BÁO CÁO TUẦN 4

Bài 1:

- *Chương trình:*

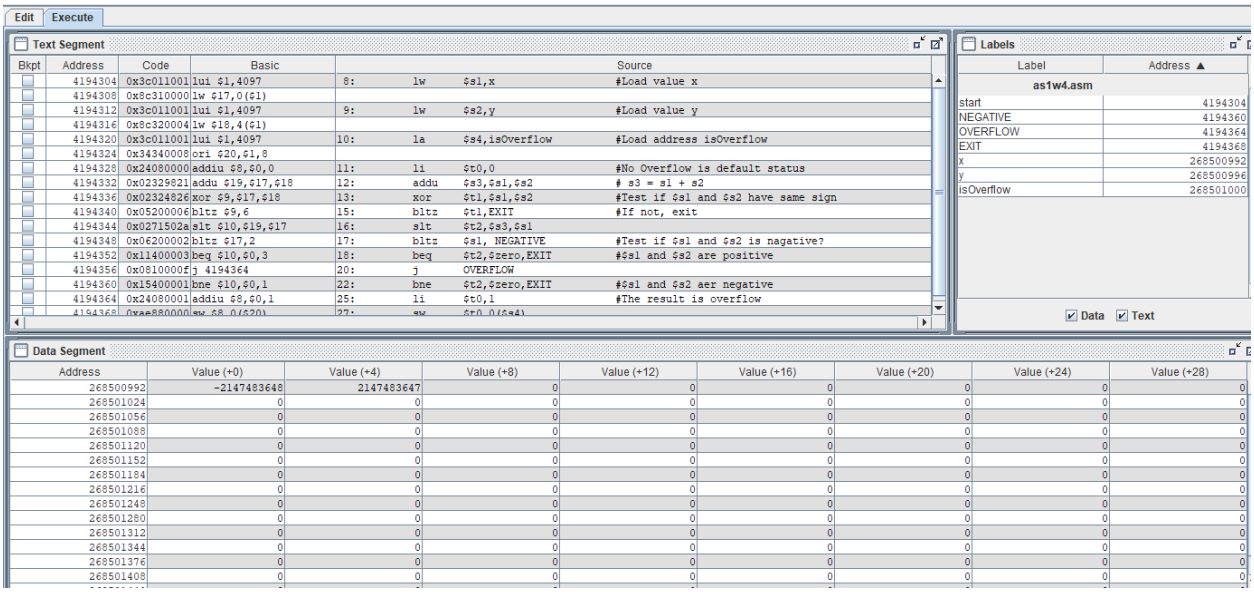
```
#Laboratory Exercise 4, Home Assignment 1
.data
x:              .word   0x80000000
y:              .word   0x7fffffff
isOverflow:     .word   0
.text
start:
        lw      $s1,x                   #Load value x
        lw      $s2,y                   #Load value y
        la      $s4,isOverflow          #Load address isOverflow
        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 same sign

        bltz    $t1,EXIT                #If not, exit
        slt     $t2,$s3,$s1
        bltz    $s1, NEGATIVE           #Test if $s1 and $s2 is nagative?
        beq     $t2,$zero,EXIT          #$s1 and $s2 are positive
                #if $s3 > $s1 then the result is not overflow
        j       OVERFLOW
NEGATIVE:
        bne     $t2,$zero,EXIT          #$s1 and $s2 aer negative
                #if $s3 < $s1 then the result is not overflow
OVERFLOW:
        li      $t0,1                   #The result is overflow
EXIT:
        sw      $t0,0($s4)
```

TH1: x, y trái dấu

- *Chương trình:*

```
#Laboratory Exercise 4, Home Assignment 1
.data
x:              .word   0x80000000
y:              .word   0x7fffffff
isOverflow:     .word   0
.text
start:
        lw      $s1,x                   #Load value x
        lw      $s2,y                   #Load value y
        la      $s4,isOverflow          #Load address isOverflow
        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 same sign

        bltz    $t1,EXIT                #If not, exit
        slt     $t2,$s3,$s1
        bltz    $s1, NEGATIVE           #Test if $s1 and $s2 is nagative?
        beq     $t2,$zero,EXIT          #$s1 and $s2 are positive
                #if $s3 > $s1 then the result is not overflow
        j       OVERFLOW
NEGATIVE:
        bne     $t2,$zero,EXIT          #$s1 and $s2 aer negative
                #if $s3 < $s1 then the result is not overflow
OVERFLOW:
        li      $t0,1                   #The result is overflow
EXIT:
        sw      $t0,0($s4)
```

- *Kết quả:*



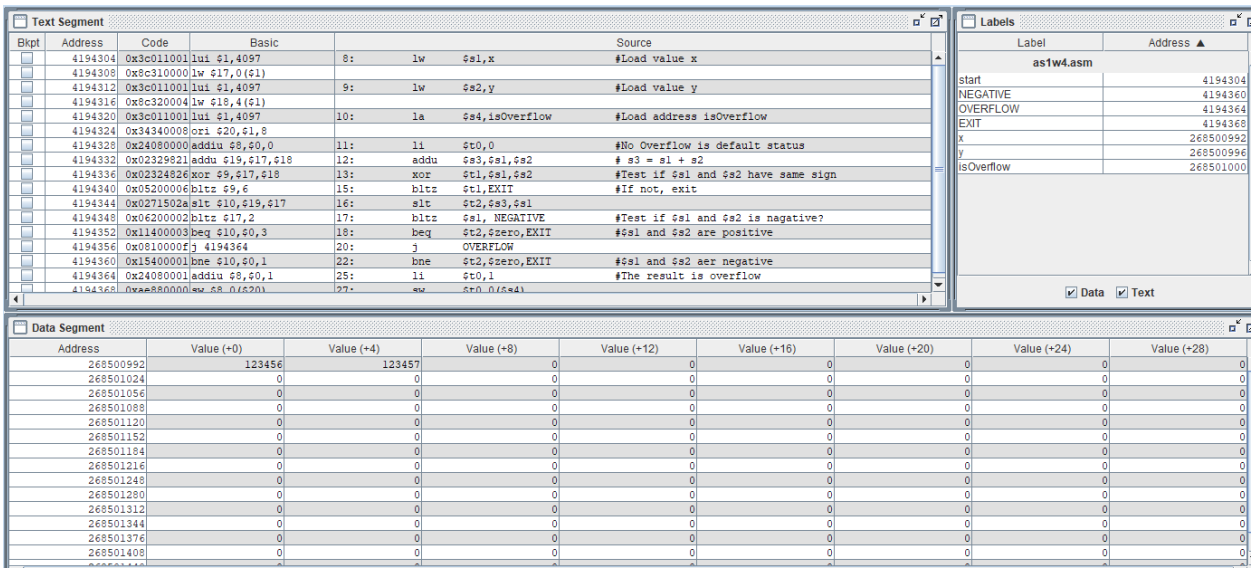
$s1	17	-2147483648
$s2	18	2147483647
$s3	19	-1
$s4	20	268501000

TH2: x, y cùng dương và x+y không tràn

- *Chương trình:*

```
#Laboratory Exercise 4, Home Assignment 1
.data
x:              .word   123456
y:              .word   123457
isOverflow:     .word   0
.text
start:
        lw      $s1,x                   #Load value x
        lw      $s2,y                   #Load value y
        la      $s4,isOverflow          #Load address isOverflow
        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 same sign

        bltz    $t1,EXIT                #If not, exit
        slt     $t2,$s3,$s1
        bltz    $s1, NEGATIVE           #Test if $s1 and $s2 is nagative?
        beq     $t2,$zero,EXIT          #$s1 and $s2 are positive
                #if $s3 > $s1 then the result is not overflow
        j       OVERFLOW
NEGATIVE:
        bne     $t2,$zero,EXIT          #$s1 and $s2 aer negative
                #if $s3 < $s1 then the result is not overflow
OVERFLOW:
        li      $t0,1                   #The result is overflow
EXIT:
        sw      $t0,0($s4)
```

- *Kết quả:*



Text Segment

Bkpt	Address	Code	Basic				Source
	4194304	0x3c011001	lui $1,4097	8:	lw	$s1,x	#Load value x
	4194308	0x8c310000	lw $17,0($1)				
	4194312	0x3c011001	lui $1,4097	9:	lw	$s2,y	#Load value y
	4194316	0x8c320004	lw $18,4($1)				
	4194320	0x3c011001	lui $1,4097	10:	la	$s4,isOverflow	#Load address isOverflow
	4194324	0x34340008	ori $20,$1,8				
	4194328	0x24080000	addiu $8,$0,0	11:	li	$t0,0	#No Overflow is default status
	4194332	0x02329821	addu $19,$17,$18	12:	addu	$s3,$s1,$s2	# s3 = s1 + s2
	4194336	0x02324826	xor $9,$17,$18	13:	xor	$t1,$s1,$s2	#Test if $s1 and $s2 have same sign
	4194340	0x05200006	bltz $9,6	15:	bltz	$t1,EXIT	#If not, exit
	4194344	0x0271502a	slt $10,$19,$17	16:	slt	$t2,$s3,$s1	
	4194348	0x06200002	bltz $17,2	17:	bltz	$s1, NEGATIVE	#Test if $s1 and $s2 is nagative?
	4194352	0x11400003	beq $10,$0,3	18:	beq	$t2,$zero,EXIT	#$s1 and $s2 are positive
	4194356	0x0810000f	j 4194364	20:	j	OVERFLOW	
	4194360	0x15400001	bne $10,$0,1	22:	bne	$t2,$zero,EXIT	#$s1 and $s2 aer negative
	4194364	0x24080001	addiu $8,$0,1	25:	li	$t0,1	#The result is overflow
	4194368	0xae880000	sw $8,0($20)	27:	sw	$t0,0($s4)	

Labels

Label	Address ▲
as1w4.asm	
start	4194304
NEGATIVE	4194360
OVERFLOW	4194364
EXIT	4194368
x	268500992
y	268500996
isOverflow	268501000

☑ Data ☑ Text

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+12)	Value (+16)	Value (+20)	Value (+24)	Value (+28)
268500992	123456	123457	0	0	0	0	0	0
268501024	0	0	0	0	0	0	0	0
268501056	0	0	0	0	0	0	0	0
268501088	0	0	0	0	0	0	0	0
268501120	0	0	0	0	0	0	0	0
268501152	0	0	0	0	0	0	0	0
268501184	0	0	0	0	0	0	0	0
268501216	0	0	0	0	0	0	0	0
268501248	0	0	0	0	0	0	0	0
268501280	0	0	0	0	0	0	0	0
268501312	0	0	0	0	0	0	0	0
268501344	0	0	0	0	0	0	0	0
268501376	0	0	0	0	0	0	0	0
268501408	0	0	0	0	0	0	0	0

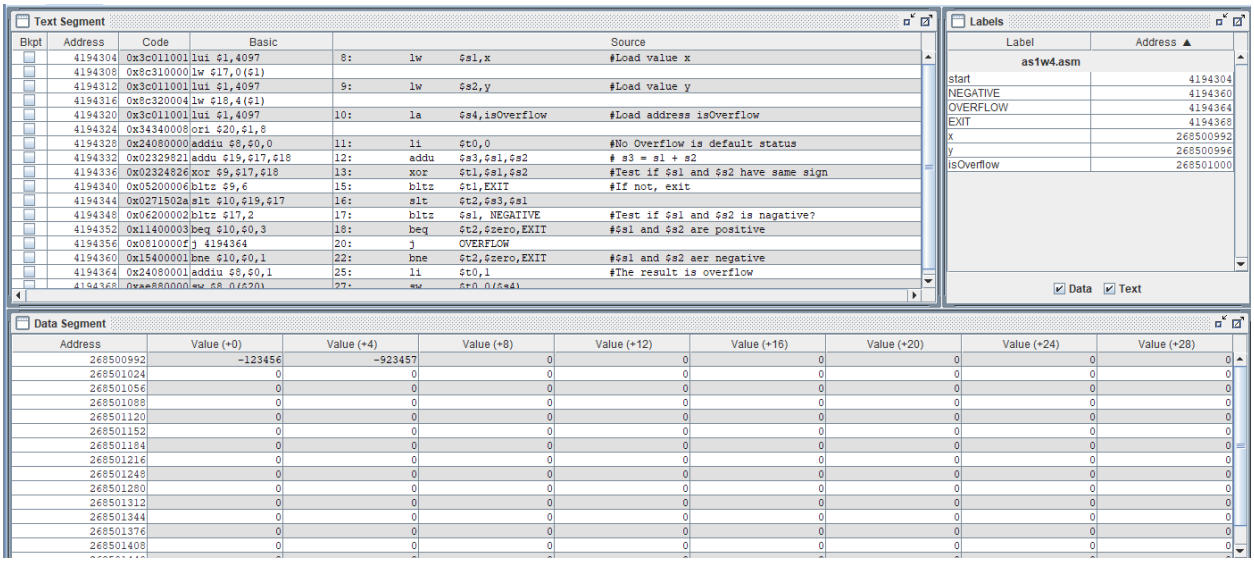
$s1	17	123456
$s2	18	123457
$s3	19	246913
$s4	20	268501000

TH3: x, y cùng âm và x+y không tràn

- *Chương trình:*

```
#Laboratory Exercise 4, Home Assignment 1
.data
x:              .word   -123456
y:              .word   -923457
isOverflow:     .word   0
.text
start:
        lw      $s1,x                   #Load value x
        lw      $s2,y                   #Load value y
        la      $s4,isOverflow          #Load address isOverflow
        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 same sign

        bltz    $t1,EXIT                #If not, exit
        slt     $t2,$s3,$s1
        bltz    $s1, NEGATIVE           #Test if $s1 and $s2 is nagative?
        beq     $t2,$zero,EXIT          #$s1 and $s2 are positive
                #if $s3 > $s1 then the result is not overflow
        j       OVERFLOW
NEGATIVE:
        bne     $t2,$zero,EXIT          #$s1 and $s2 aer negative
                #if $s3 < $s1 then the result is not overflow
OVERFLOW:
        li      $t0,1                   #The result is overflow
EXIT:
        sw      $t0,0($s4)
```

- *Kết quả:*



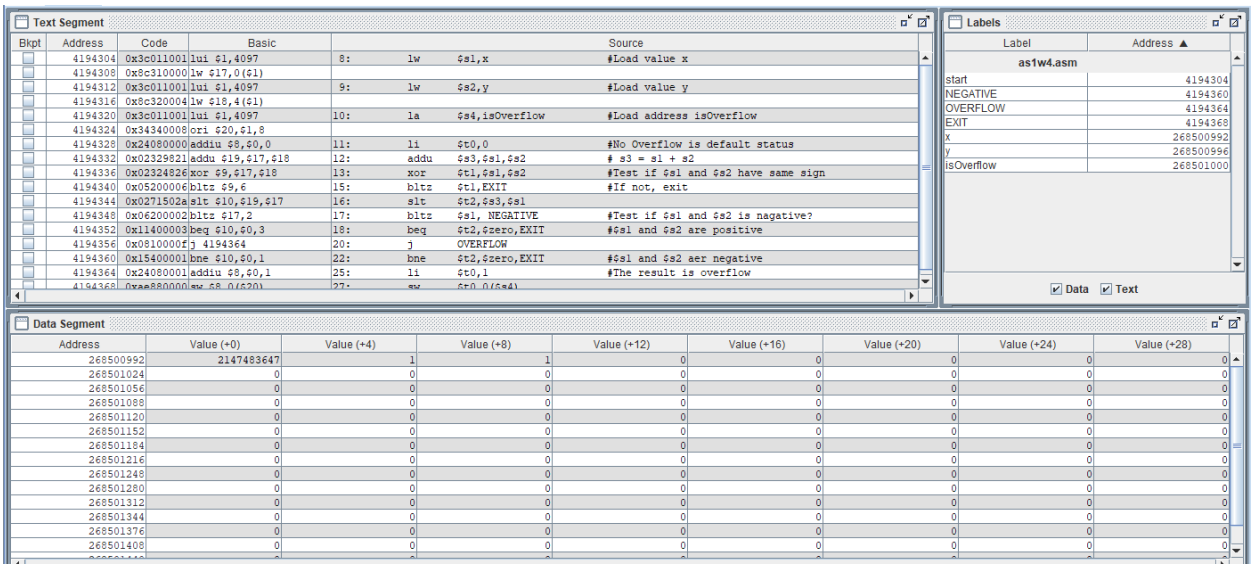
$s1	17	-123456
$s2	18	-923457
$s3	19	-1046913
$s4	20	268501000

TH4: x, y cùng dương và x+y tràn

- *Chương trình:*

```
#Laboratory Exercise 4, Home Assignment 1
.data
x:              .word   0x7fffffff
y:              .word   1
isOverflow:     .word   0
.text
start:
        lw      $s1,x                   #Load value x
        lw      $s2,y                   #Load value y
        la      $s4,isOverflow          #Load address isOverflow
        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 same sign

        bltz    $t1,EXIT                #If not, exit
        slt     $t2,$s3,$s1
        bltz    $s1, NEGATIVE           #Test if $s1 and $s2 is nagative?
        beq     $t2,$zero,EXIT          #$s1 and $s2 are positive
                #if $s3 > $s1 then the result is not overflow
        j       OVERFLOW
NEGATIVE:
        bne     $t2,$zero,EXIT          #$s1 and $s2 aer negative
                #if $s3 < $s1 then the result is not overflow
OVERFLOW:
        li      $t0,1                   #The result is overflow
EXIT:
        sw      $t0,0($s4)
```

- *Kết quả:*



Text Segment

Bkpt	Address	Code	Basic				Source
	4194304	0x3c011001	lui $1,4097	8:	lw	$s1,x	#Load value x
	4194308	0x8c310000	lw $17,0($1)				
	4194312	0x3c011001	lui $1,4097	9:	lw	$s2,y	#Load value y
	4194316	0x8c320004	lw $18,4($1)				
	4194320	0x3c011001	lui $1,4097	10:	la	$s4,isOverflow	#Load address isOverflow
	4194324	0x34340008	ori $20,$1,8				
	4194328	0x24080000	addiu $8,$0,0	11:	li	$t0,0	#No Overflow is default status
	4194332	0x02329821	addu $19,$17,$18	12:	addu	$s3,$s1,$s2	# s3 = s1 + s2
	4194336	0x02324826	xor $9,$17,$18	13:	xor	$t1,$s1,$s2	#Test if $s1 and $s2 have same sign
	4194340	0x05200006	bltz $9,6	15:	bltz	$t1,EXIT	#If not, exit
	4194344	0x0271502a	slt $10,$19,$17	16:	slt	$t2,$s3,$s1	
	4194348	0x06200002	bltz $17,2	17:	bltz	$s1, NEGATIVE	#Test if $s1 and $s2 is nagative?
	4194352	0x11400003	beq $10,$0,3	18:	beq	$t2,$zero,EXIT	#$s1 and $s2 are positive
	4194356	0x0810000f	j 4194364	20:	j	OVERFLOW	
	4194360	0x15400001	bne $10,$0,1	22:	bne	$t2,$zero,EXIT	#$s1 and $s2 aer negative
	4194364	0x24080001	addiu $8,$0,1	25:	li	$t0,1	#The result is overflow
	4194368	0xae880000	sw $8,0($20)	27:	sw	$t0,0($s4)	

