# BÁO CÁO THỰC HÀNH KIẾN TRÚC MÁY TÍNH - TUẦN 4

Họ và tên: Nguyễn Mạnh Tùng

MSSV: 20225682

#### **ASSIGNMENT 1:**

```
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 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:</pre>
```

#### - TH1:

```
.text
addi $s1, $zero, -100
addi $s2, $zero, 2000
+ Kết quả:
```

=> Không tràn số, kết quả đúng với lí thuyết.

#### - TH2:

```
.text
addi $s1, $zero, 100
addi $s2, $zero, 200
```

+ Kết quả:



=> Không tràn số, kết quả đúng với lí thuyết.

#### - TH3:

```
.text
addi $s1, $zero, -100
addi $s2, $zero, -200
+ Kết quả:
```

=> Không tràn số, kết quả đúng với lí thuyết.

#### - TH4:

```
.text
addi $s1, $zero, 0x7fffffff
addi $s2, $zero, 200
+Kết quả:
$t0 8 1
```

=> Có tràn số, kết quả đúng với lí thuyết.

#### - TH5:

```
.text
addi $s1, $zero, 0x80000002
addi $s2, $zero, -200
+ Kết quả:
$t0 8 1
```

=> Có tràn số, kết quả đúng với lí thuyết.

#### **ASSIGNMENT 2:**

\$s0 = 0x12345678

- Trích xuất 8-bit đầu của \$s0:

.text

li \$s0, 0x12345678

andi \$t0, \$s0, 0x00ffffff

+ Kết quả:

\$tO	8	0x12000000

- => Kết quả đúng với lí thuyết
- Xóa 8-bit cuối của \$s0:

.text

li \$s0, 0x12345678

andi \$t0, \$s0, 0xffffff00

+ Kết quả:

\$t0	8	0x12345600

- => Kết quả đúng với lí thuyết
- Đặt 8-bit cuối thành bit 1:

.text

li \$s0, 0x12345678

ori \$t0, \$s0, 0x000000ff

+ Kết quả:

\$t0 8 0x123456ff

=> Kết quả đúng với lí thuyết

### - Xóa \$s0:

.text

li \$s0, 0x12345678

andi \$t0, \$s0, 0

+ Kết quả:



=> Kết quả đúng với lí thuyết

#### **ASSIGNMENT 3:**

a) abs \$s0, \$s1 (\$s0 <= |\$s1|)

.text

*li* \$s1, 0xffffffb #\$s1=-5

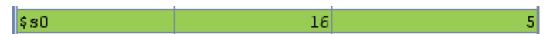
# abs \$s0, \$s1

sra \$at, \$s1, 31 # dịch phải 31-bit

xor \$s0, \$at, \$s1 # xor \$at với \$s1 và gán kq vào \$s0

subu \$s0, \$s0, \$at # thực hiện phép trừ không dấu với \$at=-1 => kết quả

+ Kết quả:



=> Kết quả đúng với lí thuyết.

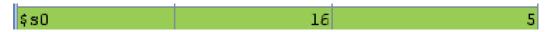
b) move \$s0, \$s1 (\$s0 <= \$s1)

.text

*li* \$\$1, 0x00000005 #\$\$1=5

# addu \$s0, \$zero, \$s1 # thực hiện cộng không dấu \$s1 với \$zero và ghi kq vào \$s0

## + Kết quả:



- => Kết quả đúng với lí thuyết.
- c) not \$s0, \$s1 (\$s0 <= bit invert \$s1)

.text

li \$s1, 0x0f0f0f0f #gán giá trị cho \$s1

nor \$s0, \$s1, \$zero #\$s1 or \$zero => not => kq

## + Kết quả:

\$s0	16	0xf0f0f0f0
\$sl	17	0x0f0f0f0f

- => Kết quả đúng với lí thuyết.
- d) ble \$s1, \$s2, label (if \$s1 <= \$s2; j label)

.text

li \$s1, 0x0000005

li \$s2, 0x00000008

slt \$s0, \$s2, \$s1 # so sánh \$s2 và \$s1

beq \$s0, \$zero, label # nếu \$s0=0 j tới label

label:

exit:

=> Kết quả đúng với lí thuyết

#### **ASSIGNMENT 4:**

.text

li \$t0, 0

*li* \$s1, 0x7fffffff

li \$s2,1

addu \$s3, \$s1, \$s2 # cộng không dấu \$s3=\$s1+\$s2

xor \$t1, \$s1, \$s2 # kiểm tra \$s1 và \$s2 có cùng dấu không

bltz \$t1, exit # nếu \$t1<0, exit

xor \$t2, \$s3, \$s1 # kiểm tra xem \$s1 và \$s3 có cùng dấu không

bgtz \$t2, exit # nếu \$t2>0, exit

j overFlow

overFlow:

li \$t0, 1 # nếu tràn số, \$t0=1

exit:

# + Kết quả:

\$t0	8	0x00000001
------	---	------------

=> Kết quả đúng với lí thuyết.

#### **ASSIGNMENT 5:**

.text

addi \$s0, \$zero, 5 # số để nhân

addi \$s1, \$zero, 1024 # lũy thừa 10 của 2

addi \$t0,\$zero, 1 # \$t0=1

loop:

j loop

exit:

add \$s2, \$zero, \$s0 # ghi kq vào \$s2

# + Kết quả:

\$s0	16	5120
\$s1	17	1
\$s2	18	5120

=> Kết quả đúng với lí thuyết.