BÁO CÁO THỰC HÀNH TUẦN 3

Lương Thị Tâm
20194663

Bài 1:

-TH1: Cộng hai số trái dấu

*Code:

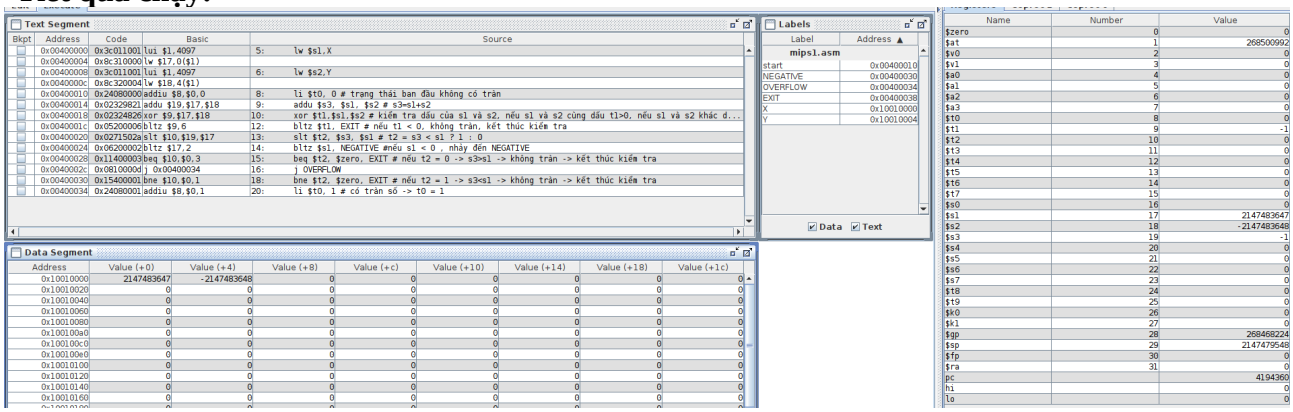
mips1.asm

```
.data
X: .word 0x7fffffff
Y: .word 0x80000000
.text
     lw $s1,X
     lw $s2,Y
start:
     li $t0, 0 # trạng thái ban đầu không có tràn
     addu $s3, $s1, $s2 # s3=s1+s2
     xor $t1,$s1,$s2 # kiểm tra dấu của s1 và s2, nếu s1 và s2 cùng dấu t1>0, nếu s1 và s2 khác dấu t1 < 0

     bltz $t1, EXIT # nếu t1 < 0, không tràn, kết thúc kiểm tra
     slt $t2, $s3, $s1 # t2 = s3 < s1 ? 1 : 0
     bltz $s1, NEGATIVE #nếu s1 < 0 , nhảy đến NEGATIVE
     beq $t2, $zero, EXIT # nếu t2 = 0 -> s3>s1 -> không tràn -> kết thúc kiểm tra
     j OVERFLOW
NEGATIVE:
     bne $t2, $zero, EXIT # nếu t2 = 1 -> s3<s1 -> không tràn -> kết thúc kiểm tra
OVERFLOW:
     li $t0, 1 # có tràn số -> t0 = 1
EXIT:

```

*Kết quả chạy:



-TH2: Cộng hai số dương có tràn số

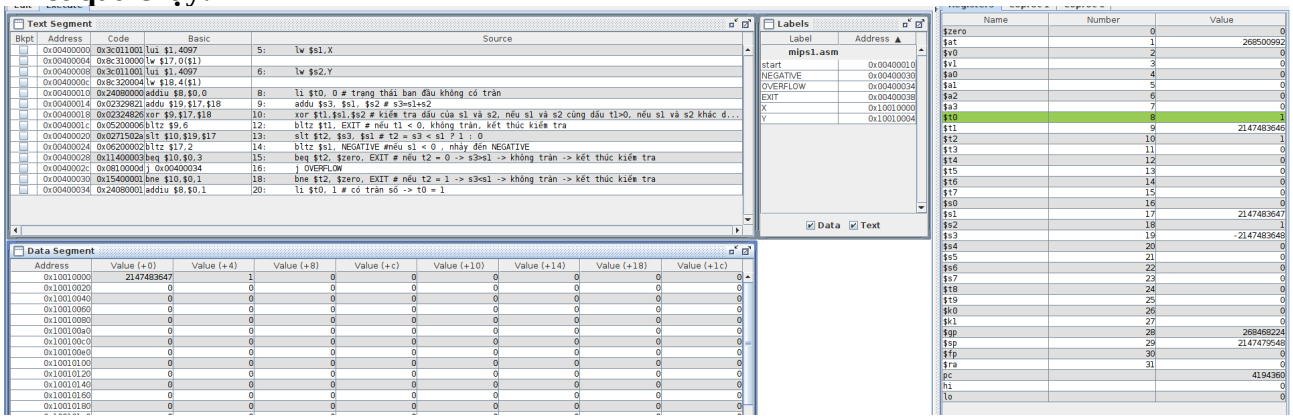
*Code:

```
.data
X: .word 0x7fffffff
Y: .word 1
.text
     lw $s1,X
     lw $s2,Y
start:
     li $t0, 0 # trạng thái ban đầu không có tràn
     addu $s3, $s1, $s2 # s3=s1+s2
     xor $t1,$s1,$s2 # kiểm tra dấu của s1 và s2, nếu s1 và s2 cùng dấu t1>0, nếu s1 và s2 khác dấu t1 < 0

     bltz $t1, EXIT # nếu t1 < 0, không tràn, kết thúc kiểm tra
     slt $t2, $s3, $s1 # t2 = s3 < s1 ? 1 : 0
     bltz $s1, NEGATIVE #nếu s1 < 0 , nhảy đến NEGATIVE
     beq $t2, $zero, EXIT # nếu t2 = 0 -> s3>s1 -> không tràn -> kết thúc kiểm tra
     j OVERFLOW
NEGATIVE:
     bne $t2, $zero, EXIT # nếu t2 = 1 -> s3<s1 -> không tràn -> kết thúc kiểm tra
OVERFLOW:
     li $t0, 1 # có tràn số -> t0 = 1
EXIT:

```

*Kết quả chạy:



-TH3: Cộng hai số dương không tràn số

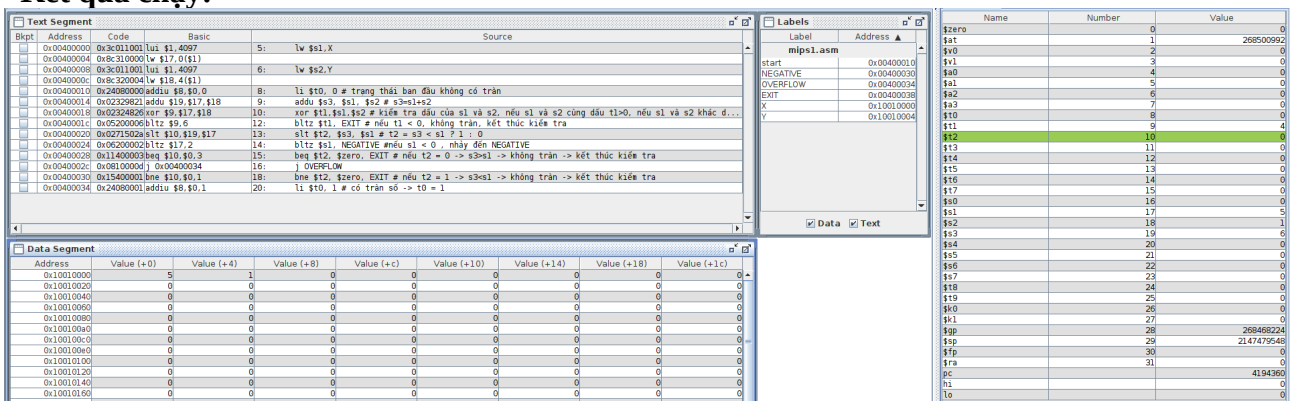
*Code:

```
.data
X: .word 5
Y: .word 1
.text
     lw $s1,X
     lw $s2,Y
start:
     li $t0, 0 # trạng thái ban đầu không có tràn
     addu $s3, $s1, $s2 # s3=s1+s2
     xor $t1,$s1,$s2 # kiểm tra dấu của s1 và s2, nếu s1 và s2 cùng dấu t1>0, nếu s1 và s2 khác dấu t1 < 0

     bltz $t1, EXIT # nếu t1 < 0, không tràn, kết thúc kiểm tra
     slt $t2, $s3, $s1 # t2 = s3 < s1 ? 1 : 0
     bltz $s1, NEGATIVE #nếu s1 < 0 , nhảy đến NEGATIVE
     beq $t2, $zero, EXIT # nếu t2 = 0 -> s3>s1 -> không tràn -> kết thúc kiểm tra
     j OVERFLOW
NEGATIVE:
     bne $t2, $zero, EXIT # nếu t2 = 1 -> s3<s1 -> không tràn -> kết thúc kiểm tra
OVERFLOW:
     li $t0, 1 # có tràn số -> t0 = 1
EXIT:

```

*Kết quả chạy:



-TH4: Cộng hai số âm có tràn số

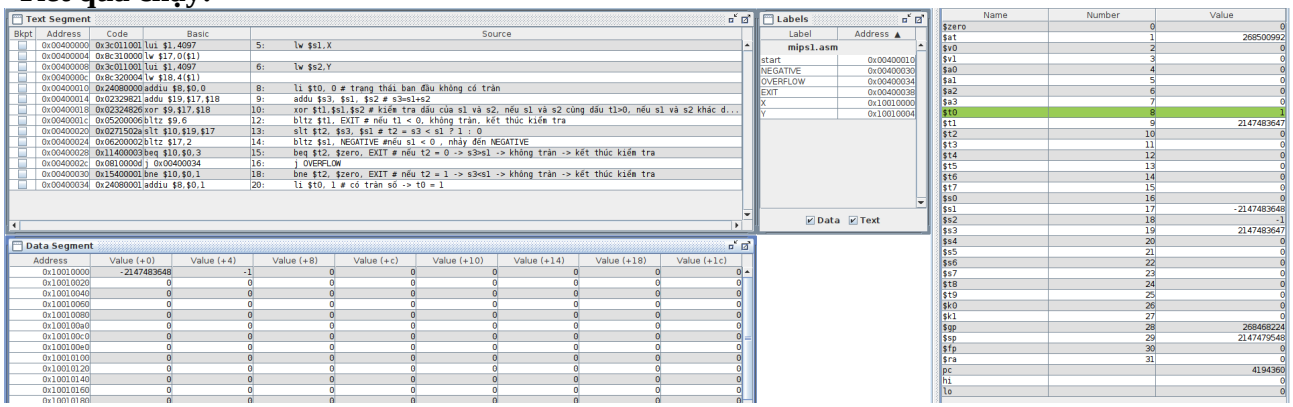
*Code:

```
.data
X: .word 0x80000000
Y: .word -1
.text
     lw $s1,X
     lw $s2,Y
start:
     li $t0, 0 # trạng thái ban đầu không có tràn
     addu $s3, $s1, $s2 # s3=s1+s2
     xor $t1,$s1,$s2 # kiểm tra dấu của s1 và s2, nếu s1 và s2 cùng dấu t1>0, nếu s1 và s2 khác dấu t1 < 0

     bltz $t1, EXIT # nếu t1 < 0, không tràn, kết thúc kiểm tra
     slt $t2, $s3, $s1 # t2 = s3 < s1 ? 1 : 0
     bltz $s1, NEGATIVE #nếu s1 < 0 , nhảy đến NEGATIVE
     beq $t2, $zero, EXIT # nếu t2 = 0 -> s3>s1 -> không tràn -> kết thúc kiểm tra
     j OVERFLOW
NEGATIVE:
     bne $t2, $zero, EXIT # nếu t2 = 1 -> s3<s1 -> không tràn -> kết thúc kiểm tra
OVERFLOW:
     li $t0, 1 # có tràn số -> t0 = 1
EXIT:

