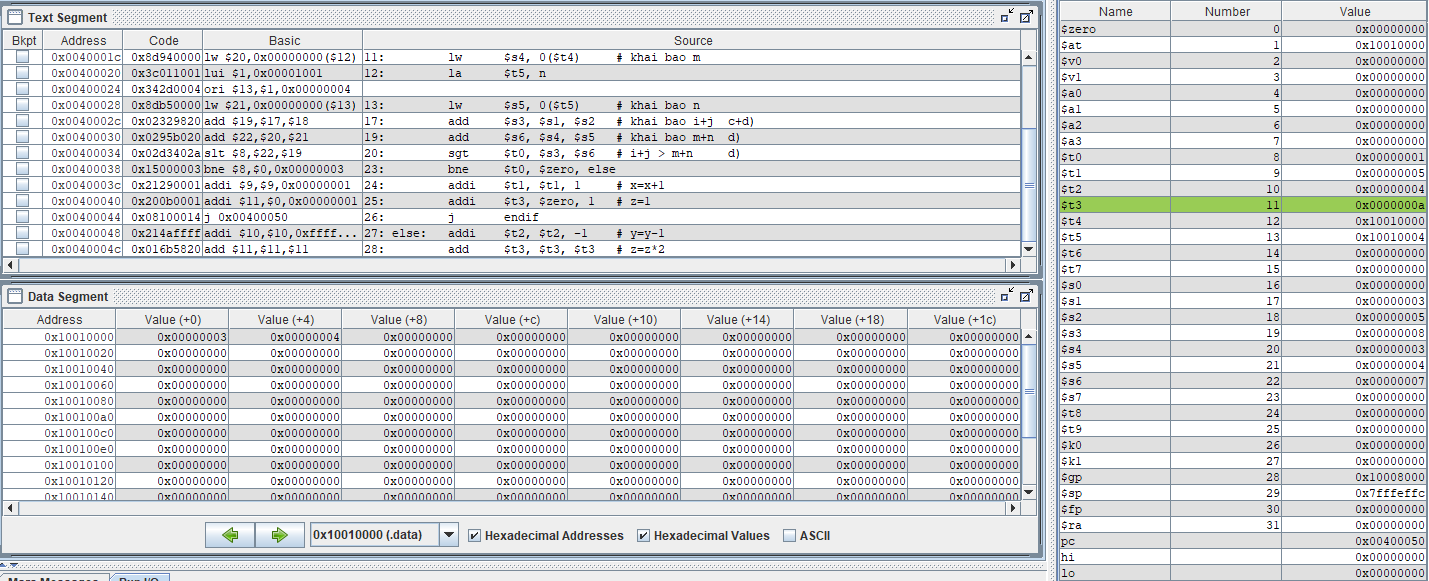
j endif

else: addi $t2, $t2, -1 # y=y-1

add $t3, $t3, $t3 # z=z\*2

endif:





Assignment 5:

1. i <= n

* Code:

.data

array: .word 1, 2, 3, 4, 5

.text

addi $s5, $zero, 0 # khoi tao sum

addi $s1, $zero, 0 # khoi tao i

addi $s3, $zero, 4 # khoi tao n

addi $s4, $zero, 1 # khoi tao step

la $s2, array # khoi tao array

loop: sle $t2, $s1, $s3 # i<=n

beq $t2, $zero, endloop

add $t1, $s1, $s1

add $t1, $t1, $t1 # t1 = 4\*i dia chi address them vao

add $t1, $t1, $s2

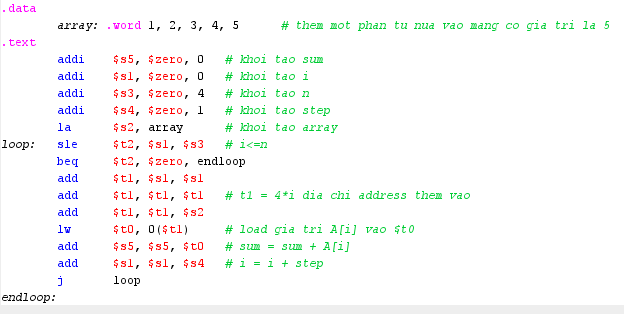
lw $t0, 0($t1) # load gia tri A[i] vao $t0

