COMP25212

Two hours

UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE

System Architecture

Date: Thursday 23rd May 2013

Time: 14:00 - 16:00

Please answer any THREE Questions from the FOUR questions provided

Use a SEPARATE answerbook for each SECTION

This is a CLOSED book examination

The use of electronic calculators is permitted provided they are not programmable and do not store text.

[PTO]

Section A

1. **Caches**

a) Explain why CPU caches are important in the design of modern computer systems. (2 marks)

b) Explain the differences between fully associative, direct mapped and set associative caches. (3 x 2 marks)

c) Explain how cache misses can be categorized as Compulsory, Capacity and Conflict misses. (3 x 2 marks)

A typical program accesses a number of different memory regions, one, for example containing its instructions, another containing local variables in the current stack frame, and a third containing data stored on a heap. Explain how such a program might cause Conflict cache misses in

d) a direct-mapped cache (1 mark)

e) a 2-way set-associative cache (2 marks)

f) a 4-way set-associative cache (2 marks)

g) a fully-associative cache (1 mark)

2. **Virtualization**

a) What is "System Virtualization"? (1 mark)

b) What are the major goals of System Virtualization? (2 marks)
Why are these goals not achievable within existing software components? (1 mark)

What do the following terms mean, and how can they be implemented using System Virtualization?

c) Rapid provisioning (4 marks)

d) Checkpointing and restoring (4 marks)

e) Live migration (4 marks)

A "Reverse debugger" allows a program that is being debugged to run backwards, so that a programmer can, for example, step backwards from a program crash to examine how the program reached that point. Reverse debuggers are typically implemented by storing a trace of every instruction executed and its results, so that most recent instructions can be "undone" to simulate stepping backwards. This is necessarily slow.

f) Outline how a "reverse debugger" could be implemented using, as a basis, one of the terms you have described in parts (c), (d) or (e)? (4 marks)

[PTO]

Section B

3. **Pipelining**

a) Explain the concept of pipelining in the context of processor design and the benefits its utilisation provides. (4 Marks)

b) Illustrate the operation of a classic 5-stage pipeline with the following example. Assume all memory accesses hit the cache and that all types of forwarding are implemented. (4 Marks)

```
ldr r1 x
ldr r2 y
add r4 r1 r2
ldr r3 z
sub r5 r4 r3
str r5 a
```

c) Rearrange the instructions in question b) to accelerate the execution of the code and show how the pipeline would behave with your new arrangement. (6 Marks)

d) Describe the problems that can arise with the use of pipelining and how they can be solved / alleviated. (6 Marks)

4. **Multithreading / Multicore**

a) Explain the differences between superscalar, multithreading and multicore. Are these techniques compatible with each other? (7 Marks)

b) Imagine you need to design a system which will be used to compute 4 different outputs from a single stream of input data, e.g. an embedded system to control a plantation which takes input from a distributed sensor network and decides in real-time when to perform processes such as fertilization, irrigation, fumigation or application of herbicides. If the computation of the 4 outputs features low ILP, which of the 3 previous approaches seems to be the most efficient and why? (4 Marks)

c) Enumerate and describe the different types of multithreading. (3 Marks)

d) Explain what is meant by the Memory Wall (2 Marks)

e) Explain the concept of cache coherence. (2 Marks)

f) Why is cache coherence important in multicore processing systems? (2 Marks)

END OF EXAMINATION