

Họ và Tên: Hà Trung Chiến

MSSV: 20225794

Assignment 1:

TH1: \$s1 và \$s2 trái dấu:

The screenshot shows the MIPS simulator interface. The assembly code in the editor is as follows:

```
1 .text
2 start:
3 add $s1, $zero, 0x80000001 # 1000 0000 0000 0000 0000 0000 0001
4 addi $s2, $zero, 2 # 0000 0000 0000 0000 0000 0000 0010
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 bnez $t1, EXIT #if not, exit
9 sll $t2, $s1, 31
10 bltz $s1, NEGATIVE #Test if $s1 and $s2 is negative?
11 beq $t2, $zero, EXIT #s1 and $s2 are positive
12 # if $s1 > $s2 then the result is not overflow
13 } OVERFLOW
14 NEGATIVE:
15 bne $t2, $zero, EXIT #s1 and $s2 are negative
16 # if $s1 < $s2 then the result is not overflow
17 OVERFLOW:
18 li $t0, 1 #the result is overflow
19 EXIT:
```

The register window on the right shows the following values:

Registers	Coproc 1	Coproc 0	Value
\$zero	0		0
\$at	1		-2147483647
\$v0	2		0
\$v1	3		0
\$a0	4		0
\$a1	5		0
\$a2	6		0
\$a3	7		0
\$t0	8		0
\$t1	9		-2147483645
\$t2	10		0
\$t3	11		0
\$t4	12		0
\$t5	13		0
\$t6	14		0
\$t7	15		0
\$s0	16		0
\$s1	17		-2147483647
\$s2	18		2
\$s3	19		-2147483645
\$s4	20		0
\$s5	21		0
\$s6	22		0
\$s7	23		0
\$s8	24		0
\$s9	25		0
\$k0	26		0
\$k1	27		0
\$gp	28		268468224
\$cp	29		2147479540
\$fp	30		0
\$ra	31		0
\$pc			4194360
\$hi			0
\$lo			0

→ Kết quả thu được \$t0 = 0 -> không bị tràn số

TH2: \$s1 và \$s2 cùng dấu dương và không bị tràn:

The screenshot shows the MIPS simulator interface. The assembly code in the editor is as follows:

```
1 .text
2 start:
3 add $s1, $zero, 12 # $s1 = 12
4 addi $s2, $zero, 2 # $s2 = 2
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 bnez $t1, EXIT #if not, exit
9 sll $t2, $s1, 31
10 bltz $s1, NEGATIVE #Test if $s1 and $s2 is negative?
11 beq $t2, $zero, EXIT #s1 and $s2 are positive
12 # if $s1 > $s2 then the result is not overflow
13 } OVERFLOW
14 NEGATIVE:
15 bne $t2, $zero, EXIT #s1 and $s2 are negative
16 # if $s1 < $s2 then the result is not overflow
17 OVERFLOW:
18 li $t0, 1 #the result is overflow
19 EXIT:
```

The register window on the right shows the following values:

Registers	Coproc 1	Coproc 0	Value
\$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		0
\$t1	9		14
\$t2	10		0
\$t3	11		0
\$t4	12		0
\$t5	13		0
\$t6	14		0
\$t7	15		0
\$s0	16		0
\$s1	17		12
\$s2	18		2
\$s3	19		14
\$s4	20		0
\$s5	21		0
\$s6	22		0
\$s7	23		0
\$s8	24		0
\$s9	25		0
\$k0	26		0
\$k1	27		0
\$gp	28		268468224
\$cp	29		2147479540
\$fp	30		0
\$ra	31		0
\$pc			4194352
\$hi			0
\$lo			0

→ Kết quả thu được \$t0 = 0 -> không bị tràn số

TH3: \$s1 và \$s2 cùng dấu dương và bị tràn:

EditExecute

mips1.asm

```

1 .text
2 start:
3 add $a1, $zero, 0x7fffffff # 0111 1111 1111 1111 1111 1111 1111
4 addi $a2, $zero, 1 # 0000 0000 0000 0000 0000 0000 0001
5 li $t0, 0 #No Overflow is default status
6 addu $a3, $a1, $a2 # a3 = a1 + a2
7 xor $t1, $a1, $a2 #Test if $a1 and $a2 have the same sign
8 bltz $t1, EXIT #if not, exit
9 slt $t2, $a3, $a1
10 bltz $a1, NEGATIVE #Test if $a1 and $a2 is negative?
11 beq $t2, $zero, EXIT #a1 and $a2 are positive
12 # if$a3>$a1then the result is not overflow
13 } OVERFLOW
14 NEGATIVE:
15 bne $t2, $zero, EXIT #a1 and $a2 are negative
16 # if $a3<$a1 then the result is not overflow
17 OVERFLOW:
18 li $t0, 1 #the result is overflow
19 EXIT:

```

Line: 4 Column: 70 Show Line Numbers

Mars MessagesRun I/O

RegistersCoproccoproc 0

Name	Number	Value
\$zero	0	0
\$at	1	2147483647
\$v0	2	0
\$v1	3	0
\$a0	4	0
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$t0	8	1
\$t1	9	2147483646
\$t2	10	1
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$a0	16	0
\$a1	17	2147483647
\$a2	18	1
\$a3	19	-2147483648
\$s0	20	0
\$s1	21	0
\$s2	22	0
\$s3	23	0
\$s4	24	0
\$s5	25	0
\$s6	26	0
\$s7	27	0
\$gp	28	268460224
\$sp	29	2147479548
\$fp	30	0
\$ra	31	4194360
\$hi		0
\$lo		0

→ Kết quả thu được \$t0 = 1 -> bị tràn số

TH4: \$s1 và \$s2 cùng dấu âm và bị tràn:

EditExecute

mips1.asm

```

1 .text
2 start:
3 add $a1, $zero, 0x80000001 # 1000 0000 0000 0000 0000 0000 0001
4 add $a2, $zero, 0x80000002 # 1000 0000 0000 0000 0000 0000 0010
5 li $t0, 0 #No Overflow is default status
6 addu $a3, $a1, $a2 # a3 = a1 + a2
7 xor $t1, $a1, $a2 #Test if $a1 and $a2 have the same sign
8 bltz $t1, EXIT #if not, exit
9 slt $t2, $a3, $a1
10 bltz $a1, NEGATIVE #Test if $a1 and $a2 is negative?
11 beq $t2, $zero, EXIT #a1 and $a2 are positive
12 # if$a3>$a1then the result is not overflow
13 } OVERFLOW
14 NEGATIVE:
15 bne $t2, $zero, EXIT #a1 and $a2 are negative
16 # if $a3<$a1 then the result is not overflow
17 OVERFLOW:
18 li $t0, 1 #the result is overflow
19 EXIT:

```

Line: 4 Column: 71 Show Line Numbers

Mars MessagesRun I/O

RegistersCoproccoproc 0

Name	Number	Value
\$zero	0	0
\$at	1	-2147483646
\$v0	2	0
\$v1	3	0
\$a0	4	0
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$t0	8	1
\$t1	9	3
\$t2	10	0
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$a0	16	0
\$a1	17	-2147483647
\$a2	18	-2147483646
\$a3	19	3
\$s0	20	0
\$s1	21	0
\$s2	22	0
\$s3	23	0
\$s4	24	0
\$s5	25	0
\$s6	26	0
\$s7	27	0
\$gp	28	268460224
\$sp	29	2147479548
\$fp	30	0
\$ra	31	4194368
\$hi		0
\$lo		0

→ Kết quả thu được \$t0 = 1 -> bị tràn số

TH5: \$s1 và \$s2 cùng dấu âm và không bị tràn:

EditExecute

mips1.asm

```

1 .text
2 start:
3 add $a1, $zero, 0xffffffff # 1000 0000 0000 0000 0000 0000 0001
4 add $a2, $zero, 0xffffffff # 1000 0000 0000 0000 0000 0000 0010
5 li $t0, 0 #No Overflow is default status
6 addu $a3, $a1, $a2 # a3 = a1 + a2
7 xor $t1, $a1, $a2 #Test if $a1 and $a2 have the same sign
8 bltz $t1, EXIT #if not, exit
9 slt $t2, $a3, $a1
10 bltz $a1, NEGATIVE #Test if $a1 and $a2 is negative?
11 beq $t2, $zero, EXIT #a1 and $a2 are positive
12 # if$a3>$a1then the result is not overflow
13 } OVERFLOW
14 NEGATIVE:
15 bne $t2, $zero, EXIT #a1 and $a2 are negative
16 # if $a3<$a1 then the result is not overflow
17 OVERFLOW:
18 li $t0, 1 #the result is overflow
19 EXIT:

```

Line: 4 Column: 28 Show Line Numbers

Mars MessagesRun I/O

RegistersCoproccoproc 0

Name	Number	Value
\$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	0
\$t1	9	0
\$t2	10	1
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$a0	16	0
\$a1	17	-1
\$a2	18	-1
\$a3	19	-2
\$s0	20	0
\$s1	21	0
\$s2	22	0
\$s3	23	0
\$s4	24	0
\$s5	25	0
\$s6	26	0
\$s7	27	0
\$gp	28	268460224
\$sp	29	2147479548
\$fp	30	0
\$ra	31	4194352
\$hi		0
\$lo		0

→ Kết quả thu được \$t0 = 0 -> không bị tràn số

Assignment 2:

- Trích xuất 8 bit đầu:

The screenshot shows the MIPS simulator interface. The assembly code in the editor is as follows:

```
1 .text
2 li $a0, 0x12345678
3 andi $t0, $a0, 0xf0000000
4 srl $t0, $t0, 24
```

The register window on the right shows the state of the registers. The register \$t0 is highlighted, showing its value as 0x00000012.