```

*Kết quả chạy:



-TH5: Cộng hai số âm không tràn số

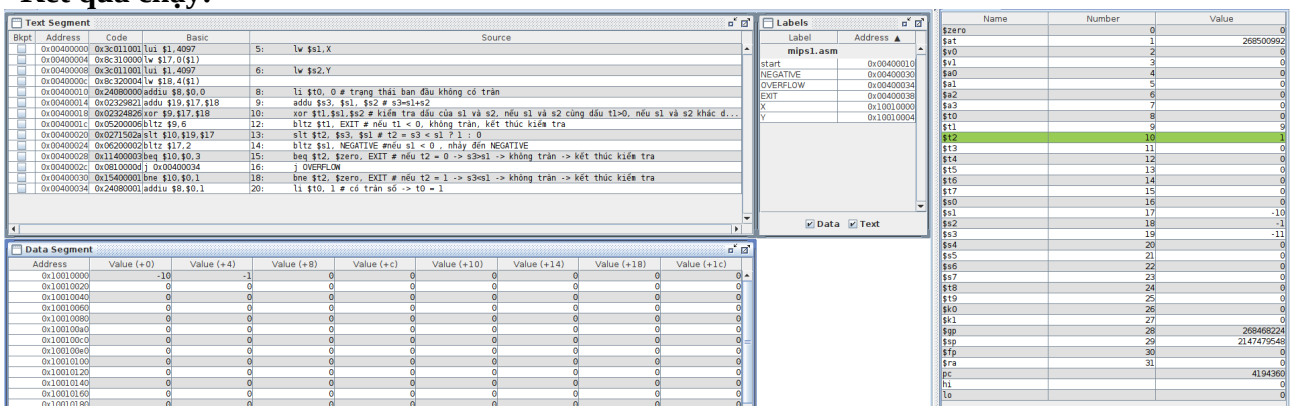
*Code:

```
.data
X: .word -10
Y: .word -1
.text
     lw $s1,X
     lw $s2,Y
start:
     li $t0, 0 # trạng thái ban đầu không có tràn
     addu $s3, $s1, $s2 # s3=s1+s2
     xor $t1,$s1,$s2 # kiểm tra dấu của s1 và s2, nếu s1 và s2 cùng dấu t1>0, nếu s1 và s2 khác dấu t1 < 0

     bltz $t1, EXIT # nếu t1 < 0, không tràn, kết thúc kiểm tra
     slt $t2, $s3, $s1 # t2 = s3 < s1 ? 1 : 0
     bltz $s1, NEGATIVE #nếu s1 < 0 , nhảy đến NEGATIVE
     beq $t2, $zero, EXIT # nếu t2 = 0 -> s3>s1 -> không tràn -> kết thúc kiểm tra
     j OVERFLOW
NEGATIVE:
     bne $t2, $zero, EXIT # nếu t2 = 1 -> s3<s1 -> không tràn -> kết thúc kiểm tra
OVERFLOW:
     li $t0, 1 # có tràn số -> t0 = 1
EXIT:

```

*Kết quả chạy:



Bài 2:

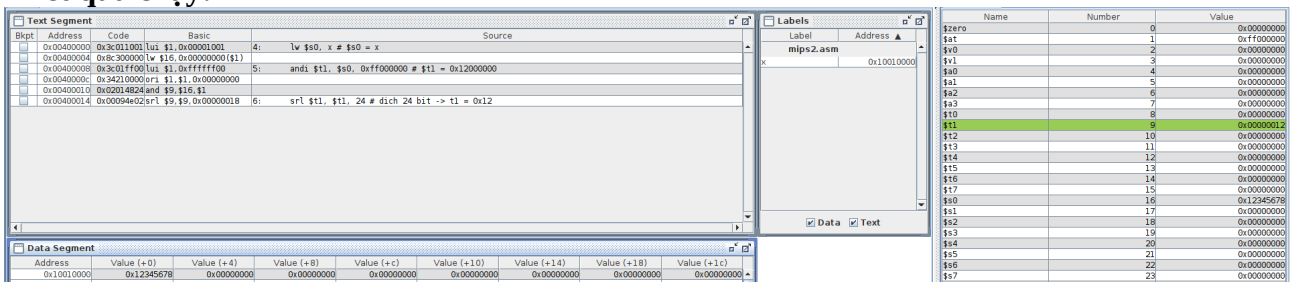
a, Extract MSB of$s0

*Code:

```
.data
x: .word 0x12345678
.text
     lw $s0, x # $s0 = x
     andi $t1, $s0, 0xff000000 # $t1 = 0x12000000
     srl $t1, $t1, 24 # dich 24 bit -> t1 = 0x12


