

Laboratory 3

Họ tên: Vũ Quốc Bảo
MSSV: 20225694

Assignment 1:

```
.data
.text
main:
li $s1,4 # i=4
li $s2,5 # j=5
li $t1,1 # x=1
li $t2,2 # y=2
li $t3,3 # z=3
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:\Users\W\Documents\2023.2\mips2.asm - MARS 4.5

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Edit Execute

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x24120005	addiu \$t3,\$0,0x00000005	4: li \$s2,4 # i=4
	0x00400004	0x24120005	addiu \$t2,\$0,0x00000005	5: li \$s2,5 # j=5
	0x00400008	0x24090001	addiu \$t1,\$0,0x00000001	6: li \$t1,1 # x=1
	0x0040000c	0x240a0002	addiu \$t2,\$0,0x00000002	7: li \$t2,2 # y=2
	0x00400010	0x240b0003	addiu \$t3,\$0,0x00000003	8: li \$t3,3 # z=3
	0x00400014	0x0251402a	slt \$t0,\$t2,\$t1	9: slt \$t0,\$t2,\$t1 # j<i
	0x00400018	0x015b0001	bne \$t0,\$zero,else	10: bne \$t0,\$zero,else # branch to else if j<i
	0x0040001c	0x21290001	addi \$t1,\$t1,1	11: addi \$t1,\$t1,1 # then part: x=x+1
	0x00400020	0x200b0001	addi \$t3,\$zero,1	12: addi \$t3,\$zero,1 # z=1
	0x00400024	0x00812000	j endif	13: j endif # skip "else" part
	0x00400028	0x21a4ffff	addi \$t2,\$t2,-1	15: addi \$t2,\$t2,-1 # begin else part: y=y-1
	0x0040002c	0x014b5220	add \$t3,\$t3,\$t3	16: add \$t3,\$t3,\$t3 # z=2*z

Labels

Label	Address
mips2.asm	0x00400000
main	0x00400000
else	0x00400028
endif	0x00400030

Registers

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x00000000
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0x00000000
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x00000000
\$t1	9	0x00000001
\$t2	10	0x00000002
\$t3	11	0x00000003
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x00000000
\$s1	17	0x00000004
\$s2	18	0x00000005
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$s6	22	0x00000000
\$s7	23	0x00000000
\$s8	24	0x00000000
\$s9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$k2	28	0x00000000
\$k3	29	0x00000000
\$k4	30	0x00000000
\$k5	31	0x00000000
\$ra		0x00000000
\$pc		0x00400030
\$hi		0x00000000
\$lo		0x00000000

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010004	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010008	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x1001000c	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010010	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010014	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010018	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x1001001c	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010024	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010028	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x1001002c	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010030	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010034	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010038	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x1001003c	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010044	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010048	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x1001004c	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010050	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010054	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010058	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x1001005c	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

Mars Messages Run I/O

program is finished running (dropped off bottom) --

Reset: reset completed.

Clear

program is finished running (dropped off bottom) --

program is finished running (dropped off bottom) --

67% Party sunny

Search

8:27 AM 3/7/2024

- Khởi tạo giá trị: $i=4, j=5$

Ta cho $x=1, y=2, z=3$

- Lệnh slt so sánh 2 thanh ghi \$s1 và \$s2 tương ứng. Khi đó, nếu $s2 < s1$ thì $t0 = 1$, còn nếu không thì $t0 = 0$.

- Lệnh bne so sánh thanh ghi \$t0 với thanh ghi 0, nếu 2 thanh ghi khác nhau thì sẽ nhảy sang phần else. Còn không thì sẽ tiếp tục phần dưới. Do ở đây $j > i$ nên sẽ tiếp tục phần dưới.

- $x = x + 1$ nên x bằng 2

- y không thay đổi.

- $z = 1$.

Assignment 2:

.data

A: .word 2,4,6,8,10 # mảng A

n: .word 5 # số lượng phần tử

step: .word 1 # giá trị bước

.text

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

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

la \$s2, A # Địa chỉ bắt đầu của mảng A

lw \$s3, n # Số lượng phần tử trong mảng

lw \$s4, step # Giá trị bước

loop:

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

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

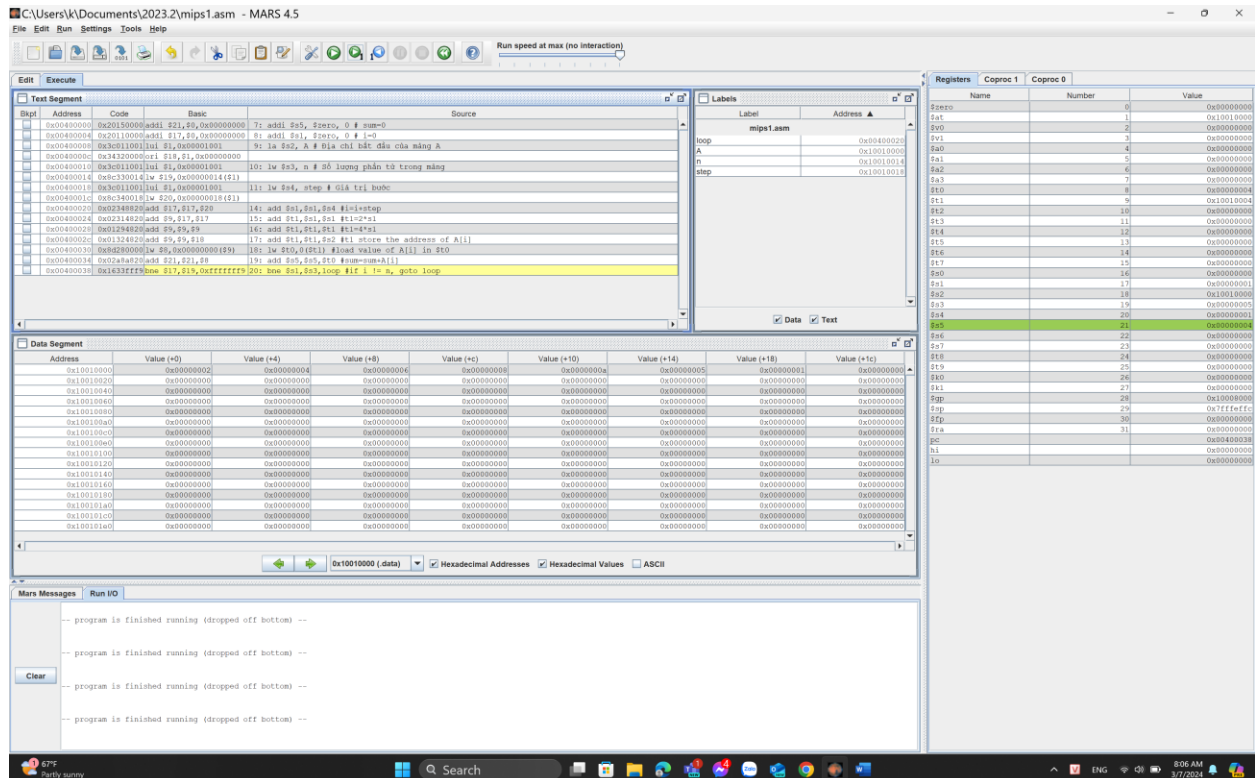
add \$t1, \$t1, \$t1 # $t1=4*s1$

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 # $sum=sum+A[i]$

bne \$s1, \$s3, loop # if $i \neq n$, goto loop



Sau khi chạy từng bước, ta thấy:

- \$a0 chuyển thành 0x10010000.
- \$t0 lưu giữ giá trị tạm thời của các phần tử thuộc mảng A.
- \$t1 lưu giữ giá trị tạm thời địa chỉ của phần tử A[i].
- \$s1 là i, lưu giữ chỉ số phần tử hiện tại mà đang xét của mảng A. Ban đầu i=0 và dần dần được cập nhật giá trị sau từng vòng lặp $i=i+step$, với $step=1$.
- \$s5=sum lưu giữ tổng các phần tử của A sau mỗi vòng lặp. $sum=sum+A[i]$.
- Cuối mỗi vòng lặp, sẽ có lệnh bne so sánh sự khác nhau của i và n. Do i chạy tăng dần từ 0 nên khi i chạy đến bằng giá trị với n thì vòng lặp sẽ dừng lại.

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

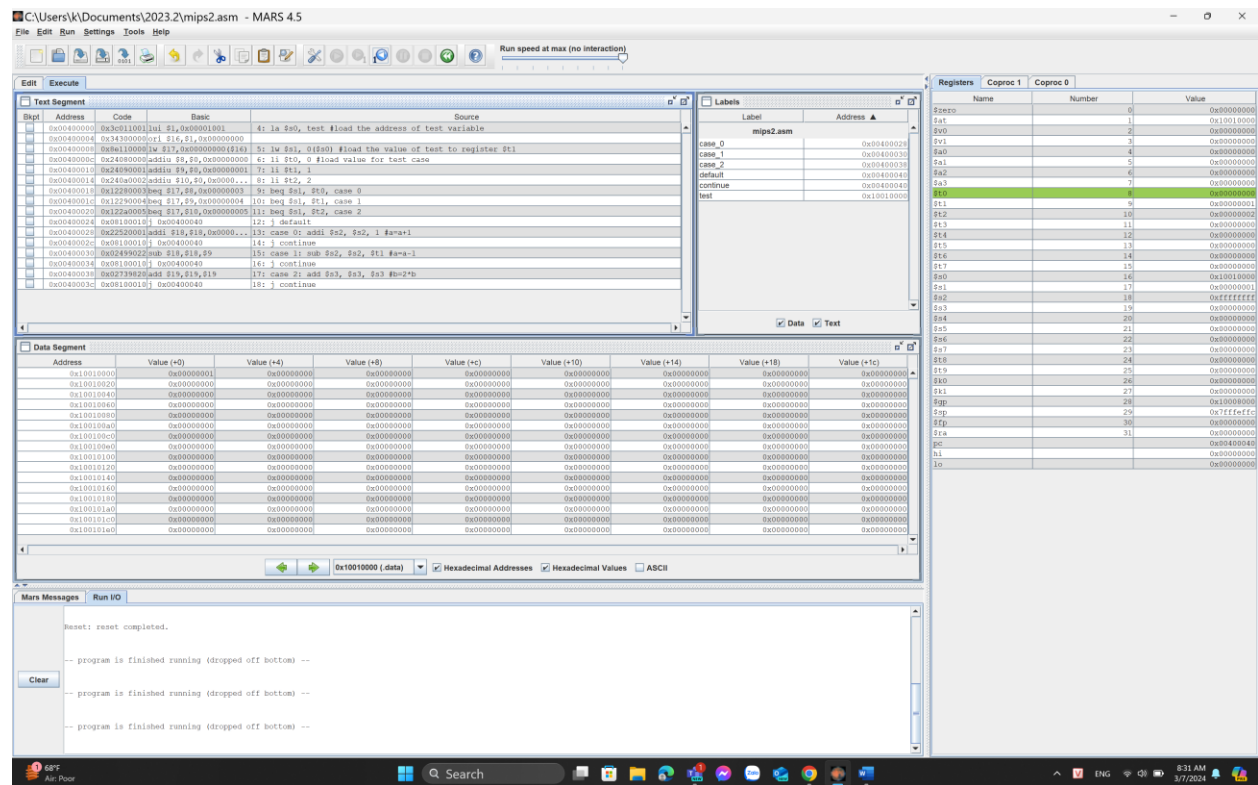
j default

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:

```



- Lệnh beq so sánh 2 thanh ghi có bằng nhau hay không. Nếu bằng nhau thì sẽ chuyển sang case tương ứng. VD như beq \$s1, \$t0, case_0 nếu \$s1=\$t0 thì sẽ thực hiện case_0.

- Ở đây ta thấy test được gán giá trị bằng 1, khi đó chương trình sẽ thực hiện dòng beq thứ 2 và đưa đến trường hợp case_1. Thực hiện phép tính a=a-1.

Assignment 4:

```

a) i<j:
.data
.text
main:
li $s1,4 # i=4
li $s2,5 # j=5
li $t1,1 # x=1
li $t2,2 # y=2
li $t3,3 # z=3
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:

```

b) $i \geq j$:

```

.data
.text
main:
li $s1,4 # i=5
li $s2,5 # j=5
li $t1,1 # x=1
li $t2,2 # y=2
li $t3,3 # z=3
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) $i+j \leq 0$:

```

.text
li $s1, 3 # i = 3
li $s2, -4 # j = -4
li $t1,1 # x=1
li $t2,2 # y=2
li $t3,3 # z=3
add $t0, $s1, $s2 # Tính i + j
bgtz $t0, else # Nhảy đến else nếu i + j > 0
then:
addi $t1, $t1, 1 # x = x + 1
addi $t3, $zero, 1 # z = 1
j endif # Bỏ qua phần "else"
else:
addi $t2, $t2, -1 # y = y - 1
add $t3, $t3, $t3 # z = 2 * z
endif:

```

d) $i+j > m+n$

