

CS202: Computer Organization

Lab assignment1

11911827 张皓淇

Q1: The output is different from the sample picture below.

Reason1: After the command

```
nor $t1, $zero, $zero
```

the value in t1 would be 0xffffffff.

And after command

```
sra $t2, $t1, 31
```

The value in t2 would also be 0xffffffff.

However we need the value in t2 to be 0x00000001, in order to get the correct output after the bitwise and operation.

So the above two command should be modify to

```
li $t2,1
```

So that we can get the correct output in a0.

Reason2: In the command

```
print_string("it is an odd number (0:false,1:true) : ")
```

We actually change the value in a0 to 0x1001001b, which will result in wrong output. So we need to change the sequence of our program.

```
.include "macro_print_str.asm"
.text
main:
    print_string("please input an integer : ")
    li $v0,5
    syscall
    move $t0, $v0
    print_string("it is an odd number (0:false,1:true) : ")
    li $t2,1
    and $a0, $t2, $t0
    li $v0,1
    syscall
    end
```

Q2:

Address: 0x10010000
The first operand

Address: 0x10010004
The second operand

Address	Value (+0)	Value (+4)	Value (+8)	Value (+C)	Value (+10)	Value (+14)	Value (+18)	Value (+1C)
0x10010000	0x00044240	0x001889F	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x10010000	0x10010004	0x00202420	0x00203420	0x24200000	0x34200020	0x00000020	0x00202420
0x10010040	0x68203420	0x68283469	0x64617865	0x64696365	0x20296e61	0x6f6c2000	0x6568283a	0x65646178
0x10010060	0x61646963	0x0020296e	0x2f200000	0x34200020	0x72200020	0x69616465	0x72656464	0x00000020
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010140	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010160	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010180	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

Fig.1. screenshot of data segment before running

Address: 0x10010000
The first operand

Address: 0x10010008
The result of addition

Address: 0x1001000C
The result of subtraction

Address: 0x1001001C
The result of division(lo):quotient

Address	Value (+0)	Value (+4)	Value (+8)	Value (+C)	Value (+10)	Value (+14)	Value (+18)	Value (+1C)
0x10010000	0x00044240	0x001889F	0x0010e8ff	0x000443a1	0x00000017	0x4867a6e0	0x0000000a	0x0000000a
0x10010020	0x10010000	0x10010004	0x00202420	0x00203420	0x24200000	0x34200020	0x00000020	0x00202420
0x10010040	0x68203420	0x68283469	0x64617865	0x64696365	0x20296e61	0x6f6c2000	0x6568283a	0x65646178
0x10010060	0x61646963	0x0020296e	0x2f200000	0x34200020	0x72200020	0x69616465	0x72656464	0x00000020
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010140	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010160	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010180	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

Address: 0x10010004
The second operand

Address: 0x10010010
The result of multiplication(hi)

Address: 0x10010014
The result of multiplication(lo)

Fig.2. screenshot of data segment after running