Estimated time: 90 mins Closed book/notes exam	**DWE: Systems / Fall 1996** **Computer Architecture**	Prof: Azer Bestavros Email: best@cs.bu.edu

Answer enough questions to secure a 100-point total

State any asumptions clearly

a. Briefly explain why RISC processors often employ a hardwired control unit design, whereas CISC processors often employ microprogramming. [10]

b. Derive the average distance that a message needs to travel between a randomly selected source and a randomly selected destination in an n-dimensional hypercube. [15]

c. A performance study of a *directly mapped, write back* 4K cache system that uses LRU cache replacement yields the following results.

Block Size in Bytes	8	16	32	64	128	256	512
Miss Rate %	9.1	7.9	7.3	7.1	7.5	8.1	9.5

Despite these results, and to be compatible with a predecessor design, the decision was to adopt a block size of 16 bytes. One approach to reducing misses in a situation like this would be to implement a prefetching technique. Luckily (or unluckily), you were assigned the design of the cache controller. Sketch the prefetching protocol you would implement. In particular, specify the actions to be taken when a read hit, read miss, write hit, and write miss occur. [15]

d. What is the *worst-case* increase in latency caused by switch contention when routing messages from n distinct sources to n distinct destinations (*i.e.* when performing a permutation) through an $n \times n$ shuffle-exchange network? [10]

Assume that the cost of a 2×2 switch in a shuffle-exchange network is twice the cost of a crosspoint in a crossbar switch and that the delay through the 2×2 switch is the same as the delay through a crossbar switch. Considering a performance/cost index measured in terms of the *worst-case bandwidth per unit cost for a random permutation*, for what value of n would you choose an $n \times n$ crossbar switch over an $n \times n$ shuffle network built using the 2×2 switch? [15]

e. A pipelined processor uses the delayed branch technique. You are asked to recommend one of two designs. In the first, the processor has a 4-stage pipeline and one delay slot, and in the second it has a 6-stage pipeline with two delay slots. Compare the performance of these two alternatives, taking only the branch penalty into account. Assume that 20% of the instructions are branch instructions and that an optimizing compiler has an 80% success rate in filling a single delay slot and a 25% success rate in filling a second delay slot. Assume that the machine with the deeper pipeline has a clock that is 10% faster than the other one. [20]

f. While the Tomasulo algorithm for dynamic scheduling of pipelines allows multiple operations that have been waiting for a single result to start simultaneously, it does not allow multiple operations that are finished execution to complete (*i.e.* write back their results) simultaneously. Write a short code segment that illustrates these 2 situations (*i.e.* give (1) a code segment that shows how multiple operations could start executing simultaneously and (2) a code segment that shows how multiple finished operations must serialize). [15]

Is it possible to overcome the Tomasulo's limitation that multiple finished operations must serialize? If yes, then how, and if no, then why. [15]

g. Multithreading is often used to tolerate long latencies in multiprocessor design. This has been implemented in the MIT Alewife multiprocessor and the Tera computer. The basic idea is to allow each processor in the system to run a large number of threads and to provide very fast context switching so that when a thread encounters a high-latency operation (*e.g.* a remote memory operation), the processor simply switches context to a different thread. Let L denote the communication latency in cycles for a remote memory access, N denote the number of threads that can be interleaved in each processor, C denote the context switching overhead in cycles, and R denote the average number of cycles between remote accesses. Derive a lower bound on N so that latency due to remote memory access is totally hidden. What is the effect of this latency-hiding technique on (1) the execution time of a single thread and (2) on the network traffic. [20]