國立交通大學 資訊工程系 計算機組織 第一次考試 (2023/4/13 1:20PM~3:20PM)

請注意：1. 不可看書及任何參考資料(Close book)。 2. 請詳述設計或計算過程(detail your design or computation process)。 3. 請依題號作答，否則高題號答案出現後，之後的低題號答案將不予計分 (answer each problem in problem sequence. The answers of small ID problems that appear behind the answers of large ID problems will be ignored)。

1. **Domain Background (6 pts)**

   Consider the computer history, mini-computers were gradually eliminated by workstations in the market. Explain the reason for the elimination of mini-computers from the viewpoint of semiconductor technology. Explain the reason why the multi-core CPUs replaces the single-core CPU to become the mainstream of CPU market from the viewpoint of performance.

2. **Clock Frequency (8 pts)**

   Assume that there are types of instructions whose required execution times are 3 and 5 *ns*, respectively. One code sequence consists of five instructions i_{11}, i_{22}, i_{32}, i_{41}, and i_{51}, where i_{51} represents that the fifth instruction is of type 1. If the clock period is 4*ns* and the execution time of an instruction is 3*ns*, then the idle time of executing this instruction is 1*ns*. (a) If the clock period is set as 3 *ns*, what is the required execution time for the code sequence (2 pts)? What is the total idle time during the overall execution time (2 pts)? (b) explain why MIPS only supports *beq* and *bne* instead of *blt* or *bgt*. (4 pts)

3. **Performance (19 pts)**

   The table below shows the relative frequencies of two programs using instructions in classes *A*, *B* and *C*. (*must show all computations to get full credits*)

Class	A	B	C
CPI for class	1	3	2
Relative frequency in program 1	0.3	0.5	0.2
Relative frequency in program 2	0.4	0.3	0.3

   (a) (7 pts) If a daily operation is to run program 1 and program 2 once and three times, respectively, and the instruction count of program 1 is twice that of program 2. What is the weighted average CPI of the daily operation?

   (b) (7 pts) Consider four loads, 100%, 80%, 50%, 0%, which loading is the most operation load of the data-center servers (3 pts)? Why is it important to design the CPU of the data-center servers that operates at 0 load (4 pts)?

   (c) (5 pts) Consider the performance metric MIPS. List the formula to compute the MIPS. Which kind of CPU has the superiority if we use MIPS to evaluate the performance of a CPU. You have to use the MIPS formula to explain your reason to get full credit.

4. **Design Principles (6 pts)**

   Use the instruction set or MIPS CPU design as an example to explain the following two design principles. (a) Make the common case fast; (b) Good design demands good compromise.

```
unsigned int fib (unsigned int n) {
 if (n < 2) return(n);
 else return(fib(n-1)+fib(n-2));
}
```

```
1.      fib:
2.      addi    $sp, $sp, -12
3.      sw      $ra, 0($sp)
4.      sw      $s1, 4($sp)
5.      sw      $a0, 8($sp)
6.      slti    $t0, $a0, 2
7.      beq     $t0, $0, L1
8.      addi    $v0, $a0, 0
9.      j       EXIT
10.     L1:
11.     addi    $a0, $a0, -1
12.     jal     fib
13.     addi    $s1, $v0, 0
14.     addi    $a0, $a0, -1
15.     jal     fib
16.     add     $v0, $v0, $s1
17.     EXIT:
18.     lw      $ra, 0($sp)
19.     lw      $a0, 8($sp)
20.     lw      $s1, 4($sp)
21.     addi    $sp, $sp, 12
22.     jr      $ra
```

5. Instructions and Decoding/Encoding (35 pts)

The above figure shows a C program and its MIPS assembly program.

(a) (3 pts) In this code, why does it need to save registers $ra and $a0?

(b) (3 pts) In this code, $s1 is used as a temporary to hold fib(n-1)+fib(n-2), so why not use register $t1 instead?

(c) (4 pts) It takes one instruction to compare $a0 with 2, and one branch instruction to skip the computations if $a0 is less than 2. If we have a blti $a0, 2, L1 (branch less than immediate) instruction, one instruction could be eliminated. Should the MIPS ISA consider adding this instruction?

(d) (9 pts) Turn the following MIPS machine instructions into assembly language form based on the following table and Table 1. You can define label name by yourself, if necessary.

