UNIVERSITY OF LONDON
IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 2004

BEng Honours Degree in Computing Part III
MEng Honours Degree in Electrical Engineering Part IV
BEng Honours Degree in Information Systems Engineering Part III
MEng Honours Degree in Information Systems Engineering Part III
MEng Honours Degree in Information Systems Engineering Part IV
MSci Honours Degree in Mathematics and Computer Science Part III
MSc in Advanced Computing
for Internal Students of the Imperial College of Science, Technology and Medicine

This paper is also taken for the relevant examinations for the Associateship of the City and Guilds of London Institute
This paper is also taken for the relevant examinations for the Associateship of the Royal College of Science

PAPER C332=I3.26=I4.56=E4.31

ADVANCED COMPUTER ARCHITECTURE

Tuesday 4 May 2004, 14:30
Duration: 120 minutes

Answer THREE questions

Paper contains 4 questions
Calculators not required

1 This question concerns branch prediction in the Intel Itanium 2 processor, as described in the paper "Itanium 2 Processor Microarchitecture" (McNairy and Soltis, IEEE Micro March-April 2003), which you should have available to you in the examination. **See, in particular, pages 47–48.** Where the paper is incomplete, you are invited to speculate using your understanding of the underlying architectural principles.

a Under what *four* conditions might the Itanium 2 suffer a non-zero branch prediction penalty when executing a taken branch instruction?

b When is the Level 2 Branch Cache (L2B) updated?

c What could go wrong due to the lack of tags in the L2B? Why is this not a serious problem?

d Despite often suffering no branch prediction penalty, taken branches lead to unused instruction issue slots. What measures can the compiler (or programmer) take to address this problem?

The four parts carry, respectively, 40%, 10%, 15%, and 35% of the marks.

2 This question concerns fetching instructions in the Intel Itanium 2 processor, as described in the paper "Itanium 2 Processor Microarchitecture" (McNairy and Soltis, IEEE Micro March-April 2003), which you should have available to you in the examination. **See, in particular, pages 45–47.** Where the paper is incomplete, you are invited to speculate using your understanding of the underlying architectural principles.

a The Itanium 2's L1I TLB has 32 entries. The 16KB L1I cache is four-way set-associative with 64-byte lines and prevalidated tags. How many bits are occupied by tags?

b A conventional alternative to this L1I cache design would be a virtually-indexed, physically-tagged cache (physical addresses are 50 bits), accessed concurrently with a TLB. How many bits need to be compared to identify a cache hit? How many bits have to be compared in the Itanium 2's prevalidated tags design to identify a cache hit?

c The Itanium 2's streaming instruction prefetch mechanism is initiated when a "br.many" branch instruction is encountered (the ".many" is a hint). When should this hint be used?

d The Itanium 2's "brp.many" instruction initiates an L2 access to fetch two L2 cache lines into the L1I cache. It is intended as a prefetch, to be used in advance of an actual branch. To hide the latency of an L2 hit fully, it must be issued 9 cycles before the branch. Suggest an example situation where latency hiding might actually be achieved.

e What is the cache pollution problem with prefetching? Can cache pollution occur with the prefetching in your answer to part(d) above?

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

3 This question concerns instruction scheduling in the Intel Itanium 2 processor, as described in the paper "Itanium 2 Processor Microarchitecture" (McNairy and Soltis, IEEE Micro March-April 2003), which you should have available to you in the examination. **See, in particular, pages 48–51.** Where the paper is incomplete, you are invited to speculate using your understanding of the underlying architectural principles.

a First, *in contrast to* the Itanium 2, consider a single-issue, dynamically-scheduled processor with speculative execution (based on Tomasulo's algorithm extended with a re-order buffer (ROB).

i) In what *two* ways are the issue-side registers updated?

ii) In what *two* ways are ROB registers updated?

iii) In what circumstances, apart from conditional branches, can the Commit stage discard instruction results?

iv) What circumstances lead to a stall in the instruction Commit stage of the pipeline?

v) What circumstances lead to a stall in the instruction issue stage of the pipeline?

b Now consider Itanium 2.

i) In what circumstances will the DET stage discard instruction results?

ii) What circumstances lead to a stall in the DET stage of the Itanium 2 pipeline?

iii) What circumstances lead to a stall in the EXP (instruction expand) stage of the Itanium 2 pipeline?

The two parts carry, respectively, 60%, and 40% of the marks.

4a You are designing a search engine which will handle 540 search requests per minute on average. The response-time requirement for the engine is that all requests should be answered within 0.25 seconds (not including network delays).

i) What is the upper bound on the average number of requests in the system if the response-time requirement is to be met?

ii) If you assume that a top-level abstraction for the search engine will be an M/M/1 queue, how fast must the search engine process requests?

iii) Now using an M/D/1 model assumption, where the mean queue length, $\mathbb{E}(N) = \rho + \frac{\rho^2}{2(1-\rho)}$, recalculate the minimum service rate of the search engine. ρ is the system utilisation.

b Given a web-graph, G, representing the link structure of the internet and a matrix P defined by $P_{ij} = 1/\deg(u_i)$ if a link exists from page u_i to u_j in G and 0 otherwise.

i) Explain what modification is made to P to get P' in order to overcome the problem of pages that have no links to other pages.

ii) Describe the two roles of the personalisation vector in the PageRank algorithm.

iii) Given the matrix equation representing the modified transition matrix, A, in the PageRank iteration $\vec{x}_{(k+1)} = \vec{x}_{(k)}A$:

$$A = cP' + (1-c)E$$

Explain why the iteration can not be executed directly using A.

iv) In the PageRank algorithm, itself:

$$\vec{x}_{(x+1)} = c\vec{x}_{(k)}P + (1 - c\,\|\vec{x}_{(k)}P\|_1)\vec{p}$$

Explain with careful reference to the equation why this is a more scalable implementation of the matrix–vector iteration.

The two parts carry, respectively, 35%, and 65% of the marks.