Computer Architecture Screening Exam

Fall 1988

Directions

For the **Breadth** exam, answer questions (1), (2), (3), (4) and (5).
For the **Depth** exam, answer all 8 questions.

(1) Suppose you want to design a 4-bit slice that will be used by people building CPUs from 16 to 64 bits wide.

(a) List the inputs and outputs of your bit-slice, explaining each signal.

(b) What limits the speed of a CPU designed with your bit-slices?

(c) Can you suggest some auxiliary chips that would either speed up the bit-sliced CPU or make its design simpler?

(2) The Test-and-Set instruction is an atomic read-modify-write operation that sets a bit in memory to one. Is it necessary for a uniprocessor? Give an example of how it is used. Why is it necessary that the operation be atomic? Give an example of failure if the operation were not atomic.

(3) You've just designed a floating point addition and multiplication unit in which both floating point addition and multiplication take roughly the same amount of time. Your boss, an expert in integer functional unit design, thinks that you must have made some mistake since in most integer functional unit designs, addition and multiplication have significantly different execution times. Explain to him why it is possible that floating point addition and multiplication can have roughly the same execution times even though that may not be the case for integer addition and multiplication. In your explanation, you must give a clear description of the steps involved in each case as well as a justification of the speed of each step and each hardware unit.

(4) Machine A has a faster clock cycle than machine B, but also has a longer pipeline. What factors determine which machine is faster?

(5) Consider a virtual memory system.

(a) What are the advantages and disadvantages of using a larger page size?

(b) If you wrote a new operating system, how would you do page replacement and why?

(c) What are page-dirty bits?

(6) Most current-day computers are still based on von Neumann's model of a computer in which a CPU is connected to a memory through a single bus. It is well known that the peak performance of such computers is limited by: (i) the bandwidth of the memory and (ii) the bandwidth of the interconnecting bus. This is also known as the *von Neumann bottleneck*.

In such a computer, the CPU must rely less on memory (and bus) bandwidth if it is to obtain higher performance. The traditional approach to decrease this bandwidth requirement has been to have compact, complex instructions.

Several modern processors go against this conventional approach. Not only are the processors much faster than older processors (thereby increasing the demand on the memory), complex instructions are broken into several simple instructions that require many more bits to encode than a single complex instruction!

Discuss the ways that modern processors use to alleviate the bandwidth bottleneck. Discuss the pros and cons of each approach.

(7) Suppose an unusual device has been developed that can store approximately 10^{30} bits of information with a cost of only 10,000 dollars. Access time is about 10 microseconds per bit. All bits can be randomly accessed, meaning there is no advantage to accessing bits in a particular order.

(a) What would you do with such a storage device? How would it impact the way current systems are built? Would this make any new applications feasible?

(b) If you wanted to turn it into a peripheral, what sort of controller or other hardware would you add? What commands would this controller respond to?

(8) Consider a function that can be pipelined into n stages, where the time to perform the sub-function in stage i is t_i, for $i = 1, n$. Let buffers between pipeline stages require delays of T seconds. Assume that there are no data dependencies between successive operations, that is, no pipeline hazards.

(a) What are the latency and throughput for a serial implementation of the function?

(b) Neglecting clock skew considerations, what are the latency and throughput for an n-stage pipelined implementation of the function?

(c) Neglecting clock skew considerations, can reducing the number of stages in the pipeline improve latency or throughput?

(d) How does clock skew affect the pipeline latency and throughput?