Week 4

Assignment 1

```
#Laboratory Exercise 4, Assignment 1

.text

      addi $s1, $zero, 1     #Initialize $s1 and $s2
      addi $s2, $zero, 2
start:
      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 $s1 and $s2 dont have same sign, 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:
```

Kết quả của chương trình:

- Với $s1 = 1, $s2 = 2

$v0	2	0x00000000
$v1	3	0x00000000
$a0	4	0x00000000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x00000000
$t1	9	0x00000003
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000000
$s1	17	0x00000001
$s2	18	0x00000002
$s3	19	0x00000003
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000

- Với $s1 = 2^31, $s1 = 2^31

$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x00000001
$t1	9	0x00000000
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000000
$s1	17	0x80000000
$s2	18	0x80000000
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000

Assignment 2

```
#Laboratory Exercise 4, Assignment 2

.text
      addi $s0, $zero, 0x99999999  #Initialize $s0
      andi $t0, $s0, 0xFF000000    #Logical instruction to extract 2 MSB
to $t0
      #Shift bytes in $t0 24 bytes to the right
      srl $t0, $t0, 24       #Shift right by 24 bits

      andi $s0, $s0, 0xFFFFFF00    #Clear 2 LSB from $s0
      xor $s0, $s0, 0x000000FF     #Set 2 LSB of $s0 to 1111 1111
      andi $s0, $s0, 0x00000000    #Clear $s0 to 0
```

Kết quả chương trình trước và sau khi clear s0:

$a3	7	0x00000000
$t0	8	0x00000099
$t1	9	0x00000000
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x999999ff
$s1	17	0x00000000
$s2	18	0x00000000

$a3	7	0x00000000
$t0	8	0x00000099
$t1	9	0x00000000
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000000
$s1	17	0x00000000
$s2	18	0x00000000
$s3	19	0x00000000

Assignment 3

A) `abs $s0,s1`

```
#Laboratory Exercise 4, Assignment 3
```

```
.text
      #Initialize $s1
      addi $s1, $zero, -2

      sra $at, $s1, 31 #Set $at to $s1 shifted right by 31 bytes
      xor $s0, $at, $s1#Logical exclusive or
      subu $s0,$s0,$at #Set $s0 = $s0 - $at
```

B) move $s0,s1

```
#Laboratory Exercise 4, Assignment 3

.text
      #Initialize $s1
      addi $s1, $zero, -2

      addu $s0, $zero, $s1   #Move content of $s1 to $s0
```

C) not $s0

```
#Laboratory Exercise 4, Assignment 3

.text
      #Initialize $s1
      addi $s1, $zero, -2

      nor $s0, $s1, $zero    #Basic instruction of bit inversion
```

D) ble $s1,s2,L

```
#Laboratory Exercise 4, Assignment 3

.text
      #Initialize $s1
      addi $s1, $zero, -2
      addi $s0, $zero, 1

      slt $at, $s1, $s0
      beq $at, $zero, Label

Label:
```

Assignment 4

```
#Laboratory Exercise 4, Assignment 4
.text
start:
      li    $s1, 2147483648        # $s1 = 2147483648
      li    $s2, 2147483648        # $s2 = 2147483648
      li    $t0, 0                 # Overflow indicator (default=0)
      addu  $s3, $s1, $s2          # $s3 = $s1 + $s2
```

```
      xor   $t1, $s1, $s2          # Test if $s1 and $s2 have the same
sign
      bltz  $t1, EXIT        # Branch to EXIT if they have different signs
      xor   $t2, $s1, $s3          # Test if $s1 and $s3 have the same
sign
      bltz  $t2, OVERFLOW          # Branch to OVERFLOW if they have
different signs
      xor   $t2, $s2, $s3          # Test if $s2 and $s3 have the same
sign
      bltz  $t2, OVERFLOW          # Branch to OVERFLOW if they have the
different signs
      j     EXIT
OVERFLOW:
    li            $t0, 1                # Set $t0 to 1 indicating overflow
EXIT:
```

Kết quả chương trình

$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x00000001
$t1	9	0x00000000
$t2	10	0x80000000
$t3	11	0x00000000
$t4	12	0x00000000

Assignment 5

```
#Laboratory Exercise 4, Assignment 5


.text #Calculate n power of 2
      li $t0, 5   #n = 5
      li $t1, 1   #result (temporary set to 1 to shift bytes)

loop:
      sll $t1, $t1, 1        # Shift left by 1 byte
        addi $t0, $t0, -1    # Decrement the exponent
        bnez $t0, loop       # Check if exponent is zero
```

Calculate power of 2 by shifting 1 byte to the left after each loop.