CS-323
HIGH PERFORMANCE MICROPROCESSORS
Attempt 2 questions out of 3

Question 1

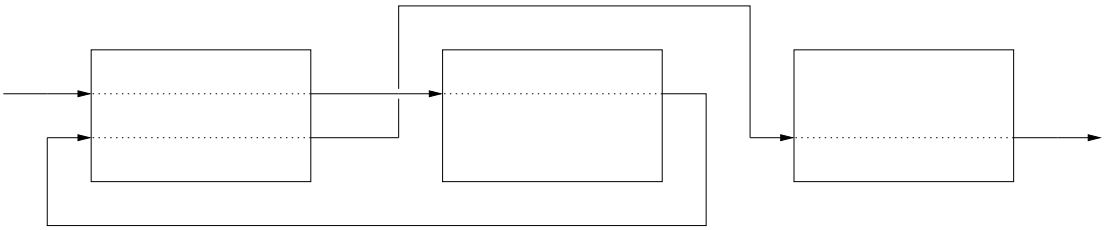
(a) Describe *Read after Write* (RAW) hazards by using a simple example.

[5 marks]

(b) Describe some *simple* strategies for minimizing the effects of RAW hazards. Also, describe a more advanced strategy that is not yet employed in commercial processors.

[5 marks]

(c) Consider the following pipeline.



Construct the reservation table, determine the forbidden latencies, and hence derive the collision vector.

[5 marks]

(d) Describe how to implement a simple 'greedy' control algorithm for the pipeline in (c) using OR gates and shifters.

[5 marks]

(e) The algorithm described in (d) is *sometimes* not optimal. Under what circumstances might you still wish to use it if that is the case?

[5 marks]

Question 2

(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 1 to 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 not be 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]

Question 3

(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,

(i) direct mapping;

(ii) associative mapping;

(iii) 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:

(i) capacity;

(ii) conflict;

(iii) compulsory.

Explain what each of these means.

[5 marks]