Labels

Label	Address ▲
as1w4.asm	
start	4194304
NEGATIVE	4194360
OVERFLOW	4194364
EXIT	4194368
x	268500992
y	268500996
isOverflow	268501000

☑ Data ☑ Text

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+12)	Value (+16)	Value (+20)	Value (+24)	Value (+28)
268500992	2147483647	1	1	0	0	0	0	0
268501024	0	0	0	0	0	0	0	0
268501056	0	0	0	0	0	0	0	0
268501088	0	0	0	0	0	0	0	0
268501120	0	0	0	0	0	0	0	0
268501152	0	0	0	0	0	0	0	0
268501184	0	0	0	0	0	0	0	0
268501216	0	0	0	0	0	0	0	0
268501248	0	0	0	0	0	0	0	0
268501280	0	0	0	0	0	0	0	0
268501312	0	0	0	0	0	0	0	0
268501344	0	0	0	0	0	0	0	0
268501376	0	0	0	0	0	0	0	0
268501408	0	0	0	0	0	0	0	0

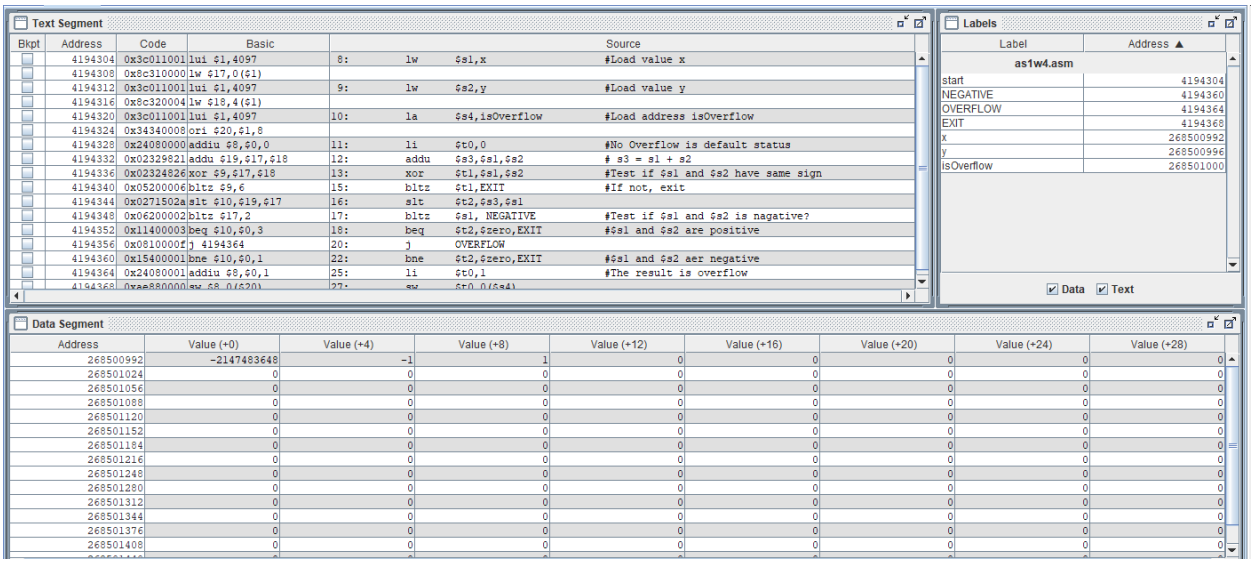
$s1	17	2147483647
$s2	18	1
$s3	19	-2147483648
$s4	20	268501000

TH5: x, y cùng âm và x+y tràn

- *Chương trình:*

```
#Laboratory Exercise 4, Home Assignment 1
.data
x:              .word   0x80000000
y:              .word   -1
isOverflow:     .word   0
.text
start:
        lw      $s1,x                   #Load value x
        lw      $s2,y                   #Load value y
        la      $s4,isOverflow          #Load address isOverflow
        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 same sign

        bltz    $t1,EXIT                #If not, exit
        slt     $t2,$s3,$s1
        bltz    $s1, NEGATIVE           #Test if $s1 and $s2 is nagative?
        beq     $t2,$zero,EXIT          #$s1 and $s2 are positive
                #if $s3 > $s1 then the result is not overflow
        j       OVERFLOW
NEGATIVE:
        bne     $t2,$zero,EXIT          #$s1 and $s2 aer negative
                #if $s3 < $s1 then the result is not overflow
OVERFLOW:
        li      $t0,1                   #The result is overflow
EXIT:
        sw      $t0,0($s4)
```

- *Kết quả:*



Text Segment

Bkpt	Address	Code	Basic			Source
	4194304	0x3c011001	lui $1,4097	8:	lw $s1,x	#Load value x
	4194308	0x8c310000	lw $17,0($1)			
	4194312	0x3c011001	lui $1,4097	9:	lw $s2,y	#Load value y
	4194316	0x8c320004	lw $18,4($1)			
	4194320	0x3c011001	lui $1,4097	10:	la $s4,isOverflow	#Load address isOverflow
	4194324	0x34340008	ori $20,$1,8			
	4194328	0x24080000	addiu $8,$0,0	11:	li $t0,0	#No Overflow is default status
	4194332	0x02329821	addu $19,$17,$18	12:	addu $s3,$s1,$s2	# s3 = s1 + s2
	4194336	0x02324826	xor $9,$17,$18	13:	xor $t1,$s1,$s2	#Test if $s1 and $s2 have same sign
	4194340	0x05200006	bltz $9,6	15:	bltz $t1,EXIT	#If not, exit
	4194344	0x0271502a	slt $10,$19,$17	16:	slt $t2,$s3,$s1	
	4194348	0x06200002	bltz $17,2	17:	bltz $s1, NEGATIVE	#Test if $s1 and $s2 is nagative?
	4194352	0x11400003	beq $10,$0,3	18:	beq $t2,$zero,EXIT	#$s1 and $s2 are positive
	4194356	0x0810000f	j 4194364	20:	j OVERFLOW	
	4194360	0x15400001	bne $10,$0,1	22:	bne $t2,$zero,EXIT	#$s1 and $s2 aer negative
	4194364	0x24080001	addiu $8,$0,1	25:	li $t0,1	#The result is overflow
	4194368	0xae880000	sw $8,0($20)	27:	sw $t0,0($s4)	

Labels

Label	Address ▲
as1w4.asm	
start	4194304
NEGATIVE	4194360
OVERFLOW	4194364
EXIT	4194368
x	268500992
y	268500996
isOverflow	268501000

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+12)	Value (+16)	Value (+20)	Value (+24)	Value (+28)
268500992	-2147483648	-1	1	0	0	0	0	0
268501024	0	0	0	0	0	0	0	0
268501056	0	0	0	0	0	0	0	0
268501088	0	0	0	0	0	0	0	0
268501120	0	0	0	0	0	0	0	0
268501152	0	0	0	0	0	0	0	0
268501184	0	0	0	0	0	0	0	0
268501216	0	0	0	0	0	0	0	0
268501248	0	0	0	0	0	0	0	0
268501280	0	0	0	0	0	0	0	0
268501312	0	0	0	0	0	0	0	0
268501344	0	0	0	0	0	0	0	0
268501376	0	0	0	0	0	0	0	0
268501408	0	0	0	0	0	0	0	0

$s1	17	-2147483648
$s2	18	-1
$s3	19	2147483647
$s4	20	268501000

Bài 2:

a, Extract MSB of $s0

- *Chương trình:*

```
#Laboratory Exercise 4, Home Assignment 2
.text
        li      $s0,0x20194543          #load test value for these function
        andi    $t0, $s0, 0xff000000    #Extract MSB of $s0
```

- *Kết quả:*

Text Segment

Bkpt	Address	Code	Basic	Source
	4194304	0x3c012019	lui $1,0x00002019	3: li $s0,0x20194543 #load test value for these function
	4194308	0x34304543	ori $16,$1,0x00004543	
	4194312	0x3c01ff00	lui $1,0xffffff00	4: andi $t0, $s0, 0xff000000 #Extract MSB of $s0
	4194316	0x34210000	ori $1,$1,0x00000000	
	4194320	0x02014024	and $8,$16,$1	

$t0	8	0x20000000

b, Clear LSB of $s0

- *Chương trình:*

```
#Laboratory Exercise 4, Home Assignment 2
.text
        li      $s0,0x20194543          #load test value for these function
        andi    $t0, $s0, 0xffffff00    #Clear LSB of $s0
```

- *Kết quả:*

Text Segment

Bkpt	Address	Code	Basic	Source
	4194304	0x3c012019	lui $1,0x00002019	3: li $s0,0x20194543 #load test value for these function
	4194308	0x34304543	ori $16,$1,0x00004543	
	4194312	0x3c01ffff	lui $1,0xffffffff	4: andi $t0, $s0, 0xffffff00 #Clear LSB of $s0
	4194316	0x3421ff00	ori $1,$1,0x0000ff00	
	4194320	0x02014024	and $8,$16,$1	

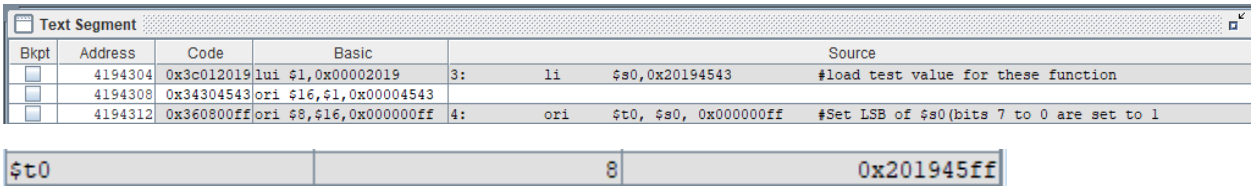
$t0	8	0x20194500

c, Set LSB of $s0(bits 7 to 0 are set to 1)

- *Chương trình:*

```
#Laboratory Exercise 4, Home Assignment 2
.text
        li      $s0,0x20194543          #load test value for these function
        ori     $t0, $s0, 0x000000ff    #Set LSB of $s0(bits 7 to 0 are set to 1
```

- *Kết quả:*



Text Segment

Bkpt	Address	Code	Basic	Source
	4194304	0x3c012019	lui $1,0x00002019	3: li $s0,0x20194543 #load test value for these function
	4194308	0x34304543	ori $16,$1,0x00004543	
	4194312	0x360800ff	ori $8,$16,0x000000ff	4: ori $t0, $s0, 0x000000ff #Set LSB of $s0(bits 7 to 0 are set to 1

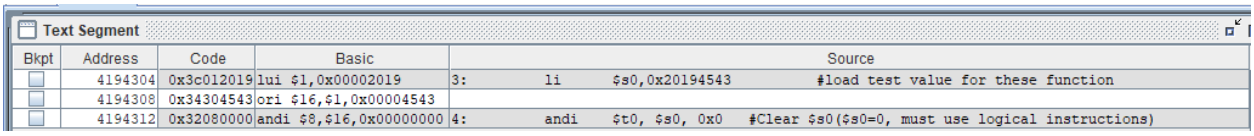
$t0	8	0x201945ff

d, Clear $s0 ($s0 =0, must use logical instructions)

- *Chương trình:*

```
#Laboratory Exercise 4, Home Assignment 2
.text
        li      $s0,0x20194543          #load test value for these function
        andi    $t0, $s0, 0x0   #Clear $s0($s0=0, must use logical instructions)
```

- Kết quả:



Text Segment

Bkpt	Address	Code	Basic	Source
	4194304	0x3c012019	lui $1,0x00002019	3: li $s0,0x20194543 #load test value for these function
	4194308	0x34304543	ori $16,$1,0x00004543	
	4194312	0x32080000	andi $8,$16,0x00000000	4: andi $t0, $s0, 0x0 #Clear $s0($s0=0, must use logical instructions)

$t0	8	0x00000000

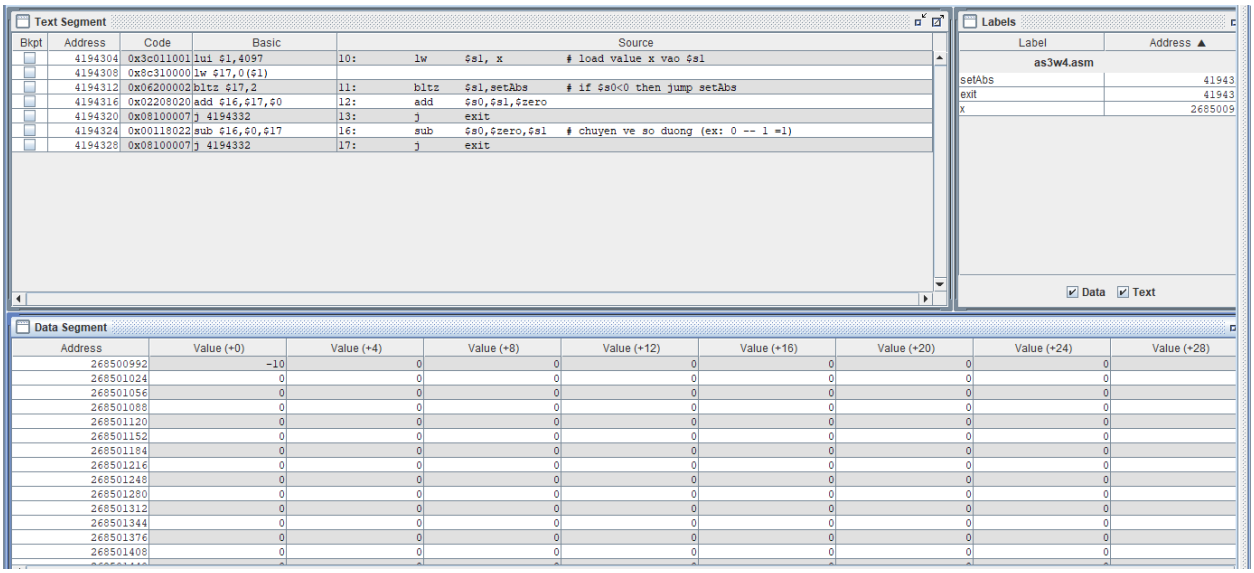
***Bài 3*:**

a, abs $s0,$s1 $s0<= |$s1|

- *Chương trình ($s1 <0):*

```
.data
x:      .word   -10
.text
        lw      $s1, x          # load value x vao $s1
        bltz    $s1,setAbs      # if $s0<0 then jump setAbs
        add     $s0,$s1,$zero
        j       exit

setAbs:
        sub     $s0,$zero,$s1   # chuyen ve so duong (ex: 0 -- 1 =1)
        j       exit
exit:
```

- *Kết quả ($s1<0):*

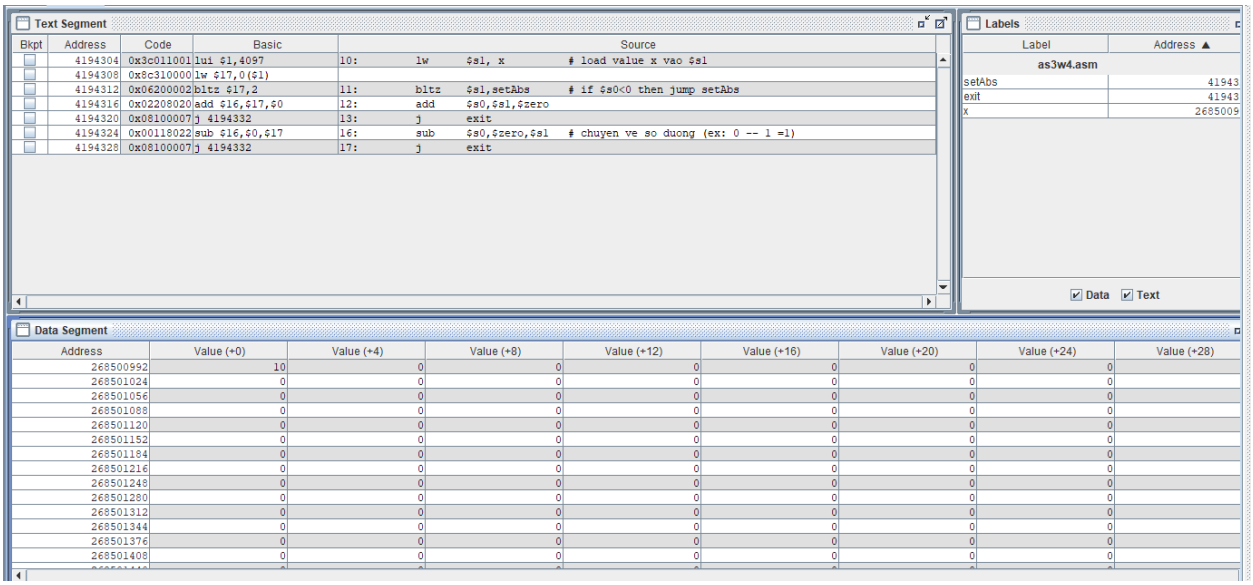


$s0	16	10
$s1	17	-10

- *Chương trình ($s1>0):*

```
.data
x:      .word   10
.text
        lw      $s1, x          # load value x vao $s1
        bltz    $s1,setAbs      # if $s0<0 then jump setAbs
        add     $s0,$s1,$zero
        j       exit

setAbs:
        sub     $s0,$zero,$s1   # chuyen ve so duong (ex: 0 -- 1 =1)
        j       exit
exit:
```

- *Kết quả ($s1>0):*



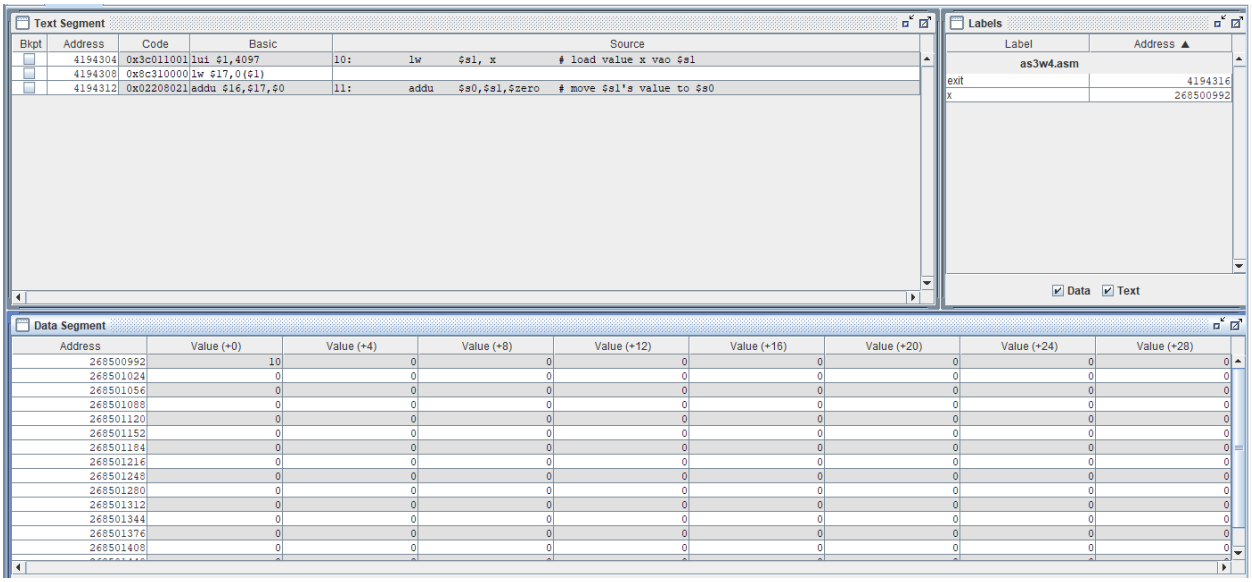
$s0	16	10
$s1	17	10

b, move $s0,$s1 $s0 <=$s1

- *Chương trình:*

```
.data
x:      .word   10
.text
        lw      $s1, x          # load value x vao $s1
        addu    $s0,$s1,$zero   # move $s1's value to $s0
exit:
```

- *Kết quả:*



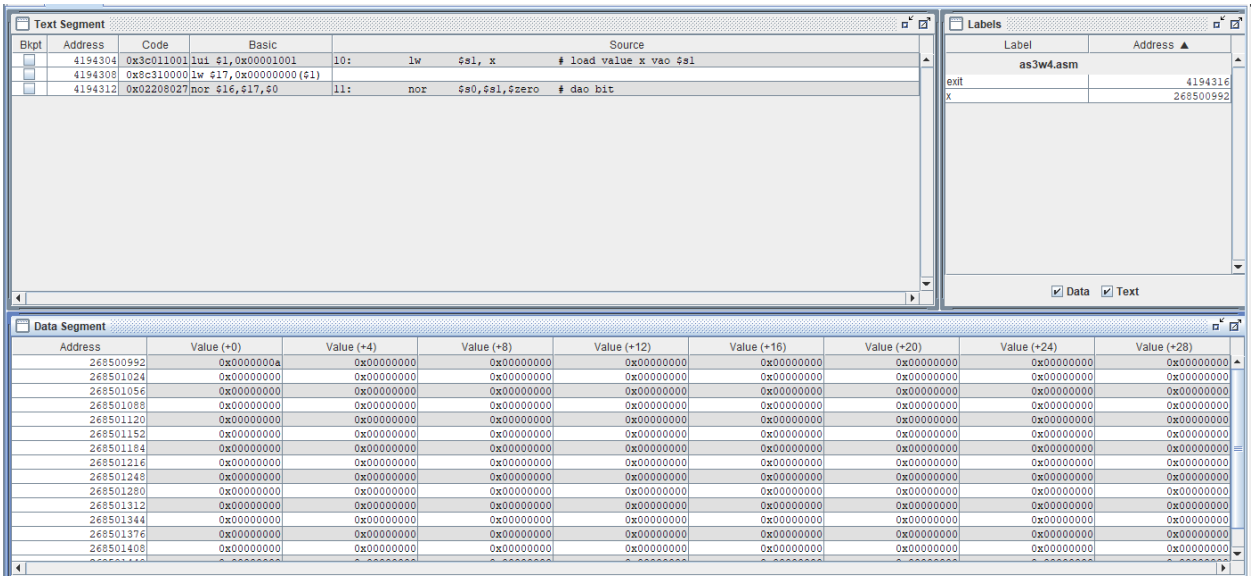
$s0	16	10
$s1	17	10

c, not $s0, $s1 $s0 <= bit invert ($s1)

- *Chương trình:*

```
.data
x:      .word   10
.text
        lw      $s1, x          # load value x vao $s1
        nor     $s0,$s1,$zero   # dao bit
exit:
```

- *Kết quả:*



$s0	16	0xfffffff5
$s1	17	0x0000000a

d, ble $s1,$s2,label if ($s1 <=$s2) j label

- *Chương trình:*

  + TH1: $s1 < $s2

```
.data
x:      .word   -20
y:      .word   10
.text
        lw      $s1, x          # load value x vao $s1
        lw      $s2, y          # load value y vao $s2
        slt     $t0,$s2,$s1     # if $s2 < $s1 then
        beq     $t0,$zero,label
        j       exit

label:
        j       exit
exit:
```

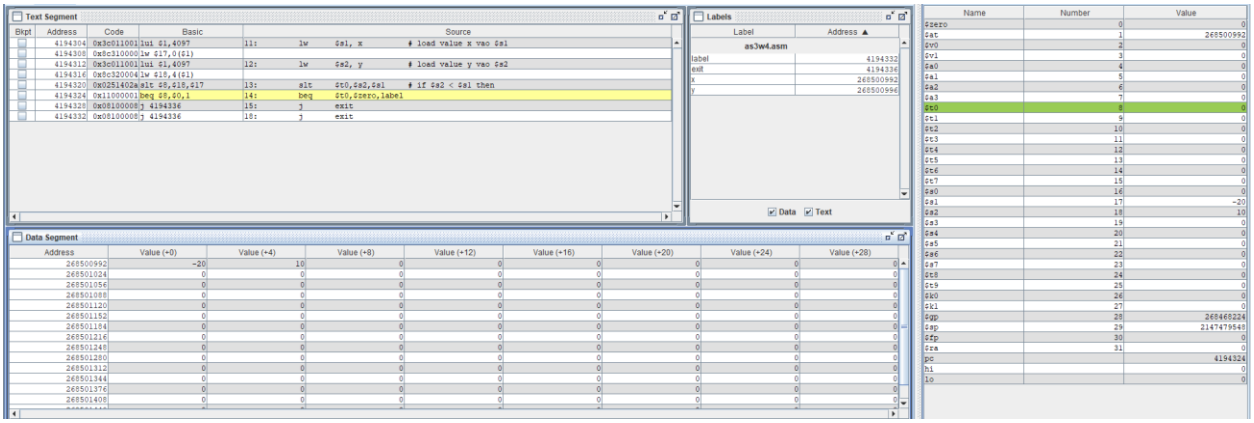
  + TH2: $s1 = $s2

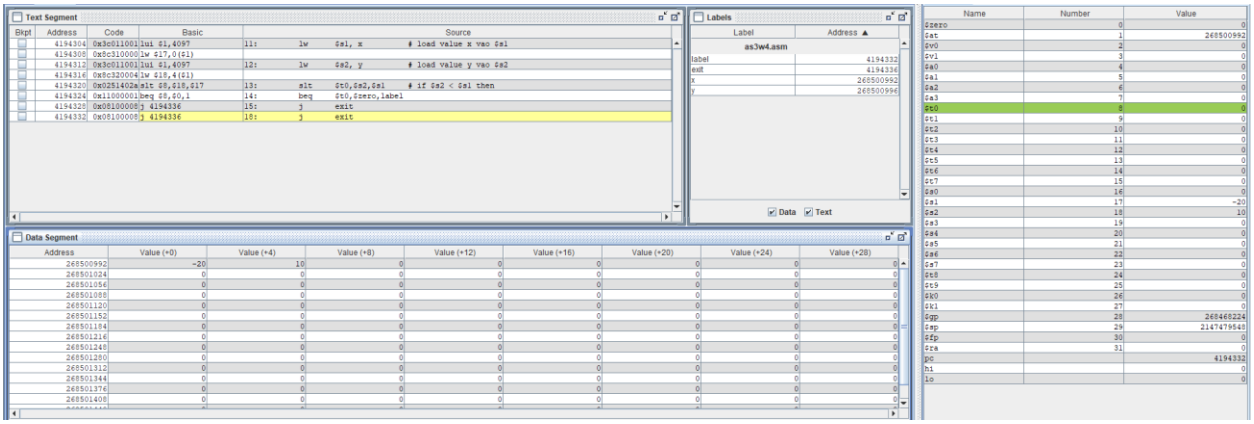
```
.data
x:      .word   10
y:      .word   10
.text
        lw      $s1, x          # load value x vao $s1
        lw      $s2, y          # load value y vao $s2
        slt     $t0,$s2,$s1     # if $s2 < $s1 then
        beq     $t0,$zero,label
        j       exit

label:
        j       exit
exit:
```

+ TH3: $s1 > $s2

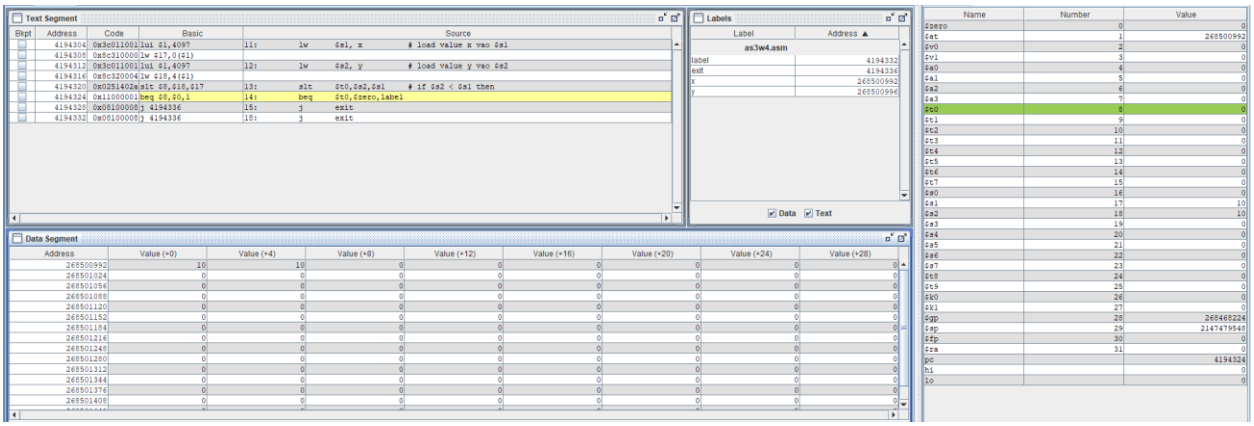
```
.data
x:      .word   20
y:      .word   10
.text
        lw      $s1, x          # load value x vao $s1
        lw      $s2, y          # load value y vao $s2
        slt     $t0,$s2,$s1     # if $s2 < $s1 then
        beq     $t0,$zero,label
        j       exit

label:
        j       exit
exit:
```

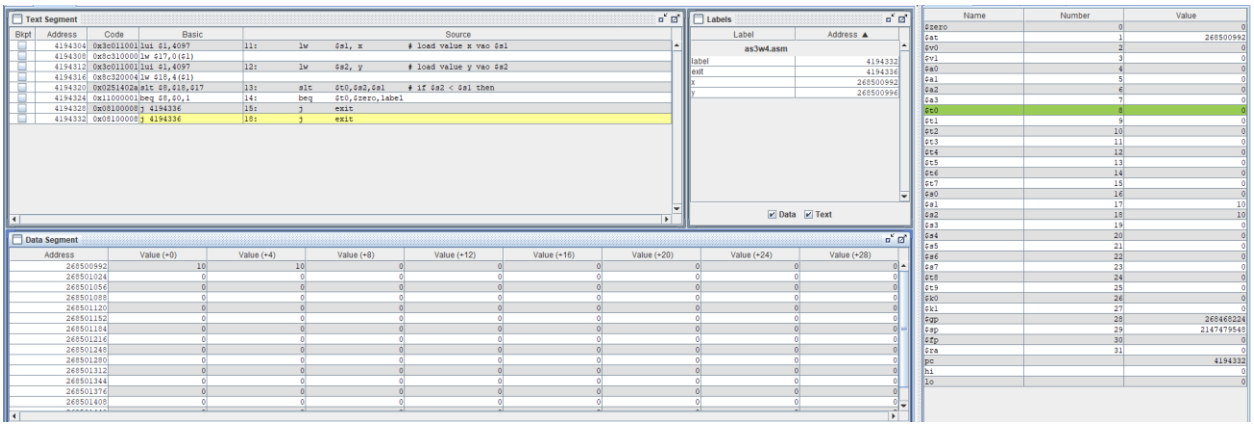
- *Kết quả:*
- + TH1: $s1 < $s2


Text Segment
Bkpt Address Code Basic Source
4194304 0x3c011001 lui $1,4097 11: lw $s1, x # load value x vao $s1
4194308 0x8c310000 lw $17,0($1)
4194312 0x3c011001 lui $1,4097 12: lw $s2, y # load value y vao $s2
4194316 0x8c320004 lw $18,4($1)
4194320 0x0251402a slt $8,$18,$17 13: slt $t0,$s2,$s1 # if $s2 < $s1 then
4194324 0x11000001 beq $8,$0,1 14: beq $t0,$zero,label1
4194328 0x08100008 j 4194336 15: j exit
4194332 0x08100008 j 4194336 16: j exit
Labels
Label Address
as3w4.asm
label 4194332
exit 4194336
x 268500992
y 268500996
Data Text
Data Segment
Address Value (+0) Value (+4) Value (+8) Value (+12) Value (+16) Value (+20) Value (+24) Value (+28)
268500992 -20 10 0 0 0 0 0 0
268501024 0 0 0 0 0 0 0 0
268501056 0 0 0 0 0 0 0 0
268501088 0 0 0 0 0 0 0 0
268501120 0 0 0 0 0 0 0 0
268501152 0 0 0 0 0 0 0 0
268501184 0 0 0 0 0 0 0 0
268501216 0 0 0 0 0 0 0 0
268501248 0 0 0 0 0 0 0 0
268501280 0 0 0 0 0 0 0 0
268501312 0 0 0 0 0 0 0 0
268501344 0 0 0 0 0 0 0 0
268501376 0 0 0 0 0 0 0 0
268501408 0 0 0 0 0 0 0 0
Name Number Value
$zero 0 0
$at 1 268500992
$v0 2 0
$v1 3 0
$a0 4 0
$a1 5 0
$a2 6 0
$a3 7 0
$t0 8 0
$t1 9 0
$t2 10 0
$t3 11 0
$t4 12 0
$t5 13 0
$t6 14 0
$t7 15 0
$s0 16 0
$s1 17 -20
$s2 18 10
$s3 19 0
$s4 20 0
$s5 21 0
$s6 22 0
$s7 23 0
$t8 24 0
$t9 25 0
$k0 26 0
$k1 27 0
$gp 28 268468224
$sp 29 2147479548
$fp 30 0
$ra 31 0
pc 4194324
hi 0
lo 0

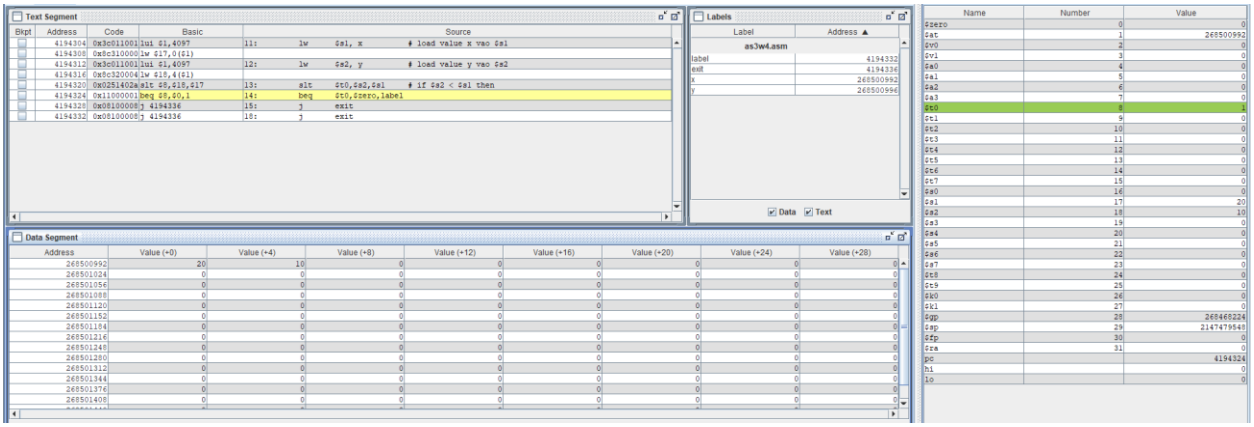

Text Segment
Bkpt Address Code Basic Source
4194304 0x3c011001 lui $1,4097 11: lw $s1, x # load value x vao $s1
4194308 0x8c310000 lw $17,0($1)
4194312 0x3c011001 lui $1,4097 12: lw $s2, y # load value y vao $s2
4194316 0x8c320004 lw $18,4($1)
4194320 0x0251402a slt $8,$18,$17 13: slt $t0,$s2,$s1 # if $s2 < $s1 then
4194324 0x11000001 beq $8,$0,1 14: beq $t0,$zero,label1
4194328 0x08100008 j 4194336 15: j exit
4194332 0x08100008 j 4194336 16: j exit
Labels
Label Address
as3w4.asm
label 4194332
exit 4194336
x 268500992
y 268500996
Data Text
Data Segment
Address Value (+0) Value (+4) Value (+8) Value (+12) Value (+16) Value (+20) Value (+24) Value (+28)
268500992 -20 10 0 0 0 0 0 0
268501024 0 0 0 0 0 0 0 0
268501056 0 0 0 0 0 0 0 0
268501088 0 0 0 0 0 0 0 0
268501120 0 0 0 0 0 0 0 0
268501152 0 0 0 0 0 0 0 0
268501184 0 0 0 0 0 0 0 0
268501216 0 0 0 0 0 0 0 0
268501248 0 0 0 0 0 0 0 0
268501280 0 0 0 0 0 0 0 0
268501312 0 0 0 0 0 0 0 0
268501344 0 0 0 0 0 0 0 0
268501376 0 0 0 0 0 0 0 0
268501408 0 0 0 0 0 0 0 0
Name Number Value
$zero 0 0
$at 1 268500992
$v0 2 0
$v1 3 0
$a0 4 0
$a1 5 0
$a2 6 0
$a3 7 0
$t0 8 0
$t1 9 0
$t2 10 0
$t3 11 0
$t4 12 0
$t5 13 0
$t6 14 0
$t7 15 0
$s0 16 0
$s1 17 -20
$s2 18 10
$s3 19 0
$s4 20 0
$s5 21 0
$s6 22 0
$s7 23 0
$t8 24 0
$t9 25 0
$k0 26 0
$k1 27 0
$gp 28 268468224
$sp 29 2147479548
$fp 30 0
$ra 31 0
pc 4194332
hi 0
lo 0

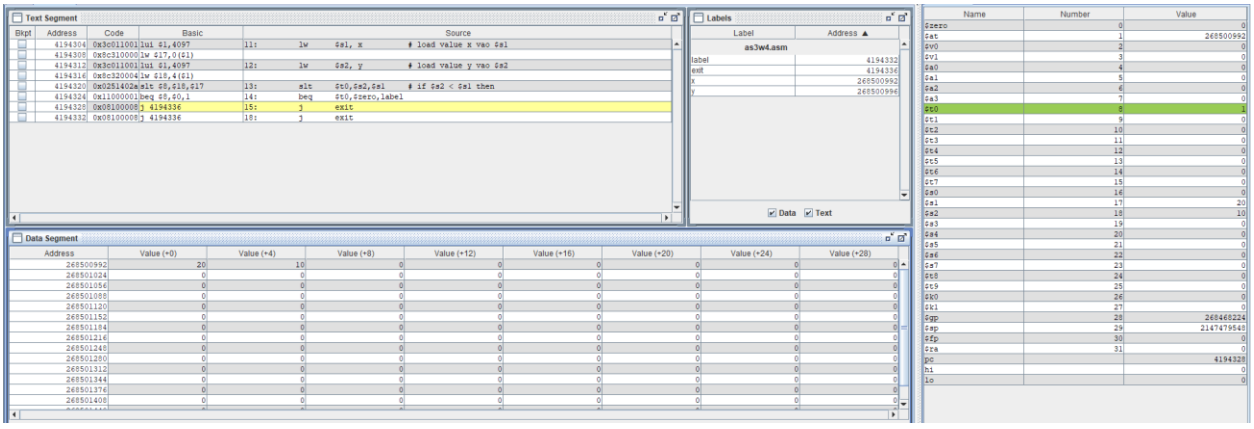
- + TH2: $s1 = $s2


Text Segment
Bkpt Address Code Basic Source
4194304 0x3c011001 lui $1,4097 11: lw $s1, x # load value x vao $s1
4194308 0x8c310000 lw $17,0($1)
4194312 0x3c011001 lui $1,4097 12: lw $s2, y # load value y vao $s2
4194316 0x8c320004 lw $18,4($1)
4194320 0x0251402a slt $8,$18,$17 13: slt $t0,$s2,$s1 # if $s2 < $s1 then
4194324 0x11000001 beq $8,$0,1 14: beq $t0,$zero,label1
4194328 0x08100008 j 4194336 15: j exit
4194332 0x08100008 j 4194336 18: j exit
Labels
Label Address
as3w4.asm
label 4194332
exit 4194336
x 268500992
y 268500996
Data Text
Data Segment
Address Value (+0) Value (+4) Value (+8) Value (+12) Value (+16) Value (+20) Value (+24) Value (+28)
268500992 10 10 0 0 0 0 0 0
268501024 0 0 0 0 0 0 0 0
268501056 0 0 0 0 0 0 0 0
268501088 0 0 0 0 0 0 0 0
268501120 0 0 0 0 0 0 0 0
268501152 0 0 0 0 0 0 0 0
268501184 0 0 0 0 0 0 0 0
268501216 0 0 0 0 0 0 0 0
268501248 0 0 0 0 0 0 0 0
268501280 0 0 0 0 0 0 0 0
268501312 0 0 0 0 0 0 0 0
268501344 0 0 0 0 0 0 0 0
268501376 0 0 0 0 0 0 0 0
268501408 0 0 0 0 0 0 0 0
Name Number Value
$zero 0 0
$at 1 268500992
$v0 2 0
$v1 3 0
$a0 4 0
$a1 5 0
$a2 6 0
$a3 7 0
$t0 8 0
$t1 9 0
$t2 10 0
$t3 11 0
$t4 12 0
$t5 13 0
$t6 14 0
$t7 15 0
$s0 16 0
$s1 17 10
$s2 18 10
$s3 19 0
$s4 20 0
$s5 21 0
$s6 22 0
$s7 23 0
$t8 24 0
$t9 25 0
$k0 26 0
$k1 27 0
$gp 28 268468224
$sp 29 2147479548
$fp 30 0
$ra 31 0
pc 4194324
hi 0
lo 0



- + TH3: $s1 > $s2





Bài 4:

Chương trình:

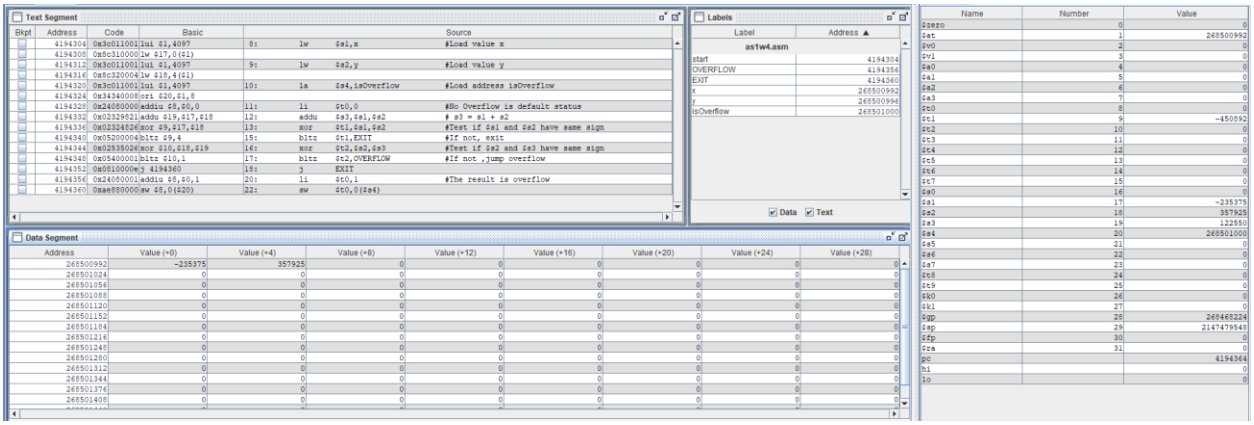
```
#Laboratory Exercise 4, Home Assignment 1
.data
x:              .word   -235375
y:              .word   357925
isOverflow:     .word   0
.text
start:
        lw      $s1,x                   #Load value x
        lw      $s2,y                   #Load value y
        la      $s4,isOverflow          #Load address isOverflow
        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 same sign

        bltz    $t1,EXIT                #If not, exit
        xor     $t2,$s2,$s3             #Test if $s2 and $s3 have same sign
        bltz    $t2,OVERFLOW            #If not ,jump overflow
        j       EXIT
OVERFLOW:
        li      $t0,1                   #The result is overflow
EXIT:
        sw      $t0,0($s4)
```

TH1: x,y trái dấu

- *Chương trình:*

```
#Laboratory Exercise 4, Home Assignment 1
.data
x:              .word   -235375
y:              .word   357925
isOverflow:     .word   0
.text
start:
        lw      $s1,x                   #Load value x
        lw      $s2,y                   #Load value y
        la      $s4,isOverflow          #Load address isOverflow
        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 same sign

        bltz    $t1,EXIT                #If not, exit
        xor     $t2,$s2,$s3             #Test if $s2 and $s3 have same sign
        bltz    $t2,OVERFLOW            #If not ,jump overflow
        j       EXIT
OVERFLOW:
        li      $t0,1                   #The result is overflow
EXIT:
        sw      $t0,0($s4)
```

- *Kết quả:*

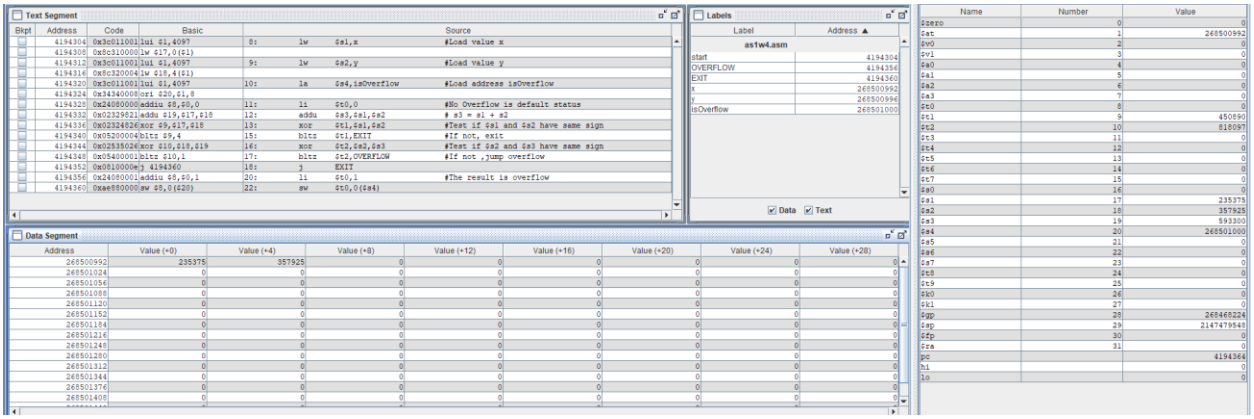


TH2: x,y cùng dương, không tràn

- *Chương trình:*

```
#Laboratory Exercise 4, Home Assignment 1
.data
x:              .word   235375
y:              .word   357925
isOverflow:     .word   0
.text
start:
        lw      $s1,x                   #Load value x
        lw      $s2,y                   #Load value y
        la      $s4,isOverflow          #Load address isOverflow
        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 same sign

        bltz    $t1,EXIT                #If not, exit
        xor     $t2,$s2,$s3             #Test if $s2 and $s3 have same sign
        bltz    $t2,OVERFLOW            #If not ,jump overflow
        j       EXIT
OVERFLOW:
        li      $t0,1                   #The result is overflow
EXIT:
        sw      $t0,0($s4)
```

