1. Division

Unlike multiplication, division generally requires too many gates to be implemented in full combinational logic. Instead, most computers implement division using an iterative algorithm that decomposes the problem into smaller steps.

a) Enumerate and describe the steps of the simple binary *restoring* division algorithm.

b) Explain how the *non-restoring* division algorithm differs from the restoring division algorithm. What is the fundamental idea here?

c) Reciprocal approximation is a different approach to fast iterative division. Briefly describe the essential ideas of how it works? Why is it faster? Why isn't it used in most computers?

2. Cache Memory

A 64K-byte cache is being designed. Physical addresses are 24 bits. The cache will either have lines of either 32 or 64 bytes. It will either be direct-mapped, or 2-way set associative. If it is set associative, replacement policy will be LRU. For each of the four possible combinations above, determine the minimum number of bits required for all memory other than the data. In each case, indicate the size and purpose of each field.

3. Synchronization in Multiprocessors

A paper from the reading list by Goodman, Vernon, and Woest discusses several synchronization scenarios. Consider three synchronization event scenarios: semaphores, pairwise data sharing, and barrier synchronization.

For each of the above scenarios, describe and discuss efficient synchronization primitives for a cache-coherent multiprocessor with a general interconnection network. Describe why the chosen primitive is efficient for the chosen scenario, that is, what problem does the chosen primitive solve (that other primitives do not solve).

4. Branch Prediction

Branch prediction is an important performance optimization for highly-pipelined and superscalar processors. There are a variety of ways to predict branches, each with its own set of pros and cons.

a) Branch prediction schemes are usually classified as either *static*, *dynamic*, or a combination of both. Define what is meant by *static* and *dynamic* in this context. Give an example of a hybrid scheme. Discuss the advantages and disadvantages of static vs. dynamic schemes.

b) Describe an implementation of the classic one-bit dynamic branch prediction algorithm.

c) How does the two-bit dynamic algorithm differ from the one-bit dynamic scheme and why?

d) Overall branch prediction accuracy is not the only issue that an architect must consider when deciding which scheme to implement. Discuss some of the other factors that must be considered before deciding upon a branch prediction scheme.

This document was created with FrameMaker 4.0.4

5. Instruction-Level Parallelism

Vector processors use vector instructions with which many operations can be issued for execution, or launched, with one instruction. Over the years, vector machines have been very successful at executing scientific applications. More recently, several people have argued that the demise of vector machines is imminent, because multiple instruction issuing machines, such as superscalar and VLIW machines, provide a (technically) superior approach to launching multiple operations.

a) Argue why vector instructions are technically superior to the superscalar/VLIW approach for launching multiple operations.

b) Argue why the superscalar/VLIW approach is a technically superior to the vector instruction approach for launching multiple operations.

For both the above questions, limit your answers to technical reasons.

6. RISC/CISC

John Mashey has argued that "most RISCs" have certain characteristics, among which are the following:

- One size of instruction in an instruction stream, and that size is 4 bytes
- A handful (1-4) of addressing modes.
- No indirect addressing in any form.
- No operations that combine load/store with arithmetic.
- Use of an MMU for a data address no more than once per instruction.
- Have ≥ 5 bits per integer register specifier.

Mashey also notes that this list distinguishes architectures first introduced before and after the middle of the 1980s. One might conclude, therefore, that these properties characterize current opinions about best architectures when relieved of the constraint of compatibility.

a) For each of the points, give an example of *any* real architecture that violates the characteristic.

b) Assume you are to design a new instruction set that is intended to survive multiple generations of new technology. For each of the characteristics, explain why your new instruction set should (or should not) comply.