

Lab 4

Arithmetic and Logical Operations

Đinh Ngọc Khánh Huyền - 20225726

Assignment 1

```
mips1.asm*  mips2.asm*
1  .text
2  li $s0,0x12345678      #assign a 32-bit number to $s0
3  andi $t0,$s0,0xff000000 # Extract the MSB of $s0
4  andi $t3,$s0,0x00ffff00
5  andi $t4,$s0,0x000000ff
6  sll $t4,$t4,24
7  srl $t0,$t0,24
8  andi $t1,$s0,0xffffffff #Clear the LSB of s0
9  ori $t2,$s0,0x000000ff  # Set the LSB of $s0 (all bits are set to 1)
10 andi $s0,$s0,0          # Clear register $s0 ($s0=0, must use logical instructi
11 or $s0,$s0,$t3
12 or $s0,$s0,$t0
13 or $s0,$s0,$t4 #Exchange the MSB of $s0 with the LSB
14
15
16
```

Giải thích:

- Lấy MSB của \$s0, ta sử dụng lệnh andi để lấy riêng phần MSB sang \$t0, để srl để dịch phải 24 bit, khi đó còn 8 bit còn lại sẽ là MSB của \$s0
- Xóa LSB của \$s0, ta and từng bit giá trị thanh ghi \$s0 với 0xffffffff00, kết quả sẽ là 8 bit cuối thanh ghi \$s0 sẽ nhận giá trị là 0
- Đặt LSB của \$s0 là 1, ra or từng bit \$s0 với giá trị 0x000000ff, kết quả trả ra là từ 8 bit cuối sẽ từ 0 thành 1
- Xóa giá trị thanh ghi \$s0, ta dùng phép XOR (phép XOR sẽ trả về 0 khi 2 bit chung giá trị) để XOR giá trị thanh ghi \$s0 với chính nó
- Lấy riêng 3 phần MSB, phần giữa và LSB của \$s0, rồi dùng sll và srl để dịch 2 bit MSB cả LSB đổi chỗ cho nhau, sau đó or 3 cái với 0 để thu được kết quả

Registers	Coproc 1	Coproc 0	
Name	Number	Value	
\$zero	0	0x00000000	
\$at	1	0xffffffff00	
\$v0	2	0x00000000	
\$v1	3	0x00000000	
\$a0	4	0x00000000	
\$a1	5	0x00000000	
\$a2	6	0x00000000	
\$a3	7	0x00000000	
\$t0	8	0x00000012	
\$t1	9	0x12345600	
\$t2	10	0x123456ff	
\$t3	11	0x00345600	
\$t4	12	0x78000000	
\$t5	13	0x00000000	
\$t6	14	0x00000000	
\$t7	15	0x00000000	
\$s0	16	0x78345612	
\$s1	17	0x00000000	
\$s2	18	0x00000000	
\$s3	19	0x00000000	
\$s4	20	0x00000000	
\$s5	21	0x00000000	
\$s6	22	0x00000000	
\$s7	23	0x00000000	
\$t8	24	0x00000000	
\$t9	25	0x00000000	
\$k0	26	0x00000000	
\$k1	27	0x00000000	
\$gp	28	0x10008000	
\$sp	29	0x7fffffc	
\$fp	30	0x00000000	
\$ra	31	0x00000000	
pc		0x0040004c	
hi		0x00000000	
lo		0x00000000	

Assignment 2

mips1.asm*	mips2.asm
1	<code>.text</code>
2	<code>start:</code>
3	<code>li \$t0,0</code>
4	<code>li \$s1,0x7fffffff</code>
5	<code>li \$s2,1808</code>
6	<code>addu \$s3,\$s2,\$s1</code>
7	<code>xor \$t1,\$s1,\$s2</code>
8	<code>bltz \$t1,exit</code>
9	<code>xor \$t2,\$s3,\$1</code>
10	<code>bgtz \$t2,exit</code>
11	<code>j overflow</code>
12	
13	<code>overflow:</code>
14	<code>li \$t0,1</code>
15	<code>exit: </code>
16	

Giải thích:

Sử dụng phép xor để nếu 2 giá trị trên 2 thanh ghi \$s1 và \$s2 trái dấu, chương trình sẽ nhảy đến label EXIT. Nếu cùng dấu ta sẽ tiếp tục kiểm tra giá trị 1 trong 2 thanh ghi trên với giá trị thanh ghi \$s3. Nếu giá trị đó trái dấu nghĩa là phép cộng trên bị tràn số, chương trình sẽ nhảy đến nhãn OVERFLOW và đặt giá trị thanh ghi \$t0 bằng 1

Registers	Coproc 1	Coproc 0	
Name	Number	Value	
\$zero	0	0x00000000	
\$at	1	0x7fff0000	
\$v0	2	0x00000000	
\$v1	3	0x00000000	
\$a0	4	0x00000000	
\$a1	5	0x00000000	
\$a2	6	0x00000000	
\$a3	7	0x00000000	
\$t0	8	0x00000001	
\$t1	9	0x7ffff8ef	
\$t2	10	0xffff070f	
\$t3	11	0x00000000	
\$t4	12	0x00000000	
\$t5	13	0x00000000	
\$t6	14	0x00000000	
\$t7	15	0x00000000	
\$s0	16	0x00000000	
\$s1	17	0x7fffffff	
\$s2	18	0x00000710	
\$s3	19	0x8000070f	
\$s4	20	0x00000000	
\$s5	21	0x00000000	
\$s6	22	0x00000000	
\$s7	23	0x00000000	
\$t8	24	0x00000000	
\$t9	25	0x00000000	
\$k0	26	0x00000000	
\$k1	27	0x00000000	
\$gp	28	0x10008000	
\$sp	29	0x7fffeffc	
\$fp	30	0x00000000	
\$ra	31	0x00000000	
pc		0x0040002c	
hi		0x00000000	
lo		0x00000000	

Assignment 3

mips1.asm	mips2.asm	mips3.asm	
<pre>1 .data 2 number: .word 0 3 .text 4 main: 5 li \$s0, 0x12345678 6 li \$v0, 5 7 syscall 8 move \$t0, \$v0 9 srlv \$s0, \$s0, \$t0 10 andi \$s0, \$s0, 0x1ffff 11 sw \$s0, number 12 li \$v0, 10 13 syscall</pre>			

Giải thích:

Gán giá trị 0x12345678 cho thanh ghi \$s0. Dùng lệnh li \$v0, 5 và syscall để thực hiện đọc số nguyên từ bàn phím. Kết quả trả về sẽ được lưu tại \$v0. Sau đó ta chuyển giá trị này về \$t0. Dùng lệnh srlv để dịch phải logic biến đổi giá trị trong \$t0 bit \$s0. Sau đó andi với 0x1ffff để giữ lại 17 bit cuối của \$s0. Sau đó lưu giá trị vào \$s0 rồi kết thúc chương trình với lệnh li \$v0, 10.

Registers	Coproc 1	Coproc 0	
Name	Number	Value	
\$zero	0	0x00000000	
\$at	1	0x10010000	
\$v0	2	0x0000000a	
\$v1	3	0x00000000	
\$a0	4	0x00000000	
\$a1	5	0x00000000	
\$a2	6	0x00000000	
\$a3	7	0x00000000	
\$t0	8	0x00000005	
\$t1	9	0x00000000	
\$t2	10	0x00000000	
\$t3	11	0x00000000	
\$t4	12	0x00000000	
\$t5	13	0x00000000	
\$t6	14	0x00000000	
\$t7	15	0x00000000	
\$s0	16	0x0001a2b3	
\$s1	17	0x00000000	
\$s2	18	0x00000000	
\$s3	19	0x00000000	
\$s4	20	0x00000000	
\$s5	21	0x00000000	
\$s6	22	0x00000000	
\$s7	23	0x00000000	
\$t8	24	0x00000000	
\$t9	25	0x00000000	
\$k0	26	0x00000000	
\$k1	27	0x00000000	
\$gp	28	0x10008000	
\$sp	29	0x7ffffc	
\$fp	30	0x00000000	
\$ra	31	0x00000000	
pc		0x00400034	
hi		0x00000000	
lo		0x00000000	

(Trường hợp nhập vào 5. Số hiển thị trên thanh ghi \$s0 là 0x0001a2b3 = 1101000101011001₂)