1 This question concerns cache architecture in the IBM Power5 processor, as described in the paper “IBM Power5 Chip: A Dual-core Multithreaded Processor” (IEEE Micro March-April 2004), which you should have available to you in the examination. **See, in particular, page 42.** Where the paper is incomplete, you are invited to speculate using your understanding of the underlying architectural principles.

a What is stored in the L3 cache directory?

b The Power5’s L3 cache directory is on-chip, although the cached data is stored off-chip. The L3 cache is 9-way set-associative, with 512-byte cache blocks managed as four 128-byte sectors. Physical addresses are 42 bits. How many bits of data storage are required for the on-chip directory? Explain carefully any other assumptions you have to make.

c The decision to place the L3 cache directory on-chip does not improve L3 cache access time. How does it improve average memory access time?

d A major difference between the Power5 and the earlier Power4 design is that the L3 cache is on the processor side of the fabric controller, instead of the memory side. Why is this particularly beneficial in very large SMP configurations?

e If the Power5 had a victim cache, where would it fit in the memory hierarchy, and what would its purpose be?

The five parts carry, respectively, 15%, 25%, 20%, 20%, and 20% of the marks.

2 This question requires access to the paper “IBM Power5 Chip: A Dual-core Multithreaded Processor” (IEEE Micro March-April 2004), which you should have available to you in the examination. Where the paper is incomplete, you are invited to speculate using your understanding of the underlying architectural principles.

This question concerns an alternative to the Power5 design philosophy (based loosely on the CELL POWERPC Processing Element). Assume, for the purposes of this question, the following architectural features:

- In-order, dual-issue pipeline
- Integer pipeline with 11 stages
- One fully-pipelined floating-point add/subtract unit
- One fully-pipelined floating-point multiply unit
- Two-way simultaneous multithreading (SMT)
- L1, L2, L3 caches and memory system similar to Power5
- A tournament branch predictor like the Power5’s
- clock rate twice as fast as the Power5

a Suggest a sensible structure for the integer pipeline. Explain briefly the function of each of the 11 stages. Take care to include how the branch predictor is accessed.

b Assuming the pipeline structure you have described, what is the minimum load-use delay? Explain carefully any assumptions you have to make.

c Assuming the pipeline structure you have described, what is the minimum conditional branch misprediction penalty in this design? Explain carefully any assumptions you have to make.

d This design supports two-way simultaneous multithreading. At which pipeline stage does this processor select between the two threads? What information is required to make this decision?

e Outline briefly the characteristics of an application program that would run faster on this design than on the Power5. Justify your answer carefully.

The five parts carry, respectively, 50%, 10%, 10%, 10%, and 20% of the marks.

3 This question concerns branch prediction in the IBM Power5 processor, as described in the paper “IBM Power5 Chip: A Dual-core Multithreaded Processor” (IEEE Micro March-April 2004), which you should have available to you in the examination. **See, in particular, pages 43–44.** Where the paper is incomplete, you are invited to speculate using your understanding of the underlying architectural principles.

a The Power5 uses three branch history tables in a “tournament” mechanism to predict branch directions. Explain how the three branch history tables are updated when a branch outcome becomes known.

b Give an example of a situation where a branch outcome is easily predicted correctly, but the branch target is likely mispredicted.

c Consider a “while” loop which is always executed exactly three times. What would the misprediction rate be for a (2,2) *gselect* predictor, assuming no interference from other branches in the program? Justify your answer carefully.

d A “trace cache” is an instruction cache indexed by a combination of instruction address and branch predictor state. The idea is to store in it compacted eight-instruction fetch packets from predicted-taken control-flow paths. What problem does this proposal address? What additional hardware would be needed?

The four parts carry, respectively, 40%, 20%, 20%, and 20% of the marks.