**BÁO CÁO THỰC HÀNH KIẾN TRÚC MÁY TÍNH TUẦN 4**

**Assignment 1**

**Code:**

#Laboratory Exercise 4, Home Assignment 1

.text

start:

li $s1,1

li $s2,-2

li $t0,0 #No Overflow is default status

addu $s3,$s1,$s2 # s3 = s1 + s2

xor $t1,$s1,$s2 #Test if $s1 and $s2 have the same sign

bltz $t1,EXIT #If not: t1<0, exit

slt $t2,$s3,$s1 # if $s3 < $s1 => $t2 = 1

bltz $s1,NEGATIVE #Test if $s1 and $s2 is negative?

beq $t2,$zero,EXIT #if t2=0=> s3>s1 (s1 and $s2 are positive)

# if $s3 > $s1 then the result is not overflow =>jump EXIT

j OVERFLOW

NEGATIVE:

bne $t2,$zero,EXIT #if t2!=0 => $s3<s1 (s1 and $s2 are negative)

# if $s3 < $s1 then the result is not overflow =>jump EXIT

OVERFLOW:

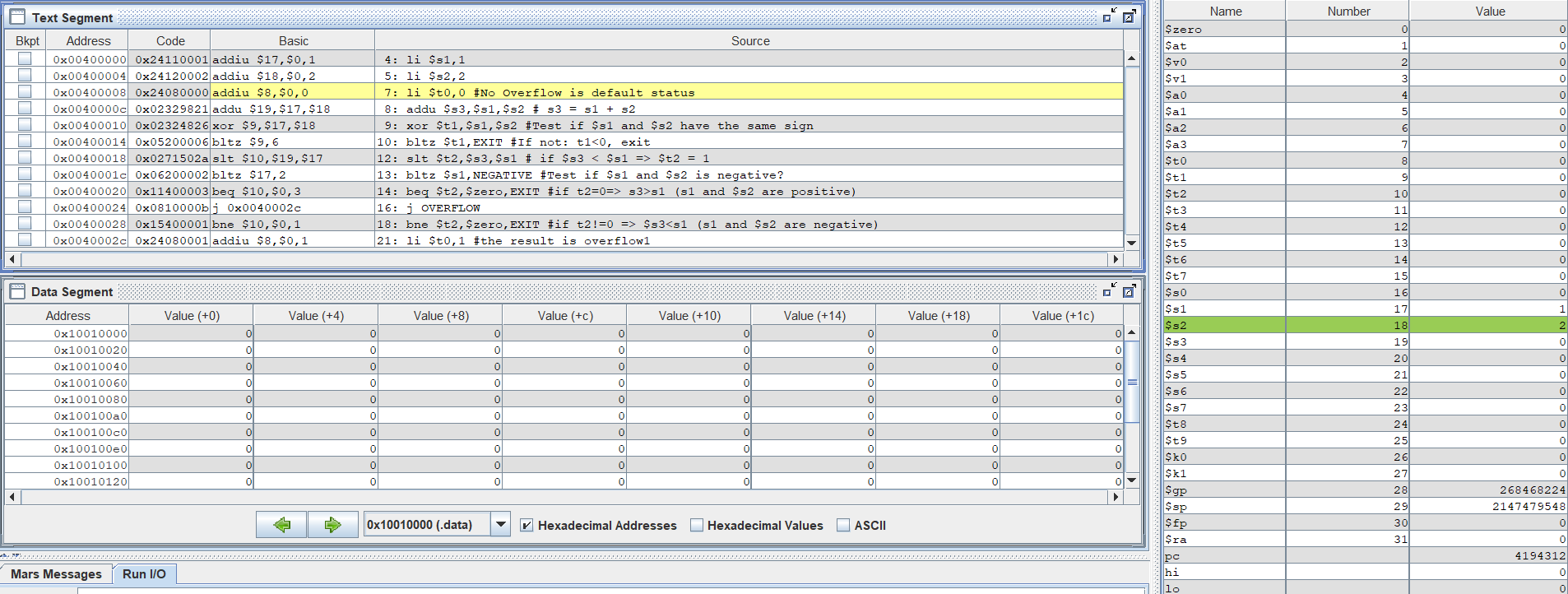
li $t0,1 #the result is overflow1

EXIT:

