Exam 26 April 2016, questions

Computer Architecture (The University of Edinburgh)

UNIVERSITY OF EDINBURGH

COLLEGE OF SCIENCE AND ENGINEERING

SCHOOL OF INFORMATICS

INFR09009 COMPUTER ARCHITECTURE

Tuesday 26th April 2016

09:30 to 11:30

INSTRUCTIONS TO CANDIDATES

Answer any TWO questions.

All questions carry equal weight.

CALCULATORS MAY NOT BE USED IN THIS EXAMINATION

Year 3 Courses

Convener: C. Stirling
External Examiners: A. Cohn, T. Field

THIS EXAMINATION WILL BE MARKED ANONYMOUSLY

1. (a) An *orthogonal ISA*, as discussed in class, is one in which operations, registers, and addressing modes are all independent of each other. In other words, any instruction can access any register or memory location in the same manner as any other type of instruction.

   i. List one benefit and one drawback of having an orthogonal ISA. [*4 marks*]

   ii. Is MIPS an orthogonal ISA? Why do you think its designers made it that way? [*3 marks*]

   (b) The following questions pertain to Amdahl's Law.

   i. In your own words, explain Amdahl's Law. [*3 marks*]

   ii. A seemingly easy way to speed up program execution is to parallelize it and execute on multiple processors concurrently. Alas, every parallel program has some inherently sequential (that is, non-parallel) sections; for instance, at program initialization and at the end when results are produced.
   For Program X, the sequential portions add up to 20% of its execution time on a single processor. Use Amdahl's Law to determine the useful limit of parallelizing Program X by considering parallelizing it across 10, 100, and 1,000 processors. Of these three parallelization options, which one do you think is the practical sweet spot for this program? You must show your work and justify your conclusion. [*5 marks*]

   (c) Today's out-of-order processors that extend Tomasulo's algorithm with speculation have in-order fetch, out-of-order execute, and in-order commit.

   i. What is the problem with allowing the instructions to commit out of order? [*3 marks*]

   ii. What is the problem with allowing the instructions to be fetched out of order? [*3 marks*]

   (d) Processor A has a CPI of 2, while Processor B has a CPI of 3. What can you conclude about the relative performance of Processor B relative to A? Explain your answer. [*2 marks*]

   (e) Does the MIPS compiler need to know the pipeline details of the target machine (e.g., number of pipeline stages or the hit time for the L1 data cache) to generate correct code? Justify your answer. [*2 marks*]

2. (a) The following questions relate to hazards (hint: do not confuse these with data dependencies)

   i. Name and define the three types of pipeline hazards. [*3 marks*]

   ii. For each type of hazard, give an example of a hardware technique to mitigate it. [*3 marks*]

   iii. For each type of hazard, give an example of a software technique to mitigate it. [*3 marks*]
   Note: for (ii) and (iii), the same technique may be used to mitigate more than one type of hazard.

   iv. Imagine a program with abundant ILP (i.e., few true data dependencies). Which of the hazards will greatly limit the extent of reordering that both Scoreboarding and Tomasulo's algorithm can achieve on this program? Explain your answer. [*2 marks*]

   v. Name one *software* mechanism to partly mitigate the problem in part iv. Explain your answer. [*2 marks*]

   (b) A MIPS processor with a 5-stage pipeline as studied in class has a CPI of 1 in the absence of data cache misses. Loads and stores represent 20% of all instructions. The data cache miss rate is 10% and the memory latency is 100 cycles.

   i. What is the relative CPI of this processor compared to one that never misses in the cache? You must show your work to receive credit. [*4 marks*]

   ii. In order to improve performance, the cache architect suggests increasing the size of the cache, which will cut the miss rate but will double the hit time. In order to avoid reducing the frequency of the entire processor to accommodate the slower cache, the cache architect suggests adding an additional pipeline stage for the memory access (i.e., the original M stage would be split into two stages - M1 and M2, with the load value becoming available at the end of M2).

      A. Discuss what may limit the extent of performance gain provided by the cache architect's solution. Ignore clock skew and other circuit-level issues. [*3 marks*]

      B. Suggest a way to help alleviate the problem created by the extra pipeline stage. Either a software or hardware solution is acceptable. Succinctly state why your suggestion is useful. [*2 marks*]

   (c) Does speculation by the processor ever become visible to the application, operating system, or the user? If yes, describe in what way it is manifested. If not, explain the mechanism that hides it. [*3 marks*]

3. (a) This problem relates to the 3 C's of cache misses.

   i. List and define each of the three C's. [*3 marks*]

   ii. For each C, list a mitigating technique (either software or hardware). [*3 marks*]

   (b) Consider a computer system with a very small first-level data cache with the following characteristics: direct-mapped; 64B line size; 128B total capacity (i.e., 2 cache lines). The system has a separate instruction cache.

   This system is used to run the following code:

```
for (i=1; i<=1024; i++)
  X[i] = X[i] + Y[i]
```

   Assume that both X and Y have 1024 elements, each consisting of 4 bytes (single precision floating point). Both arrays are allocated consecutively in physical memory.

   The assembly code generated by a naive compiler is the following:

```
loop: lw f2, 0(r1)       # load X[i]
      lw f4, 0(r2)       # load Y[i]
      addd f2, f2, f4    # perform the addition
      sw 0(r1), f2       # store the new value of X[i]
      addi r1, r1, 4     # update address of X
      addi r2, r2, 4     # update address of Y
      addi r3, r3, 1     # increment loop counter
      bne r3, 1025, loop # branch back if not done
```

   i. Coming back to the three C's, how many misses of each type will this code generate? Your answer must have a number for each C, and show the work or an explanation for each. [*6 marks*]

   ii. Would doubling the number of blocks that the cache can store help reduce any type of misses. If so, which type(s). Justify your answer. [*3 marks*]

   iii. Would doubling the block size help reduce any type of misses. If so, which type(s). Justify your answer. [*3 marks*]

   (c) Most contemporary ISAs provide non-cacheable versions of load and store instructions (also called *non-temporal* loads and stores). Give one use case for when such instructions might be useful. Focus on the single-processor case (i.e., ignore issues of cache coherence and consistency). [*3 marks*]

   (d) Today's high-end processors integrate an ever-growing number of cores per chip. Given the trend, what kind of on-chip caches do you think architects should use in these chips: *write-back* or *write-through*? Justify your answer. [*4 marks*]