Very short answer questions. You must use 10 or fewer words. "True", "False", "Yes", "No", etc. are all considered very short answers.

(1) [1] What principle makes Virtual Memory possible?

(2) [1] Does an average program's locality behavior remain the same during the entire run of the program?

(3) [1] Which type of cache miss can be changed by altering the mapping scheme?

(4) [1] Which type of cache miss can be reduced by using longer lines?

(5) [1] Which type of cache miss can be increased by using longer lines?

(6) [1] What does UMA stand for?

(7) [3] What does each of the following acronyms stand for (write your answer below each):

UMA DSM VMM

(8) [2] What are the two instructions used in RISC machines to support atomicity?

(9) [2] There are two major challenges to obtaining a substantial decrease in response time when using the MIMD approach. What are they?

(10) [2] If I add processors but keep the job size the same, am I measuring strong or weak scaling? Does this correspond most closely to response time or throughput?

(11) [2] Give a one-word definition of coherence, and a one-word definition of consistency.

Short Answers (20 or fewer words)

(12) [2] What is the main difference between a commodity cluster and a custom cluster?

(13) [2] How do two processes communicate when running on a shared memory machine?

(14) [4] There are some hardware structures which do use true LRU, even over a large number of addresses - what is an example, and why would designers do that?

(15) [4] What does TLB stand for? What does it do?

(16) [3] It is harder to write a Virtual Machine Monitor for some instruction sets than it is for others. Why?

(17) [3] Writing to a cache is inherently slower than reading from a cache. Why?

(18) [3] Why is it difficult/impossible to create a benchmark that will work across all classes of parallel processors?

(19) [3] Which is harder to write a program for, a shared memory machine or a message passing machine? Why?

(20) [3] Which is more expensive to build - a shared memory machine, or a message passing machine? Why?

(21) [6] Snooping is one approach to providing coherence - state what the other main approach is, and briefly outline how each of them work.

(22) [6] The designer has the choice of using a physically addressed cache or a virtually addressed cache. Explain the difference (drawing a picture is fine!), and give 1 advantage for each.

(23) [6] Find the Average Memory Access Time (AMAT) for a processor with a 1 ns clock cycle time, a miss penalty of 20 clock cycles, a miss rate of 0.10 misses per instruction, and a cache access time (including hit detection) of 1 clock cycle. Assume that the read and write miss penalties are the same and ignore other write stalls.

(24) [6] You are trying to decide if you should use a write through or a write back cache. Suppose going to memory requires 100 extra cycles, and 10% of the instructions are stores - if the CPI without cache misses was 2, what would the CPI become if you use the write through approach?

(25) [13] For each of the following techniques, circle the arrow(s) associated with each of the terms in the Average Memory Access Time which indicates how (on average) that term is affected. If there is more than one answer, then circle more than one term. For example, if the HT goes up, you would circle the up arrow - if the HT goes down, circle the down arrow. If the HT is unaffected, do not circle anything. Assume there is a single L1 cache.

Increasing cache size (HT: ↑ ↓ MR: ↑ ↓ MP: ↑ ↓)

Decreasing Associativity (HT: ↑ ↓ MR: ↑ ↓ MP: ↑ ↓)

Decreasing Line/Block size (HT: ↑ ↓ MR: ↑ ↓ MP: ↑ ↓)

Hardware Prefetching (HT: ↑ ↓ MR: ↑ ↓ MP: ↑ ↓)
(Assume prefetched data arrives before needed)

Compiler optimizations (HT: ↑ ↓ MR: ↑ ↓ MP: ↑ ↓)
(Excluding prefetching)

Nonblocking cache (HT: ↑ ↓ MR: ↑ ↓ MP: ↑ ↓)

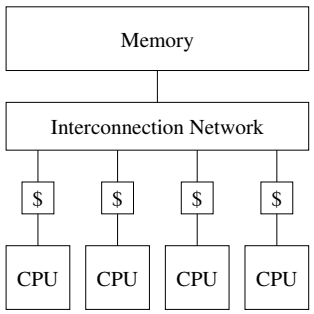
Adding an L2 cache (HT: ↑ ↓ MR: ↑ ↓ MP: ↑ ↓)
(Affect on L1 parameters)

Physically Addressed Cache (HT: ↑ ↓ MR: ↑ ↓ MP: ↑ ↓)

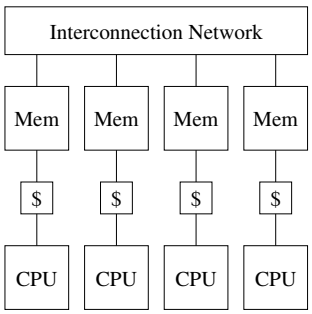
Adding a Victim Cache (HT: ↑ ↓ MR: ↑ ↓ MP: ↑ ↓)
(Assume it is not accessed in parallel)

---

(26) [2] Look at the figure below, then write next to each machine if it is more likely to be a message passing or a shared memory design.



Design Approach A



Design Approach B

(27) [3] Assuming a 25-bit address and a 512-byte Direct Mapped cache with a linesize=2, show how an address is partitioned/interpreted by the cache.

(28) [3] Assuming a 25-bit address and a 96-byte 3-way Set Associative cache with a linesize=4, show how an address is partitioned/interpreted by the cache.

(29) [2] Assuming a 20-bit address and a 792-byte Fully Associative cache with a linesize=8, show how an address is partitioned/interpreted by the cache.

(30) [5] Given a 2 Megabyte physical memory, a 23 bit Virtual address, and a page size of 2K bytes, write down the number of entries in the Page Table, and the width of each entry. Is there a problem with this configuration? If so, how can you fix it?

(31) [5] Given a 1 Gigabyte physical memory, a 46 bit Virtual address, and a page size of 8K bytes, write down the number of entries in the Page Table, and the width of each entry. Is there a problem with this configuration? If so, how can you fix it?