- *Kết quả:*

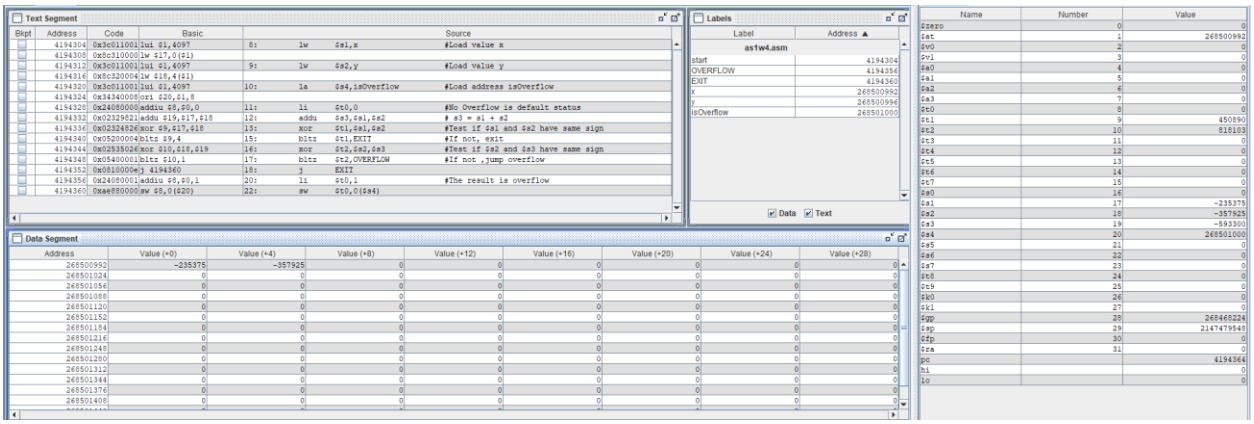


TH3: x,y cùng âm, không tràn

- *Chương trình:*

```
#Laboratory Exercise 4, Home Assignment 1
.data
x:              .word   -235375
y:              .word   -357925
isOverflow:     .word   0
.text
start:
        lw      $s1,x                   #Load value x
        lw      $s2,y                   #Load value y
        la      $s4,isOverflow          #Load address isOverflow
        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 same sign

        bltz    $t1,EXIT                #If not, exit
        xor     $t2,$s2,$s3             #Test if $s2 and $s3 have same sign
        bltz    $t2,OVERFLOW            #If not ,jump overflow
        j       EXIT
OVERFLOW:
        li      $t0,1                   #The result is overflow
EXIT:
        sw      $t0,0($s4)
```

- *Kết quả:*

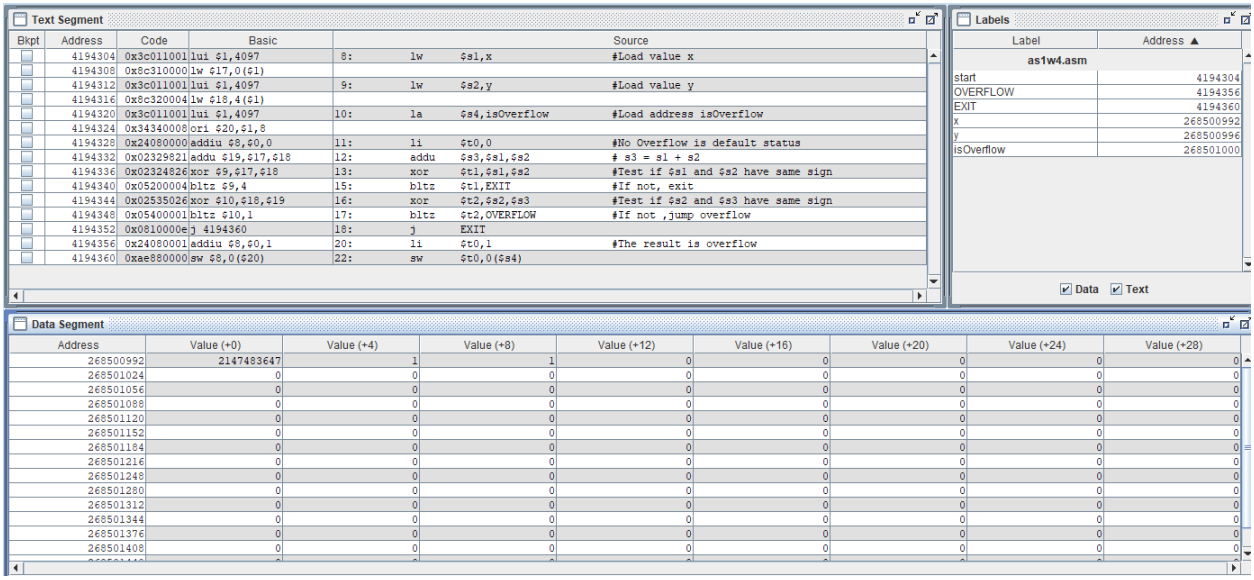


TH4: x,y cùng dương, tràn

- *Chương trình:*

```
#Laboratory Exercise 4, Home Assignment 1
.data
x:              .word   0x7fffffff
y:              .word   1
isOverflow:     .word   0
.text
start:
        lw      $s1,x                   #Load value x
        lw      $s2,y                   #Load value y
        la      $s4,isOverflow          #Load address isOverflow
        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 same sign

        bltz    $t1,EXIT                #If not, exit
        xor     $t2,$s2,$s3             #Test if $s2 and $s3 have same sign
        bltz    $t2,OVERFLOW            #If not ,jump overflow
        j       EXIT
OVERFLOW:
        li      $t0,1                   #The result is overflow
EXIT:
        sw      $t0,0($s4)
```

- *Kết quả:*

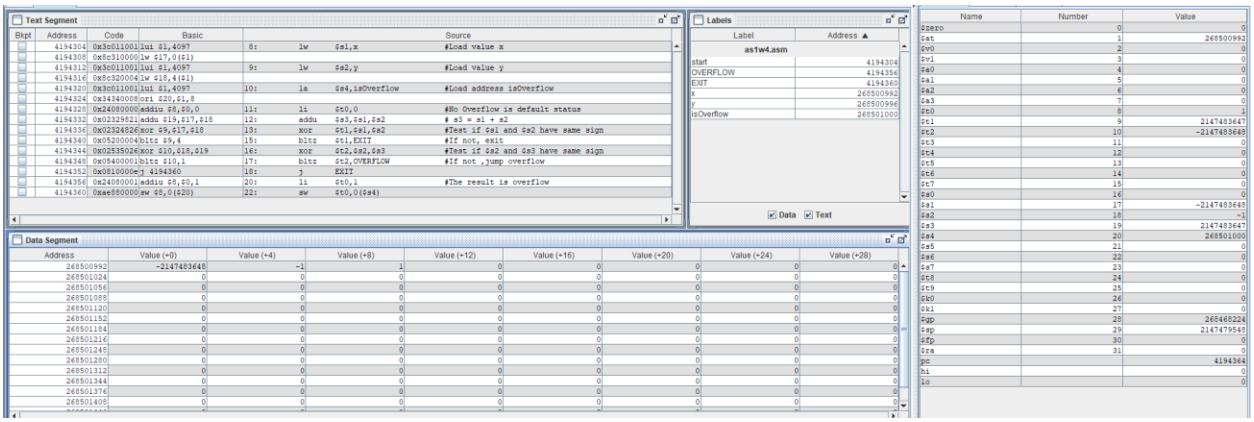


TH5: x,y cùng âm, tràn

- *Chương trình:*

```
#Laboratory Exercise 4, Home Assignment 1
.data
x:              .word   0x80000000
y:              .word   -1
isOverflow:     .word   0
.text
start:
        lw      $s1,x                   #Load value x
        lw      $s2,y                   #Load value y
        la      $s4,isOverflow          #Load address isOverflow
        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 same sign

        bltz    $t1,EXIT                #If not, exit
        xor     $t2,$s2,$s3             #Test if $s2 and $s3 have same sign
        bltz    $t2,OVERFLOW            #If not ,jump overflow
        j       EXIT
OVERFLOW:
        li      $t0,1                   #The result is overflow
EXIT:
        sw      $t0,0($s4)
```

- *Kết quả:*



Bài 5:

- *Chương trình:*

```
#Laboratory Exercise 5
.data
x:      .word   12      #x=12
y:      .word   4       #y=4
i:      .word   0       # bien chay i=0
sum:    .word   0       # tich sum=0
.text
        lw      $s0,x   # load value x
        lw      $s1,y   # load value y
        lw      $s2,i   # load value i
        la      $s3,sum # load address sum
        li      $s4,1   # load 1 to $s4
loop:
        beq     $s1,$s4,endloop # if y=1 then endloop
        srl     $s1,$s1,1       # y=y:2
        addi    $s2,$s2,1       # i=i+1
        j       loop
endloop:
        sllv    $s5,$s0,$s2     # sum=x*y
        sw      $s5,0($s3)      # return sum
```

- *Kết quả:*