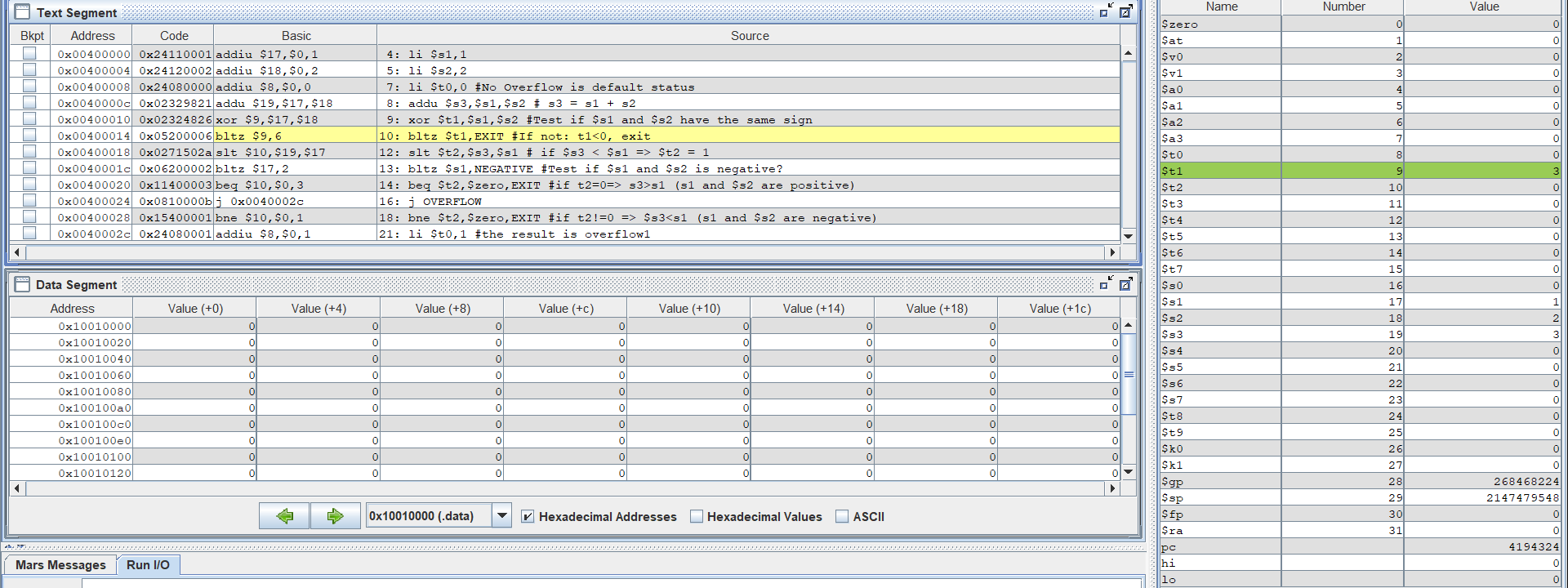
**Nhận xét**

* **TH1: Cộng 2 số dương không tràn bộ nhớ**
* Lệnh: li $s1,1

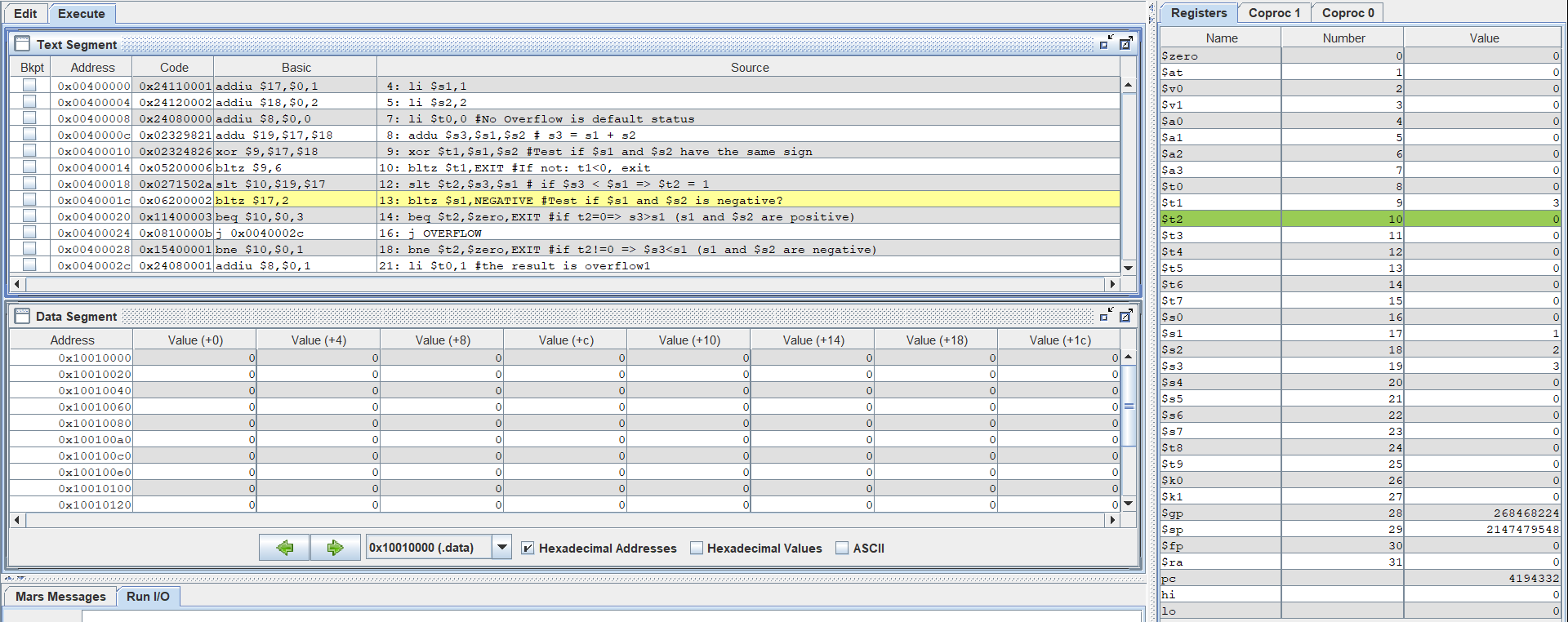
li $s2,2



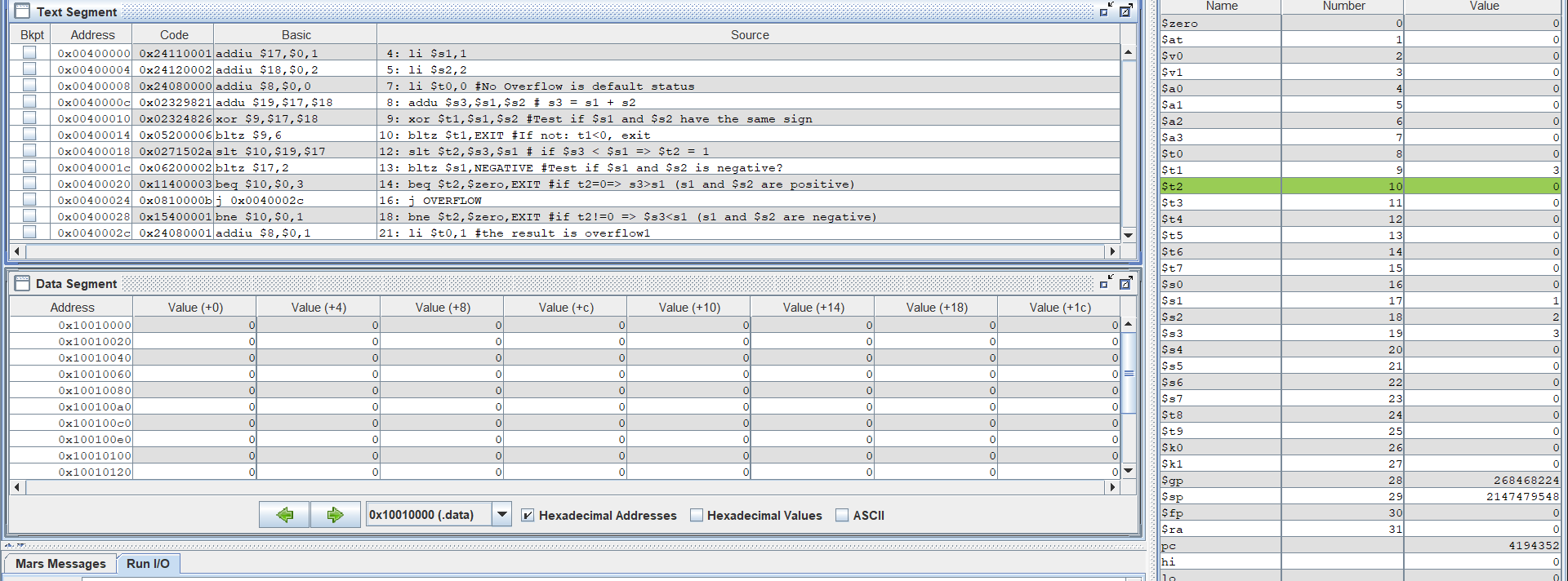
* Lệnh: li $t0,0 #No Overflow is default status
* Gán giá trị 0 là trạng thái không tràn bộ nhớ cho $t0
* addu $s3,$s1,$s2
* s3 = s1 + s2
* xor $t1,$s1,$s2 #Test if $s1 and $s2 have the same sign

