Two hours

UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE

High Performance Microprocessors

Date: Wednesday 21st May 2008
Time: 09:45 – 11:45

Please answer any THREE Questions from the FIVE questions provided
This is a CLOSED book examination

The use of electronic calculators is NOT permitted.
1. a) Explain how splitting instruction execution into a sequence of a small number of (typically 5) phases and then implementing these in the form of an instruction pipeline can lead potentially to an instruction execution rate of one instruction per clock cycle at a high clock rate. 

b) Apart from technological advances (such as smaller feature size), what architectural change can be made to an instruction pipeline that will allow use of a higher clock rate (i.e. a shorter clock period).

c) Explain why simple instruction pipelines rarely achieve an instruction execution rate of one instruction per clock cycle. Be specific about the problems that can occur and that may cause the pipeline to stall. (Note: there are several distinct kinds of such problems; full marks will be awarded only for complete coverage of these.)

d) For each problem identified in your answer to part (c), describe how you might alleviate it, by changing the way instructions are executed, by adding extra hardware, or by recompiling in order to create more suitable machine code.
2. a) The following source code extract (written in C) implements a square dense matrix multiplication, where \( A \), \( B \), and \( C \) are \( n \times n \) arrays of double-precision floating point values.

```
for (i := 0; i < n; i++)
    for (j := 0; j < n; j++)
        \{t := 0;
        for (k := 0; k < n; k++)
            t += A[i, k] * B[k, j];
        C[i, j] := t;
\}
```

List the unoptimised sequence of machine code instructions that you would expect a compiler to generate for the innermost `for` loop of this program (you should use a MIPS-like instruction set; if you have difficulty remembering the mnemonic codes for such an instruction set, please make it clear what each instruction mnemonic that you use does). State any assumptions you make.

(8 marks)

b) The unoptimised machine code executes \( 2n+1 \) branch instructions while executing the innermost `for` loop once. Describe a simple code optimisation that will reduce the number of executed branch instructions to \( n+1 \) per execution of the innermost `for` loop.

(2 marks)

c) The above code is executed in a 2-way superscalar architecture which can issue up to one integer and one floating point instruction per clock cycle; all load and store instructions are executed in the integer unit. The architecture implements aggressive dynamic scheduling and speculation so that issue of eligible instructions is immediate; it also implements non-deferred branches. How many clock cycles will it take to complete a single execution of the innermost `for` loop? Explain your answer clearly and state any assumptions you need to make.

(4 marks)

d) Explain how compiler-based loop unrolling of the innermost loop will affect the above execution behaviour. Assess the number of clock cycles required for a single execution of the innermost `for` loop when it is unrolled 2, 4 and 8 times. You may assume in each case that \( n \) is a multiple of the number of times the loop is unrolled.

(6 marks)
3.  

a) Outline the principles of value prediction and explain how it can lead to improved processor performance. You do not need to give details of the hardware structure of a processor using the technique.  

(3 marks)

b) Given a sequence of integer values of the general form: \(x(0), x(1), x(2), x(3), \ldots, x(i), x(i+1), \ldots\), where \(x(i+1) = x(i) + k*i\) and \(k\) is a constant, show how it is possible to predict the next value in the sequence from the last three values produced. What is predicted when fewer than three prior values have been produced?  

(4 marks)

c) An integer instruction in a program produces the values 0, 0, 1, 3, 6, 10, 15, 21, 28, 0, 0, 1, 3, 6, … in a repeating sequence. Explain how the value predictor in part (b) will react to this repeating sequence.  

(4 marks)

d) A context value predictor is proposed which maintains a partial history of prior values by storing the least significant two bits of the values produced by the integer instruction on its previous two executions (for simplicity, assume that the history for this instruction is captured in full). Sketch the storage structure required to implement this new predictor and show the contents of the key elements in the structure after the sequence has repeated a number of times. Describe how this context value predictor will react to the repeating sequence given above. Compare this behaviour with that described in your answer to part (c).  

(6 marks)

e) Explain how you would modify the above context value predictor so that it can correctly predict all values in the given repeating sequence, once the prior history table has been appropriately filled.  

(3 marks)
4. a) Describe the techniques of Direct Mapped, Set Associative and Fully Associative caching, comparing the potential performance and cost of each. (7 marks)

b) Discuss the techniques of hardware and software cache pre-fetching, with emphasis on the benefits and possible drawbacks of the two approaches. (4 marks)

c) What is a 'Stream Buffer' and why is it useful in the context of pre-fetching. (3 marks)

d) A two level cache system has a Level 1 (L1) cache with a 1 ns (nanosecond) access time and a level 2 (L2) cache with 6 ns access time. The main memory access time is 40 ns. The following tables show the read hit rates for different possible sizes of the two caches.

<table>
<thead>
<tr>
<th>L1</th>
<th></th>
<th>L2</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>Hit Rate</td>
<td>Size</td>
<td>Hit Rate</td>
</tr>
<tr>
<td>32 kbytes</td>
<td>96%</td>
<td>512k</td>
<td>75%</td>
</tr>
<tr>
<td>64 kbytes</td>
<td>98%</td>
<td>1024k</td>
<td>90%</td>
</tr>
</tbody>
</table>

The L2 cache hit rate given in the table is the local hit rate; that is, the percentage of accesses referred to it from the L1 cache that result in a hit. Assuming that chip area can be devoted to either the 32k + 1024k or the 64k + 512k combination, evaluate which is the better choice, in terms of read access performance, by calculating the average read access time for each combination. (6 marks)
5. a) Describe the problem of *cache coherence*. Explain how this problem may arise in a *scalar* microprocessor if it has separate Level 1 data and instruction caches. (3 marks)

b) Explain how a 'snoopy bus' can be used to overcome the above problem. (2 marks)

c) Describe how cache coherence becomes a more serious problem in the context of shared memory multiprocessors. In particular, explain why a 'snoopy bus' cannot in general be used to overcome the problem. (5 marks)

d) Discuss the difference between the 'write invalidate' and the 'write update' protocols used to maintain coherence in a shared memory multiprocessor. (4 marks)

e) Describe the operation of the MESI (Modified; Exclusive; Shared; Invalid) cache coherence protocol. Your answer should include a discussion of the MESI protocol states and an informal description of the processor and bus actions that cause transitions between them. There is no need to reproduce a complete state transition diagram. (6 marks)