Báo cáo thực hành môn Kiến trúc máy tính

Họ tên: Nguyễn Thanh Hưng

MSSV: 20225633

1. Assignment 1

Code:

#Laboratory Exercise 3, Home Assignment 1

start:

addi $s1, $zero, 3 # initialize i to 3

addi $s2, $zero, 5 # initialize j to 5

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:

Giải thích:

* 2 dòng lệnh addi thực hiện việc gán giá trị $s1 = 3 và $s2 = 5

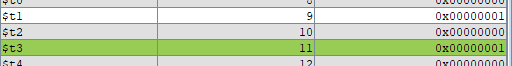
A green and white rectangle

Description automatically generated

* slt $t0, $s2, $s1: So sánh $s1 với s3. Nếu i nhỏ hơn n, $t0 sẽ bằng 1, ngược lại bằng 0.
* beq $t0, $zero, endloop: Nếu $s1 không nhỏ hơn $t0, tức là đã duyệt hết mảng, chương trình sẽ nhảy đến nhãn else và thực hiện các lệnh có trong else.

Việc thực hiện 2 lệnh này sẽ không thay đổi giá trị của bất cứ thanh ghi nào mà chỉ có thanh ghi pc các bước mà thôi (1 bước = 4 bytes)

* 2 lệnh add thực hiện việc tính kết quả



* Lệnh j endif nhảy đến nhãn endif sau khi thực hiện xong các lệnh và kết thúc chương trình.

1. Assignment 2

Code:

.data

A: .word 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

size: .word 10

sum: .word 0

.text

main:

    la $t0, A         # Load the address of array A into $t0

    lw $t1, size      # Load the size of array A into $t1

    li $t2, 0         # Initialize the sum to 0

loop:

    beq $t1, $zero, end   # If size is 0, exit the loop

    lw $t3, 0($t0)        # Load the current element of array A into $t3

    add $t2, $t2, $t3    # Add the current element to the sum

    addi $t0, $t0, 4     # Move to the next element of array A

    addi $t1, $t1, -1    # Decrement the size counter

    j loop

end:

    sw $t2, sum      # Store the sum in the sum variable

Giải thích các câu lệnh có trong vòng lặp:

* Lệnh beq $t1, $zero, end có tác dụng kiểm tra xem nếu độ dài mảng là 0 thì sẽ nhảy tới nhãn end là kết thúc vòng lặp
* Lệnh lw $t3, 0($t0) và add $t2, $t2, $t3 thực hiện lần lượt việc load giá trị hiện tại là A[0] và $t3 và thực hiện phép cộng vào sum

A green and grey stripes

Description automatically generated

* Lệnh addi $t0, $t0, 4 thực hiện việc nhảy tới giá trị tiếp theo là A[1] sau khi thực hiện phép cộng (nhảy 4 bytes)



* Lệnh addi $t1, $t1, -1 thực hiện việc giảm các bước của vòng lặp từ 10 bước xuống 9 bước



* Vòng lặp được kết thúc khi giá trị $t1 = 0 và lưu kết quả sum vào thanh ghi $t2



* Kết quả ở đây: 0x37 = 3 \* 16 + 7 = 55 bằng tổng từ 1 đến 10.

1. Assignment 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

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:

Giải thích:

* Lệnh la $s0, test lưu địa chỉ của biến test vào $s0 và lw $s1, 0($s0) lưu giá trị của test là 1 vào $s1



* 3 lệnh li lưu các giá trị đại diện cho các case vào các thanh ghi

A green and white rectangle with a white stripe

Description automatically generated

* 3 lệnh beq tiếp theo sẽ kiểm tra $s1 sẽ thuộc vào case nào dựa theo các giá trị đã lưu ở $t0, $t1, $t2 ở đây lệnh case 1 sẽ được thực hiện và thực hiện các lệnh có trong case 1



1. Assignment 4

a, i > j

#Laboratory Exercise 3, Home Assignment 1

start:

addi $s1, $zero, 5 # initialize i to 5

addi $s2, $zero, 3 # initialize j to 3

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

bne $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:

b, i >= j

#Laboratory Exercise 3, Home Assignment 1

start:

addi $s1, $zero, 6 # initialize i to 6

addi $s2, $zero, 6 # initialize j to 6

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:

c, i + j <= 0

#Laboratory Exercise 3, Home Assignment 1

start:

addi $s1, $zero, 5 # initialize i to 5

addi $s2, $zero, -6 # initialize j to -6

add $t4, $s1, $s2 # i + j

blez $t4, else # branch to else if i + j <= 0

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:

* Lệnh blez có tác dụng so sánh 1 số > 0, nếu không nhảy tới nhãn của nó

d, i + j > m + n

#Laboratory Exercise 3, Home Assignment 1

start:

addi $s1, $zero, 3 # initialize i to 3

addi $s2, $zero, 5 # initialize j to 5

addi $s3, $zero, 1 # initialize m to 1

addi $s4, $zero, 2 # initialize n to 2

add $t4, $s1, $s2 # i + j

add $t5, $s3, $s4 # m + n

bgt $t5, $t4, else # branch to else if i + j > m + n

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. Assignment 5

a, i < n

.data

A: .word 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

size: .word 10

sum: .word 0

.text

main:

    li $t2, 0         # Initialize the sum to 0

    li $t1, 0         # Initialize the counter i to 0

loop:

    lw $t0, size      # Load the size of array A into $t0

    slt $t4, $t1, $t0   # Check if i < size

    beq $t4, $zero, end   # If i >= size, exit the loop

    lw $t3, A($t1)        # Load the current element of array A into $t3

    add $t2, $t2, $t3    # Add the current element to the sum

    addi $t1, $t1, 4     # Increment the counter i by 1

    j loop

end:

    sw $t2, sum      # Store the sum in the sum variable

b, i <= n

#Laboratory 3, Home Assigment 2

.data

    n: .word 3

    step: .word 1

    A: .word 1,9,5  #Load array

.text

    #Load n, step, starting address A[i]

    la $t7, n

    la $t8, step

    lw $s3, 0($t7)

    lw $s4, 0($t8)

    la $s2, A

    addi $s5, $zero, 0  # sum = 0

    addi $s1, $zero, 0  # i = 0

    loop:

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

    beq $t2, 1, endloop # i <= n

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

    add $t1, $t1, $t1   # $t1 = 4 \* $s1 - 4 byte word

    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:

c, sum >= 0

.data

A: .word 1, 2, 3, 4, 5, 6, 7, 8, 9, -1000

size: .word 10

sum: .word 0

.text

main:

    la $t0, A         # Load the address of array A into $t0

    lw $t1, size      # Load the size of array A into $t1

    li $t2, 0         # Initialize the sum to 0

loop:

    lw $t3, 0($t0)        # Load the current element of array A into $t3

    add $t2, $t2, $t3    # Add the current element to the sum

    addi $t0, $t0, 4     # Move to the next element of array A

    bltz $t3, end        # If the current element is negative, exit the loop

    j loop

end:

    sw $t2, sum      # Store the sum in memory

* Lệnh bltz thực hiện việc kiểm tra xem một giá trị có phải là số âm hay không, nếu đúng sẽ nhảy tới nhãn của nó.

d, A[i] == 0

.data

A: .word 1, 2, 3, 4, 5, 0, 7, 8, 9, 10

size: .word 10

sum: .word 0

.text

main:

    la $t0, A         # Load the address of array A into $t0

    lw $t1, size      # Load the size of array A into $t1

    li $t2, 0         # Initialize the sum to 0

loop:

    beq $t1, $zero, end   # If size is 0, exit the loop

    lw $t3, 0($t0)        # Load the current element of array A into $t3

    beq $t3, $zero, end   # If A[i] == 0, exit the loop

    add $t2, $t2, $t3    # Add the current element to the sum

    addi $t0, $t0, 4     # Move to the next element of array A

    addi $t1, $t1, -1    # Decrement the size counter

    j loop

end:

    sw $t2, sum      # Store the sum in the sum variable

1. Assignment 6

.data

array: .word 1, -5, 3, -7, 2, -4, 6, -8, 9, -1

size: .word 10

.text

main:

    la $t0, array     # Load the base address of the array

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

    li $t2, 0         # Initialize the maximum absolute value to 0

    li $t3, 0         # Initialize the index of the maximum absolute value to 0

loop:

    beq $t1, $zero, end   # If the size is 0, exit the loop

    lw $t4, 0($t0)        # Load the current element

    abs $t4, $t4         # Take the absolute value of the current element

    bgt $t4, $t2, update  # If the absolute value is greater than the maximum absolute value, update the maximum

    addi $t0, $t0, 4     # Move to the next element

    addi $t1, $t1, -1    # Decrease the size

    j loop

update:

    move $t2, $t4        # Update the maximum absolute value

    sub $t3, $t3, $t1    # Update the index of the maximum absolute value

    j loop

end:

Kết quả:

