

Assignment 1

```
#Laboratory Exercise 4, Home Assignment 1
.text
li $s1 4588
li $s2 10

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 (base on first bit)
bltz $t1,EXIT #If t1 < 0 (not same sign), exit
slt $t2,$s3,$s1 #If s3 < s1 then t2 =1, else t2 = 0
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:
```

TH1:Cùng dấu và tràn

\$s1	17	4588
\$s2	18	2147483600
\$s3	19	-2147479108

Giải thích:

- $s3 = s1 + s2 = 2147488188$
- lệnh xor $\Rightarrow t1 > 0$ (do cùng dấu)
- slt $\Rightarrow t2 = 0$ do $s3 > s1$
- Tất cả các lệnh so sánh để exit đều không đạt \Rightarrow jump đến Overflow
- Gán $t0 = 1$ (overflow)

TH2: Khác dấu và không tràn

\$s1	17	-4588
\$s2	18	2147483600
\$s3	19	2147479012

Giải thích:

- $s3 = s1 + s2 = 2147479012$
- lệnh xor $\Rightarrow t1 < 0$ (do khác dấu)
- Do $t1 < 0 \Rightarrow$ EXIT \Rightarrow not overflow

Assignment 2

Ý a: Extract MSB of \$s0

#Laboratory Exercise 4, Home Assignment 2

.text

```
li $s0, 0x20194588 #load test value for these function
andi $t0, $s0, 0xff000000 #Extract the MSB of $s0
andi $t1, $s0, 0x0400 #Extract bit 10 of $s0
```

Kết quả:

\$t0	8	0x20000000
------	---	------------

Giải thích:

AND s0 với 0xff000000 => Giữ lại 2 bit đầu, chuyển các bit sau về 0

Ý b: Clear LSB of \$s0

#Laboratory Exercise 4, Home Assignment 2

.text

```
li $s0, 0x20194588 #load test value for these function
andi $t0, $s0, 0xffffffff00 #Clear LSB
andi $t1, $s0, 0x0400 #Extract bit 10 of $s0
```

Kết quả:

\$t0	8	0x20194500
------	---	------------

Giải thích: AND s0 với 0xffffffff00 => Giữ lại 6 bit đầu, chuyển 2 bit sau về 0

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

#Laboratory Exercise 4, Home Assignment 2

.text

```
li $s0, 0x20194588 #load test value for these function
ori $t0, $s0, 0xff #Set LSB of $s0 (bits 7 to 0 are set to 1)
andi $t1, $s0, 0x0400 #Extract bit 10 of $s0
```

Kết quả:

\$t0	8	0x201945ff
------	---	------------

Giải thích: Or s0 với 0xff => 2 bit cuối chuyển hết về f, 6 bit đầu giữ nguyên

Ý d: Clear \$s0 (s0=0, must use logical instructions)

```
#Laboratory Exercise 4, Home Assignment 2
.text

li $s0, 0x20194588 #load test value for these function
andi $t0, $s0, 0x00000000 #Clear $s0
andi $t1, $s0, 0x0400 #Extract bit 10 of $s0
```

Kết quả:

\$t0	8	0x00000000
------	---	------------

Giải thích:

AND s0 với 0x00000000 => Tất cả các bit chuyển về 0

Assignment 3

Khai báo:

```
.data
aa: .word 4588
bb: .word -4588
```

Ý a:

```
lw $s2, bb          # load bb to s2, result is s4
bltz $s2, setAbs     # if(s2 < 0) setAbs
# s2 >= 0
addi $s4, $s2, 0     # result to s4
j exit
nop
# s2 < 0
setAbs:
sub $s4, $zero, $s2
j exit
```

Kết quả: $s4 = |s2|$

\$s2	18	-4588
\$s3	19	0
\$s4	20	4588

Giải thích:

- Bltz => Nếu $s2 < 0$ thì gọi setAbs
- setAbs: $s4 = 0 - s2 = 4588$
- Nếu $s2 > 0$ thì $s4 = s2$ rồi exit

Ý b:

bb = 4588

```
lw $s2, bb
addu $s4, $s2, $zero
j exit
```

Kết quả:

\$s2	18	-4588
\$s3	19	0
\$s4	20	-4588

Giải thch:

- Đơn giản là dùng $s4 = s2 + 0$ để gán s2 cho s4

Ý c:

```
lw $s2, bb
nor $s4, $s2 $zero
j exit
```

Kết quả:

\$s2	18	4588
\$s3	19	0
\$s4	20	-4589

Giải thích:

- Nor s2 và 0 => ra not s2 => gán vào s4

Ý d:

```
case4:
li $s0,4588
li $s1,1
li $s2,2019
sle $s3, $s1, $s2
bne $s3,$zero,L
j exit

L:
li $s0,0
exit:
```

Kết quả:

Name	Number	Value
\$zero	0	0
\$at	1	1
\$v0	2	0
\$v1	3	0
\$a0	4	0
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$t0	8	4
\$t1	9	0
\$t2	10	0
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$s0	16	0
\$s1	17	1
\$s2	18	2019
\$s3	19	1
\$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		4194448
hi		0
lo		0

Giải thích:

- Sle=> s1 <=s2 thì s3 =1
- Bne=> nếu s3 khác 0 (s1 <=s2) thì jump đến L
- Còn nếu không thì exit

Assignment 4

```

.text
li $t0, 0          # overflow check, default = 0
li $s1, 4588       # load a
li $s2, 2147483600 # load b

addu $s3, $s1, $s2 # s3 = s1 + s2
xor $t1, $s1, $s2  # t1 = s1 xor s2
bltz $t1, exit     # if (s1 xor s2) < 0 => s1, s2 don't have the same sign => exit
checksum:
xor $t1, $s1, $s3  # check if s1 and (s1 + s2) have the same sign or not
bltz $t1, overflow # if s1 and (s1 + s2) don't have the same sign, then overflow occur
j exit            # else, overflow not occur

overflow:
li $t0, 1

exit:

```

Kết quả: Cùng dấu và tràn

\$s1	17	4588
\$s2	18	2147483600
\$s3	19	-2147479108

Giải thích:

- Addu => $s3 = s1 + s2$
- Xor => xem xem 1 và s2 có cùng dấu hay không (chú ý bit đầu), nếu cùng dấu thì $t1 > 0$ và ngược lại.
- Bltz => nếu $t1 < 0$ => ngược dấu => không thể tràn => exit
- Nếu cùng dấu thì checksum
- checksum: Tiếp tục so sánh s1 và s3 ($= s1 + s2$) và lưu giá trị so sánh vào t1 như trên
- bltz => nếu $t1 < 0$ => Ngược dấu => Tràn

Assignment 5

```

.text
li $s0, 1 #s0=1
sll $s1, $s0, 3 #s1=s0*(2^3)

```

Kết quả:

\$s0	16	1
\$s1	17	8

Giải thích:

- Sử dụng sll (dịch bit trái) => Dịch x bit thì tương đương với nhân với 2^x