```
.text
li $s1, 3 # i = 3
li $s2, 4 # j = 4
li $s3, 5 # m=5
li $s4, 9 # n=9
li $t1, 1 # x=1
li $t2, 2 # y=2
li $t3, 3 # z=3
add $t0, $s1, $s2 # Tính i + j
add $t4, $s3, $s4 # Tính m+n
slt $t5, $t0, $t4 # t0 < t4? 1:0
bne $t5, $zero, else # Nhảy đến else nếu i + j <= m+n
then:
addi $t1, $t1, 1 # x = x + 1
addi $t3, $zero, 1 # z = 1
j endif # Bỏ qua phần "else"
else:
addi $t2, $t2, -1 # y = y - 1
add $t3, $t3, $t3 # z = 2 * z
endif:
```

Assignment 5:

a) $i < n$:

```
.data
A: .word 2,4,6,8,10 # mảng A
n: .word 5 # số lượng phần tử
step: .word 1 # giá trị bước

.text
addi $s5, $zero, 0 # sum=0
addi $s1, $zero, 0 # i=0
la $s2, A # Địa chỉ bắt đầu của mảng A
lw $s3, n # Số lượng phần tử trong mảng
lw $s4, step # Giá trị bước

loop:
add $s1, $s1, $s4 # i=i+step
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]
bne $s1, $s3, loop #if i != n, goto loop
```

b) $i \leq n$:

.data

A: .word 2,4,6,8,10 # mảng A

n: .word 5 # số lượng phần tử

step: .word 1 # giá trị bước

.text

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

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

la \$s2, A # Địa chỉ bắt đầu của mảng A

lw \$s3, n # Số lượng phần tử trong mảng

lw \$s4, step # Giá trị bước

loop:

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

bne \$t2, \$zero, enloop # nếu \$t2 khác 0 thì chuyển đến enloop

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

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]

j loop # nhảy về loop

enloop:

c) $sum \geq 0$:

.data

A: .word 0,2,4,-10,5,6,8,-6,11,-5

.text

li \$s3, 10 #n=10

li \$s4, 1 #step=1

la \$s2,A # địa chỉ A[0]

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

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

loop:

bltz \$s5, endloop # endloop nếu $sum < 0$

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

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

add \$t1,\$t1,\$s2 #t1 lưu trữ địa chỉ của A[i]

lw \$t0,0(\$t1) #load giá trị A[i]

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

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

j loop #goto loop

endloop:

d) $A[i] == 0$:

```

.data
A: .word 12,15,9,7,0,-8,23,1,56
.text
li $s3, 9 #n=9
li $s4, 1 #step=1
la $s2,A # địa chỉ A[0]
addi $s5, $zero, 0 #sum = 0
addi $s1, $zero, 0 #i = 0
loop:
add $t1,$s1,$s1 #$t1=2*$s1
add $t1,$t1,$t1 #$t1=4*$s1
add $t1,$t1,$s2 #$t1 lưu trữ địa chỉ của A[i]
lw $t0,0($t1) #load giá trị A[i] vào $t0
beqz $t0, endloop # endloop nếu A[i] == 0
add $s5,$s5,$t0 #sum=sum+A[i]
add $s1,$s1,$s4      #i=i+step
j loop #goto loop
endloop:

```

Assignment 6:

```

.data
A: .word 1,2,3,4,5,6,8,9,10
.text
li $s7,10 # $s7=10
la $s6,A # gán nhãn A cho $s6
lw $s1,0($s6) # lưu A[0] vào $s1
abs $s2, $s1 # max=|A[0]|
addi $s3,$zero,1 # i=1
loop:
beq $s3,$s7,enloop # nếu i=n thoát
sll $s5,$s3,2 # $s5=i*4
add $s5,$s6,$s5
lw $s1,0($s5) # $s1=A[i];
abs $s1,$s1 # $s1=|A[i]|
slt $s0,$s1,$s2 # nếu A[i]>max thì $s0=0
bne $s0,$zero, L2 # nếu $s0!=0 thì nhảy đến else
add $s2,$zero,$s1 # max=A[i]
else:
addi $s3,$s3,1 # i=i+1
j for # quay lại for
enloop:

```