****

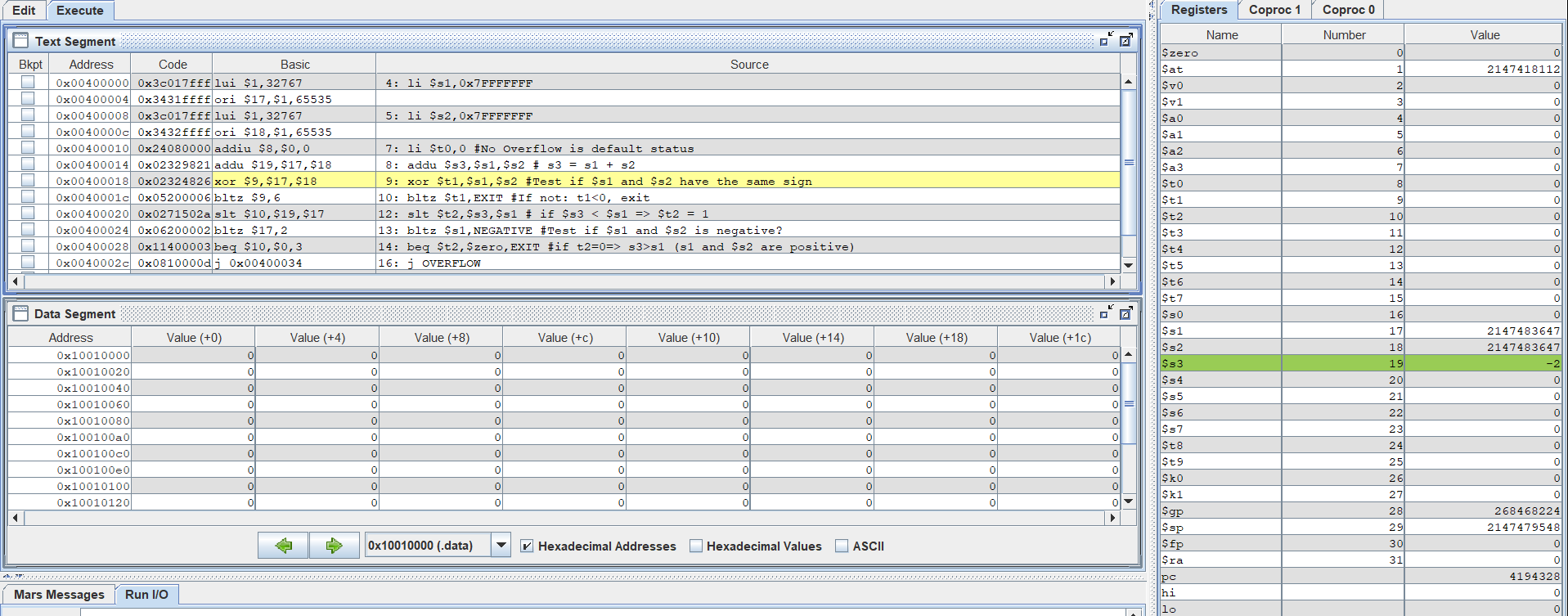
* $t1 =3 => $s1 và $s2 cùng dấu
* bltz $t1,EXIT #If not: t1<0, exit
* vì $t1=3 >0 => chạy tiếp lệnh dưới
* slt $t2,$s3,$s1 # if $s3 < $s1 => $t2 = 1



