

# BÁO CÁO THỰC HÀNH KIẾN TRÚC MÁY TÍNH TUẦN 3

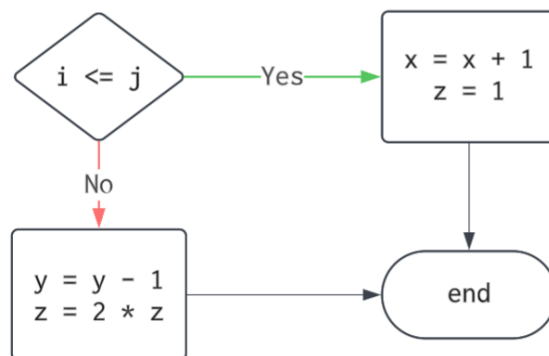
Họ và tên: Nguyễn Quý Đức MSSV: 20235682

Mã HP: IT3280

Mã lớp: 156788

## 1. Assignment 1

- Lưu đồ:



- Nhập lệnh:

```
1  # Laboratory Exercise 3, Home Assignment 1
2  .text
3  start:
4      # Lần lượt khởi tạo x,y,z,i,j
5      addi t1, zero, 2
6      addi t2, zero, 3
7      addi t3, zero, 4
8      addi s1, zero, 5
9      addi s2, zero, 3
10     blt s2, s1, else
11 then:
12     addi t1, t1, 1
13     addi t3, zero, 1
14     j endif
15 else:
16     addi t2, t2, -1
17     add t3, t3, t3
18 endif:
```

- Thực hiện từng dòng lệnh

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

|    |    |            |
|----|----|------------|
| t1 | 6  | 0x00000002 |
| t2 | 7  | 0x00000003 |
| t3 | 28 | 0x00000004 |
| s1 | 9  | 0x00000005 |
| s2 | 18 | 0x00000003 |

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

|    |    |            |
|----|----|------------|
| t2 | 7  | 0x00000002 |
| t3 | 28 | 0x00000008 |

+ Thanh ghi pc luôn nhảy thêm 4 sau mỗi lệnh

|    |  |            |
|----|--|------------|
| pc |  | 0x00400000 |
| pc |  | 0x00400004 |
| pc |  | 0x00400008 |
| pc |  | 0x0040000c |
| pc |  | 0x00400010 |
| pc |  | 0x00400014 |

(do  $i \leq j$  return false nên chương trình nhảy đến nhánh else có địa chỉ 0x00400024)

|    |  |            |
|----|--|------------|
| pc |  | 0x00400024 |
| pc |  | 0x00400028 |

| Bkpt                     | Address    | Code        | Basic                 | Source               |
|--------------------------|------------|-------------|-----------------------|----------------------|
| <input type="checkbox"/> | 0x00400000 | 0x00200313  | addi x6,x0,2          | 5: addi t1, zero, 2  |
| <input type="checkbox"/> | 0x00400004 | 0x00300393  | addi x7,x0,3          | 6: addi t2, zero, 3  |
| <input type="checkbox"/> | 0x00400008 | 0x00400e13  | addi x28,x0,4         | 7: addi t3, zero, 4  |
| <input type="checkbox"/> | 0x0040000c | 0x00500493  | addi x9,x0,5          | 8: addi s1, zero, 5  |
| <input type="checkbox"/> | 0x00400010 | 0x00300913  | addi x18,x0,3         | 9: addi s2, zero, 3  |
| <input type="checkbox"/> | 0x00400014 | 0x00994863  | blt x18,x9,0x00000010 | 10: blt s2, s1, else |
| <input type="checkbox"/> | 0x00400018 | 0x00130313  | addi x6,x6,1          | 12: addi t1, t1, 1   |
| <input type="checkbox"/> | 0x0040001c | 0x00100e13  | addi x28,x0,1         | 13: addi t3, zero, 1 |
| <input type="checkbox"/> | 0x00400020 | 0x00c0006f  | jal x0,0x0000000c     | 14: j endif          |
| <input type="checkbox"/> | 0x00400024 | 0xffff38393 | addi x7,x7,0xffffffff | 16: addi t2, t2, -1  |
| <input type="checkbox"/> | 0x00400028 | 0x01ce0e33  | add x28,x28,x28       | 17: add t3, t3, t3   |