→ Kết quả thu được tại \$t0 = 0x00000012

- Xóa 8 bit cuối:

The screenshot shows the MIPS simulator interface. The assembly code in the editor is as follows:

```
1 .text
2 li $a0, 0x12345678
3 andi $t0, $a0, 0xf0000000
4 srl $t0, $t0, 8
```

The register window on the right shows the state of the registers. The register \$t0 is highlighted, showing its value as 0x00123456.

→ Kết quả thu được tại \$t0 = 0x00123456

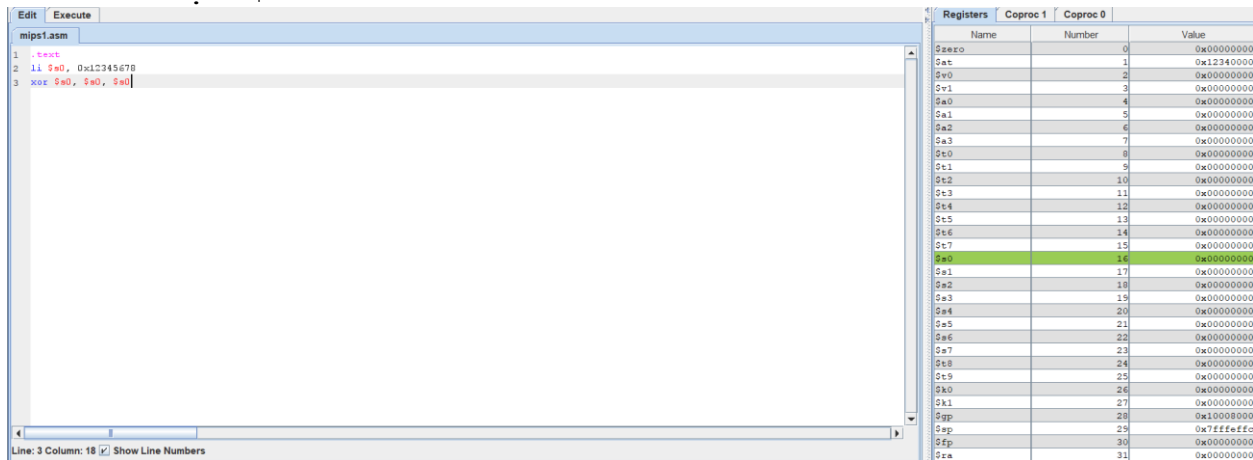
Biến 8 bit cuối thành 1:

The screenshot shows the MIPS simulator interface. The assembly code in the editor is as follows:

```
1 .text
2 li $a0, 0x62345678
3 andi $t0, $a0, 0xf0000000
4 addi $t0, $t0, 0x01111111
```

The register window on the right shows the state of the registers. The register \$t0 is highlighted, showing its value as 0x61111111.

- Thu được kết quả tại \$t0 = 0x61111111.
 - Xóa sạch \$s0:



→ Thu được kết quả tại \$s0 = 0x00000000

Assignment 3:

- a. `.text`
 `sra $t0, $s1, 0x0000001f`
 `xor $t1, $t0, $s1`
 `subu $t1, $t1, $t0`
 b. `addu $s0, $zero, $s1`
 c. `nor $s1, $s2, $0`
 d. `slt $at, $s2, $s1`
 `beq $at, $zero, L`

Assignment 4:

```

.text
start:
li $s1, 0xffffffff # s1 = -1
li $s2, 0xffffffff # s2 = -1
li $t0, 0
addu $s3, $s1, $s2      # s3 = s1 + s2
xor $t1, $s1, $s2      # Test if $s1 and $s2 have the same sign  $t1 = 1
bltz $t1, EXIT          # t1 < 0 -> EXIT
xor $t2, $s3, $s1      # if tong cua s3 va s1 khac dau thi t2 < 0
bltz $t2, OVERFLOW     # t2 < 0 -> OVF
j EXIT
OVERFLOW:
li $t0, 1
  
```

Assignment 5:

```

.text
start:
li $s0, 1 # $s0 = 1
li $s1, 1024 # $s1 = 1024
li $s3, 1 # s3 = 1
  
```

```

loop:
beq $s1, $s3 , exit   # $s1 = 1 → exit
j continue
continue:
sll $s0, $s0, 1# nhan 2
srl $s1, $s1, 1# chia 2
j loop
exit:
add $t0, $0 , $s0

```

+ Khởi tạo:

\$s0	16	1
\$s1	17	1024

+ Kết quả:

\$s0	16	1024
\$s1	17	1

+ Giải thích:

Khởi tạo \$s0 là 1 và \$s1 là 1024. Bước vào vòng lặp, ta sẽ dịch trái 1 bit tức nhân 2 đối với \$s0 và dịch phải 1 bit tức chia 2 đối với \$s1. Vòng lặp kết thúc khi \$s1 = 1.