Department of Computer Science Qualifying Examination

Computer Architecture

Solutions

The intent is that it should take approximately one hour to answer the questions on this exam. The questions are equally weighted.

1. Four of the classic network topologies are the bus, ring, crossbar, and hypercube.

   A. Compare the bandwidth, latency, and cost of these topologies.

Topology	***Bandwidth***	***Latency***	***Cost***
Bus	*O(1)*	***O(1)***	***O(n)***
Ring	***O(n)***	*O(n)*	***O(n)***
Crossbar	***O(n)***	***O(1)***	*O(n^2)*
Hypercube	***O(n)***	*O(log n)*	*O(n log n)*

   (optimal paramaters are in bold for each topology)

   The bus, ring, and crossbar provide three extrema in the design space: each is optimal for two of the three metrics, but quite poor in the third. The hypercube is optimal for bandwidth, and is a compromise between good latency and good cost.

   B. How have developments in network implementation (hardware cost, wormhole routing, etc.) affected the importance of the metrics in part A?

2. Suppose you have a computer system with the following characteristics:

Address Width	43 bits
Page Size	8K
Size of Table Entries	8
Number of Entries in Page Table	1K
Size of Level 1 Cache	16K
Level 1 Cache Block Size	16
Level 1 Cache Associativity	8

   A. How many levels of page tables are there (all levels of paging have the characteristics shown above)

   There will be three levels of paging.

   B. How is an address divided into fields for virtual memory lookups (be sure to give the width of each field)?

   The offset into a page has to be 13 bits, since a page is 8K. The number of entries in each level of table is 1K, so 10 bits are required to index the table. The result looks like this:

10	*10*	*10*	*13*

The 10-bit wide fields are indexes into each of the three levels of page table. The 13-bit wide field is the offset into the page.

C. How is an address divided into fields for level 1 cache lookups (be sure to give the width of each field)?

As the block size is 16, four bytes are required to index an address within a block. Since the cache size is 16K, the number of blocks is (cache size/blocksize) = 1K and the number of sets is (number of blocks/asociativity) = 128. Seven bits are required to index the proper set, leaving 43-7-4 = 32 bits for the tag. The result looks like this:

32	*7*	*4*

D. If the cache is in the physical address space, is it feasible to do a TLB lookup for the virtual memory at the same time as a cache lookup?

Yes. Since the total width of the set index plus the offset index is less than the width of the page offset field, it is possible to retrieve the cache block simultaneously with doing a TLB lookup, and then do the tag compare based on what is returned by the TLB.

3. Acquiring and releasing locks on shared data can be thought of as claiming and relinquishing ‘‘ownership’’ of the data. When you request a lock you request ownership; when you release the lock you are relinqushing the ownership. How does this relate to message passing? What operation corresponds the acquiring a lock? What operation corresponds to releasing a lock? Which mechanism gives the programmer more control over the ownership of the data?

   Releasing a lock corresponds to a send; acquiring a lock corresponds to a receive. A lock release/acquire pair transfers ‘‘ownership’’ of the data from one process to another; a send/receive pair similarly transfers this ownership. Send/receive gives more control, since the data is sent to a particular process rather than being made available for any process to grab.

4. The following somewhat provocative statement appeared in the comp.arch newsgroup:

```
Subject:  Re: Is RISC dead? (was: Re: K7’s FP performance inches
past P-III’s)

Date: 1999/05/19

Author:  Piercarlo Grandi <piercarl@Dial.PIPEX.com>

Well, my taqke here (and I said so in "comp.arch" quite a while
ago) is that _architecture_ is dead: with essentially infinite
silicon/design budgets virtually any architecture, including x86,
can be made to perform; in other words pnly implementation matters.
```

   Respond to Piercarlo’s comment. Have we reached the point that the instruction set is irrelevant, because with current technology any pig can be made to fly? Are some aspects of the architecture still important, and others less so?

   Important point here is the quality of the student’s discussion.

   My own commentary would be that there is a lot of truth to the statement. Since complex instructions can be decoded into simpler RISC-like instructions, and these RISC-like instructions reordered as necessary, the actual high-level instruction set becomes much less important so long as the *pipelines can be kept full. A few things remain important, in particular the address and data sizes: 32 bit addresses have pretty much run out of room, so an enhancement to 64 bits is needed.*