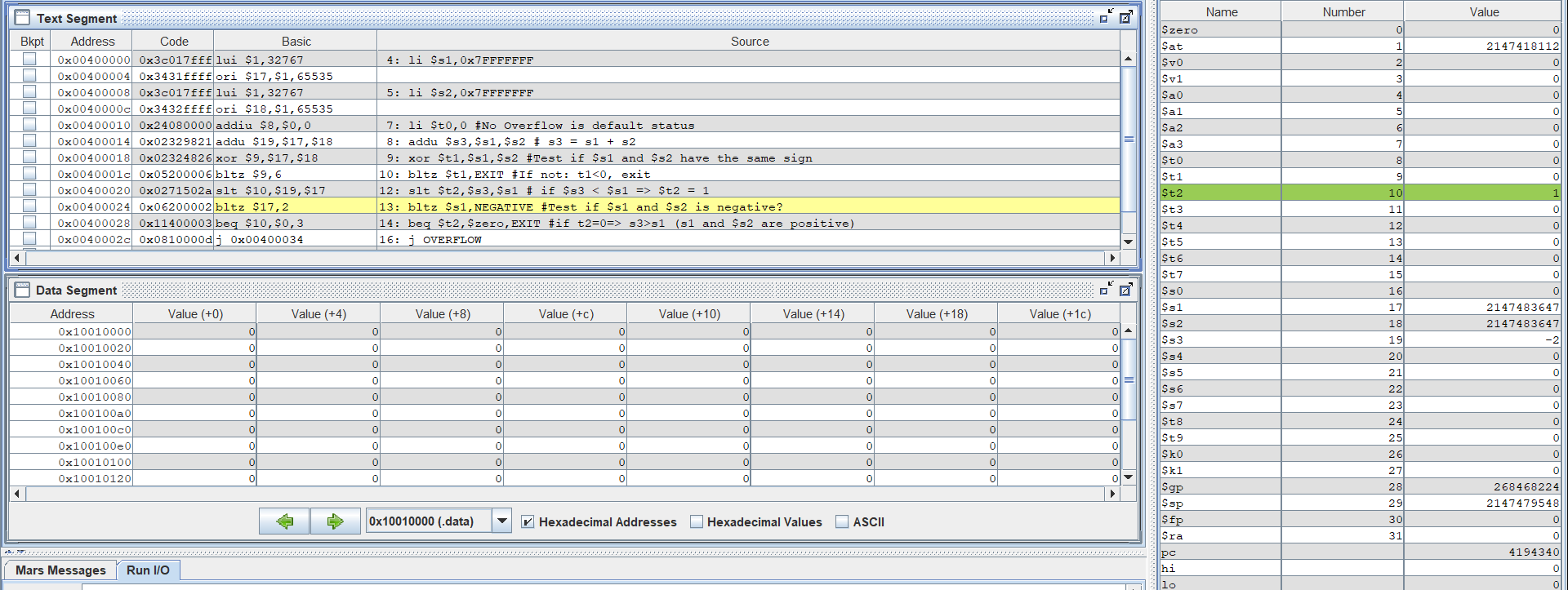
* Vì $s3>$s1 => $t2 =0
* bltz $s1,NEGATIVE #Test if $s1 and $s2 is negative?
* Vì $s1 =1 không <0 => tiếp tục chạy lệnh bên dưới
* beq $t2,$zero,EXIT #if t2=0=> s3>s1 (s1 and $s2 are positive)



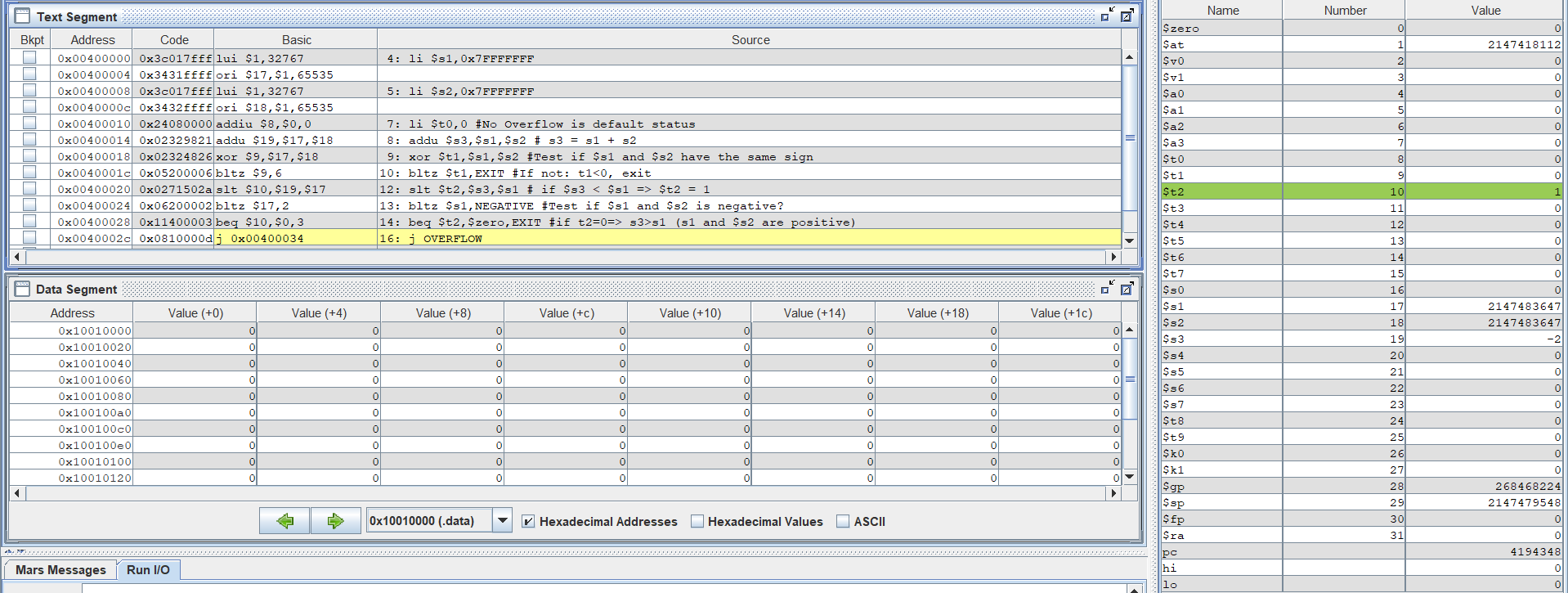
* Vì $t2 =0=> Nhảy tới EXIT để kết thúc chương trình
* $t0 = 0 => Không bị tràn bộ nhớ
* **TH2 : Cộng 2 số dương tràn bộ nhớ**
* Tương tự như chương trình cộng 2 số dương nhưng tới lệnh:
* addu $s3,$s1,$s2 # $s1=0x7fffffff, $s2
* s3 = s1 + s2 = -2 ( overflow )

