Laboratory Exercise 4

Arithmetic and Logical operation

Họ tên : Đỗ Hải Dương

MSSV : 20194528

Assignment 1:

Assignment 1.asm | mips2.asm

```
.text
start:
        li      $s1, 2147483647
        li      $s2, 4528
        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:
```

*) Trường hợp 1:

- 2 số s1, s2 cùng dấu dương có xảy ra overflow:

$s0	16	0
$s1	17	2147483647
$s2	18	4528
$s3	19	-2147479121

- Hai số s1, s2 cùng dấu nên t1 cùng dấu với s1, s2 .Vì hai số đều dương nên t1 dương => Thực hiện so sánh s3 với s1 . s3 < s1 nên t2 = 1 . Vì t2 = 1 nên overflow => t0 = 1

$t0	8	1
$t1	9	2147479119
$t2	10	1

*) Trường hợp 2 : s1, s2 trái dấu => Không overflow

Assignment 1.asm | mips2.asm

```
.text
start:
        li      $s1, 2147483647
        li      $s2, -4528
        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:
```

- Hai số s1, s2 trái dấu nên giá trị thanh ghi $t1 âm => nhảy đến EXIT luôn mà không kiểm tra overflow

Trường hợp 3 : 2 số s1, s2 cùng dấu âm và không xảy ra overflow:

Assignment 1.asm | mips2.asm

```
.text
start:
        li      $s1, -2019
        li      $s2, -4528
        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:
```

- Thanh ghi:

$s1	17	-2019
$s2	18	-4528
$s3	19	-6547

- Vì 2 số cùng âm nên tại dòng 11 chương trình rẽ sang nhánh NEGATIVE
- Vì s3 < s1 + s2 => $t2 = 1 (không xảy ra overflow)

$t2	10	1
$t3	11	0

- Do đó thanh ghi $t0 vẫn có giá trị bằng 0:

$t0	8	0

Assignment 2 :

a. Extract MSB of $s0

Assignment 1.asm | Assignment 2_a.asm

```
.text
        li      $s0, 0x45282019         #load test value for these function
        andi    $t0, $s0, 0xff000000            #Extract the LSB of $s0
```

Kết quả:

$s0	16	0x45282019
$t0	8	0x45000000

b. Clear LSB of $s0

Assignment 1.asm | Assignment 2_a.asm | Assignment 2_b.asm

```
.text
        li      $s0, 0x45282019         #load test value for these function
        andi    $t0, $s0, 0xffffff00    #Extract the LSB of $s0
```

Kết quả :

$s0	16	0x45282019
$t0	8	0x45282000

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

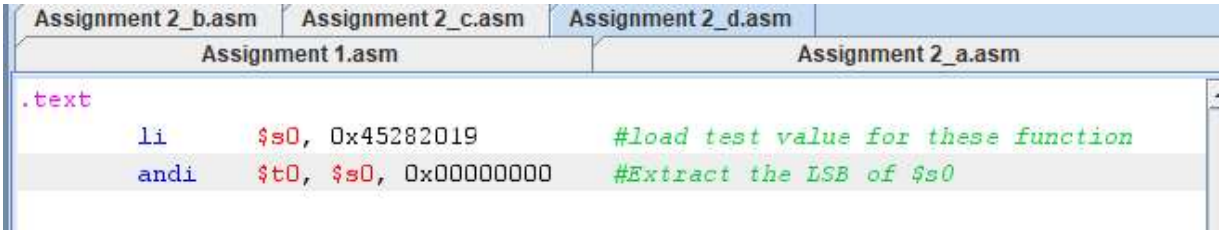
Assignment 1.asm | Assignment 2_a.asm | Assignment 2_b.asm | Assignment 2_c.asm

```
.text
        li      $s0, 0x45282019         #load test value for these function
        ori     $t0, $s0, 0xff          #Extract the LSB of $s0
```

Kết quả :

$s0	16	0x45282019
$t0	8	0x452820ff

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



Kết quả :

$s0	16	0x45282019
$t0	8	0x00000000

Assignment 3:

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

- Kết quả trường hợp $s1 = -4528

```
.text
        li      $s1, -4528      # test value
        bltz    $s1, Convert    # neu s1 < 0
        add     $s0, $s1, 0     # luu result vao $s0
        j       Exit
Convert:
        xori    $s0, $s1, 0xffffffff
        add     $s0, $s0, 1
Exit:
```

$s0	16	4528
$s1	17	-4528
$s2	18	0

- Kết quả trường hợp $s1 = 4528

$s0	16	4528
$s1	17	4528
$s2	18	0

b. **move $s0,s1**

s0 <= $s1

```
.text
        li $s1, 4528     # test value
        add $s0, $s1, 0
```

Kết quả:

$s0	16	4528
$s1	17	4528
$s2	18	0

c. not $s0

s0 <= bit invert (s0)

```
.text
        li      $s0, 0x4528     #test value
        nor     $s1, $s0, $zero
```

Kết quả :

$s0	16	0x00004528
$s1	17	0xffffbad7

Hệ nhị phân :

4528 : 0100010100101000

bad7 : 1011101011010111

⇨ Chính xác

d. ble $s1, s2, L

if (s1 <= $s2)

j L

```
.text
        li      $s0, 4528       # test value
        li      $s1, 1
        li      $s2, 2
        slt     $s3, $s2, $s1
        beq     $s3, $zero, L
        j       EXIT
L:
        li      $s0, 0
EXIT:
```

Kết quả : s1 = 1 < s2 = 2 ➔ Nhảy tới L . s0 = 0

$s0	16	0
$s1	17	1
$s2	18	2

Assignment 4:

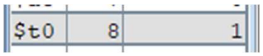
TH1 : Hai số dương :

```
.text
start:
        li      $s1, 2147483647
        li      $s2, 4528
        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     $t2,$s3,$s1
        bltz    $t2, OVERFLOW
        j       EXIT

OVERFLOW:
        li      $t0,1           #the result is overflow
EXIT:
```

Cộng hai số dương . s1 s2 cùng dấu nên t1 >0. Đến bước tiếp , s3 với s1 khác dấu nên t2 <0 ➔ nhảy tới overflow . => t0 = 1



$t0	8	1

TH2 : Hai số trái dấu

```
.text
start:
        li      $s1, 2147483647
        li      $s2, -4528
        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     $t2,$s3,$s1
        bltz    $t2, OVERFLOW
        j       EXIT

OVERFLOW:
        li      $t0,1           #the result is overflow
EXIT:
```

Kết quả : Không overflow

- hai số khác dấu nên t1 < 0 . t1 < 0 nên nhảy luôn sang Exit và thoát .t0 = 0

$t0	8	0

Assignment 5:

- Nhân 2:

```
1  .text
2          li      $s0, 4528       # s0 = 1
3          sll     $s1, $s0, 1     #$s1 = s0 * 2
```

Kết quả:

$s0	16	4528
$s1	17	9056

- Nhân 4:

```
.text
        li      $s0, 4528       # s0 = 1
        sll     $s1, $s0, 2     #$s1 = s0 * 4
```

Kết quả:

$s0	16	4528
$s1	17	18112

- Nhân 8:

```
.text
        li      $s0, 4528       # s0 = 1
        sll     $s1, $s0, 3     #$s1 = s0 * 8
```

Kết quả:

$s0	16	4528
$s1	17	36224

- Nhân 16:

```
.text
        li      $s0, 4528       # s0 = 1
        sll     $s1, $s0, 4     #$s1 = s0 * 16
```

Kết quả:

$s0	16	4528
$s1	17	72448