國立交通大學試題紙

一百學年度第一次
博士班資格考

科目：計算機架構(A)　　日期：101年1月16日　第1頁 共1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. **(15 points)** Multithreading and Multicore.
   (a) What is multithreading? What are the hardware design issues that should be considered in order to support multithreading?
   (b) What is "simultaneous multithreading" (SMT, or so-called hyper-threading in Intel)? What is the basic idea and how does it work?
   (c) Give brief difference comparisons for executing a multithreading program between a 2-way SMT processor and a dual-core machine.

2. **(20 points)** Assume that *Average-memory-access-time (AMAT) = Hit time + Miss rate x Miss penalty*. For the following organization, please explain how the AMAT is impacted by the three factors (using "↗" for increase, "↘" for decrease, and "=" for no change) and also discuss if the hardware complexity will be increased or there are other negative impacts. (No credits will be given without explanation)

   (a) caches with larger associativity
   (b) victim caches
   (c) non-blocking caches
   (d) multi-level caches
   (e) virtual address cache (avoiding address translation during indexing the cache)

3. **(15 points)** Control hazard: There are several methods for dealing with stalls caused by branch delay.
   (a) What is the scheme of "delayed branch"? Explain "branch delay slot".
   (b) What three strategies of instruction scheduling (by compiler) can be employed to reduce the branch penalty? Briefly explain the basic idea of each strategy and its limitation.
   (c) Compare advantages and disadvantages of the above delayed branch scheme with any branch prediction mechanism (hint: consider hardware/software cost or complexity)

◎ 請用深黑色鋼筆或原子筆出題

命題老師簽名：

國立交通大學試題紙

一百學年度第一次
博士班資格考

科目：計算機架構(B)　　日期： 101 年 1 月 16 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.

To answer the questions, you may use either Chinese or English, and are encouraged to use the language that you are better at, to avoid any confusion/misunderstanding.

1. (8 points) This question is about dynamic power and energy consumptions:
   (a) List the dynamic power equation of a clocked MOS digital circuit.
   (b) What makes us worry about dynamic power consumption being too high? Point out a few major reasons.
   (c) Repeat (a) for dynamic energy.
   (d) Then repeat (b) for dynamic energy.

2. (24 points) Parallelisms exist in many forms between instructions, basic blocks, loop iterations, procedures/subroutines, programs, or whatever. On another hand, many techniques have been developed to take advantages of different sorts of parallelisms to speed up computing. For each of the following techniques, explain what it is about, and point out what kind of parallelism it is intended to take advantage of:
   (a) Pipelining?—Overlap instruction cycles of sequentially executed instructions as much as possible, to shorten the average time occupied by each of the instructions in series. (Copy this down, and then you will get two points for free.)
   (b) Superscalar processing?
   (c) Loop unrolling?
   (d) Register renaming?
   (e) Branch prediction?
   (f) Coarse multi-threading?
   (g) Fine multi-threading?
   (h) Simultaneous multithreading?

◎ 請用深黑色鋼筆或原子筆出題

命題老師簽名：

3. (9 points) Pipelining design must face three fundamental hazards.
   (List only the name(s) is fine if you don't think explanation is necessary, but don't run the risk of confusing me.)
   (a) Please give the names of these hazards.
   (b) Explain what problems these hazards may result into.
   (c) What techniques can be used to deal with these hazards respectively?

4. (9%) Let us visit the Tomasulo's algorithm. Given the original design form in IBM 360/91, and assume that there are 4 floating-point registers, ***A*** floating-point adder reservation stations, ***M*** floating-point multiplier reservation stations, ***L*** load buffer entries, and ***S*** store buffer entries.
   (a) What advantages can this algorithm bring?
   (b) What is the most noticeable performance limiting factor of the algorithm?
   (c) How many effective (or you may call them renameable) registers can you see in this example?