Thực hành Kiến trúc máy tính tuần 4

Họ tên: Đỗ Hoàng Minh Hiếu

MSSV: 20225837

<u>Bài 1</u>

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

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

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 \$\$1 là số dương lớn nhất trong dải biểu diễn nên khi cộng tổng 2\$% số lại thì \$\$\$3 lưu giá trị 0\$\$80000000, là một số âm =>\$% tràn số =>\$\$10 lưu giá trị 1.

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

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

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

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?
beg $t2,$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 (0xfffffffd) => 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.

<u>Bài 2</u>

.text

li \$s0, 0x12345678

andi \$t0, \$s0, 0xff000000 # dung lenh and de giu lai MSB

srl \$t0, \$t0, 24 #dich phai de dua MSB ve cuoi

addi \$t1, \$s0, 0xffffff00 #xoa 2 byte cuoi cua \$s0

ori \$t2,\$s0,0x000000ff #dat 2 bit cuoi

xor \$s0, \$s0, \$s0 # dung lenh xor voi chinh no de dua het ve 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:

EXIT:

<u>Bài 5</u>

.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

\$v0	\$zero	0	0
\$v1 3 \$a0 4 \$a1 5 \$a2 6 \$a2 6 \$a3 7 \$5t0 8 \$a5t1 9 \$5t2 10 \$5t3 11 \$5t4 12 \$5t5 13 \$5t6 14 \$6t7 15 \$6s0 16 \$2355 \$6s1 17 \$6s2 18 \$6s3 19 \$6s4 20 \$6s5 21 \$6s6 22 \$6s7 23 \$6t8 24 \$5t9 25 \$6k0 26 \$kk1 27 \$6gp 28 26846822 \$6sp 29 214747954 \$6fp 30 \$5ra 31 \$pc 419433 \$hi	\$at	1	0
\$a0	\$v0	2	0
\$a1	\$vl	3	0
\$a2 6 \$a3 7 \$t0 8 \$t1 9 \$t2 10 \$t2 10 \$t3 11 \$t4 12 \$t5 13 \$t6 14 \$t7 15 \$s0 16 \$s3 19 \$s4 20 \$s3 19 \$s4 20 \$s5 21 \$s6 22 \$s7 23 \$t8 2 \$t8 2 \$t8 2 \$t8 2 \$t8 2 \$s7 2 \$s6 2 \$s7 2 \$s6 2 \$s7 2 \$s6 2 \$t7 2 \$t8 3 \$t8 3 \$t8 3 \$t9 3 \$t8 4 \$t9 3 \$t8 5 \$t8 5 5 5 \$t8 5 5 \$t8 5 5 5 \$t8 5 5 5 \$t8 5 5 5 \$t8 5 5 5 5 \$t8 5 5 5 5 5 5 \$t8 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5	\$a0	4	0
\$a3 7 \$t0 8 \$t1 9 \$t2 10 \$t2 10 \$t3 11 \$t4 12 \$t5 13 \$t6 14 \$t7 15 \$s0 16 2355 \$s1 17 \$s2 18 \$s2 18 \$s3 19 \$s4 20 \$s5 21 \$s6 22 \$s7 23 \$st8 24 \$t0 26 \$k1 27 \$t0 26 \$k1 27 \$s6 22 \$s7 23 \$s6 22 \$s7 23 \$s7 23 \$s6 22 \$s7 23 \$s7 23 \$s7 23 \$s8 24 \$t0 26 \$s8 24 \$t0 25 \$s8 24 \$t0 26 \$s8 24 \$t0 26 \$s8 24 \$t0 26 \$s8 24 \$t0 26 \$s8 26 \$s8 26 \$s8 26 \$s8 36	\$al	5	0
\$t0 8 \$t1 9 \$t2 10 \$t3 11 \$t4 12 \$t5 13 \$t6 14 \$t7 15 \$s0 16 2355 \$s1 17 \$s2 18 \$s3 19 \$s4 20 \$s5 21 \$s6 22 \$s7 23 \$t6 24 \$t7 25 \$s6 22 \$s7 23 \$t8 24 \$t9 25 \$s6 24 \$t9 25 \$s6 27	\$a2	6	0
\$t1 9 \$t2 10 \$t3 11 \$t4 12 \$t5 13 \$t6 14 \$t7 15 \$s0 16 2355 \$s1 17 \$s2 18 \$s3 19 \$s3 19 \$s4 20 \$s5 21 \$s6 22 \$s7 23 \$t8 24 \$t9 25 \$k0 26 \$k1 27 \$gp 28 26846822 \$sp 29 214747954 \$fp 30 \$ra 31 \$pc 419433	\$a3	7	0
\$t2	\$t0	8	1
\$t3	\$t1	9	0
\$t4	\$t2	10	0
\$t5	\$t3	11	0
\$t6	\$t4	12	0
\$t7	\$t5	13	0
\$80 16 2355 \$81 17 \$82 18 \$\$2 18 \$\$3 19 \$\$4 20 \$\$55 21 \$\$56 22 \$\$57 23 \$\$54 24 \$\$59 25 \$\$60 26 \$\$1 27 \$\$9p 28 26846822 \$\$19 \$\$5p 29 214747954 \$\$5p 30 \$\$5ra 31 \$\$pc 419433	\$t6	14	0
\$s1 17 \$s2 18 \$s3 19 \$s4 20 \$s5 21 \$s6 22 \$s7 23 \$t8 24 \$t9 25 \$k0 26 \$k1 27 \$gp 28 26846822 \$sp 29 214747954 \$fp 30 \$ra 31 pc 419433	\$t7	15	0
\$\$2	\$80	16	23552
\$33	\$sl	17	1
\$s4 20 \$s5 21 \$s6 22 \$s7 23 \$t8 24 \$t9 25 \$k0 26 \$k1 27 \$gp 28 26846822 \$sp 29 214747954 \$fp 30 \$ra 31 pc 419433	\$s2	18	0
\$85 21 \$86 22 \$87 23 \$t8 24 \$t9 25 \$k0 26 \$k1 27 \$gp 28 26846822 \$sp 29 214747954 \$fp 30 \$ra 31 pc 419433	\$s3	19	0
\$86 22 \$87 23 \$t8 24 \$t9 25 \$k0 26 \$k1 27 \$gp 28 26846822 \$sp 29 214747954 \$fp 30 \$ra 31 pc 419433	\$s4	20	0
\$87 23 \$t8 24 \$t9 25 \$k0 26 \$k1 27 \$gp 28 26846822 \$sp 29 214747954 \$fp 30 \$ra 31 pc 419433	\$85	21	0
\$t8	\$86	22	0
\$t9	\$87	23	0
\$k0 26 \$k1 27 \$gp 28 26846822 \$sp 29 214747954 \$fp 30 \$ra 31 pc 419433	\$t8	24	0
\$k1 27 \$gp 28 26846822 \$sp 29 214747954 \$fp 30 \$ra 31 pc 419433 hi	\$t9	25	0
\$gp 28 26846822 \$sp 29 214747954 \$fp 30 \$ra 31 pc 419433	\$k0	26	0
\$sp 29 214747954 \$fp 30 \$ra 31 pc 419433 hi	\$kl	27	0
\$sp 29 214747954 \$fp 30 \$ra 31 pc 419433 hi	\$gp	28	268468224
\$ra 31 pc 419433 hi	\$sp	29	2147479548
pc 419433 hi	\$fp	30	0
hi .	\$ra	31	0
hi	pc		4194332
	hi		0
10	10		0

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