**20184270\_Phi Quoc Hung**

**Lab Session 4 Assembly**

**Assignment 1:**

* Biến $t0 được thành lập để xét sự tràn số, với mặc định là không
* Cộng 2 giá trị của biến ở thanh ghi $s1, $s2 và lưu vào $s3.
* Kiểm tra xem 2 giá trị ở $s1 và $s2 có cùng dấu hay không, nếu có thì $t1 = 1, còn lại $t1 = 0
* Kiểm tra nếu $t1 =0 thì tiếp tục chương trình, còn nếu $t1=1 thì kết thúc chương trình vì hai số ngược dấu sẽ không dẫn đến overflow
* Nếu $t1=0( cùng dấu):

- Xét xem cùng dấu dương hay âm :

- Nếu cùng âm nhảy đến NEGATIVE, kiểm tra nếu $s3 < $s1 thì không bị overflow và ngược lại

- Nếu cùng dương , kiểm tra nếu $s3 > $s1 thì không bị overflow và ngược lại

**Assignment 2:**

a) Extract MSB of $s0:

.text

li $s0, 0x12345678

andi $t0, $s0, 0xff000000

b) Clear LSB of $s0:

.text

li $s0,0x12345678

andi $t0, $s0, 0xff

sub $s0, $s0, $t0

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

.text

li $s0, 0x12345678

andi $t0, $s0, 0xff

sub $s0, $s0, $t0

addi $s0, $s0, 0x11

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

.text

li $s0, 0x12345678

sll $s0, $s0, 31

**Assignment 3:**

a) abs $s0, $s1

sra $1, $17, 0x0000001f

xor $16, $1, $17

subu $16, $16, 1

b) move $s0, $s1

addu $s0, $zero, $s1

c) not $s0

nor $s0, $s0, $zero

d) le $s1, $s2, L

slt $1, $s2, $s1

beq $1, $zero, L

L:

**Assignment 4:**

(Để cho đẹp phần trình bày, code ở trang dưới)

.text

start:

li $t0, 0

addu $s3, $s1, $s2

xor $t1, $s1, $s2

bltz $t1, EXIT

xor $t2, $s3, $s1

bgtz $s1, EXIT

Overflow:

li $t0, 1

EXIT:

**Assignment 5**:

.text

li $s0, 0x12345678

sll $s0, $s0, 4

Explanation:

Khi nhận một số với 4 trong hệ nhị phân, ta dịch số đó sang trái 2 bit, bởi mỗi lần dịch trái(shift left) thì giá trị của số được nhân lên 2 lần

**Question:**

Sự khác biệt giữa lệnh sll và sllv, srlv và srl:

* Trong lệnh sll, số bit cần dịch sẽ lấy trong trường shamt, còn trong lệnh sllv, số bit cần dịch sẽ lấy trong trường rs
* Trong lệnh srl, số bit cần dịch sẽ lấy trong trường shamt, còn trong lệnh srlv, số bit cần dịch sẽ lấy trong trường rs