HO CHI MINH UNIVERSITY OF TECHNOLOGY

FACULTY OF COMPUTER SCIENCE AND ENGINEERING

COMPUTER ARCHITECTURE

Practical session - Week 5

Lecturers: Assoc. Prof. Dr. Cuong Pham Quoc

Mr. Kieu Do Nguyen Binh

Class: CC04 Group: 8

Members: Nguyễn Minh Hùng Student ID: 1952737

Phạm Nhật Hoàng Student ID: 1952703

Nguyễn Bá Minh Hưng Student ID: 1952748

Question 1. Given the following Multiplication and Division program with MIPS assembly.

```
######### Multiplication ###############
        lui $t0, 0x5000
        li $t1, 4
        mult $t0, $t1
        mflo $s0
        mfhi $s1

######### Division #####################
        li $t0, 10
        li $t1, 7
        div $t0, $t1
        mflo $s0
        mfhi $s1

######### Exit #########################
        li $v0, 10
        syscall
```

Students run the program and answer the following question
1. What is the function of mflo and mfhi instruction?
2. What is the value of $s0 and $s1 after the multiplication.
3. What is the value of $s0 and $s1 after the division?
4. What is the role of lo and hi register in multiplication?
5. What is the role of lo and hi register in division?
6. Do hi and lo is a part of 32 general purpose register of MIPS processors?

Answer:
1)
mflo : move from LO register to $s0 (the 32 least significant bits are held in LO after mult or div instructions)
mfhi : move from HI register to $s1 (the 32 most significant bits are held in HI after mult or div instructions)
2)
$s0 : 0x40000000
$s1 : 0x00000001
3)
$s0 : 0x00000001
$s1 : 0x00000003
4)
The result of the multiplication of two 32-bit numbers yields a 64-number,

therefore the result automatically goes into two special registers called HI and LO after mult instruction.
The 32 least and most significant bits are held in LO and HI, respectively after mult instructions.
5) The register LO and HI after div instruction will store Quotient and Remainder, respectively after div instruction.
6) yes

Question 2. Given the following log 2 procedure in C:

```
int log_2(int n)
{
int ret = 0;
for (; n/2 != 0; ret = ret + 1) n = n / 2;
return ret;
}
```

Assume that all input is the exponent of 2. Implement a MIPS program that:
1. Receive input value from user.
2. Call the log 2 procedure.
3. Get the return value of log_2 procedure and print to screen.

```
.text
.globl main
main:

li          $v0,4                   # print string
la          $a0,input
            syscall


li          $v0,5                   # read input
            syscall
move        $a0,$v0                 # $a0 = n

jal         log_2                   # call log_2
j           end
#-----------------------------
log_2:                              # procedure
li          $t0,0                   # ret = 0
loop:
srl         $t2,$a0,1               # m = n/2
beq         $t2,0,exit              # m != 0 ?
srl         $a0,$a0,1               # n = n / 2
addi        $t0,$t0,1               # i++
j           loop
exit:
move        $v0,$t0
```

```
jr          $ra
# ------------------------
end:
move        $s1,$v0

li          $v0,1
move        $a0,$s1
            syscall

li            $v0  ,  10
syscall


.data
input:      .asciiz "Input n: "
```

q2.asm

Question 3. Implement the following C code by using MIPS code. Assume that b and c are 10 and 7, respectively while input variable is read from keyboard. Print value of a to the terminal.

---

switch (input){
case 0: a = b + c; break;
case 1: a = b - c; break;
case 2: a = b * c; break; // print both low and high word
case 3: a = b / c; break;
case 4: a = b % c; break;
default: printf{"Your choice is invalid"}; break;
}

```
.text

.globl main

main:

li          $s0,10
li          $s1,7


li          $v0,4
la          $a0,input
            syscall
li          $v0,5
            syscall
move        $t0,$v0


beq         $t0,0,case0
beq         $t0,1,case1
beq         $t0,2,case2
beq         $t0,3,case3
beq         $t0,4,case4
```

```
24
25 default:
26 li          $v0,4
27 la          $a0,invalid
28             syscall
29 j           end
30 case0:
31 add         $t1,$s0,$s1
32 li          $v0,1
33 move        $a0,$t1
34             syscall
35 j           end
36 case1:
37 sub         $t1,$s0,$s1
38 li          $v0,1
39 move        $a0,$t1
40             syscall
41 j           end
42 case2:
43 mult        $s0,$s1
44 mfhi        $t2
45 mflo        $t3
46 li          $v0,4
47 la          $a0,hi
48             syscall
49 li          $v0,1
50 move        $a0,$t2
51             syscall
52
53 li          $v0,4
54 la          $a0,lo
55             syscall
56 li          $v0,1
57 move        $a0,$t3
58             syscall
59 j           end
60 case3:
61 div         $s0,$s1
62 mflo        $t2
63 li          $v0,1
64 move        $a0,$t2
65             syscall
66 j           end
67 case4:
68 div         $s0,$s1
69 mfhi        $t2
70 li          $v0,1
71 move        $a0,$t2
72             syscall
73 j           end
74
75 end:
76
77 .data
78 input:     .asciiz "input: "
79 hi:        .asciiz "\nhi register: "
80 lo:        .asciiz "\nlo register: "
81 invalid:   .asciiz "Your choice is invalid\n"
```

q3.asm