****

* slt $t2,$s3,$s1

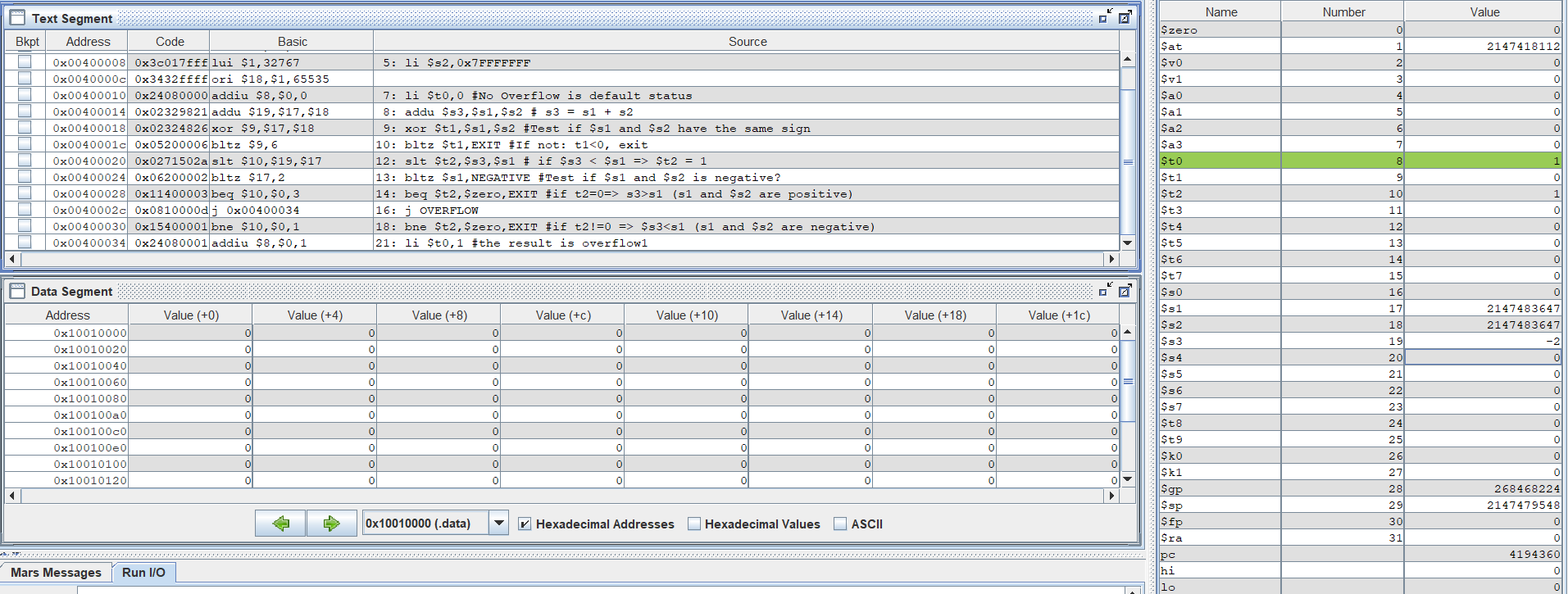
****

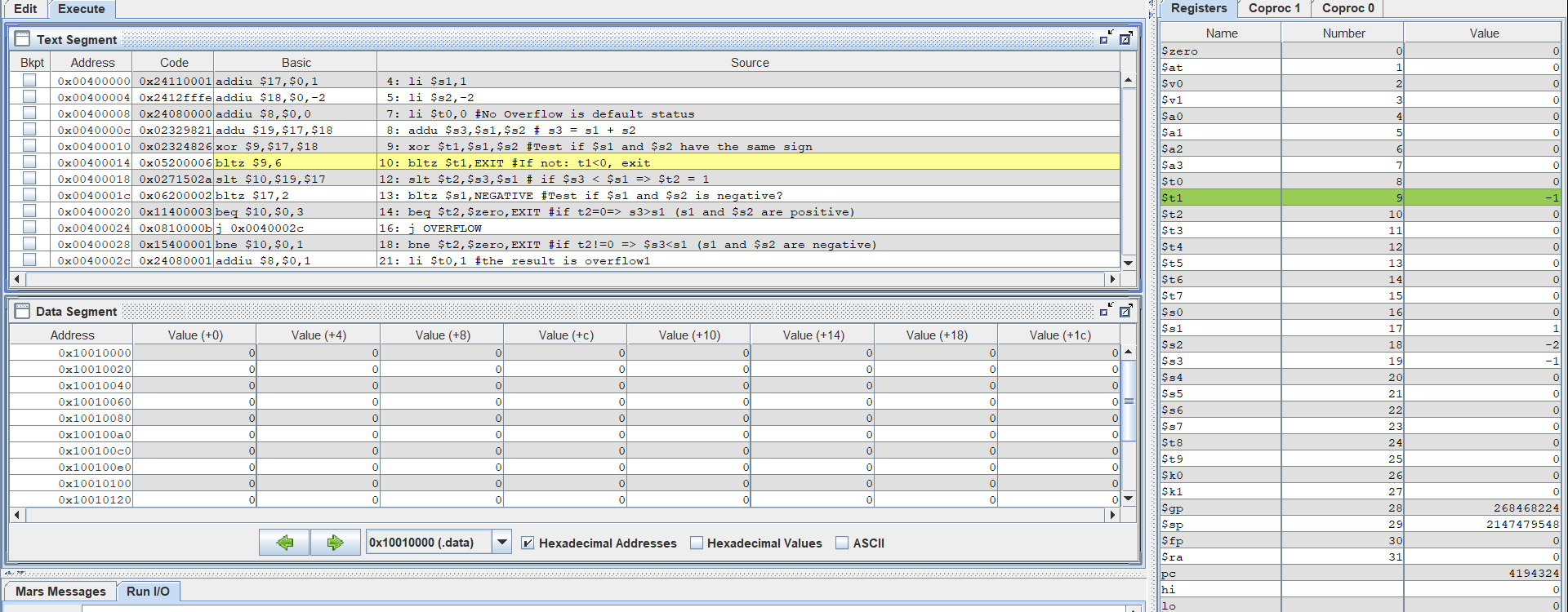
* Vì $s3 < $s1( -2< 0x7fffffff) => $t2 =1
* beq $t2,$zero,EXIT

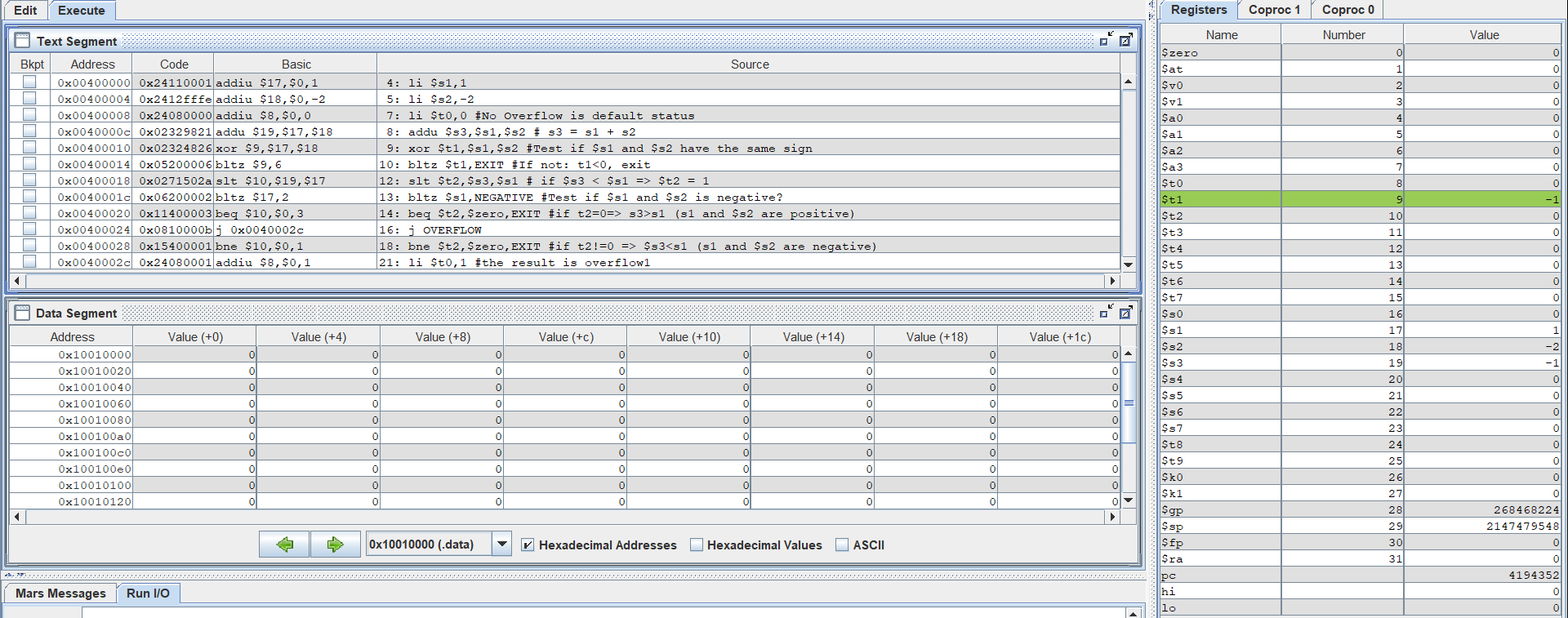
****

* Vì $t2 =1 !=0 => chương trình sẽ jump tới OVERFLOW
* OVERFLOW:

li $t0,1 #the result is overflow is 1



* Gán giá trị 1 cho biến $t0=> Chương trình bị tràn bộ nhớ
* TH3 : Cộng 2 số khác dấu
* Tương tự chương trình cộng 2 số dương cho đến lệnh:
* xor $t1,$s1,$s2 
* $t1 =-1 ( $s1 =1 , $s2 =-2)
* bltz $t1,EXIT #If not: t1<0, exit



* Vì trong TH này $t1 = <0 => Nhảy tới lệnh EXIT
* Vì $t0 = 0 => Chương trình không bị tràn bộ nhớ
* TH cộng 2 số âm không bị tràn bộ nhớ và bị tràn bộ nhớ sẽ lọt vào NEGATIVE, kết quả tương tự như TH1 và TH2

**Assignment 2**

Kết quả được chứa ở $s1:

* Lấy ra MSB của $s0

.text

li $s0, 0x312ACD81 # Set value to s0

li $t0, 0xFF000000 # Mask to get MSB

and $s1, $s0, $t0 # Get MSB

* Xóa LSB của $s0

.text

li $s0, 0x312ACD81 # Set value to s0

li $t0, 0xffffff00 # Mask to clear LSB

and $s1, $s0, $t0 # Clear LSB

* Đặt 8 bit bên phải của $s0 là 1

.text

li $s0, 0x312ACD81 # Set value to s0

li $t0, 0x000000ff # Mask to Set LSB

or $s1, $s0, $t0 # Set LSB

* Xóa $s0 ($s0 = 0)

.text

li $s0, 0x312ACD81 # Set value to s0

and $s0, $s0, $zero # Clear $s0

**Assignment 3**

.text

li $s1, 10

bltz $s1, NEGSET # Branch to set $s0 = -$s1 if s1 < 0

add $s0, $s1, $zero # s0 = s1 if s1 > 0

j EXIT

NEGSET:

li $t0, 0xffffffff # Load Mask to t0

xor $s3, $s1, $t0 # Set s0 = -s1

addi $s0, $s3, 1

EXIT:

.text

add $s0, $s1, $zero # s0 = s1

.text

li $s2, 0xffffffff

xor $s0, $s1, $s2 # s0 = not(s1)

.text

slt $t1, $s1, $s2

bne $t1, $zero, label

beq $s1, $s2, label

**Assignment 4**

.text

li $s1,0x7fffffff

li $s2,0x7fffffff

li $t0, 0 # Dedault status is no overflow

xor $t1, $s1, $s2 # Check s1 and s2 have the same sign

bltz $t1, EXIT # Exit if $t1<0 <=> different sign

# Else s1 and s2 have same sign

addu $s3, $s1, $s2 # s3 = s1 + s2

xor $t1, $s3, $s1 # Check sum and s1 have the same sign

bltz $t1, OVERFLOW # t1 < 0 <=> sum and s1 does not have the same sign => overflow

j EXIT

OVERFLOW:

li $t0, 1 # Overflow

EXIT:

**Assignment 5**

.text

li $s0, 1 # set first result s0 = 1

li $s1, 8 # set second factor s1 = 8 (small power of X2)

li $s2, 1 # value to determine stop time of loop

LOOP:

beq $s1, $s2, END # End if s1 = 1

sll $s0, $s0, 1 # s0 = s0 \* 2

srl $s1, $s1, 1 # s1 = s1/2

j LOOP

END: