Exam 2 May 2017, questions

Computer Architecture (The University of Edinburgh)

UNIVERSITY OF EDINBURGH

COLLEGE OF SCIENCE AND ENGINEERING

SCHOOL OF INFORMATICS

INFR09009 COMPUTER ARCHITECTURE

Tuesday 2$^{\underline{\text{nd}}}$ May 2017

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: A. Cohn, A. Donaldson, S. Kalvala

THIS EXAMINATION WILL BE MARKED ANONYMOUSLY

1. (a) Consider the following code snippet:

```
for (i=0; i<9; i++)
 X[i] = X[i] + 1
```

The compiler has produced the following assembly:

```
loop: lw r5, 0(r1)       # load X[i]
      addi r5, r5, 1     # perform the addition
      sw 0(r1), r5       # store the new value of X[i]
      addi r1, r1, 4     # update address of X
      addi r3, r3, 1     # increment loop counter
      bne r3, 9, loop    # branch back if not done
```

Given a processor with a dynamic branch prediction unit, comprised of a branch direction predictor (e.g., bimodal, two-level, etc.) and a BTB, answer the following questions:

i. The first time the loop is executed, the processor fetches some number of sequential instructions after `bne` until the misprediction is discovered. Give **two** distinct reasons for why the dynamic branch prediction unit may have mispredicted the branch. [*4 marks*]

ii. Do you expect a 2-bit (bimodal) or a 2-level (adaptive) branch direction predictor to perform better on the loop above? Assume all entries of both predictors are initialized to *strongly not taken* prior to the loop's execution. Justify your answer. [*4 marks*]

iii. Some researchers have suggested tagging branches in the dynamic branch predictor as either *low confidence* or *high confidence*. A high confidence branch is very likely to have an accurate prediction, whereas the prediction for a low confidence branch may well be wrong. There are many potential uses for branch confidence to improve processor performance, energy-efficiency and/or area-efficiency. Give one possible way to use branch confidence information. Be sure to explain your answer. [*4 marks*]

(b) Processor X is a wide super-scalar with a large ROB and advanced dynamic branch prediction. On a particular code sequence, it achieves a very low IPC (i.e., high CPI) despite no branch mispredictions, no cache misses and no structural hazards. Explain how this can occur. [*4 marks*]

QUESTION CONTINUES ON NEXT PAGE

(c) As discussed in class, x86 uses a dedicated *flags* register to store the branch condition codes. The various bits in the register are set as a side-effect of executing regular arithmetic instructions. Branch instructions then check the appropriate flag bits to determine the branch outcome. E.g., the `ble` instruction would check bits N (negative) and Z (zero). List one advantage and one disadvantage of using the flags register. [*4 marks*]

(d) Give one use of PC-relative addressing for data accesses (i.e., loads/stores). Explain the benefit of using this addressing mode over the alternatives. [*3 marks*]

(e) Are WAR and WAW hazards a concern in the class 5-stage MIPS pipeline? If so, how are they handled? [*2 marks*]

2. (a) Intel recently came up with a new multi-core processor which can support up to 72 cores. Your architect colleague at Fastclock PLC has proposed a new 72-issue super-scalar processor to compete with Intel.

   i. What are the challenges in architecting a 72-issue super-scalar? [*4 marks*]

   ii. Under what situation would a 72-core single-issue processor perform better than a 72-issue super-scalar processor? State your assumptions. [*2 marks*]

   iii. Under what situation would a 72-issue super-scalar processor perform better than a 72-core single-issue processor? State your assumptions. [*2 marks*]

(b) State whether the following assertions are true or false. Justify your answer with reasons.

   i. All other things being equal, a VLIW processor of the same issue width as a dynamically scheduled super-scalar processor tends to consume significantly higher power than the super-scalar. [*2 marks*]

   ii. A prefetcher can possibly cause the miss-rate to increase. [*2 marks*]

   iii. It is possible for multiple virtual addresses to map to the same physical address. [*2 marks*]

   iv. It is possible for the same virtual address to map to multiple physical addresses. [*2 marks*]

(c) Consider an L1 cache which is 2-way set associative, has a capacity of 16 KB, and a block size of 64 bytes. Assume that both virtual addresses and physical addresses have 32 bits. Also, assume that the system has a page size of 4 KB.

   i. Compute the size of the index and the size of the tag in bits. [*1 mark*]

   ii. Illustrate, with examples, the problems that can arise if the above L1 cache were to be implemented with a VI-PT (virtually indexed physically tagged) organisation. [*4 marks*]

   iii. Describe two ways to fix the above problems. [*4 marks*]

3. (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. What do you think is the parallelism sweet spot for this program? You must show your work and explain your conclusion. [*5 marks*]

   (c) Today's out-of-order processors that extend Tomasulo's 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 number of stages and in which stages key events happen (e.g., branches resolve and loads complete) to generate correct code? Justify your answer. [*2 marks*]