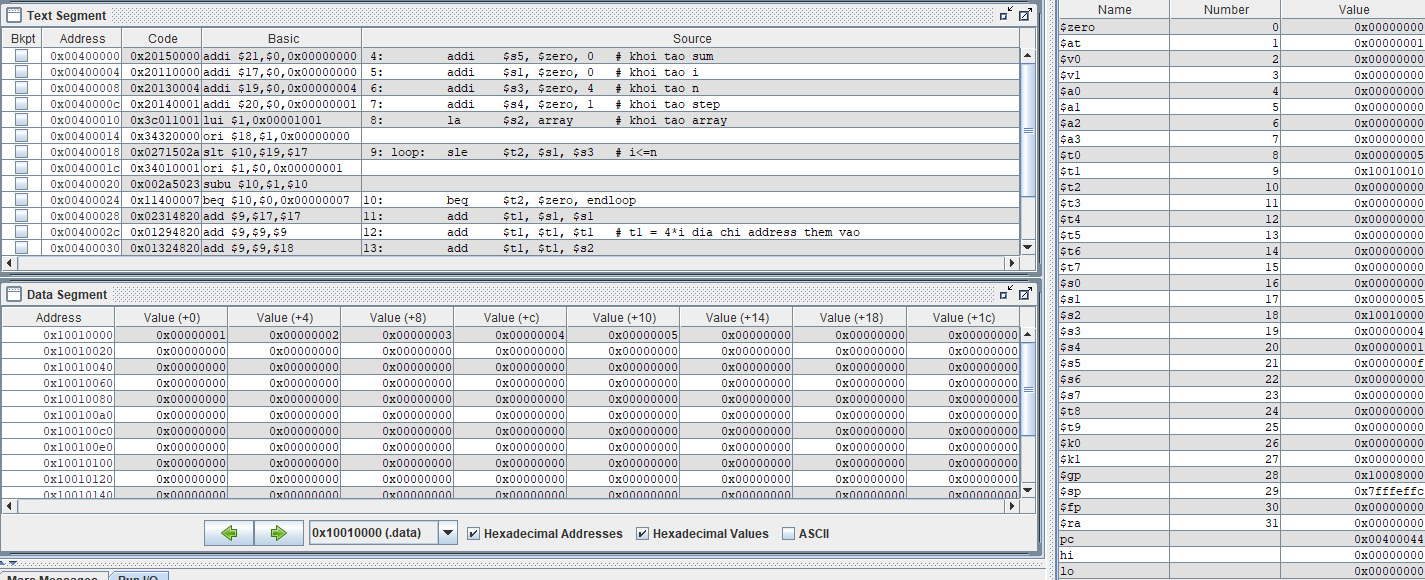
add $s5, $s5, $t0 # sum = sum + A[i]

add $s1, $s1, $s4 # i = i + step

j loop

endloop:





* Nhận xét: Thanh ghi $s5 giờ có giá trị là 0x0000000f = 15, đúng với phép tính 1 + 2 + 3 + 4 + 5 = 15.

1. sum >= 0

* Code:

.data

array: .word -1, 0, 0, 0, 0 # them mot phan tu nua vao mang co gia tri la 5

.text

addi $s5, $zero, 0 # khoi tao sum

addi $s1, $zero, 0 # khoi tao i

addi $s3, $zero, 4 # khoi tao n

addi $s4, $zero, 1 # khoi tao step

la $s2, array # khoi tao array

loop: sle $t2, $s1, $s3 # i<=n

beq $t2, $zero, endloop

add $t1, $s1, $s1

add $t1, $t1, $t1 # t1 = 4\*i dia chi address them vao

add $t1, $t1, $s2

lw $t0, 0($t1) # load gia tri A[i] vao $t0

add $s5, $s5, $t0 # sum = sum + A[i]

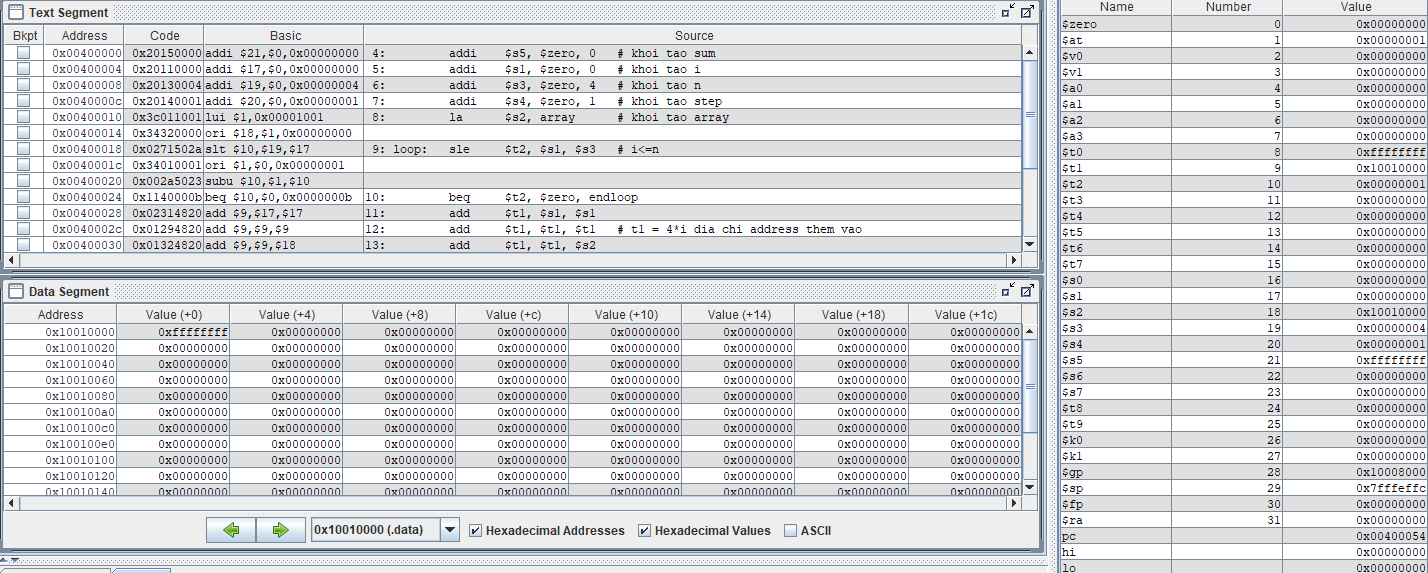
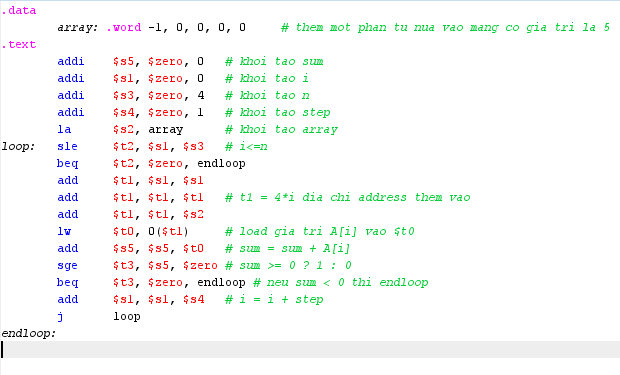
sge $t3, $s5, $zero # sum >= 0 ? 1 : 0

beq $t3, $zero, endloop # neu sum < 0 thi endloop

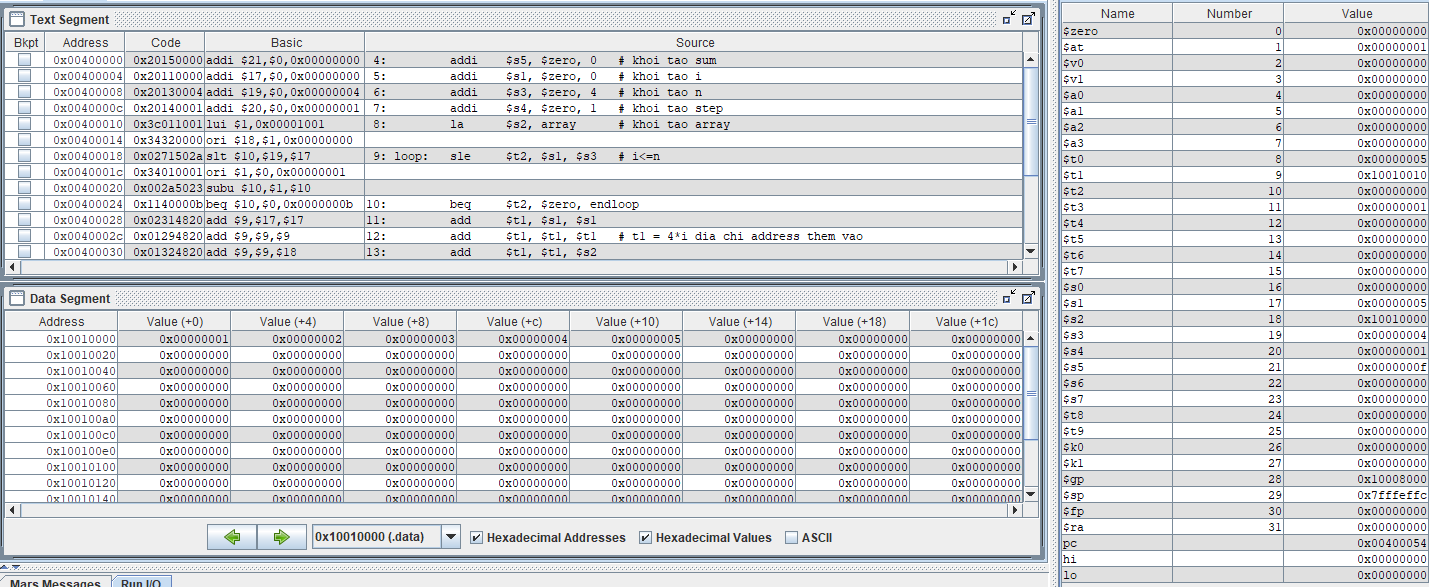
add $s1, $s1, $s4 # i = i + step

j loop

endloop:



*Dãy -1, 0, 0, 0, 0*

******

*Dãy 1, 2, 3, 4, 5*

* Nhận xét:
* Với dãy -1, 0, 0, 0, 0 thì vòng lặp đã ngay lập tức dừng khi sum = -1 < 0 => Thoả mãn.
* Với dãy 1, 2, 3, 4, 5 thì vòng lặp vẫn chạy bình thường và cho ra kết quả đúng là 0x0000000f = 15.

1. A[i] != 0

* Code:

.data

array: .word 1, 2, 3, 0, 5 # them mot phan tu nua vao mang co gia tri la 5

.text

addi $s5, $zero, 0 # khoi tao sum

addi $s1, $zero, 0 # khoi tao i

addi $s3, $zero, 4 # khoi tao n

addi $s4, $zero, 1 # khoi tao step

la $s2, array # khoi tao array

loop: sle $t2, $s1, $s3 # i<=n

beq $t2, $zero, endloop

add $t1, $s1, $s1

add $t1, $t1, $t1 # t1 = 4\*i dia chi address them vao

add $t1, $t1, $s2

lw $t0, 0($t1) # load gia tri A[i] vao $t0

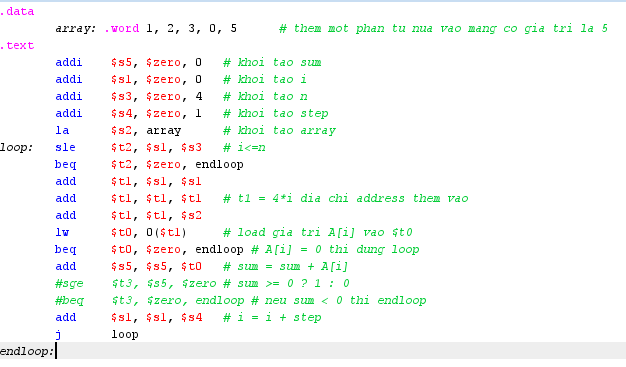
beq $t0, $zero, endloop # A[i] = 0 thi dung loop

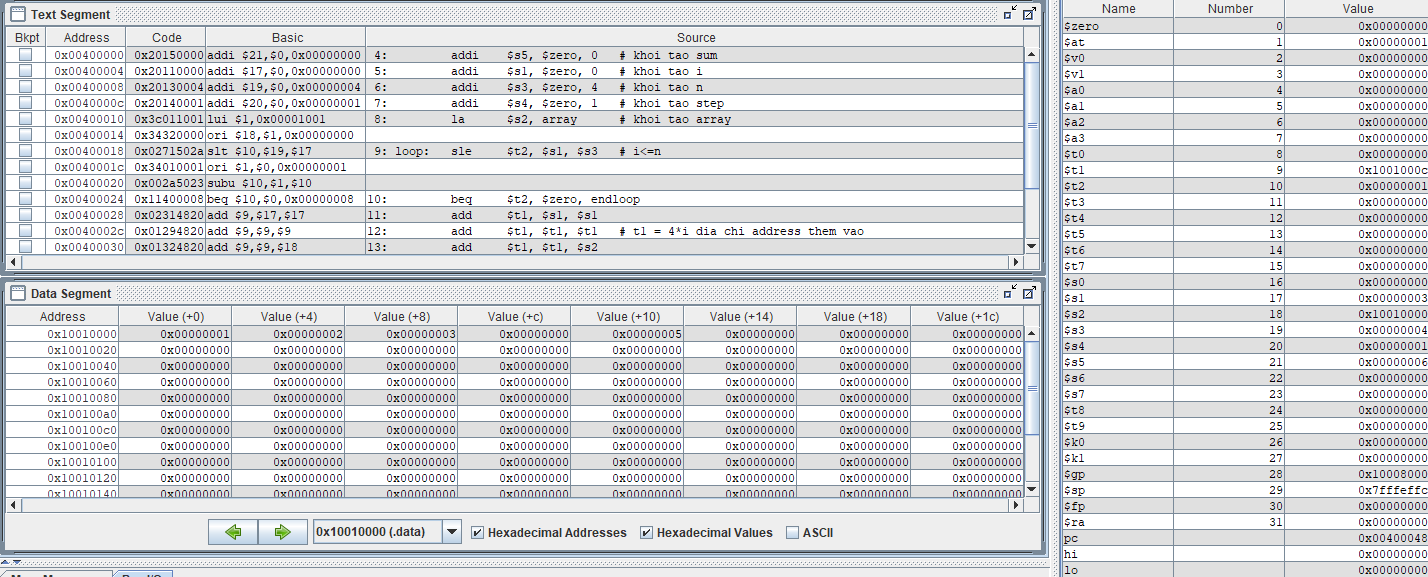
add $s5, $s5, $t0 # sum = sum + A[i]

add $s1, $s1, $s4 # i = i + step

j loop

endloop:





* Nhận xét: Xét mảng gồm 5 phần tử có các giá trị 1, 2, 3, 0, 5. Vòng lặp đã dừng lại khi A[3] = 0, và tổng lúc này ta tính được là 1 + 2 + 3 = 6, mà $s5 = 0x00000006 = 6 => Thỏa mãn đề bài.

Assignment 6:

* Code:

.data

array: .word 3, 6, -2, -8, -15, 10 # khoi tao mang va cac phan tu

.text

addi $s1, $zero, 0 # khoi tao i

addi $s2, $zero, 6 # khoi tao n

addi $s3, $zero, 1 # khoi tao step

la $s4, array # khoi tao array

addi $s5, $zero, 0 # khoi tao max

loop:

slt $t2, $s1, $s2

beq $t2, $zero, endloop

add $t1, $s1, $s1

add $t1, $t1, $t1 # t1 = 4 \* i

add $t1, $t1, $s4

lw $t0, 0($t1) # gia tri cua A[i] luu vao $t0

if:

slti $t3, $t0, 0 # A[i] < 0 ? 1 : 0

beq $t3, $zero, else # A[i} >= 0 thi else

sub $t0, $zero, $t0 # A[i] = 0 - A[i}

j endif

else: addi $t0, $t0, 0

endif:

if1:

sge $t4, $t0, $s5 # A[i] >= max ? 1 : 0

beq $t4, $zero, endif1 # neu A[i] < max thi ket thuc if1

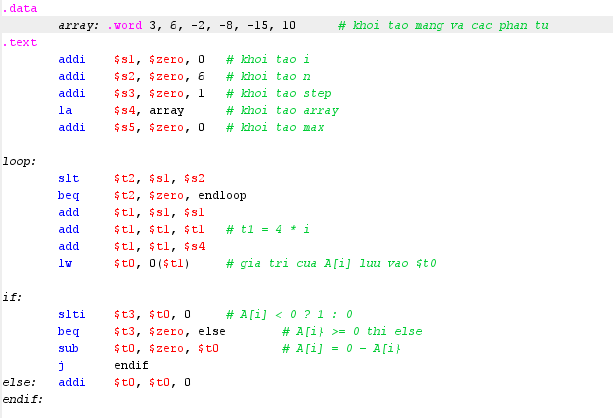
add $s5, $zero, $t0 # gan max = A[i]

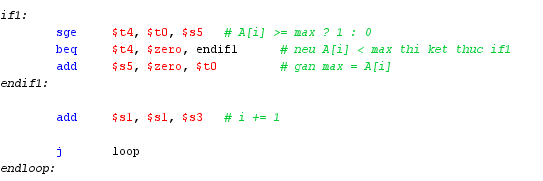
endif1:

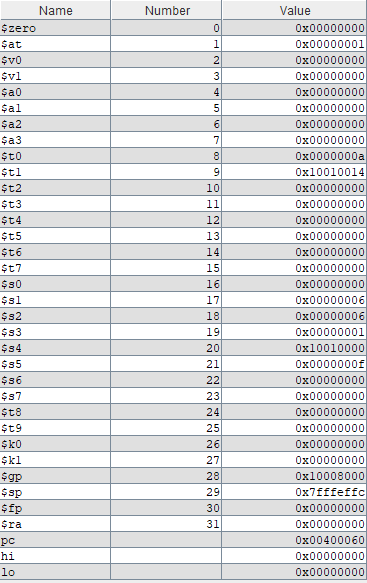
add $s1, $s1, $s3 # i += 1

j loop

endloop:







* Nhận xét: Xét mảng 6 phần tử gồm các giá trị: 3, 6, -2, -8, -15, 10, ta tìm được giá trị tuyệt đối lớn nhất của mảng này là |-15| = 15. Mặt khác, max được lưu vào thanh ghi $s5 = 0x0000000f = 15 => Kết quả sau khi chạy chương trình đã đúng với kết quả lý thuyết.