國立交通大學試題紙

九十八學年度第二次
博士班資格考

科目：計算機結構(A)　　日期： 99 年 7 月 28 日　第 1 頁 共 2 頁

請 "✓" 明　✓不可看書　可看書

* 請將答案依題號順序寫入答案卷

答題時字跡需工整，否則不予計分。Write your answers legibly; otherwise you will get zero score.

1. (11%)
   (a) (7%) The execution times of programs P1 and P2 on machines A and B are listed in the following table.

	Machine A	Machine B
Program P1 (secs)	4	32
Program P2 (secs)	240	120

   i. Calculate the arithmetic and geometric means of the normalized execution times to different reference machines (machine A or B) for summarizing performance. (5%)
   ii. According to the results in (a), please point out one major advantage and one major disadvantage of geometric mean. (2%)

   (b) (4%) Consider a computer running programs with CPU times shown in the following table:

FP instr.	INT ALU instr.	L/S instr.	Branch instr.	Total time
450 s	300 s	150 s	100 s	1000 s

   i. What is the overall speedup if the time for FP instructions is reduced by 40%?
   ii. By how much must we improve the speed of the INT ALU instructions if we want this program run 1.25 times faster? How about making it 1.5 times faster?

2. (7%)
   (a) (2%) A 2-GHz processor was used to execute a benchmark program with the following instruction mix and clock cycle counts per instruction of that type:

Instruction type	Frequency	Clock cycle count
Integer ALU ops	40%	1
Floating-point ops	20%	5
Loads & Stores	10%	1
Branches	30%	2

   Assume that the total number of instructions executed is 4,000,000, determine the effective CPI and execution time for this program.
   (b) (2%) Assume that a design enhancement is to reduce the CPI of the FP operations to 3. Determine the effective CPI and the speedup after such enhancement is applied.
   (c) (3%) Assume that we build an optimizing compiler to discard 50% of the branch instructions from the original instruction mix without affecting other parts of the program, determine the instruction count, the effective CPI, and the speedup of the enhancement.

3. (8%)
   (a) (4%) Define the following four different instruction set architectures: Stack architecture, Accumulator architecture, Register-memory architecture, and Register-register (Load-store) architecture.
   (b) (4%) Describe the advantages and disadvantages of a register-register (load-store) architecture.

4. (9%)
   (a) (3%) Define "Data Hazard", "Control Hazard", and "Structure Hazard."
   (b) (2%) Control hazards can cause a great performance loss for MIPS pipeline. Describe briefly two of the methods for reducing the pipeline stalls caused by branches.
   (c) (4%) For data hazards involving registers, three forms exist: RAW (read after write), WAR, and WAW. Describe briefly two possible ways to avoid each of these hazards.

5. (6%) Describe the characteristics, one major advantage, and one major disadvantage for each of a VLIW (very long instruction word) processor and a dynamically scheduled superscalar processor.

6. (9%) The following loop is a code sequence for dot product (assuming the running sum in F2 is initially 0, and the result should be written into F2).

```
loop:   L.D      F0, 0(R1)
        L.D      F4, 0(R2)
        MUL.D    F0, F0, F4
        ADD.D    F2, F0, F2
        DADDUI   R1, R1, #-8
        DADDUI   R2, R2 ,#-8
        BNE      R1, R3, loop
```

   Consider the following machine features:
   - It has a single-issue pipeline.
   - Branches are resolved in the ID stage.
   - There is one branch delay slot.
   - There is full forwarding and bypassing.
   - There are as many registers, both FP and integer, as you need.
   - If unspecified, its properties are like those in a typical MIPS pipeline.
   - The use latencies of two consecutive instructions are shown as follows:

Instruction producing result	Instruction using result	Latency in clock cycles
FP ALU op	Another FP ALU op	3
FP ALU op	Store double	2
Load double	FP ALU op	1
Load double	Store double	0

   (a) (3%) Without rescheduling the code, how many stalls are there in each iteration? Please show your work by putting a stall (or stalls) where it would occur in the code.
   (b) (6%) Unroll the loop twice and reschedule it to eliminate as many of the stalls in the loop body as possible. Assume that the loop executes a nonzero, even number of times.

國立交通大學試題紙

九十八學年度第二次
博士班資格考

科目：計算機結構(B)　　日期：99年7月28日　第1頁 共3頁

請"✓"明　✓不可看書　可看書

* 請將答案依題號順序寫入答案卷

答題時字跡需工整，否則不予計分。Write your answers legibly; otherwise you will get zero score.

1. (10%) In exploiting ILP, Chapter 3 has defined an ideal processor hardware model:
   i. Infinite number of virtual registers available;
   ii. Branch prediction is perfect;
   iii. All jumps are perfectly predicted;
   iv. Perfect (memory) address alias analysis;
   v. Perfect caches.
   (a) With assumption i above, what advantage can be obtained?
   (b) To approximate the advantage in (a), what design technique do you need to use? In your answer, first clearly specify the name of the technique, and then concisely explain what it is (whether it is a static or run-time scheme, and how it should be executed).
   (c) Assumptions ii and iii together imply a very important hardware feature. What is it?
   (d) With assumption iv above, what advantage can be obtained?
   (e) What have not been assumed are the instruction fetch, decode and execution hardware bandwidths. Given also infinite instruction fetch, decode, and execution hardware bandwidths, state the CPU time of a program on this model, assuming there are no I/Os in the program.

2. (12%) Given a conventional processor design oriented at instruction-level parallelism, if we are to modify it to a multi-threaded processor so that it can exploit both instruction and thread level parallelisms,
   (a) What will this multi-threading differ from the so-called multi-programming popular in the 1960's to 1980's?
   (b) Typically, what hardware resources should be added to such a multi-threaded processor?
   (c) Simultaneous multi-threading differs from other types of multithreading in that more than one thread is active at a time. What is the distinguishing advantage this scheme intends to obtain compared with other types of multithread? Clearly specify your answer.
   (d) Continued from (c), what distinguishing design difficulties will simultaneous multi-threading face?

3. (8%) We talked a lot about multiprocessors, their memory organizations and communications mechanisms in Chapter 4.
   (a) (3%) When we say *symmetric multiprocessor (SMP)*, how should the many processors look like? How should the memory system be organized? How the processors and memory system should be interconnected? And, what communications mechanisms can be used?
   (b) (3%) SMP is very easy to build and use, yet it also experiences a few major shortcomings. Name a major one, and state what kind of design is now used to correct it.
   (c) (2%) Repeat (b) for another major shortcoming.

4. (6%) Given a typical write-back snoopy cache coherence protocol (whose details can be slightly different from the text as long as it is correct), fill the following table assuming three cache block states I (invalid), S (shared), and E (exclusive). The load/store sequence of the nodes are:
   [Event 1] A reads x;
   [Event 2] B writes 7 to x;
   [Event 3] A reads x;
   [Event 4] x in B gets replaced;
   [Event 5] A writes 9 to x;
   [Event 6] A writes 11 to x.

Event	x value in memory	x in cache A (state/value)	x in cache B (state/value)	bus request signal (control signals only)
Initial	5	I/?	I/?	none
1				
…				

5. (14%) Assume a 2^V-byte virtual address space, 2^M-byte byte-addressable physical memory, 2^P-byte page size, a physical cache of size 2^S bytes, block size 2^B bytes, and associativity can only take on 2^A values. (You may not need all these parameters in the following questions; additionally, if you need any extra parameters, define them before using.)
   (a) (8%) In accessing the cache, how many address bits are needed? Among which, how many are byte offset bits? How many are index bits? And how many are tag bits?
   (b) (4%) At most how many bits are used for the cache tag storage? And at least how many?
   (c) (2%) One advanced cache optimizations for reducing hit times says: "Use small and simple caches." What does small mean here? Define the term *small* using parameters above.