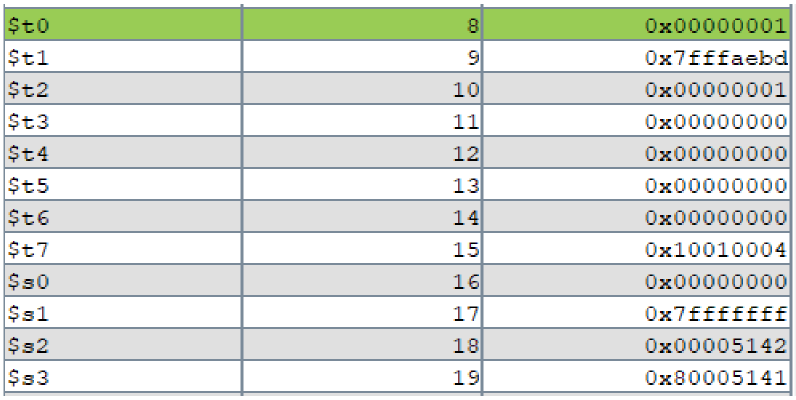
1. Assignment 1

```
#Laboratory Exercise 4, Home Assignment 1
.data
      x:    .word 0x7fffffff
      y:    .word 0x00005142
.text
start:
la $t7, x
lw $s1, 0($t7)
la $t7, y
lw $s2, 0($t7)
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, exit
slt $t2,$s3,$s1
bltz $s1,NEGATIVE #Test if $s1 and $s2 is negative?
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 are negative
# if $s3 < $s1 then the result is not overflow
OVERFLOW:
li $t0,1 #the result is overflow
EXIT:
```

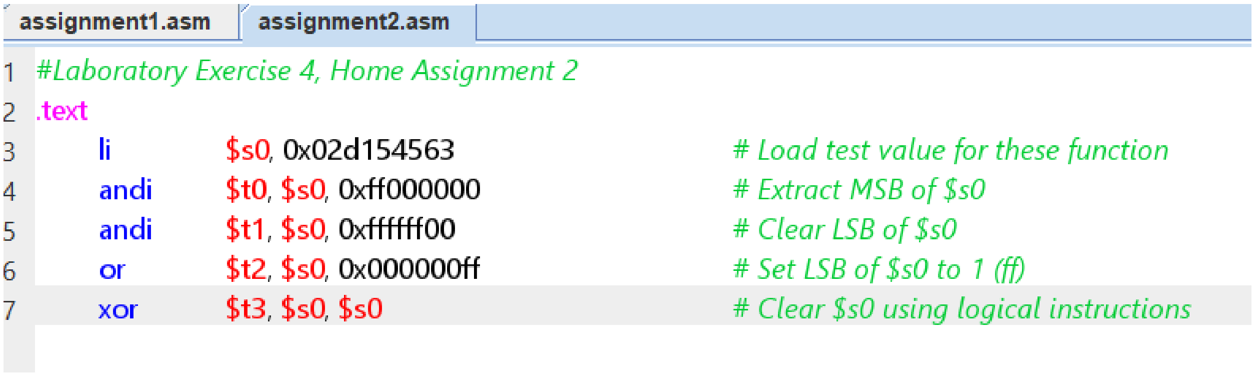
- Các thanh ghi được gán giá trị:
  - $s1 = 0x7fffffff = 01111111111111111111111111111111
  - $s2 = 0x   = 00000000000000000101000101000010
  - $s3 = $s1 + $s2 => $s3 xảy ra overflow do bit MSB = 1



$t0	8	0x00000001
$t1	9	0x7fffaebd
$t2	10	0x00000001
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x10010004
$s0	16	0x00000000
$s1	17	0x7fffffff
$s2	18	0x00005142
$s3	19	0x80005141

- s1 và s2 cùng dấu nên phép xor cho kết quả bit MSB là 0 ($t1 = 0x7fffaebd = 01111111111111111010111010111101) => bltz $t1,EXIT không nhảy đến nhãn EXIT
- slt $t2,$s3,$s1 trả về giá trị của thanh ghi t2 = 1 do s3 = 0x80005141 = 10000000000000000101000101000001 = -2147462847 < s1 = 2147483647). Ta có s1 dương nên chương trình không nhảy đến nhãn NEGATIVE mà tiếp tục thực hiện các câu lệnh tiếp theo
- beq $t2,$zero,EXIT không thực hiện nhảy đến nhãn EXIT do t2 != 0 => j OVERFLOW => thanh ghi $t0 lấy giá trị = 1 thể hiện đã xảy ra hiện tượng tràn số (overflow) do ở đây 2 số dương có dấu trả về kết quả phép cộng là 1 số âm (2147483647 + 20802 = -2147462847 chứ không phải là 2147504449)

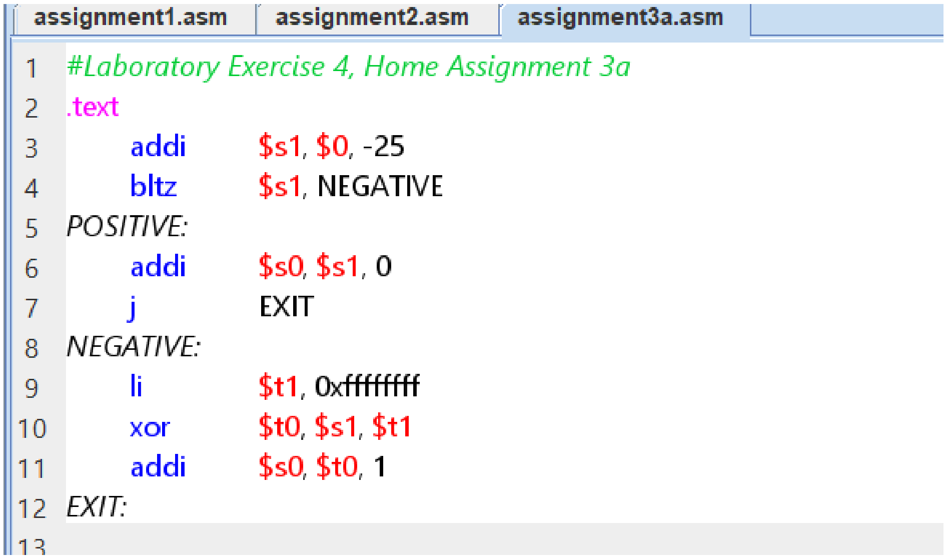
2. Assignment 2



```
#Laboratory Exercise 4, Home Assignment 2
.text
        li      $s0, 0x02d154563            # Load test value for these function
        andi    $t0, $s0, 0xff000000        # Extract MSB of $s0
        andi    $t1, $s0, 0xffffff00        # Clear LSB of $s0
        or      $t2, $s0, 0x000000ff        # Set LSB of $s0 to 1 (ff)
        xor     $t3, $s0, $s0               # Clear $s0 using logical instructions
```

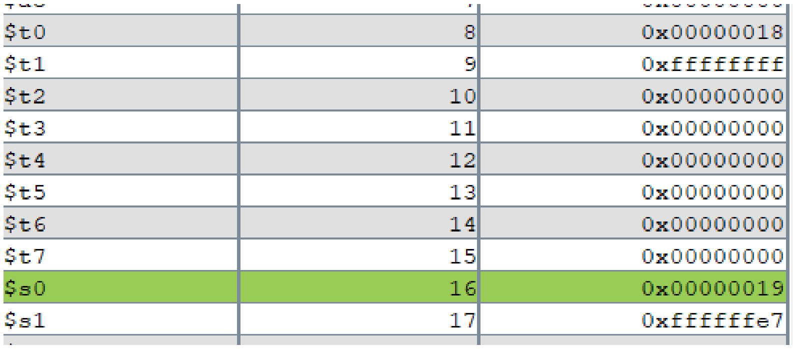
3. Assignment 3

a) Abs



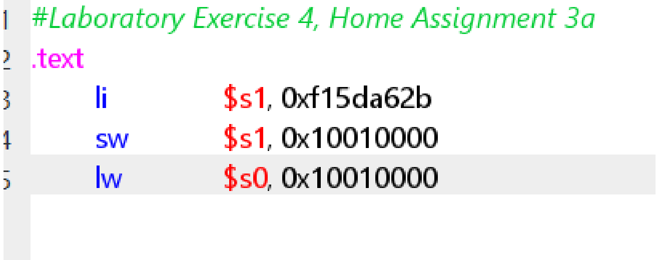
```
#Laboratory Exercise 4, Home Assignment 3a
.text
	addi	$s1, $0, -25
	bltz	$s1, NEGATIVE
POSITIVE:
	addi	$s0, $s1, 0
	j	EXIT
NEGATIVE:
	li	$t1, 0xffffffff
	xor	$t0, $s1, $t1
	addi	$s0, $t0, 1
EXIT:
```

- Kết quả:



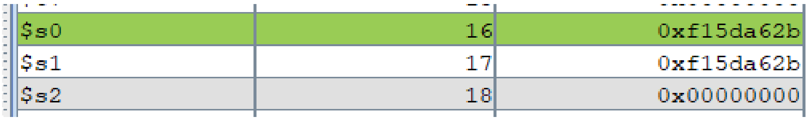
$t0	8	0x00000018
$t1	9	0xffffffff
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000019
$s1	17	0xffffffe7

b) Move



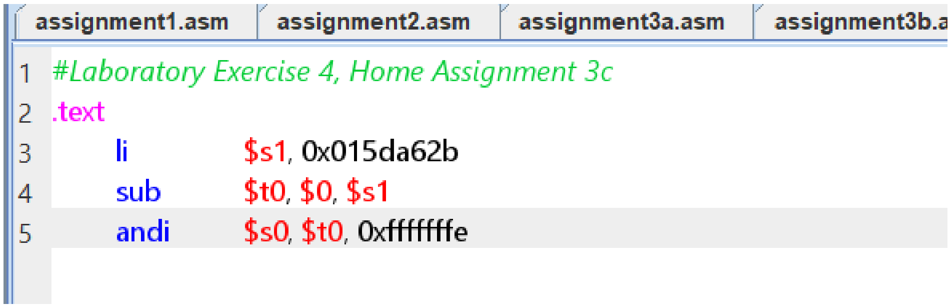
```
#Laboratory Exercise 4, Home Assignment 3a
.text
	li	$s1, 0xf15da62b
	sw	$s1, 0x10010000
	lw	$s0, 0x10010000
```

- Kết quả:



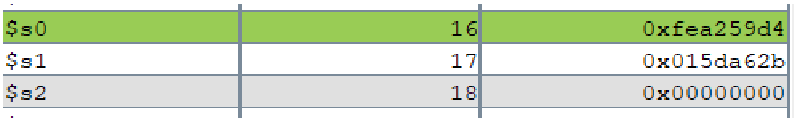
$s0	16	0xf15da62b
$s1	17	0xf15da62b
$s2	18	0x00000000

c) Not



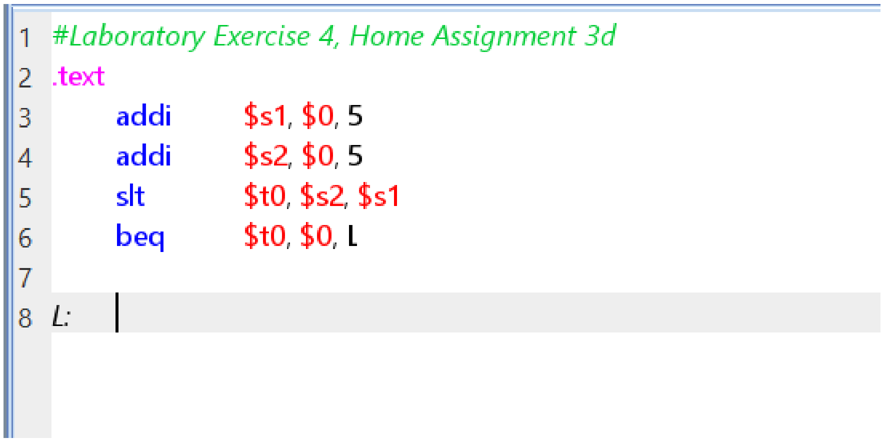
```
#Laboratory Exercise 4, Home Assignment 3c
.text
	li	$s1, 0x015da62b
	sub	$t0, $0, $s1
	andi	$s0, $t0, 0xfffffffe
```

- Kết quả:



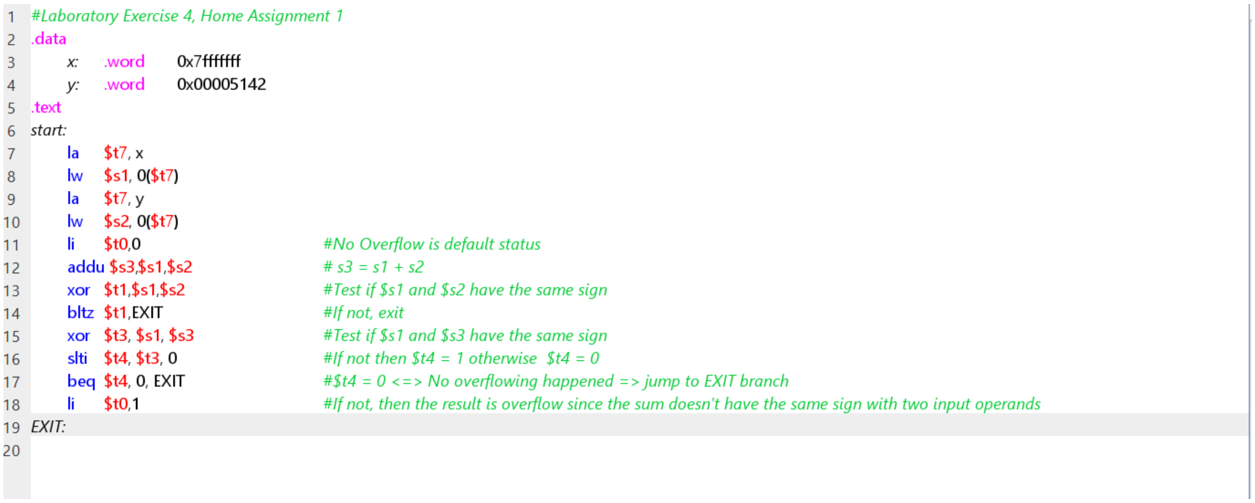
$s0	16	0xfea259d4
$s1	17	0x015da62b
$s2	18	0x00000000

d) Ble



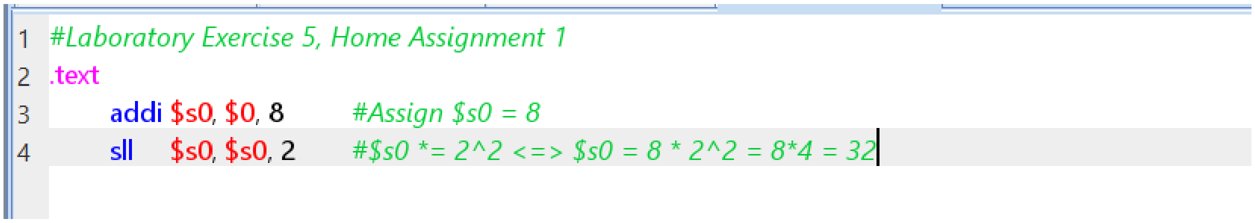
```
#Laboratory Exercise 4, Home Assignment 3d
.text
	addi	$s1, $0, 5
	addi	$s2, $0, 5
	slt	$t0, $s2, $s1
	beq	$t0, $0, L

L:
```

4. Assignment 4



```
#Laboratory Exercise 4, Home Assignment 1
.data
	x:	.word	0x7fffffff
	y:	.word	0x00005142
.text
start:
	la	$t7, x
	lw	$s1, 0($t7)
	la	$t7, y
	lw	$s2, 0($t7)
	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, exit
	xor	$t3, $s1, $s3			#Test if $s1 and $s3 have the same sign
	slti	$t4, $t3, 0			#If not then $t4 = 1 otherwise  $t4 = 0
	beq	$t4, 0, EXIT			#$t4 = 0 <=> No overflowing happened => jump to EXIT branch
	li	$t0,1				#If not, then the result is overflow since the sum doesn't have the same sign with two input operands
EXIT:
```

5. Assignment 5



```
#Laboratory Exercise 5, Home Assignment 1
.text
	addi $s0, $0, 8		#Assign $s0 = 8
	sll	$s0, $s0, 2		#$s0 *= 2^2 <=> $s0 = 8 * 2^2 = 8*4 = 32
```