**REPORT WEEK 3**

Họ và tên: Nguyễn Văn Hưng

MSSV: 20225634

Lớp: Thực hành kiến trúc máy tính – Mã lớp: 147799

1. **Assignment 1:**
2. Đề bài: Create a new project to implement the code in Home Assignment 1. Initialize for i and j variable. Compile and upload to the simulator. Run this program step by step, observe the changing of memory and the content of registers at each step.
3. Code:

#Laboratory Exercise 3, Home Assignment 1

.text

addi $s1, $zero, 3 # i = 3

addi $s2, $zero, 5 # j = 5

addi $t1, $zero, 2004 # x = 2004

addi $t2, $zero, 2022 # y = 2022

addi $t3, $zero, 5634 # z = 5634

start:

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

bne $t0,$zero,else # branch to else if j<i

addi $t1,$t1,1 # then part: x=x+1

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

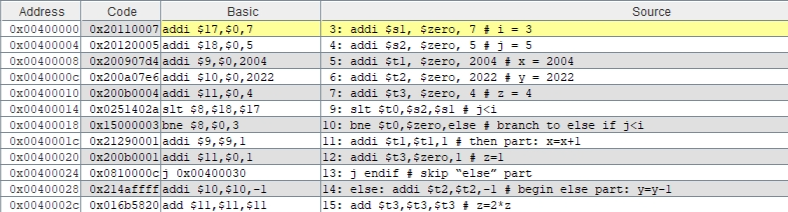
j endif # skip “else” part

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

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

endif:

1. Text segment:



1. Kết quả:
2. Trường hợp 1: Với i = 3, j = 5 (x = 2004,y = 2022,z = 5634)

* **addi $s1, $zero, 7**: Khởi tạo thanh ghi **$s1** với giá trị 7, biểu thị cho **i**.
* **addi $s2, $zero, 5**: Khởi tạo thanh ghi **$s2** với giá trị 5, biểu thị cho **j**.
* **addi $t1, $zero, 2004**: Khởi tạo thanh ghi **$t1** với giá trị 2004, biểu thị cho **x**.
* **addi $t2, $zero, 2022**: Khởi tạo thanh ghi **$t2** với giá trị 2022, biểu thị cho **y**.
* **addi $t3, $zero, 4**: Khởi tạo thanh ghi **$t3** với giá trị 4, biểu thị cho **z**.

Vòng lặp bắt đầu:

* + - **slt $t0,$s2,$s1**: Thiết lập **$t0** thành 1 nếu **j < i**, ngược lại thiết lập thành 0.
    - **bne $t0,$zero,else**: Nhảy tới nhãn **else** nếu **$t0** không bằng 0, điều này có nghĩa là **j** nhỏ hơn **i**.
    - **addi $t1,$t1,1**: Nếu **j < i**, tăng **x** lên 1.
    - **addi $t3,$zero,1**: Đặt **z** thành 1. Lưu ý rằng điều này luôn được thực hiện vì không có lệnh nhảy hoặc nhánh giữa dòng này và **endif**.
    - **j endif**: Nhảy tới nhãn **endif**, bỏ qua phần **else**.

Nếu nhánh được thực hiện (tức là nếu **j** nhỏ hơn **i**), phần **else** sẽ không được thực hiện và **z** vẫn giữ nguyên là 1. Ngược lại, phần **else** sẽ được thực hiện:

* + - **addi $t2,$t2,-1**: Giảm **y** đi 1.
    - **add $t3,$t3,$t3**: Nhân đôi giá trị của **z**, hiệu quả là nhân nó với 2.

Cuối cùng, nhãn **endif** được sử dụng như một đích cho nhánh có điều kiện và việc thực thi tiếp tục từ đó.

* Tóm lại, mã này tăng **x** lên 1 nếu **j** nhỏ hơn **i**, đặt **z** thành 1,giảm **y** đi 1 nếu **j** không nhỏ hơn **i**, và nhân đôi **z** nếu **j** không nhỏ hơn **i**

Nên kết quả sẽ là x = x +1 = 2005, y = 2022, z = 1



1. Trường hợp 2: Với i = 7, j = 5 (x = 2004,y = 2022,z = 5634)

* Tương tự trường hợp 1
* Vì i > j nên kết quả sẽ là x = 2004, y = y-1 = 2022, z = 11268 

1. **Assignment 2:**
2. Đề bài: Create a new project implementing the code in Home Assignment 2. Initialize for i, n, step, sum variables and array A. Compile and upload to the simulator. Run this program step by step, observe the changing of memory and the content of registers by each step. Try to test with some more cases (change the value of variables).
3. Code:

#Laboratory 3, Home Assigment 2

.data

A: .word 4,3,2,1,5,6

.text

la $s2, A

addi $s5, $zero, 0 # sum = 0

addi $s1, $zero, 0 # i = 0

addi $s4, $zero, 1 # step = 1

addi $s3, $zero, 6 # n = 3

loop: slt $t2, $s1, $s3 # $t2 = i < n ? 1 : 0

beq $t2, $zero, endloop

add $t1, $s1, $s1 # $t1 = 2 \* $s1

add $t1, $t1, $t1 # $t1 = 4 \* $s1

add $t1, $t1, $s2 # $t1 store the address of A[i]

lw $t0, 0($t1) # load value of A[i] in $t0

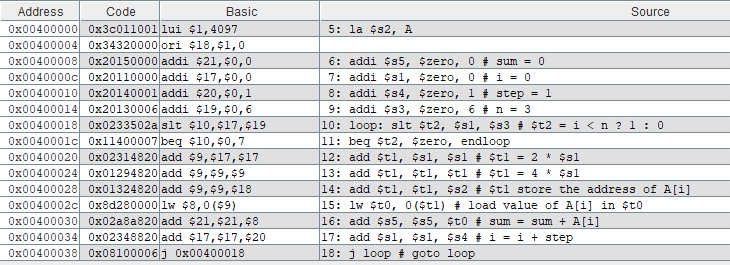
add $s5, $s5, $t0 # sum = sum + A[i]

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

j loop # goto loop

endloop:

1. Text segment:



1. Kết quả:

* **la $s2, A**: Đây là lệnh để nạp địa chỉ của mảng A vào thanh ghi $s2.
* **addi $s5, $zero, 0**: Khởi tạo biến sum bằng 0.
* **addi $s1, $zero, 0**: Khởi tạo biến i bằng 0.
* **addi $s4, $zero, 1**: Khởi tạo biến step bằng 1.
* **addi $s3, $zero, 6**: Khởi tạo biến n bằng 6.

Sau đó là vòng lặp:

* **slt $t2, $s1, $s3**: So sánh giá trị của biến i và biến n. Nếu i < n thì $t2 được set thành 1, ngược lại set thành 0.
* **beq $t2, $zero, endloop**: Nếu $t2 = 0, tức là i >= n, chương trình nhảy tới nhãn **endloop**, kết thúc vòng lặp.

Trong vòng lặp:

* **add $t1, $s1, $s1**: $t1 = 2 \* $s1
* **add $t1, $t1, $t1**: $t1 = 4 \* $s1
* **add $t1, $t1, $s2**: $t1 store địa chỉ của A[i]
* **lw $t0, 0($t1)**: Load giá trị của A[i] vào $t0.
* **add $s5, $s5, $t0**: Cộng giá trị của A[i] vào biến sum.
* **add $s1, $s1, $s4**: Tăng biến i thêm step.
* **j loop**: Nhảy lại đầu vòng lặp.

Cuối cùng, khi vòng lặp kết thúc, **sum** chứa tổng các phần tử trong mảng A.

* Giá trị của A là [4, 3, 2, 1, 5, 6]. Tổng của tất cả các phần tử trong mảng này là 4 + 3 + 2 + 1 + 5 + 6 = 21.



1. **Assignment 3:**
2. Đề bài: Create a new project implementing the code in Home Assignment 3. Compile and upload to the simulator. Run this program step by step; observe the changing of memory and the content of registers by each step. Change the value of test variable and run this program some times to check all cases.
3. Code:

#Laboratory Exercise 3, Home Assignment 3

.data

test: .word 1

.text

la $s0, test #load the address of test variable

lw $s1, 0($s0) #load the value of test to register $t1

addi $s2, $zero, 7

addi $s3, $zero, 5

li $t0, 0 #load value for test case

li $t1, 1

li $t2, 2

beq $s1, $t0, case\_0

beq $s1, $t1, case\_1

beq $s1, $t2, case\_2

j default

case\_0: addi $s2, $s2, 1 #a=a+1

j continue

case\_1: sub $s2, $s2, $t1 #a=a-1

j continue

case\_2: add $s3, $s3, $s3 #b=2\*b

j continue

default:

continue:

1. Text segment:

A screenshot of a computer

Description automatically generated

1. Kết quả:
2. Trường hợp 1:

* **.data**: Đây là phần khai báo dữ liệu.
* **test: .word 1**: Khai báo biến **test** với giá trị khởi tạo là 1.
* **.text**: Đây là phần của mã để thực thi.
* **la $s0, test**: Nạp địa chỉ của biến **test** vào thanh ghi **$s0**.
* **lw $s1, 0($s0)**: Load giá trị của biến **test** vào thanh ghi **$s1**.
* **addi $s2, $zero, 7**: Khởi tạo thanh ghi **$s2** với giá trị 7.
* **addi $s3, $zero, 5**: Khởi tạo thanh ghi **$s3** với giá trị 5.
* **li $t0, 0**: Load giá trị 0 vào thanh ghi **$t0**.
* **li $t1, 1**: Load giá trị 1 vào thanh ghi **$t1**.
* **li $t2, 2**: Load giá trị 2 vào thanh ghi **$t2**.
* **beq $s1, $t0, case\_0**: Nếu giá trị của **test** bằng 0, nhảy tới nhãn **case\_0**.
* **beq $s1, $t1, case\_1**: Nếu giá trị của **test** bằng 1, nhảy tới nhãn **case\_1**.
* **beq $s1, $t2, case\_2**: Nếu giá trị của **test** bằng 2, nhảy tới nhãn **case\_2**.
* **j default**: Nếu không thỏa mãn bất kỳ điều kiện nào, nhảy tới nhãn **default**.

Và giờ chúng ta đi qua từng nhãn:

* **case\_0**: Nếu giá trị của **test** bằng 0, thực hiện:

**addi $s2, $s2, 1**: Tăng giá trị của **$s2** lên 1 ( **$s2 = $s2 + 1**).

**j continue**: Nhảy tới nhãn **continue**.

* **case\_1**: Nếu giá trị của **test** bằng 1, thực hiện:

**sub $s2, $s2, $t1**: Giảm giá trị của **$s2** đi 1 (**$s2 = $s2 - 1**).

**j continue**: Nhảy tới nhãn **continue**.

* **case\_2**: Nếu giá trị của **test** bằng 2, thực hiện:

**add $s3, $s3, $s3**: Nhân giá trị của **$s3** với 2 (**$s3 = 2 \* $s3**).

**j continue**: Nhảy tới nhãn **continue**.

* **default**: Nếu không thỏa mãn bất kỳ điều kiện nào, không có thực hiện gì cả, và chương trình tiếp tục từ nhãn **continue**.

Vì **test** ban đầu có giá trị là 1, điều kiện thỏa mãn trong **case\_1** được thực hiện, và giá trị của **s2** sẽ giảm đi 1, trở thành 6. 

1. Trường hợp 2:

Vì **test** ban đầu có giá trị là 0, điều kiện thỏa mãn trong **case\_0** được thực hiện, và giá trị của **s2** sẽ tăng lên 1, trở thành 8. 

1. Trường hợp 3:

Vì **test** ban đầu có giá trị là 2, điều kiện thỏa mãn trong **case\_2** được thực hiện, và giá trị của **s3** sẽ nhân 2, trở thành 10.



1. **Assignment 4:**
2. Đề bài: Modify the Assignment 1, so that the condition tested is
3. i < j
4. i >= j
5. i+j <= 0
6. d. i+j > m+n (m and n stored in other registers)
7. Với i < j

* Code:

#Laboratory Exercise 3, Home Assignment 4

.text

addi $s1, $zero, 3 # i = 3

addi $s2, $zero, 5 # j = 5

addi $t1, $zero, 2004 # x = 2004

addi $t2, $zero, 2024 # y = 2024

addi $t3, $zero, 4 # z = 4

start:

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

beq $t0,$zero,else # branch to else if i<j

addi $t1,$t1,1 # then part: x=x+1

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

j endif # skip “else” part

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

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

endif:

* Kết quả:
  + Với i < j (i=3, j=5) thì x=x+1= 2004+1=2005, z=1
  + Với i = j (i=5, j=5) thì y=y-1= 2022-1=2005, z=2z=11268 Với i > j (i=5, j=5) thì y=y-1= 2022-1=2005, z=2z=11268

1. Với i >= j

* Code:

addi $s1, $0, 2004

addi $s2, $0, 2022

addi $s4 , $0, 5634

start:# s1 = i va s2 = j

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

bne $t0, $s4, else# if t0 == 1 -> else

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

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

j endif

else:

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

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

endif:

* Kết quả:
  + Với i > j thì x=x+1= 2004+1=2005, z=1
  + Với i = j thì y=y-1= 2004+1=2005, z=1 Với i < j thì y=y-1= 2022-1=2005, z=2z=11268

1. Với i +j <= 0

* Code:

.text

addi $s1, $zero, 3 # i = 3

addi $s2, $zero, 5 # j = 5

addi $t1, $zero, 2004 # x = 2004

addi $t2, $zero, 2024 # y = 2024

addi $t3, $zero, 4 # z = 5634

start:# s1 = i va s2 = j

add $s4, $s1, $s2 # s4 = i+j

slt $t0, $zero, $s4 # i+j> 0->1

bne $t0, $t4, else# if t0 == 1 -> else

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

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

j endif

else:

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

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

endif:

* Kết quả:
* Với i+j <= 0 thì x=x+1= 2004+1=2005, z=1
* Với i+j > 0 thì y=y-1= 2022-1=2005, z=2z=11268

1. Với i + j > m+n

* Code:

.text

addi $s1, $zero, 3 # i = 3

addi $s2, $zero, 5 # j = 5

addi $t1, $zero, 2004 # x = 2004

addi $t2, $zero, 2024 # y = 2024

addi $t3, $zero, 4 # z = 5634

start:# s1 = i va s2 = j  
  #s3 = m , s4 = n  
  add $t4 , $s3, $s4 # t4 = m+n  
  add $t5 , $s1, $s2 # t5 = i+j  
  slt $t0, $t4, $t5# m+n < i+j -> 1  
  bne $t0, $zero, else#  if t0 == 1 -> else  
  addi $t1, $t1,1 # t1 += 1  
  addi $t3, $zero, 1# t3 = z = 1  
  j endif  
else:  
  addi $t2, $t2, -1# t2 = y -= 1  
  add $t3, $t3, $t3# z =  t3 = 2t3  
endif:

* Kết quả:
* Với i+j > m+n thì x=x+1= 2004+1=2005, z=1
* Với i+j <= m+n thì y=y-1= 2022-1=2005, z=2z=11268

1. **Assignment 5:**
2. Đề bài: Modify the Assignment 2, so that the condition tested to keep the loop running is

a. i <= n

b. sum >= 0

c. A[i] != 0

1. Với i<=n

* Code

.data

A: .word 4,3,2,1,5,6,7

.text

la $s2, A

addi $s5, $zero, 0 # sum = 0

addi $s1, $zero, 0 # i = 0

addi $s4, $zero, 1 # step = 1

addi $s3, $zero, 6 # n = 3

loop: slt $t2, $s3, $s1

bne $t2, $zero, endloop

add $t1, $s1, $s1 # $t1 = 2 \* $s1

add $t1, $t1, $t1 # $t1 = 4 \* $s1

add $t1, $t1, $s2 # $t1 store the address of A[i]

lw $t0, 0($t1) # load value of A[i] in $t0

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

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

j loop # goto loop

endloop:

* Kết quả

Vì i<=n nên sum = 4+3+2+1+5+6+7 (từ A[0] đến A[6] là 7 phần tử )



1. Với sum>=0

* Code

.data

A: .word 4,3,2,1,5,6,7

.text

la $s2, A

addi $s5, $zero, 0 # sum = 0

addi $s1, $zero, 0 # i = 0

addi $s4, $zero, 1 # step = 1

addi $s3, $zero, 6 # n = 3

loop: slt $t2, $s1, $s3

bne $t2, $zero, endloop

add $t1, $s1, $s1 # $t1 = 2 \* $s1

add $t1, $t1, $t1 # $t1 = 4 \* $s1

add $t1, $t1, $s2 # $t1 store the address of A[i]

lw $t0, 0($t1) # load value of A[i] in $t0

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

slt $t7,$s5, $zero# t7= 1; t7 = 0   
bne $t7,$zero, endloop

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

j loop # goto loop

endloop:

* Kết quả

Vì i<=n nên sum = 4+3+2+1+5+6+7 (từ A[0] đến A[6] là 7 phần tử do sum không âm)



1. Với A[i] != 0

* Code

.data

A: .word 4,3,2,1,5,6,7

.text

la $s2, A

addi $s5, $zero, 0 # sum = 0

addi $s1, $zero, 0 # i = 0

addi $s4, $zero, 1 # step = 1

addi $s3, $zero, 6 # n = 3

loop: slt $t2, $s1, $s3

bne $t2, $zero, endloop

add $t1, $s1, $s1 # $t1 = 2 \* $s1

add $t1, $t1, $t1 # $t1 = 4 \* $s1

add $t1, $t1, $s2 # $t1 store the address of A[i]

lw $t0, 0($t1) # load value of A[i] in $t0

lw $t0, 0($t1)# t0 = A[i]  
beq $t0, $zero, endloop

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

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

j loop # goto loop

endloop:

* Kết quả

Vì i<=n nên sum = 4+3+2+1+5+6+7 (từ A[0] đến A[6] là 7 phần tử do không phần tử nào âm)



1. **Assignment 6:**
2. Đề bài: Using all of above instructions and statements, create a new project to implement this function: find the element with the largest absolute value in a list of integers. Assuming that this list is store in an integer array and we know the number of elements in it.
3. Code

.data

array: .word 1, -3, 5, -7, 9 # Example array

array\_size: .word 5 # Number of elements in the array

.text

main:

lw $t0, array\_size # Load the number of elements in the array

lw $t1, array # Load the address of the array

lw $t2, 0($t1) # Load the first element of the array

abs $t3, $t2 # Calculate the absolute value of the first element

move $t4, $t2 # Initialize $t4 with the value of the first element

loop:

addi $t0, $t0, -1 # Decrement the counter

beq $t0, $zero, end\_loop # If counter is zero, exit loop

lw $t1, 4($t1) # Load the next element of the array

abs $t5, $t1 # Calculate the absolute value of the current element

bgt $t5, $t3, update\_max # If current absolute value is greater, update max

j loop # Continue looping

update\_max:

move $t3, $t5 # Update max absolute value

move $t4, $t1 # Update the element with largest absolute value

j loop # Continue looping

end\_loop:

# At this point, $t4 contains the element with the largest absolute value

# You can perform any further operations with $t4 here

# Custom implementation of absolute value calculation

abs:

bltz $a0, negative # If the value is negative, branch to negative

move $v0, $a0 # If positive, return the same value

j abs\_end

negative:

sub $v0, $zero, $a0 # If negative, return the negation of the value

abs\_end:

jr $ra # Return to the calling function