南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Computer Organization and Design

Midterm Practice Exam

Name: _____ Student Number: _____

1. True or False (15 points)

(1) In 1965, Gordon Moore made a prediction "The number of transistors that can be integrated on a die would double every 18 to 24 months".

(2) CPI (Cycles per instruction) is only determined by instruction set architecture.

(3) In memory, the stack pointer $sp is initialized to 7ffffffc and grows up towards the data segment.

(4) The desire for speed is the reason that MIPS has 32 registers rather than many more.

(5) In MIPS, add, addi and addu can cause exception on overflow.

(6) Multiplication hardware simply shifts and add.

(7) We can accelerate division by predicting multiple quotient bits and then correcting mispredictions later.

(8) MIPS doesn't support floating-point instructions.

(9) Like integers, we can also represent exactly every number between the smallest and largest number for floating-point numbers.

(10) We can use `lw $t0, $s1` to load the value saved in the address $s1.

2. Performance (10 points)

Consider three different processors P1, P2, and P3 executing the same instruction set. P1 has a 3 GHz clock rate and a CPI of 1.5. P2 has a 2.5 GHz clock rate and a CPI of 1.0. P3 has a 4.0 GHz clock rate and has a CPI of 2.2.

a. Which processor has the highest performance expressed in instructions per second? (3 points)

b. If the processors each execute a program in 10 seconds, find the number of cycles and the number of instructions. (4 points)

c. We are trying to reduce the execution time by 30% but this leads to an increase of 20% in the CPI. What clock rate should we have to get this time reduction? (3 points)

3. Assembly Language (22 points)

Consider the following C code segment:

```
if (x<15)
   x = x+m;
else
   x = x-m;
x++;
```

(1) Write a sequence of MIPS instructions that directly correspond to this C code segment. Assume register $s0 contains the integer variable x and register $s1 contains the integer variable m. Use temporary registers if necessary. To alter the order of execution of instructions in your MIPS code, do not use jump (j) instructions, but only branch (beq and bne) instruction. (12 points)

(2) If x=3, m=2, how many MIPS instructions will be executed? (4 points)

(3) Provide the type, assembly language instruction, and binary representation of instruction described by the following MIPS fields (6 points):

op = 0, rs = 10, rt= 15, rd = 22, shamt = 0, function = 36

4. Arithmetic for computers (33 points)

(1) Assume 185 and 122 are signed 8-bit decimal integers stored in sign-magnitude format. Calculate 185-122. Is there overflow, underflow, or neither? (6 points)

(2) Fill in the blanks in the table, calculate 74 divided by 21 (the octal unsigned 6-bit integers) using the hardware described in Figure 1. You should show the contents of each register on each step. Assume both inputs are unsigned 6-bit integers. (22 points)

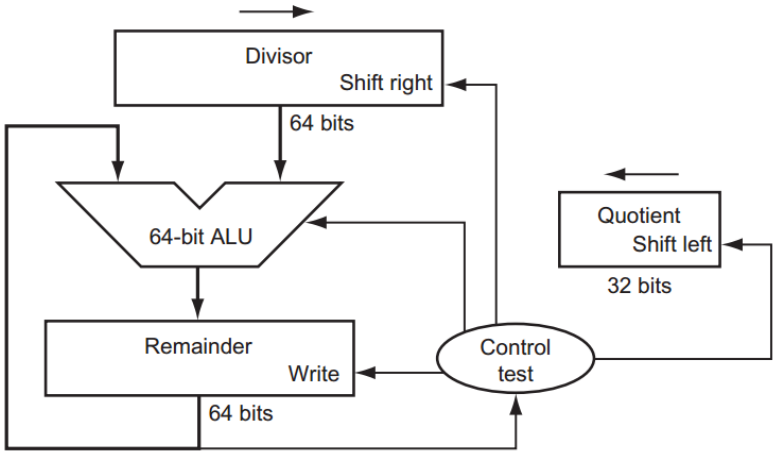

Figure 1. First version of the division hardware.

Step	Action	Quotient	Divisor	Remainder
0				
1				
2				
3				
4				
5				
6				
7				

(3) Write down the binary representation of the decimal number 63.25 assuming the IEEE 754 single precision format. (5 points)

5. Fill the blanks (20 points)

(1) List the names of the basic components of a computer: _____, _____, _____ and _____.

(2) Given that the instruction j NEXT is at address 0x004000F4, and the label NEXT is at address 0x00402AEC. Then, the 26-bit immediate stored in the jump instruction for the label NEXT is_____.

(3) There is a one-to-one correspondence between assembly language and _____ language.

(4) Indicate what the addressing mode is:

j $ra _____ addressing

bne $t0, $t1, Exit _____ addressing

sw $s1,0($t1) _____ addressing

(5) After executed the assembly language below, what is the value in the $t1? _____.

```
        addi $t0, $zero, 35
        addi $t1, $zero, 8
        sll $t1, $t1,4
        slt $t2, $t1, $t0
        bne $t2, $zero, ELSE
        addi $t1, $t1, 1
ELSE: addi $t1, $t1, -5
```