Tóm lại:

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

Do  $(i \leq j)$  return false ( $i=5, j=3$ ) nên  $y_{\text{sau}}=3-1=2$  và  $z_{\text{sau}}=4*2=8$  (thỏa mãn với  $(x,y,z)$  ban đầu là  $(2,3,4)$ )

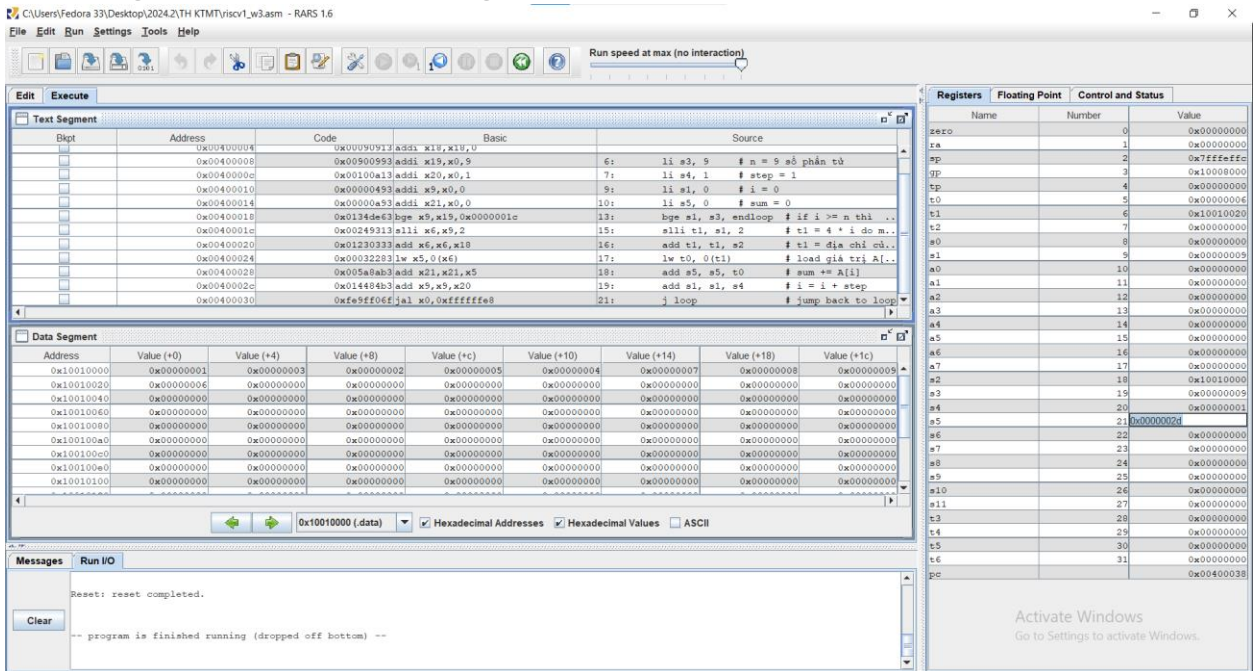
## 2. Assignment 2

```

1  .data
2      A: .word 1, 3, 2, 5, 4, 7, 8, 9, 6
3
4  .text
5      la s2, A          # lưu A vào s2
6      li s3, 9          # n = 9 số phần tử
7      li s4, 1          # step = 1
8
9      li s1, 0          # i = 0
10     li s5, 0          # sum = 0
11
12 loop:
13     bge s1, s3, endloop # if i >= n thì go to endloop
14
15     slli t1, s1, 2      # t1 = 4 * i do mỗi phần tử kiểu .word có size = 4byte
16     add t1, t1, s2      # t1 = địa chỉ của A[i]
17     lw t0, 0(t1)       # load giá trị A[i] vào t0
18     add s5, s5, t0      # sum += A[i]
19     add s1, s1, s4      # i = i + step
20
21     j loop              # jump back to loop
22
23 endloop:

```

- Nhập lệnh: (ở trang trước)
- Trạng thái các thanh ghi:



- Giải thích mã:

- Khởi tạo:  $s2 = \text{địa chỉ A}$ ,  $s1 = 0$ ,  $s3 = 9$  (số phần tử),  $s4 = 1$  (step),  $s5 = 0$  (sum).
- Vòng lặp
  - + Kiểm tra  $i \geq n$ , 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  $s1 = 0x09$  ( $\geq s3$ ), thu được tổng  $s5 = 0x2d$  (= 45 ở hệ thập phân) => thỏa mãn

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

```

1  .data
2      A: .word 1, 3, 5, 7
3
4  .text
5      la s2, A      # lưu A vào s2
6      li s3, 4      # n = 9 số phần tử
7      li s4, 1      # step = 1
8
9      li s1, 0      # i = 0
10     li s5, 0      # sum = 0
11
12 loop:
13     bge s1, s3, endloop # if i >= n thì go to endloop
14
15     slli t1, s1, 2    # t1 = 4 * i do mỗi phần tử kiểu .word có size = 4byte
16     add t1, t1, s2    # t1 = địa chỉ của A[i]
17     lw t0, 0(t1)     # load giá trị A[i] vào t0
18     add s5, s5, t0    # sum += A[i]
19     add s1, s1, s4    # i = i + step
20
21     j loop           # jump back to loop
22
23 endloop:

```

|    |    |            |
|----|----|------------|
| s5 | 21 | 0x00000010 |
|----|----|------------|

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

### 3. Assignment 3

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

```

1  # Laboratory Exercise 3, Home Assignment 3
2  .data
3      test: .word 0
4  .text
5      la s0, test # Nạp địa chỉ của biến test vào s0
6      lw s1, 0(s0) # Nạp giá trị của biến test vào s1
7      li t0, 0 # Nạp giá trị cần kiểm tra
8      li t1, 1 # Nạp giá trị cần kiểm tra
9      li t2, 2 # Nạp giá trị cần kiểm tra
10     li s2, 10 # Giả sử đã lưu a = 10 trong s2
11     li s3, 20 # Giả sử đã lưu b = 20 trong s3
12     beq s1, t0, case_0
13     beq s1, t1, case_1
14     beq s1, t2, case_2
15     j default
16 case_0:
17     addi s2, s2, 1 # a = a + 1
18     j continue
19 case_1:
20     sub s2, s2, t1 # a = a - 1
21     j continue
22 case_2:
23     add s3, s3, s3 # b = 2 * b
24     j continue
25 default:

```

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

C:\Users\Fedora 33\Desktop\2024\2\TH KTM\visrv1\_w3.asm - RARS 1.6

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Registers Floating Point Control and Status

| Name | Number | Value      |
|------|--------|------------|
| zero | 0      | 0x00000000 |
| ra   | 1      | 0x00000000 |
| sp   | 2      | 0x7ffffeff |
| gp   | 3      | 0x10008000 |
| tp   | 4      | 0x00000000 |
| t0   | 5      | 0x00000000 |
| t1   | 6      | 0x00000001 |
| t2   | 7      | 0x00000002 |
| a0   | 8      | 0x10010000 |
| a1   | 9      | 0x00000000 |
| a2   | 10     | 0x00000000 |
| a3   | 11     | 0x00000000 |
| a4   | 12     | 0x00000000 |
| a5   | 13     | 0x00000000 |
| a6   | 14     | 0x00000000 |
| a7   | 15     | 0x00000000 |
| a8   | 16     | 0x00000000 |
| a9   | 17     | 0x00000000 |
| a10  | 18     | 0x00000000 |
| a11  | 19     | 0x00000000 |
| a12  | 20     | 0x00000000 |
| a13  | 21     | 0x00000000 |
| a14  | 22     | 0x00000000 |
| a15  | 23     | 0x00000000 |
| a16  | 24     | 0x00000000 |
| a17  | 25     | 0x00000000 |
| a18  | 26     | 0x00000000 |
| a19  | 27     | 0x00000000 |
| a20  | 28     | 0x00000000 |
| a21  | 29     | 0x00000000 |
| a22  | 30     | 0x00000000 |
| a23  | 31     | 0x00000000 |
| pc   |        | 0x00400000 |

Messages Run I/O

Clear -- program is finished running (dropped off bottom) --

Activate Windows  
Go to Settings to activate Windows.

| 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 |

#### 4. Assignment 4

Khởi tạo bộ (x,y,z,i,j) = (2,3,4,5,3)

a.i < j

```

1  # Laboratory Exercise 3, Home Assignment 1
2  .text
3  start:
4      # Lần lượt khởi tạo x, y, z, i, j
5      addi t1, zero, 2
6      addi t2, zero, 3
7      addi t3, zero, 4
8      addi s1, zero, 5 #khởi tạo i
9      addi s2, zero, 3 #khởi tạo j
10     blt s1, s2, then #branch to then if i<j
11
12
13 else:
14     addi t2, t2, -1
15     add t3, t3, t3
16     j endif
17 then:
18     addi t1, t1, 1
19     addi t3, zero, 1
20 endif:
21

```

| Registers |        |            | Floating Point | Control and Status |
|-----------|--------|------------|----------------|--------------------|
| Name      | Number | Value      |                |                    |
| zero      | 0      | 0x00000000 |                |                    |
| ra        | 1      | 0x00000000 |                |                    |
| sp        | 2      | 0x7fffffc  |                |                    |
| gp        | 3      | 0x10008000 |                |                    |
| tp        | 4      | 0x00000000 |                |                    |
| t0        | 5      | 0x00000000 |                |                    |
| t1        | 6      | 0x00000002 |                |                    |
| t2        | 7      | 0x00000002 |                |                    |
| s0        | 8      | 0x00000000 |                |                    |
| s1        | 9      | 0x00000005 |                |                    |
| a0        | 10     | 0x00000000 |                |                    |
| a1        | 11     | 0x00000000 |                |                    |
| a2        | 12     | 0x00000000 |                |                    |
| a3        | 13     | 0x00000000 |                |                    |
| a4        | 14     | 0x00000000 |                |                    |
| a5        | 15     | 0x00000000 |                |                    |
| a6        | 16     | 0x00000000 |                |                    |
| a7        | 17     | 0x00000000 |                |                    |
| s2        | 18     | 0x00000003 |                |                    |
| s3        | 19     | 0x00000000 |                |                    |
| s4        | 20     | 0x00000000 |                |                    |
| s5        | 21     | 0x00000000 |                |                    |
| s6        | 22     | 0x00000000 |                |                    |
| s7        | 23     | 0x00000000 |                |                    |
| s8        | 24     | 0x00000000 |                |                    |
| s9        | 25     | 0x00000000 |                |                    |
| s10       | 26     | 0x00000000 |                |                    |
| s11       | 27     | 0x00000000 |                |                    |
| t3        | 28     | 0x00000008 |                |                    |
| t4        | 29     | 0x00000000 |                |                    |
| t5        | 30     | 0x00000000 |                |                    |
| t6        | 31     | 0x00000000 |                |                    |
| pc        |        | 0x00400030 |                |                    |

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

$b.i \geq j$

```

1 # Laboratory Exercise 3, Home Assignment 1
2 .text
3 start:
4     # Lần lượt khởi tạo x, y, z, i, j
5     addi t1, zero, 2
6     addi t2, zero, 3
7     addi t3, zero, 4
8     addi s1, zero, 5 #khởi tạo i
9     addi s2, zero, 3 #khởi tạo j
10    blt s1, s2, else #branch to "else" if !(i>=j) or (i<j)
11
12 then:
13     addi t1, t1, 1
14     addi t3, zero, 1
15     j endif
16 else:
17     addi t2, t2, -1
18     add t3, t3, t3
19 endif:
20

```

| Registers      |        |                    |
|----------------|--------|--------------------|
| Floating Point |        | Control and Status |
| Name           | Number | Value              |
| zero           | 0      | 0x00000000         |
| ra             | 1      | 0x00000000         |
| sp             | 2      | 0x7fffffe0         |
| gp             | 3      | 0x10008000         |
| tp             | 4      | 0x00000000         |
| t0             | 5      | 0x00000000         |
| t1             | 6      | 0x00000003         |
| t2             | 7      | 0x00000003         |
| s0             | 8      | 0x00000000         |
| s1             | 9      | 0x00000005         |
| a0             | 10     | 0x00000000         |
| a1             | 11     | 0x00000000         |
| a2             | 12     | 0x00000000         |
| a3             | 13     | 0x00000000         |
| a4             | 14     | 0x00000000         |
| a5             | 15     | 0x00000000         |
| a6             | 16     | 0x00000000         |
| a7             | 17     | 0x00000000         |
| s2             | 18     | 0x00000003         |
| s3             | 19     | 0x00000000         |
| s4             | 20     | 0x00000000         |
| s5             | 21     | 0x00000000         |
| s6             | 22     | 0x00000000         |
| s7             | 23     | 0x00000000         |
| s8             | 24     | 0x00000000         |
| s9             | 25     | 0x00000000         |
| s10            | 26     | 0x00000000         |
| s11            | 27     | 0x00000000         |
| t3             | 28     | 0x00000001         |
| t4             | 29     | 0x00000000         |
| t5             | 30     | 0x00000000         |
| t6             | 31     | 0x00000000         |
| pc             |        | 0x00400030         |

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

$c.i + j \leq 0$

```

1 # Laboratory Exercise 3, Home Assignment 1
2 .text
3 start:
4     # Lần lượt khởi tạo x, y, z, i, j
5     addi t1, zero, 2
6     addi t2, zero, 3
7     addi t3, zero, 4
8     addi s1, zero, 5 #khởi tạo i
9     addi s2, zero, 3 #khởi tạo j
10    add t0, s1, s2 # t0 = i + j
11    blt zero, t0, else # branch to "else" if !(i+j<=0) or i+j>0
12
13 then:
14     addi t1, t1, 1
15     addi t3, zero, 1
16     j endif
17 else:
18     addi t2, t2, -1
19     add t3, t3, t3
20
21 endif:
22

```

| Registers      |        |                    |
|----------------|--------|--------------------|
| Floating Point |        | Control and Status |
| Name           | Number | Value              |
| zero           | 0      | 0x00000000         |
| ra             | 1      | 0x00000000         |
| sp             | 2      | 0x7fffffe0         |
| gp             | 3      | 0x10008000         |
| tp             | 4      | 0x00000000         |
| t0             | 5      | 0x00000008         |
| t1             | 6      | 0x00000002         |
| t2             | 7      | 0x00000002         |
| s0             | 8      | 0x00000000         |
| s1             | 9      | 0x00000005         |
| a0             | 10     | 0x00000000         |
| a1             | 11     | 0x00000000         |
| a2             | 12     | 0x00000000         |
| a3             | 13     | 0x00000000         |
| a4             | 14     | 0x00000000         |
| a5             | 15     | 0x00000000         |
| a6             | 16     | 0x00000000         |
| a7             | 17     | 0x00000000         |
| s2             | 18     | 0x00000003         |
| s3             | 19     | 0x00000000         |
| s4             | 20     | 0x00000000         |
| s5             | 21     | 0x00000000         |
| s6             | 22     | 0x00000000         |
| s7             | 23     | 0x00000000         |
| s8             | 24     | 0x00000000         |
| s9             | 25     | 0x00000000         |
| s10            | 26     | 0x00000000         |
| s11            | 27     | 0x00000000         |
| t3             | 28     | 0x00000008         |
| t4             | 29     | 0x00000000         |
| t5             | 30     | 0x00000000         |
| t6             | 31     | 0x00000000         |
| pc             |        | 0x00400034         |

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



$d.i + j > m + n$

\*Khởi tạo  $(m,n)=(1,9)$

```

1  # Laboratory Exercise 3, Home Assignment 1
2  .text
3  start:
4      # Lần lượt khởi tạo x, y, z, i, j
5      addi t1, zero, 2
6      addi t2, zero, 3
7      addi t3, zero, 4
8      addi s1, zero, 5 #khởi tạo i
9      addi s2, zero, 3 #khởi tạo j
10     addi a0, zero, 1 #khởi tạo m = 1
11     addi a1, zero, 9 #khởi tạo n = 9
12     add a2, s1, s2 # a2 = i + j
13     add a3, a0, a1 # a3 = m + n
14     blt a3, a2, then # branch to "then" if m+n < i+j
15
16 else:
17     addi t2, t2, -1
18     add t3, t3, t3
19     j endif
20 then:
21     addi t1, t1, 1
22     addi t3, zero, 1
23
24 endif:

```

| Registers | Floating Point | Control and Status |
|-----------|----------------|--------------------|
| Name      | Number         | Value              |
| zero      | 0              | 0x00000000         |
| ra        | 1              | 0x00000000         |
| sp        | 2              | 0x7ffffc           |
| gp        | 3              | 0x10008000         |
| tp        | 4              | 0x00000000         |
| t0        | 5              | 0x00000000         |
| t1        | 6              | 0x00000002         |
| t2        | 7              | 0x00000002         |
| s0        | 8              | 0x00000000         |
| s1        | 9              | 0x00000005         |
| a0        | 10             | 0x00000001         |
| a1        | 11             | 0x00000009         |
| a2        | 12             | 0x00000008         |
| a3        | 13             | 0x0000000a         |
| a4        | 14             | 0x00000000         |
| a5        | 15             | 0x00000000         |
| a6        | 16             | 0x00000000         |
| a7        | 17             | 0x00000000         |
| s2        | 18             | 0x00000003         |
| s3        | 19             | 0x00000000         |
| s4        | 20             | 0x00000000         |
| s5        | 21             | 0x00000000         |
| s6        | 22             | 0x00000000         |
| s7        | 23             | 0x00000000         |
| s8        | 24             | 0x00000000         |
| s9        | 25             | 0x00000000         |
| s10       | 26             | 0x00000000         |
| s11       | 27             | 0x00000000         |
| t3        | 28             | 0x00000008         |
| t4        | 29             | 0x00000000         |
| t5        | 30             | 0x00000000         |
| t6        | 31             | 0x00000000         |
| pc        |                | 0x00400040         |

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

## 5. Assignment 5

$a.i < n$

```

1  .data
2      A: .word 1, 3, 2, 5, 4, 7, 8, 9, 10, 7, 3
3
4  .text
5      la s2, A
6      li s3, 9    #chỉ tính tổng 10 phần tử đầu tiên của array A
7      li s4, 1
8
9      li s1, 0
10     li s5, 0
11
12 loop:
13     bgt s1, s3, endloop
14
15     slli t1, s1, 2
16     add t1, t1, s2
17     lw t0, 0(t1)
18     add s5, s5, t0
19     add s1, s1, s4
20
21     j loop
22
23 endloop:
24

```

| Registers |        |            |
|-----------|--------|------------|
| Name      | Number | Value      |
| zero      | 0      | 0x00000000 |
| ra        | 1      | 0x00000000 |
| sp        | 2      | 0x7fffffc  |
| gp        | 3      | 0x10008000 |
| tp        | 4      | 0x00000000 |
| t0        | 5      | 0x00000007 |
| t1        | 6      | 0x10010024 |
| t2        | 7      | 0x00000000 |
| s0        | 8      | 0x00000000 |
| s1        | 9      | 0x0000000a |
| a0        | 10     | 0x00000000 |
| a1        | 11     | 0x00000000 |
| a2        | 12     | 0x00000000 |
| a3        | 13     | 0x00000000 |
| a4        | 14     | 0x00000000 |
| a5        | 15     | 0x00000000 |
| a6        | 16     | 0x00000000 |
| a7        | 17     | 0x00000000 |
| s2        | 18     | 0x10010000 |
| s3        | 19     | 0x00000009 |
| s4        | 20     | 0x00000001 |
| s5        | 21     | 0x00000038 |
| s6        | 22     | 0x00000000 |
| s7        | 23     | 0x00000000 |
| s8        | 24     | 0x00000000 |
| s9        | 25     | 0x00000000 |
| s10       | 26     | 0x00000000 |
| s11       | 27     | 0x00000000 |
| t3        | 28     | 0x00000000 |
| t4        | 29     | 0x00000000 |
| t5        | 30     | 0x00000000 |
| t6        | 31     | 0x00000000 |
| pc        |        | 0x00400038 |

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

b.  $\text{sum} < 0$

```

1  .data
2      A: .word 1, 3, -2, 5, -4, -7, -8, -9, 6
3
4  .text
5      la s2, A
6      li s3, 9
7      li s4, 1
8
9      li s1, 0
10     li s5, 0
11
12 loop:
13     blt s5, zero, endloop    # nếu sum < 0 thì endloop
14     slli t1, s1, 2
15     add t1, t1, s2
16     lw t0, 0(t1)
17     add s5, s5, t0
18     add s1, s1, s4
19
20     j loop
21
22 endloop:
23

```

| Registers |        |            |
|-----------|--------|------------|
| Name      | Number | Value      |
| zero      | 0      | 0x00000000 |
| ra        | 1      | 0x00000000 |
| sp        | 2      | 0x7fffffc  |
| gp        | 3      | 0x10008000 |
| tp        | 4      | 0x00000000 |
| t0        | 5      | 0xfffffff9 |
| t1        | 6      | 0x10010014 |
| t2        | 7      | 0x00000000 |
| s0        | 8      | 0x00000000 |
| s1        | 9      | 0x00000006 |
| a0        | 10     | 0x00000000 |
| a1        | 11     | 0x00000000 |
| a2        | 12     | 0x00000000 |
| a3        | 13     | 0x00000000 |
| a4        | 14     | 0x00000000 |
| a5        | 15     | 0x00000000 |
| a6        | 16     | 0x00000000 |
| a7        | 17     | 0x00000000 |
| s2        | 18     | 0x10010000 |
| s3        | 19     | 0x00000009 |
| s4        | 20     | 0x00000001 |
| s5        | 21     | 0xfffffff9 |
| s6        | 22     | 0x00000000 |
| s7        | 23     | 0x00000000 |
| s8        | 24     | 0x00000000 |
| s9        | 25     | 0x00000000 |
| s10       | 26     | 0x00000000 |
| s11       | 27     | 0x00000000 |
| t3        | 28     | 0x00000000 |
| t4        | 29     | 0x00000000 |
| t5        | 30     | 0x00000000 |
| t6        | 31     | 0x00000000 |
| pc        |        | 0x00400038 |

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

c.A[i] == 0

|    |   |  |  |
|----|---|--|--|
| 1  | .data   |  |  |
| 2  | A: .word 1, 3, -2, 5, 0, -7, -8, -9, 6        |  |  |
| 3  |   |  |  |
| 4  | .text   |  |  |
| 5  | la s2, A                                      |  |  |
| 6  | li s3, 9                                      |  |  |
| 7  | li s4, 1                                      |  |  |
| 8  |   |  |  |
| 9  | li s1, 0                                      |  |  |
| 10 | li s5, 0                                      |  |  |
| 11 |   |  |  |
| 12 | loop:   |  |  |
| 13 | slli t1, s1, 2                                |  |  |
| 14 | add t1, t1, s2                                |  |  |
| 15 | lw t0, 0(t1)                                  |  |  |
| 16 | beq t0, zero, endloop #nếu t0 = 0 thì endloop |  |  |
| 17 | add s5, s5, t0                                |  |  |
| 18 | add s1, s1, s4                                |  |  |
| 19 |   |  |  |
| 20 | j loop  |  |  |
| 21 |   |  |  |
| 22 | endloop:                                      |  |  |
| 23 |   |  |  |

| Registers | Floating Point | Control and Status |
|-----------|----------------|--------------------|
| Name      | Number         | Value              |
| zero      | 0              | 0x00000000         |
| ra        | 1              | 0x00000000         |
| sp        | 2              | 0x7ffffc           |
| gp        | 3              | 0x10008000         |
| tp        | 4              | 0x00000000         |
| t0        | 5              | 0x00000000         |
| t1        | 6              | 0x10010010         |
| t2        | 7              | 0x00000000         |
| s0        | 8              | 0x00000000         |
| s1        | 9              | 0x00000004         |
| a0        | 10             | 0x00000000         |
| a1        | 11             | 0x00000000         |
| a2        | 12             | 0x00000000         |
| a3        | 13             | 0x00000000         |
| a4        | 14             | 0x00000000         |
| a5        | 15             | 0x00000000         |
| a6        | 16             | 0x00000000         |
| a7        | 17             | 0x00000000         |
| s2        | 18             | 0x10010000         |
| s3        | 19             | 0x00000009         |
| s4        | 20             | 0x00000001         |
| s5        | 21             | 0x00000007         |
| s6        | 22             | 0x00000000         |
| s7        | 23             | 0x00000000         |
| s8        | 24             | 0x00000000         |
| s9        | 25             | 0x00000000         |
| s10       | 26             | 0x00000000         |
| s11       | 27             | 0x00000000         |
| t3        | 28             | 0x00000000         |
| t4        | 29             | 0x00000000         |
| t5        | 30             | 0x00000000         |
| t6        | 31             | 0x00000000         |
| pc        |                | 0x00400038         |

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

## 6. 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:

```

| Registers | Floating Point | Control and Status |
|-----------|----------------|--------------------|
| Name      | Number         | Value              |
| zero      | 0              | 0x00000000         |
| ra        | 1              | 0x00000000         |
| sp        | 2              | 0x7ffffefc         |
| gp        | 3              | 0x10008000         |
| tp        | 4              | 0x00000000         |
| t0        | 5              | 0x10010018         |
| t1        | 6              | 0x00000005         |
| t2        | 7              | 0x00000000         |
| s0        | 8              | 0x00000000         |
| s1        | 9              | 0x00000000         |
| a0        | 10             | 0x10010000         |
| a1        | 11             | 0x00000007         |
| a2        | 12             | 0x00000007         |
| a3        | 13             | 0x00000032         |
| a4        | 14             | 0x00000000         |
| a5        | 15             | 0x00000000         |
| a6        | 16             | 0x00000000         |
| a7        | 17             | 0x00000000         |
| s2        | 18             | 0x00000000         |
| s3        | 19             | 0x00000000         |
| s4        | 20             | 0x00000000         |
| s5        | 21             | 0x00000000         |
| s6        | 22             | 0x00000000         |
| s7        | 23             | 0x00000000         |
| s8        | 24             | 0x00000000         |
| s9        | 25             | 0x00000000         |
| s10       | 26             | 0x00000000         |
| s11       | 27             | 0x00000000         |
| t3        | 28             | 0x00000000         |
| t4        | 29             | 0x00000000         |
| t5        | 30             | 0x00000000         |
| t6        | 31             | 0x00000000         |
| pc        |                | 0x0040004c         |

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