Examination for Grid Hardware Infrastructure

19th December 2006 13:30-16:30

Instructions

You are required to answer question 1 plus any 2 of the remaining 3 questions. You may not bring books or lecture notes into the examination and you have three hours to complete the examination. Each question carries an equal weight in the assessment. Questions parts are labeled with percentage marks for that part in square brackets e.g. [x%]. The percentage is of the final assessment, i.e. a total of 60% for the examination or 20% for each question answered.

Question 1.

(a) In 1980 memory speed and processor speed were comparable. However, processor speed has increased at a much faster rate than that of DRAM (about 60% per year for processors and 7% per year for memory). How has this affected computer architecture during this period. Currently memory speed is about 100 times slower than processor speed, if this trend continues for the next 10 years what will the impact be on future microprocessors? [4%].

(b) A pipeline implements concurrency in an operation by dividing the operation into a number of dependent stages and performing each operation concurrently over a sequence of operations. For example, a function unit performing floating point addition requires the following sequence of operations: *subtract exponents (SE), shift mantissas(SM), add mantissas(AM), normalise(N)*. Draw a diagram, with time on one axis and the sequence of operations on the other axis, to show the concurrency in a pipelined floating point addition. What is the maximum speedup that can be obtained by this pipelined this implementation of floating-point addition compared to a non-pipelined implementation and under what circumstances will this occur? [4%]

(c) Define the parameters $r_{infinity}$ and $n_{1/2}$ that characterise the performance of a pipeline in the absence of any bubbles. Derive a formula for the time required to compute n operations in a pipeline and hence a formula for the computational rate of a pipeline as a function of n. You may assume the pipeline has L stages and a cycle time of t seconds. Compute the computational rate of this pipeline for $k^* n_{1/2}$ where k={1,2,3,4} and hence sketch a graph of this function using units of $n_{1/2}$ and 1/t. [4%]

(d) Define the following metrics used to characterise multi-computer networks, what are the units of these metrics? [8%]

(i) node degree
(ii) channel bandwidth
(iii) average distance
(iv) bisection bandwidth

Question 2.

(a) VLIW is used extensively in the embedded systems market where it achieves a high performance with low power dissipation. What features of the VLIW architecture give this power saving when compared to a superscalar architecture, which is used extensively in the commodity computer market? What major disadvantage does the VLIW architecture have when addressing the commodity computer market? [4%]

(b) and (c) Two VLIW architectures are used extensively in the commodity computer market, the Intel EPIC architecture and the Transmeta Crusoe/Eficeon architecture. For each, give a detailed description of the architecture and the way in which they process instructions - using diagrams where appropriate. Identify in your answers the manner in which these architectures differ from a conventional VLIW architecture, in overcoming the limitation you have identified in part (a) and which allows them to be used in the commodity market. [8% + 8%]

Question 3.

(a) How do the following dependencies arise in the execution of assembly/binary code and under what scheduling conditions?

(i) Data dependency [2%]
(ii) Output dependency [2%]
(iii) Anti dependency [2%]

(b) Identify all dependencies in the following fragment of assembler code where the instructions are allowed to issue and complete out of order. You should identify the type of dependency and identify the instructions by label (e.g. instruction a is dependent of instruction b – data dependency). Which of these dependencies are true data dependencies?

{For the avoidance of doubt, there are four register specifiers $0..$3; operations update the first register specifier in an instruction and the second and third specifier (where present) are the operands of the operation; a load word (lw)

instruction updates the first register specifier, the other is used as an index register; a store word instruction (sw) stores the data from the first register specifier, the other is used as an index register.} [6%]

```
1     addi $0 $0 4
2     lw $1 A($0)
3     mpy $2 $1 $1
4     lw $1 A+1($0)
5     mpy $3 $1 $1
6     add $1 $1 $3
7     sw $1 A($0)
```

(c) Register renaming logic maps a logical to a physical register address for the purpose of avoiding anti dependencies and output dependencies in an out-of-order issue processor. At any point in the execution of code, there is just one valid mapping for a given logical register specifier, and subsequent instructions will use that mapping to read their operands until a further remapping is made. Two implementations of the remapping function are possible, which map a logical address to a physical register address:

1. RAM, where the current physical address is stored in a table the size of the logical register file and read using the logical register file address.

2. CAM, where the logical address is stored in a content addressable table the size of the physical register file. In this case, the logical address is used as a tag and multiple matches to the tag when mapping a logical address to a physical address are resolved using a currently-valid bit.

Draw a diagram of both implementations of the register-renaming mapping function for a logical address of between 0 and 3 and a physical address of between 0 and 15. [4%]

As predicted branches may turn out to be wrong, the current mapping may have to be rolled back to a prior one if the prediction was incorrect and thus a copy of a valid mapping must be made for any branch that is currently undecided.

Compare the sizes of these tables in total bits of data stored, assuming that a current and two prior mappings must be maintained in the mapping function. How will this ratio change as the width of issue increases? [4%]

Question 4.

A shared-memory parallel computer is constructed using 64 microprocessors, each with a memory system shared by all other processors via the network giving non-uniform memory access. The processors are interconnected using a 2-

dimensional, torroidal, packet-routed network, comprising 64 nodes with one bidirectional channel to the processor and one bidirectional channel to each neighbouring node.

The processor has a frequency of 2GHz and can perform two floating-point operations in one cycle.

The memory system has a 64-bit data interface and a 32-bit address at each node. It requires 4 cycles from the address being made available to the memory to the data being available on at the memory's output. The memory system uses synchronous DRAM and can accept a new address on each cycle from either the processor or network interface.

The network channel has a width of 8 bits and implements remote memory reads and writes. Message formats are defined for implementing a memory read, a read response, a memory write and a write acknowledge. Each channel has a frequency of 500MHz and the router at each node can (theoretically) recieve an 8-bit flit from each channel and forward it using dimension-order routing to an output channel on for the next channel cycle. I.e. the latency of each routing step for a message is one channel cycle per node.

(a) For a single processor and its local memory, derive the following parameters [4%]:
   (i) memory bandwidth;
   (ii) memory latency;
   (iii) processor speed;
   (iv) processor performance in floating point operations per second.

(b) For the network, derive the following parameters [4%]:
   (i) node degree
   (ii) channel bandwidth
   (iii) diameter
   (iv) bisection bandwidth

(c) Define the packet formats for each message type assuming that reads and writes are implemented on 64 bits of data. What is the maximum read latency in a lightly-loaded network, assuming that messages are propagated using wormhole flow control. What will limit the memory bandwidth under these conditions? [8%]

(d) How will memory bandwidth and latency be affected as load on the network increases, and why? [4%]