INSTRUCTIONS

For the Breadth Exam, answer questions 1 through 4 and for the Depth Exam, answer questions 1 through 7. The questions are quite specific. If, however, some confusion should arise, be sure to state all your assumptions explicitly.

BREADTH EXAM

1. Consider a processor with a cache connected to a main memory with a bus (which is 32 bits wide). A read access by the processor that hits in the cache takes 1 cycle. On a miss, the entire block must be fetched from main memory over the bus. A bus transaction consists of one address cycle to send an address (32 bits) to the memory, four cycles of idle-time for main memory access, and one cycle to transfer each word (32 bytes) in the block to the cache. (Assume that the processor continues execution only after the last word of the block has arrived.) The following table gives the average cache miss rates of a 1Mbyte cache for various block sizes.

Block size (B), words	Miss ratio (m), %
1	4.5
4	2.4
8	1.6
16	1.0
32	0.75

i. What block size yields the best average memory access time?

ii. If bus contention adds 2 cycles on average to the main-memory access time, which block size yields the best average memory access time?

iii. If the bus width is doubled to 64 bits, what is the optimal block size?

2. A "cost-effective" computer design strikes a near-optimal balance between cost and performance. Clearly there is no single cost-effective design, but rather a range of solutions, running the spectrum from high-cost/high-performance to low-cost/low-performance. Give two examples of a computer component or subsystem that can be designed three different ways to achieve different levels of cost and performance. Describe the differences in approach that determine the balance of cost and performance.

3. Most computers do not include the full address of a memory reference explicitly in an instruction. Why not? Most computers allow the address to be specified in several ways. Describe the most common methods and when they might be used in a program. (The name of a method is *not* sufficient to describe it.)

4. Floating-point data is a way that computers represent an approximation to real numbers. The representation is necessarily approximate, since many real numbers cannot be represented in a finite number of bits. Rounding is the process that computers use to convert a higher-precision representation to the precision supported by the hardware. What are the different rounding techniques? What are the criteria we use to evaluate the "goodness" of a rounding scheme? How do the different rounding techniques compare under these "goodness" criteria?

DEPTH EXAM

5. Some computers, such as the CRAY-1, support vector instructions. Since the effect of executing a vector instruction can be realized with a series of regular (called *scalar* in this context) instructions, the inclusion of vector instructions does not expand the problems a computer can solve. Why then do some architectures support vector instructions? Alternatively, why do some architectures choose not to have vector instructions?

6. Consider the following hypothetical situation:

A major breakthrough has just occurred in high-temperature superconductors producing an exciting new technology called yttrium-enhanced-semiconductor (YES). YES has basic gate-delays 100 times faster than gallium arsenide (GaAs), promising processor designs that are 100-1000 times faster than current processors. However, while the density of YES logic gates is comparable to ECL, YES memory cells are *extremely* expensive, requiring over 100 times the area of a simple NAND-gate. Thus memory, including caches, registers, flip-flops, and writable control-store are much more expensive than current technologies.

You are to design the first computer to take advantage of this radical new technology. Your mandate is to design the fastest computer possible, with no regard for compatibility with previous designs.

i. How would the technology constraints affect your instruction set design? How would it compare to previous instruction sets?

ii. What type of memory system would you design for this machine? How would this be different from machines built in other technologies?

iii. Your manager has promised that your design will be 100 times faster than any machine currently on the market. Will your design fulfill this promise? Is YES capable of fulfilling this promise? Why or why not?

7. Two of the best metrics for characterizing a computer pipeline are its latency and its bandwidth. The computer architect has available techniques for increasing memory bandwidth, but many of these techniques may result in an increase in the latency as well. Alternatively, there are techniques available that can reduce the latency, but which limit the peak bandwidth available from the memory system.

i. Without going into specific techniques for increasing bandwidth and latency, explain how the architect knows when to apply these techniques. Is there a point beyond which applying these techniques is no longer justified? If so, what determines that point?

ii. Let latency be defined as the time (in seconds) from when a memory operand is requested until the operand is received. Let bandwidth be the number of bytes per second supplied by the memory system. The product of the bandwidth and the latency is a number with units of bytes. What is the significance of latency/bandwidth product? In particular, what does a large number signify?

iii. Give at least one physical interpretation for the latency/bandwidth product of a pipelined memory system?

iv. What are the implications to a programmer or compiler for a memory system that has a large latency/bandwidth product?