```

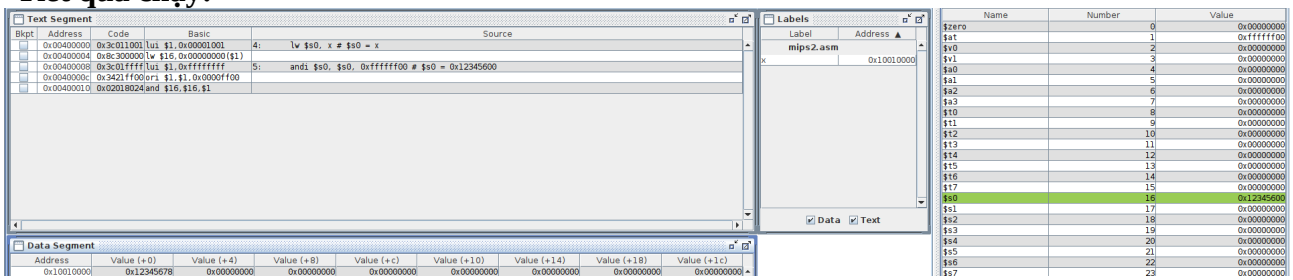
*Kết quả chạy:



b, Clear LSB of$s0
*Code:

```
.data
x: .word 0x12345678
.text
    lw $s0, x # $s0 = x
    andi $s0, $s0, 0xffffff00 # $s0 = 0x12345600
```

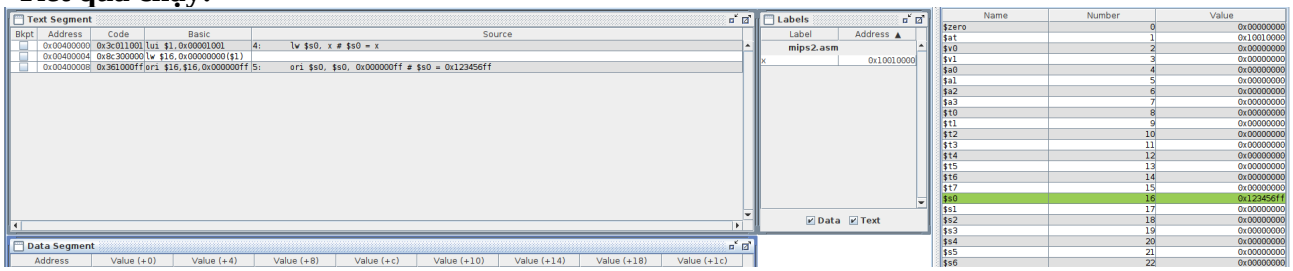
*Kết quả chạy:



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

```
.data
x: .word 0x12345678
.text
    lw $s0, x # $s0 = x
    ori $s0, $s0, 0x000000ff # $s0 = 0x123456ff
```

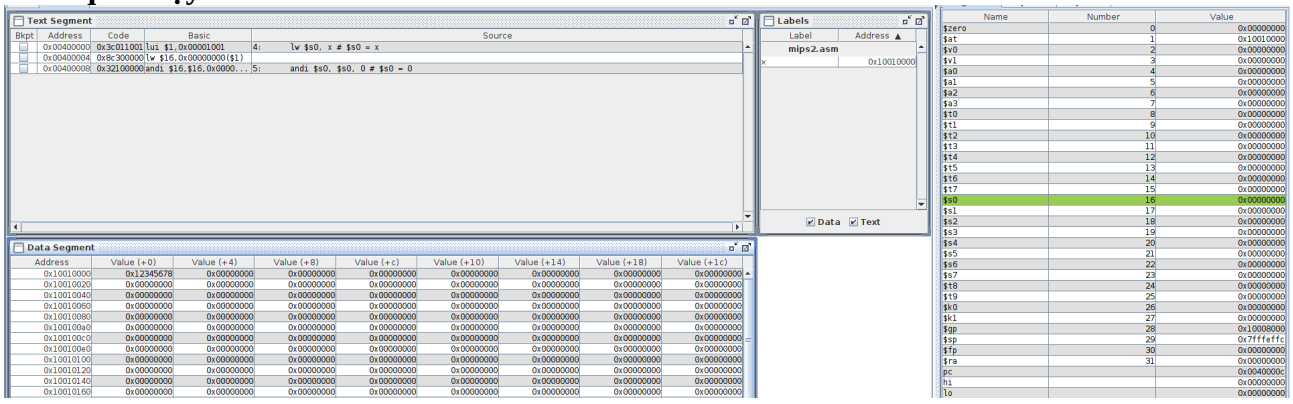
*Kết quả chạy:



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

```
.data
x: .word 0x12345678
.text
    lw $s0, x # $s0 = x
    andi $s0, $s0, 0 # $s0 = 0
```

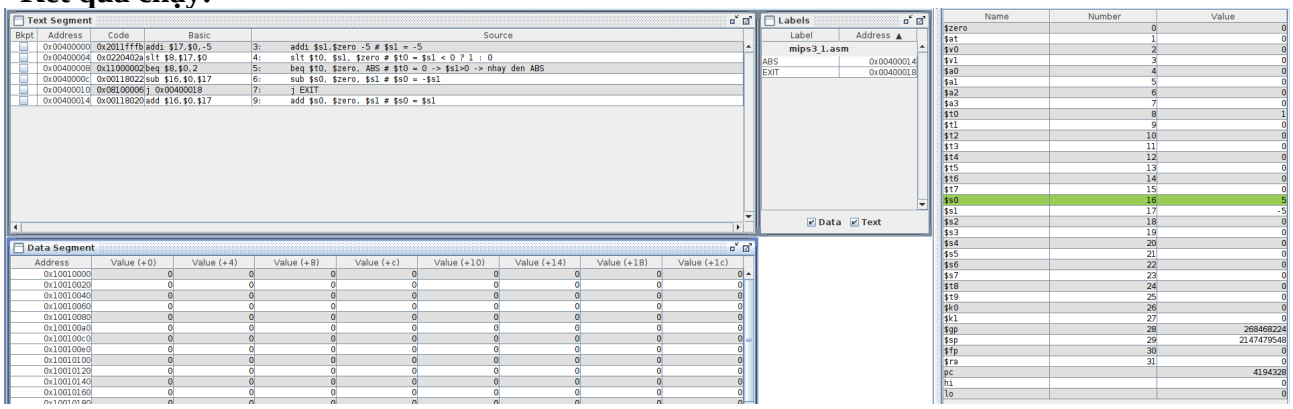
*Kết quả chạy:



Bài 3:
a, abs $s0,$s1
*Code:

```
# abs $s0, $s1
.text
      addi $s1,$zero -5 # $s1 = -5
      slt $t0, $s1, $zero # $t0 = $s1 < 0 ? 1 : 0
      beq $t0, $zero, ABS # $t0 = 0 -> $s1>0 -> nhay den ABS
      sub $s0, $zero, $s1 # $s0 = -$s1
      j EXIT
ABS:
      add $s0, $zero, $s1 # $s0 = $s1
EXIT:
```

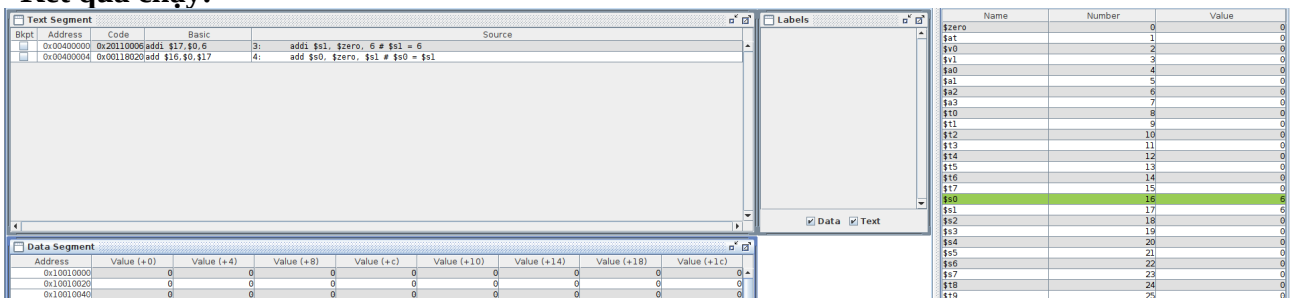
*Kết quả chạy:



b, move $s0, $s1
*Code:

```
# move $s0, $s1
.text
      addi $s1, $zero, 6 # $s1 = 6
      add $s0, $zero, $s1 # $s0 = $s1
```

*Kết quả chạy:

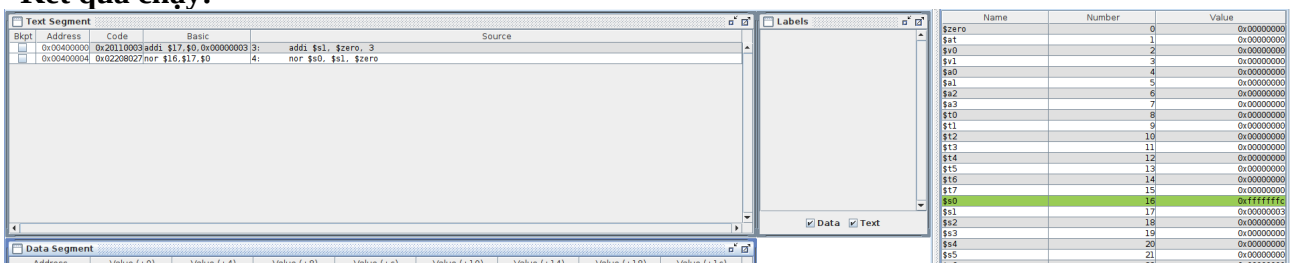


c, not $s0, $s1

*Code:

```
#not $s0, $s1
.text
      addi $s1, $zero, 3
      nor $s0, $s1, $zero
```

*Kết quả chạy:

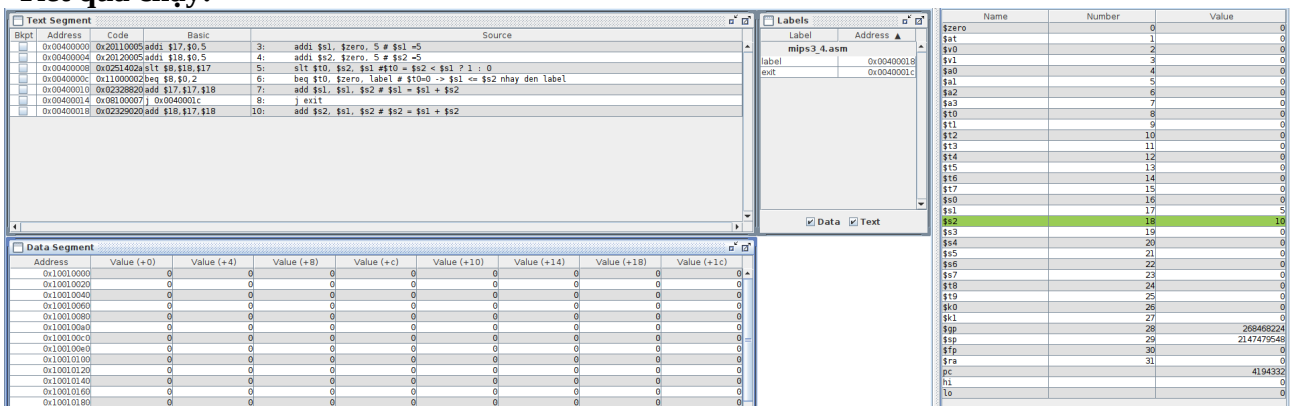


d, ble $s1, $s2, label

*Code:

```
#ble $s1, $s2, lablel
.text
      addi $s1, $zero, 5 # $s1 =5
      addi $s2, $zero, 5 # $s2 =5
      slt $t0, $s2, $s1 #$t0 = $s2 < $s1 ? 1 : 0
      beq $t0, $zero, label # $t0=0 -> $s1 <= $s2 nhay den label
      add $s1, $s1, $s2 # $s1 = $s1 + $s2
      j exit
label:
      add $s2, $s1, $s2 # $s2 = $s1 + $s2
exit:
```

*Kết quả chạy:



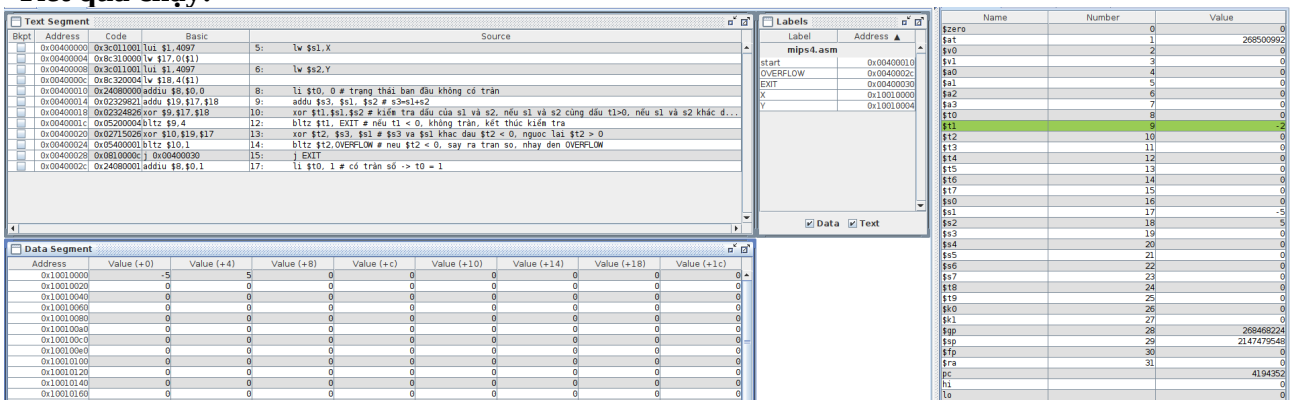
Bài 4:
-TH1: Cộng hai số khác dấu
*Code:

```
.data
X: .word -5
Y: .word 5
.text
      lw $s1,X
      lw $s2,Y
start:
      li $t0, 0 # trạng thái ban đầu không có tràn
      addu $s3, $s1, $s2 # s3=s1+s2
      xor $t1,$s1,$s2 # kiểm tra dấu của s1 và s2, nếu s1 và s2 cùng dấu t1>0, nếu s1 và s2 khác dấu t1 < 0

      bltz $t1, EXIT # nếu t1 < 0, không tràn, kết thúc kiểm tra
      xor $t2, $s3, $s1 # $s3 va $s1 khac dau $t2 < 0, nguoc lai $t2 > 0
      bltz $t2,OVERFLOW # neu $t2 < 0, say ra tran so, nhay den OVERFLOW
      j EXIT
OVERFLOW:
      li $t0, 1 # có tràn số -> t0 = 1
EXIT:

```

*Kết quả chạy:



Text Segment

Bkpt	Address	Code	Basic		Source
	0x00400000	0x3c011001	lui $1,4097	5:	lw $s1,X
	0x00400004	0x8c310000	lw $17,0($1)		
	0x00400008	0x3c011001	lui $1,4097	6:	lw $s2,Y
	0x0040000c	0x8c320004	lw $18,4($1)		
	0x00400010	0x24080000	addiu $8,$0,0	8:	li $t0, 0 # trạng thái ban đầu không có tràn
	0x00400014	0x02329821	addu $19,$17,$18	9:	addu $s3, $s1, $s2 # s3=s1+s2
	0x00400018	0x02324826	xor $9,$17,$18	10:	xor $t1,$s1,$s2 # kiểm tra dấu của s1 và s2, nếu s1 và s2 cùng dấu t1>0, nếu s1 và s2 khác d...
	0x0040001c	0x05200004	bltz $9,4	12:	bltz $t1, EXIT # nếu t1 < 0, không tràn, kết thúc kiểm tra
	0x00400020	0x02715026	xor $10,$19,$17	13:	xor $t2, $s3, $s1 # $s3 va $s1 khac dau $t2 < 0, nguoc lai $t2 > 0
	0x00400024	0x05400001	bltz $10,1	14:	bltz $t2,OVERFLOW # neu $t2 < 0, say ra tran so, nhay den OVERFLOW
	0x00400028	0x08100000c	j 0x00400030	15:	j EXIT
	0x0040002c	0x24080001	addiu $8,$0,1	17:	li $t0, 1 # có tràn số -> t0 = 1

Labels

Label	Address
mips4.asm	
start	0x00400010
OVERFLOW	0x0040002c
EXIT	0x00400030
X	0x10010000
Y	0x10010004

Data Text

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	-5	5	0	0	0	0	0	0
0x10010020	0	0	0	0	0	0	0	0
0x10010040	0	0	0	0	0	0	0	0
0x10010060	0	0	0	0	0	0	0	0
0x10010080	0	0	0	0	0	0	0	0
0x100100a0	0	0	0	0	0	0	0	0
0x100100c0	0	0	0	0	0	0	0	0
0x100100e0	0	0	0	0	0	0	0	0
0x10010100	0	0	0	0	0	0	0	0
0x10010120	0	0	0	0	0	0	0	0
0x10010140	0	0	0	0	0	0	0	0
0x10010160	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	-2
$t2	10	0
$t3	11	0
$t4	12	0
$t5	13	0
$t6	14	0
$t7	15	0
$s0	16	0
$s1	17	-5
$s2	18	5
$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		4194352
hi		0
lo		0

-TH2: Cộng hai số dương không tràn số

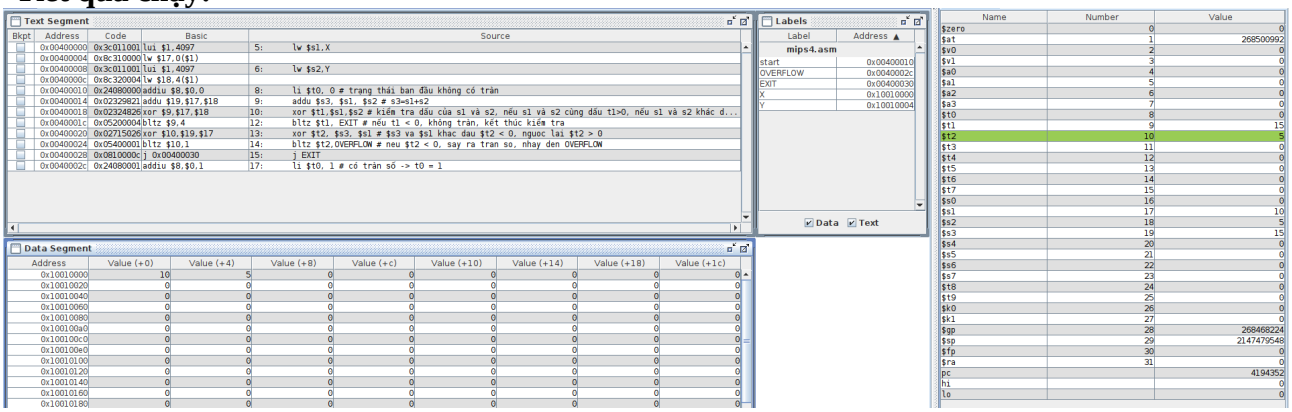
*Code:

```
.data
X: .word 10
Y: .word 5
.text
     lw $s1,X
     lw $s2,Y
start:
     li $t0, 0 # trạng thái ban đầu không có tràn
     addu $s3, $s1, $s2 # s3=s1+s2
     xor $t1,$s1,$s2 # kiểm tra dấu của s1 và s2, nếu s1 và s2 cùng dấu t1>0, nếu s1 và s2 khác dấu t1 < 0

     bltz $t1, EXIT # nếu t1 < 0, không tràn, kết thúc kiểm tra
     xor $t2, $s3, $s1 # $s3 va $s1 khac dau $t2 < 0, nguoc lai $t2 > 0
     bltz $t2,OVERFLOW # neu $t2 < 0, say ra tran so, nhay den OVERFLOW
     j EXIT
OVERFLOW:
     li $t0, 1 # có tràn số -> t0 = 1
EXIT:

```

*Kết quả chạy:



-TH3: Cộng hai số dương có tràn số

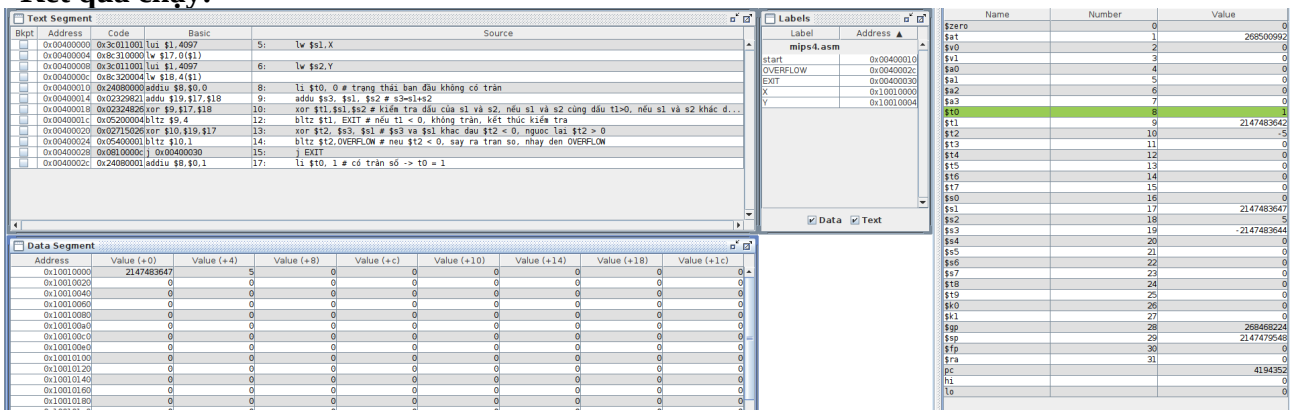
*Code:

```
.data
X: .word 0x7fffffff
Y: .word 5
.text
     lw $s1,X
     lw $s2,Y
start:
     li $t0, 0 # trạng thái ban đầu không có tràn
     addu $s3, $s1, $s2 # s3=s1+s2
     xor $t1,$s1,$s2 # kiểm tra dấu của s1 và s2, nếu s1 và s2 cùng dấu t1>0, nếu s1 và s2 khác dấu t1 < 0

     bltz $t1, EXIT # nếu t1 < 0, không tràn, kết thúc kiểm tra
     xor $t2, $s3, $s1 # $s3 va $s1 khac dau $t2 < 0, nguoc lai $t2 > 0
     bltz $t2,OVERFLOW # neu $t2 < 0, say ra tran so, nhay den OVERFLOW
     j EXIT
OVERFLOW:
     li $t0, 1 # có tràn số -> t0 = 1
EXIT:

```

*Kết quả chạy:



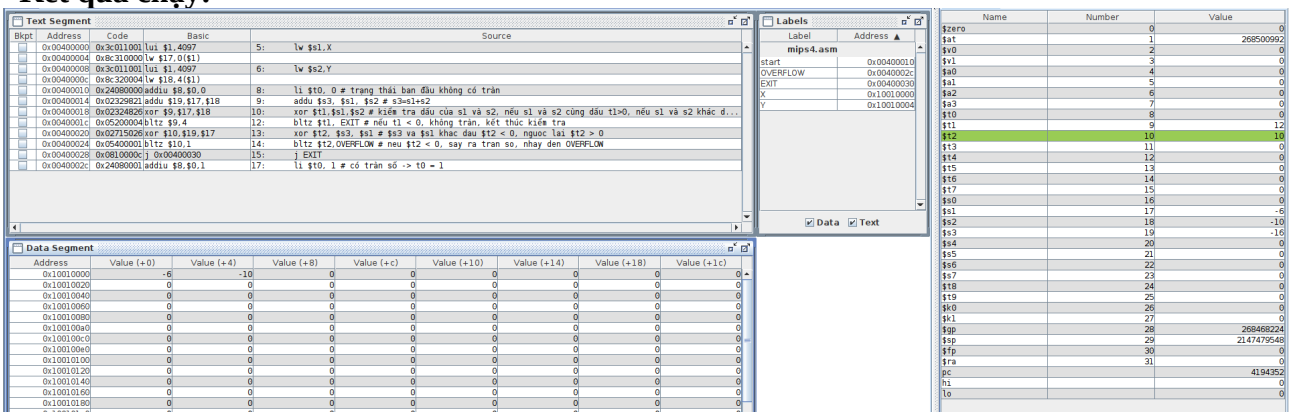
-TH4: Cộng hai số âm không tràn số

*Code:

```
.data
X: .word -6
Y: .word -10
.text
     lw $s1,X
     lw $s2,Y
start:
     li $t0, 0 # trạng thái ban đầu không có tràn
     addu $s3, $s1, $s2 # s3=s1+s2
     xor $t1,$s1,$s2 # kiểm tra dấu của s1 và s2, nếu s1 và s2 cùng dấu t1>0, nếu s1 và s2 khác dấu t1 < 0

     bltz $t1, EXIT # nếu t1 < 0, không tràn, kết thúc kiểm tra
     xor $t2, $s3, $s1 # $s3 va $s1 khac dau $t2 < 0, nguoc lai $t2 > 0
     bltz $t2,OVERFLOW # neu $t2 < 0, say ra tran so, nhay den OVERFLOW
     j EXIT
OVERFLOW:
     li $t0, 1 # có tràn số -> t0 = 1
EXIT:
```

*Kết quả chạy:



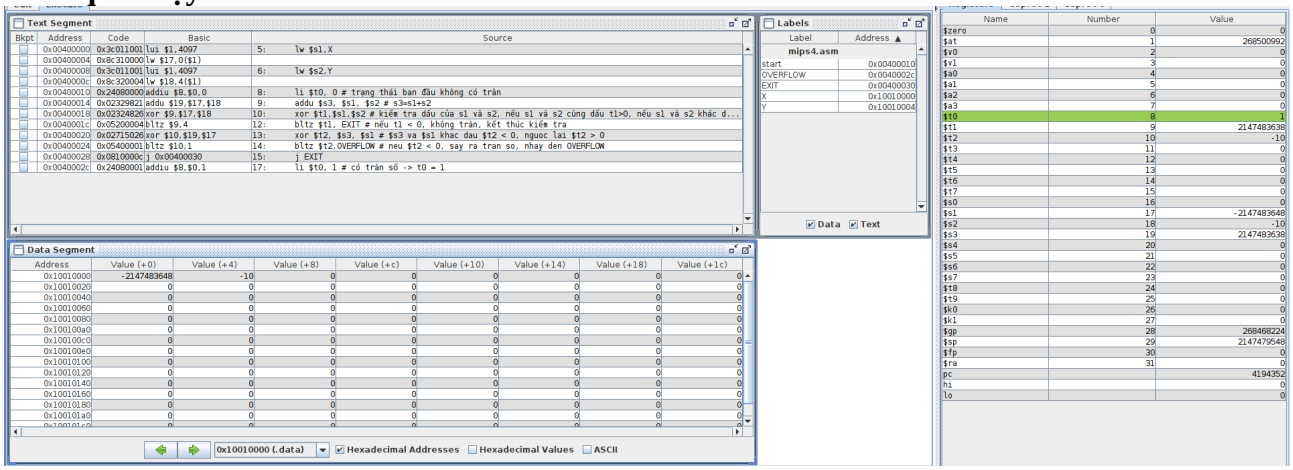
-TH5: Cộng hai số âm có tràn số

*Code:

```
.data
X: .word 0x80000000
Y: .word -10
.text
     lw $s1,X
     lw $s2,Y
start:
     li $t0, 0 # trạng thái ban đầu không có tràn
     addu $s3, $s1, $s2 # s3=s1+s2
     xor $t1,$s1,$s2 # kiểm tra dấu của s1 và s2, nếu s1 và s2 cùng dấu t1>0, nếu s1 và s2 khác dấu t1 < 0

     bltz $t1, EXIT # nếu t1 < 0, không tràn, kết thúc kiểm tra
     xor $t2, $s3, $s1 # $s3 va $s1 khac dau $t2 < 0, nguoc lai $t2 > 0
     bltz $t2,OVERFLOW # neu $t2 < 0, say ra tran so, nhay den OVERFLOW
     j EXIT
OVERFLOW:
     li $t0, 1 # có tràn số -> t0 = 1
EXIT:
```

*Kết quả chạy:



Bài 5:
*Code:

```
.data
x: .word 6 # x = 6
y: .word 8 # y = 8
p: .word 0
i: .word 0 # i = 0
.text
     lw $s1, x # $s1 = x
     lw $s2, y # $s2 = y
     lw $s3, i # $s3 = i
     la $t8, p
     li $t0, 1
loop:
     beq $s2, $t0, enloop # y = 1, ket thuc vong lap
     srl $s2, $s2, 1 # y = y : 2
     addi $s3, $s3, 1 # i = i + 1
     j loop
enloop:
     sllv $t1, $s1, $s3 # p = x*y
     sw $t1, 0($t8)
```

*Kết quả chạy: