

## Thực hành Kiến trúc máy tính tuần 4

Họ tên: Đỗ Hoàng Minh Hiếu

MSSV: 20225837

### Bài 1

TH1: \$s1=0x9, \$s2=0x8

```
1  .text
2  li $s0, 0x9
3  li $s1, 0x8
4  start:
5  li $t0,0 #No Overflow is default status
6  addu $s3,$s1,$s2 # s3 = s1 + s2
7  xor $t1,$s1,$s2 #Test if $s1 and $s2 have the same sign
8  bltz $t1,EXIT #If not, exit
9  slt $t2,$s3,$s1
10 bltz $s1,NEGATIVE #Test if $s1 and $s2 is negative?
11 beq $t2,$zero,EXIT #s1 and $s2 are positive
12 # if $s3 > $s1 then the result is not overflow
13 j OVERFLOW
14 NEGATIVE:
15 bne $t2,$zero,EXIT #s1 and $s2 are negative
16 # if $s3 < $s1 then the result is not overflow
17 OVERFLOW:
18 li $t0,1 #the result is overflow
19 EXIT:
```

Với TH1 cộng hai số cùng dấu thì \$t1 lưu giá trị 1 và \$s3 lưu giá trị 0x11. Do giá trị 0x11 vẫn thuộc dải biểu diễn nên \$t0 không thay đổi giá trị nên \$t0 bằng 0.

TH2: \$s1=0x7fffffff, \$s2=0x1

Với trường hợp này cộng hai số cùng dương với \$s1 là số dương lớn nhất trong dải biểu diễn nên khi cộng tổng 2 số lại thì \$s3 lưu giá trị 0x80000000, là một số âm => Có xảy ra tràn số => \$t0 lưu giá trị 1.

```

mips1.asm
1  .text
2  li $s1, 0x7fffffff
3  li $s2, 0x1
4  start:
5  li $t0,0 #No Overflow is default status
6  addu $s3,$s1,$s2 # s3 = s1 + s2
7  xor $t1,$s1,$s2 #Test if $s1 and $s2 have the same sign
8  bltz $t1,EXIT #If not, exit
9  slt $t2,$s3,$s1
10 bltz $s1,NEGATIVE #Test if $s1 and $s2 is negative?
11 beq $t2,$zero,EXIT #s1 and $s2 are positive
12 # if $s3 > $s1 then the result is not overflow
13 j OVERFLOW
14 NEGATIVE:
15 bne $t2,$zero,EXIT #s1 and $s2 are negative
16 # if $s3 < $s1 then the result is not overflow
17 OVERFLOW:
18 li $t0,1 #the result is overflow
19 EXIT:

```

TH3: \$s1=0xffffffff, \$s2=0x00000014

```

1  .text
2  li $s1, 0xffffffff
3  li $s2, 0x00000014
4  start:
5  li $t0,0 #No Overflow is default status
6  addu $s3,$s1,$s2 # s3 = s1 + s2
7  xor $t1,$s1,$s2 #Test if $s1 and $s2 have the same sign
8  bltz $t1,EXIT #If not, exit
9  slt $t2,$s3,$s1
10 bltz $s1,NEGATIVE #Test if $s1 and $s2 is negative?
11 beq $t2,$zero,EXIT #s1 and $s2 are positive
12 # if $s3 > $s1 then the result is not overflow
13 j OVERFLOW
14 NEGATIVE:
15 bne $t2,$zero,EXIT #s1 and $s2 are negative
16 # if $s3 < $s1 then the result is not overflow
17 OVERFLOW:
18 li $t0,1 #the result is overflow
19 EXIT:

```

Vì hai số là hai số trái dấu nên thực hiện lệnh xor và bltz \$t1,EXIT sẽ nhảy về EXIT và kết thúc chương trình => Không có hiện tượng tràn số => \$t0 vẫn giữ nguyên giá trị 0.

TH4: \$s1=0x80000001, \$s2=0xf0000000

```
.text
li $s1, 0x80000001
li $s2, 0xf0000000
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:
```

Trường hợp này, hai số âm lớn cộng với nhau. Khi cộng hai giá trị lại với nhau, giá trị \$s3 bằng 0x70000001, là một số dương => Có hiện tượng tràn số => \$t0 bằng 1.

TH5: \$s1=-1, \$s2=-2

```
.text
li $s1, -1
li $s2, -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 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 âm đều nhỏ nên tổng của chúng sẽ bằng -3 (0xffffffd) => Không tràn số, cho nên thanh ghi \$t0 bằng 0.

Kết luận: Trong cả 5 trường hợp, giá trị thanh ghi đều đúng với lý thuyết.

## **Bài 2**

.text

li \$s0, 0x12345678

andi \$t0, \$s0, 0xff000000 # dùng lệnh and để giữ lại MSB

srl \$t0, \$t0, 24 # dịch phải để đưa MSB về cuối

addi \$t1, \$s0, 0xfffff00 # xóa 2 byte cuối của \$s0

ori \$t2, \$s0, 0x000000ff # đặt 2 bit cuối

xor \$s0, \$s0, \$s0 # dùng lệnh xor với chính nó để đưa hết về 0

## **Bài 3**

a) sra \$at, \$s1, 31

    xor \$s0, \$at, \$s1

    subu \$s0, \$s0, \$at

b) add \$s0, \$zero, \$s1

c) nor \$s0, \$s1, \$zero

d) slt \$t0, \$s2, \$s1

    beq \$t0, \$zero, label

## **Bài 4**

.text

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

xor \$t2, \$s3, \$s1 # Test if \$s1 and \$s3 have the same sign

bgtz \$t2, EXIT

j OVERFLOW

OVERFLOW:

li \$t0, 1 #the result is overflow

EXIT:

## Bài 5

.text

li \$s0, 23 #so nhan

li \$s1, 1024 #so  $2^n$

li \$t0, 1

loop:

beq \$s1, \$t0, exit # stop if  $s1 = 1$

sra \$s1, \$s1, 1 #  $s1 = s1/2$

sll \$s0, \$s0, 1 #  $s0 = s0*2$

j loop

exit:

Ví dụ:  $s0 = 23$ ,  $s1 = 1024$

\$zero	0	0
\$at	1	0
\$v0	2	0
\$v1	3	0
\$a0	4	0
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$t0	8	1
\$t1	9	0
\$t2	10	0
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$s0	16	23552
\$s1	17	1
\$s2	18	0
\$s3	19	0
\$s4	20	0
\$s5	21	0
\$s6	22	0
\$s7	23	0
\$t8	24	0
\$t9	25	0
\$k0	26	0
\$k1	27	0
\$gp	28	268468224
\$sp	29	2147479548
\$fp	30	0
\$ra	31	0
pc		4194332
hi		0
lo		0

$23*1024=23552$  (Đúng với thanh \$t0 sau khi thực hiện)