106 CI 01

國立臺北科技大學 106 學年度碩士班招生考試

系所組別：2300 資訊工程系碩士班

第一節　計算機概論　試題

注意事項：

1. 本試題共五題，共 100 分。
2. 請標明大題、子題編號作答，不必抄題。
3. 全部答案均須在答案卷之答案欄內作答，否則不予計分。

一. Provide answers for the following problems:

1. (10%) The 32-bit IEEE 754 floating-point format consists of one sign bit, 8 exponent bits coded with excess-127, and 23 mantissa bits represented by implicit-one convention. Represent the factional number (10/3) by IEEE 745 floating-point format with the default rounding (round to nearest, ties to even) method.
2. (10 %) In most microprocessors, there are carry and overflow flags. Explain the differences between these two flags. In addition, give a simple algorithm to show how to detect overflow in a microprocessor.
3. (5 %) Give a precise definition of the access time of a magnetic disk.
4. (5 %) What is the difference between a process and a thread?
5. (10 %) What is a "critical section" in concurrent programming? How to implement critical section(s) in two threads?

二. Answer the following questions:

1. (2%) What is the length of the IPv4 internet address (in bits)?
2. (2%) What is the length of the Ethernet MAC address (in bits)?
3. (4%) Identify **two** main different features between **UDP** and **TCP** transport layer protocols.
4. (4%) Compare the main features between circuit and packet switching strategies in the following two aspects: (1) queueing delay inside the switching node; (2) bandwidth reservation.
5. (4%) Two hosts, Hosts A and B, are connected by a single link of rate R bps. Suppose that the two hosts are separated by d meters, and suppose the propagation speed along the link is p meters/sec. Host A is to send a packet of size L bits to Host B. Then, (1) express the propagation delay of the link. (2) determine the transmission time of the packet.
6. (4%) Identify the main functions provided by (1) DNS protocol; (2) ARP protocol.

三. Given the following MIPS assembly code segments in Figure 1, which contain a function **int f(int a, int b, int c, int d)** that calls another function **foo(a, b)**. Please answer the following problems.

MIPS Code	**f**: addi $sp,$sp,-4
	sw $ra, 0($sp)
	add $s0,$a2, $zero
	add $s1,$a3, $zero
	jal **foo**
	add $a0,$v0, $zero
	add $a1,$s0,$s1
	jal **foo**
	lw $ra, 0($sp)
	addi $sp,$sp, 4
	jr $ra

Figure 1. MIPS assembly code segment

1. (8%) This code contains some mistakes that violates some MIPS calling conventions. Please why indicate the mistake of violating calling convention, and re-write the fixed code.

2. (7%) What is the **equivalent C code** of the above MIPS assembly code of the function **f** in Figure 1? Please disassemble the MIPS code into the **equivalent C code**. Assume that the function f's four arguments **a, b, c, and d** are held in the registers **$a0 - $a3**, respectively.

四. Given a typical MIPS processor with a five-stage pipelined datapath as illustrated in Figure 2. Assume that the following fraction of these instructions have a particular type of **RAW (Read-after-Write) data dependence**, as listed in Table 1. The type of RAW data dependence is identified by the stage that produces the result (EX or MEM) and the instruction that consumes the result (1st instruction that follows the one that produces the result, 2nd instruction that follows, or both). Assume that the register-write is done in the first-half of a clock cycle, and the register-read can be done in the second-half of the cycle. Also, assume that the CPI of the processor is 1, if there are no **data hazards**.

注意：背面尚有試題

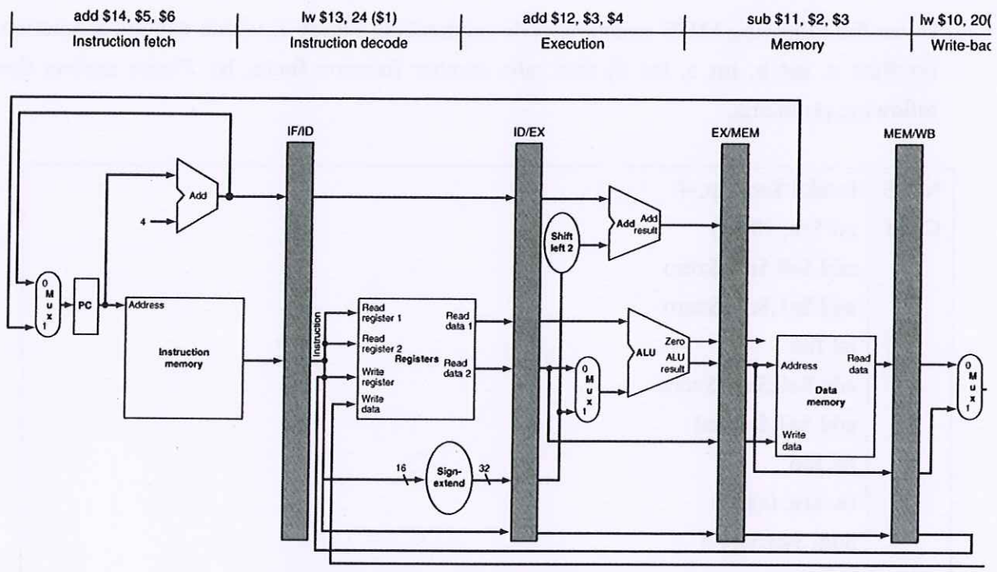

Figure 2. The pipelined datapath structure

EX to 1st Only	MEM to 1st Only	EX to 2nd Only	EX to 1st and MEM to 2nd	MEM to 2nd Only	Other RAW Dependences
10%	25%	10%	15%	15%	25%

Table 1. The fractions of RAW data dependence types of the instructions

Assume the following latencies for individual pipeline stages as depicted in Table 2. For the EX stage, latencies are given separately for a processor without forwarding and for a processor with different kinds of forwarding.

IF	ID	EX (no FW)	EX (full FW)	EX (FW from EX/MEM only)	EX (FW from MEM/ WB only)	MEM	WB
200 ps	150 ps	150 ps	250 ps	220 ps	180 ps	180 ps	150 ps

Table 2. Latencies for individual pipeline stages

1. (5%) If we use no forwarding, what is the fraction of stall cycles to total required cycles should we need to stall the pipeline due to data hazards?

2. (5%) If we use full forwarding (forward all results that can be forwarded) pipeline, what is the fraction of stall cycles to total required cycles should we need to stall the pipeline due to data hazards?

3. (5%) For the given hazard probabilities and pipeline stage latencies, what is the **speedup** achieved by adding **full forwarding** as compared to a pipeline that had **no forwarding**?

五. (10%) Consider the following pseudo codes for counting the semaphore **S**, what kind of the disadvantage does it have? Please show how you improve the disadvantage of the original code for semaphore counting by providing your modified codes of semaphore waiting and signaling.

```
wait (S) {
  while S <= 0
  ;   // no operation
  S--;
}

signal (S) {
  S++;
}
```