UNIVERSITY OF EDINBURGH

COLLEGE OF SCIENCE AND ENGINEERING

SCHOOL OF INFORMATICS

INFR09009 COMPUTER ARCHITECTURE

Tuesday 1st May 2018

09:30 to 11:30

INSTRUCTIONS TO CANDIDATES

Answer any TWO of the three questions. If more than two questions are answered, only QUESTION 1 and QUESTION 2 will be marked.

All questions carry equal weight.

CALCULATORS MAY NOT BE USED IN THIS EXAMINATION

Year 3 Courses

Convener: C. Stirling
External Examiners: S.Rogers, A. Donaldson, S. Kalvala

THIS EXAMINATION WILL BE MARKED ANONYMOUSLY

1. (a) State true or false and justify your answer with reasons.

   i. The compiler (used to generate machine code) can impact the clock cycle time of a processor. [*2 marks*]

   ii. CPU A has a clock frequency of 4 GHz, while CPU B has a clock frequency of 2 GHz. This implies that CPU A will perform twice as fast as CPU B on any given program. [*2 marks*]

   iii. CPU A has a MIPS (million instructions per second) rating of 1000, while CPU B has a MIPS rating of 500. This implies that CPU A will perform twice as fast as CPU B on any given program. [*2 marks*]

   iv. A dynamic branch predictor will always perform better than a static branch predictor. [*2 marks*]

(b) You are the principal architect of Fastclock PLC. You are tasked with enhancing the performance of floating point instructions for an important client whose workload spends 50% of its execution time on floating point instructions. You have identified two directions. One direction is to optimise one important floating point instruction – the floating point add, which is responsible for 10% of the *overall* execution time – by reducing its execution latency by a factor of 10. Another direction is to optimise all floating point instructions by reducing each of their execution latencies by a factor of 2. Given that both directions require approximately the same effort – but budget and other constraints forces you to choose one option – what direction will you take and why? [*4 marks*]

(c) The MIPS instruction set supports various types of memory operations: load byte (signed and unsigned variants), load halfword (signed and unsigned variants), load word, store byte, store halfword, and store word.

   i. Why does MIPS only support one variant of load word (unlike load byte and load halfword that support both signed and unsigned variants)? [*2 marks*]

   ii. Why does MIPS support only one variant of each store instruction (store byte, store halfword and store word)? [*2 marks*]

(d) i. What is a branch delay slot? What is its purpose? [*2 marks*]

   ii. What should the compiler do, if it cannot find a suitable instruction to insert into the branch delay slot? [*2 marks*]

(e) Starting from an unpipelined processor, how is the overall performance of a processor expected to vary with increasing pipeline stages? Justify your answer with reasons. State your assumptions, if any. [*5 marks*]

2. (a) Two techniques commonly used in VLIW processors are *predication* and *loop unrolling.*

    i. Briefly explain each of these; be sure to state what problem the technique is designed to solve. [*4 marks*]

    ii. For each technique, discuss whether or not it would be useful in an out-of-order processor with speculation. [*4 marks*]

   (b) The following questions pertain to caches.

    i. For a given cache size, a set-associative cache will likely give a lower miss rate than a direct-mapped cache. Yet, a very influential research paper in the early days of caches argued for a direct-mapped organization for first-level processor caches as a way to achieve higher processor performance. What argument(s) do you think the authors made to support their position? (**Hint**: You may want to relate your answer to the computer performance equation.) [*2 marks*]

    ii. Should a victim cache be direct-mapped or set-associative? Justify your answer with reasons. [*2 marks*]

    iii. Would a stride prefetcher be effective for an *instruction* cache? Justify your answer with reasons. [*2 marks*]

   (c) Today's high-performance processors can cache page table entries (PTEs) when a page table is accessed on a TLB miss. Because a PTE is a software data structure, normal processor caches can be very effective for this purpose. So when a TLB miss is detected, the required PTE may be found in the processor cache, rather than in main memory, thus speeding up address translation.

   One argument against *inverted page tables* is that they lack access locality compared to a conventional page table, making them less effective for caching purposes. Explain what kind of locality is the problem, why it exists in page table accesses, and why an inverted page table lacks it. [*6 marks*]

   (d) For the following questions, consider a realistic processor that has a variety of multi-cycle functional units.

    i. One type of hazard that may occur in such a processor is WAW (write-after-write hazard). Explain why this hazard might occur. [*3 marks*]

    ii. Would Tomasulo's approach be able to avoid this hazard? If so, describe how. If not, explain why. [*2 marks*]

3. For the following questions, consider a two-level cache hierarchy with the following parameters:

   - L1 cache: n_1 sets, a_1 associativity, b_1 block size, replacement policy: LRU (if applicable)
   - L2 cache: n_2 sets, a_2 associativity, b_2 block size, replacement policy: LRU (if applicable)

   (a) State whether the following assertion is true or false and justify your answer with reasons: "Architects typically opt for a much larger L1 cache compared to L2." *[3 marks]*

   (b) How is the local L2 miss-rate different from the global L2 miss-rate? Explain why local L2 cache miss rates are typically much higher than global L2 miss rates? *[4 marks]*

   (c) What does it mean for an L2 cache to be inclusive? *[2 marks]*

   (d) What does it mean for an L2 cache to be exclusive? *[2 marks]*

   (e) Is it possible for an L2 cache to be neither inclusive nor exclusive? *[2 marks]*

   (f) State one advantage and one disadvantage of having an inclusive L2 cache. *[2 marks]*

   (g) Let us suppose that both L1 and L2 are direct-mapped caches ($a_1 = a_2 = 1$). Further, let us assume $b_1 = 4$ bytes and $b_2 = 8$ bytes, and $n_1 = 4$ and $n_2 = 8$. For these parameters, is L2 guaranteed to be inclusive? If so, argue why that is the case. If inclusion is not guaranteed, provide a counter-example. *[5 marks]*

   (h) Let us suppose both L1 and L2 are 2-way set associative ($a_1 = a_2 = 2$). Further, let us assume both caches have the same block size ($b_1 = b_2 = 8$ bytes), and $n_1 = 4$ and $n_2 = 8$. For these parameters, is L2 guaranteed to be inclusive? If so, argue why that is the case. If inclusion is not guaranteed, provide a counter-example. *[5 marks]*