0x00ae8020 0x00000000 0x8fb1fffc

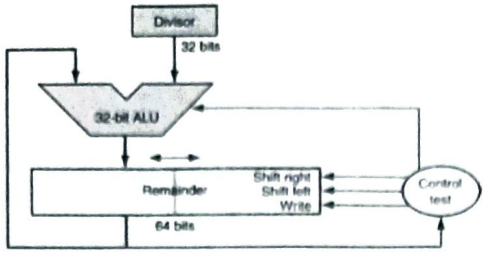
Name	$zero	$v0-$v1	$a0-$a3	$t0-$t7	$s0-$s7	$t8-$t9	$sp
Reg. no.	0	2-3	4-7	8-15	16-23	24-25	29

(e) (6 pts) For the C code and its related MIPS assembly codes as shown above. As we invoke the procedure by fib(2), list the execution sequence, like the sequence (1, 2~9, 17~22) for fib(1). Notably, if a recursive call happens, you can list the sequence like (…, 10, 11, 1 (n=k), 2 ~…) (4 pts).

(f) (10 pts) For each pseudoinstruction in the following table, produce a **minimal sequence** of actual MIPS instructions to accomplish the same thing. In the following table, big refers to a specific number that requires 32 bits to represent.

Note: you can use upper16(big) to represent the upper 16 bits of big; and lower16(big) to represent the lower 16 bits of big. For MIPS assembly instructions, please refer to the table on the last page.

	Pseudoinstruction	What it accomplishes	(Hint: Minimal sequence)
1 (3%)	bge $t5, $t3, L	if($t5 >= $t3) goto L	2 instructions
2 (4%)	bgt $t5, $t3, L	if($t5 > $t3) goto L	2 instructions
3 (3%)	Li $t5, big	$t5 = big	2 instructions



6. Division (12 pts)

Given the above division hardware.

(a) (6 pts) Explain how to update the restoring division to be a non-restoring division. You need to explain why this update is equivalent to the restoring division.

(b) (6 pts) Compare the performance of restoring and non-restoring division by showing the number of addition/subtraction operations required to do the following division using both methods:
11111 ÷ 00101

7. Floating-Point Numbers and IEEE Std. 754 (16 pts)

Give the IEEE-754 single precision floating-point operand format as follows:

Sign (s)	Exponent (e)	Fraction (f)
1 bit	8 bits	23 bits

The exponent bias is 127. There is an implied bit to the left of the binary point in the fraction. Exponents with all 0's and 1's are reserved for special conditions.

(a) (3 pts) Show the IEEE 754 binary representation of the number – 9.375 in single precision.

(b) (3 pts) What decimal number is represented by this word?
1 10000011 10101000000000000000000

(c) (3 pts)For the 32-bit IEEE 754 floating-point standard, what are the largest and the smallest positive normalized numbers?

(d) (3 pts) For the 32-bit IEEE 754 floating-point standard, what are the largest and the smallest positive denormalized numbers?

(e) (4 pts) Consider the schemes of guard bit and round bit. Now we use "digit" rather than "bit" in the following problem. Each fraction is rounded off to the second decimal place. What are the accuracies of the addition $7.74\times10^2 + 6.92\times10^4$ with and without guard and round digits in terms of ulp?

Op (31:26)								
28-26 31-29	0(000)	1(001)	2(010)	3(011)	4(100)	5(101)	6(110)	7(111)
0(000)	R-format	Bltz/gez	Jump	Jal	Beq	Bne	Blez	Bgtz
1(001)	Addi	Addiu	Slti	Sltiu	Andi	Ori	Xori	Lui
2(010)	TLB	FlPt						
3(011)								
4(100)	Lb	Lh	Lwl	Lw	Lbu	Lhu	lwr	
5(101)	Sb	Sh	Swl	Sw			Swr	
6(110)	Lwc0	Lwc1						
7(111)	Swc0	swc1						
Op (31:26)=010000 (TLB), rs(25:21)								
23-21 25-24	0(000)	1(001)	2(010)	3(011)	4(100)	5(101)	6(110)	7(111)
0(00)	mfc0		cfc0		mtc0		ctc0	
1(01)								
2(10)								
3(11)								
Op(31:26)=000000 (R-format), funct(5:0)								
2-0 5-3	0(000)	1(001)	2(010)	3(011)	4(100)	5(101)	6(110)	7(111)
0(000)	Sll		Srl	Sra	Sllv		Srlv	Srav
1(001)	Jump reg.	Jalr			Syscall	Break		
2(010)	Mfhi	Mthi	Mflo	Mtlo				
3(011)	Mult	Multu	Div	Divu				
4(100)	Add	Addu	Subtract	Aubu	And	Or	Xor	Nor
5(101)			Set l.t.					
6(110)								
7(111)								

Table 1