PRIFYSGOL CYMRU; UNIVERSITY OF WALES

DEGREE EXAMINATIONS JANUARY 2002

SWANSEA

Computer Science

CS 323   High Performance Microprocessors

Attempt 2 questions out of 3

Time allowed: 2 hours

Students are permitted to use the dictionaries provided by the University through the invigilators

CS_323
HIGH PERFORMANCE MICROPROCESSORS

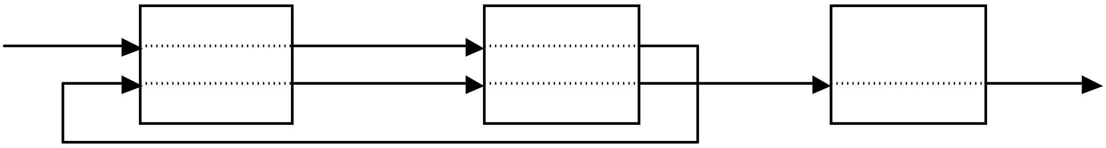
(Attempt 2 questions out of 3)

Question 1

(a) Pipelined processors attempt to overlap the execution of multiple, sequential instructions. To do this, they must resolve a number of *dependencies* that, if ignored, would disrupt execution and possibly result in erroneous results. Explain, perhaps with the aid of simple examples, the dependencies *that apply to pipelined processors only.* [**5 marks**]

(b) One of the dependencies in your answer to (a) is inherent in the sequential nature of programs. Unlike the other dependencies, in all currently-available microprocessors this particular dependency always causes a pipeline stall. Which dependency is this? In principle, pipeline stalls caused by this dependency could be removed, in at least some cases: how? [**5 marks**]

(c) Consider the following pipeline.



Construct the reservation table, determine the forbidden latencies, and hence derive the collision vector. **[5 marks]**

(d) In principle, a pipeline such as that shown in (c) should be able to complete two operations in four cycles. How can you infer this from the reservation table? How would you determine, from the reservation table, if this were actually the case? What steps would you take to modify a pipeline in order to ensure that it reached the theoretical bound on throughput? Note: you are not being asked to actually determine the minimum latency cycles.
[**10 marks**]

Question 2

(a) The traditional approach to compiler code optimisation essentially involves minimising the number and execution time of instructions. In the case of a superscalar processor, it is also necessary to minimise the number of empty instruction slots. *Software pipelining* is one technique that can be used that interleaves multiple loop iterations. Describe software pipelining, with the aid of a simple example. **[10 marks]**

(b) Consider a machine with a 4Gbyte memory address space, and a cache of 1K 64-bit words. Describe, perhaps with the aid of diagrams, *direct mapping*, *associative mapping*, and *8-way set-associative mapping* for this particular machine. Your answer should include the sizes of the *tag*, *word* and *slot* fields in each case. **[10 marks]**

(c) Cache misses can be broadly characterised as one of: *capacity*, *conflict* and *compulsory*. Explain what each of these means. **[5 marks]**

Question 3

(a) *Reorder buffers* are commonly found in superscalar processors. What problems do they solve? **[10 marks]**

(b) VLIW machines are seen to solve a problem with superscalar machines: namely that a large amount of hardware is devoted to scheduling instruction execution at runtime. However, they have their own problem: what is this? How does Intel's Itanium-series processor solve some of these?
[10 marks]

(c) Branch history mechanisms generally keep track of the behaviour of branch instructions: typically using 1-4 bits of information. When using two bits of branch history information it is common to require two successive incorrect "guesses" before changing prediction: for example, before changing a prediction from *taken* to *not taken*, the branch must be not taken on two successive occasions. An alternative strategy is to require two successive mis-predictions when changing from *taken* to *not taken*, but only one mis-prediction when changing from *not taken* to *taken*. Why might this be a useful strategy? What additional information might also be useful in such circumstances? **[5 marks]**