Problem 1. The following is a sequence of address references given as word addresses:

2,3,11,16,21,13,64,48,19,11,3,22,4,27,6,11

A. Show the hits and misses and final cache contents for a fully associative cache with one-word blocks and a total size of 16 words. Assume LRU replacement.

B. Show the hits and misses and final cache contents for a fully associative cache with *four*-word blocks and a total size of 16 words. Assume LRU replacement.

Problem 2. Cache multiple choice:

A. If a cache access requires one clock cycle and handling cache misses stalls the processor for an additional five cycles, which of the following cache hit rates comes closest to achieving an average memory access of 2 cycles?

(A) 75%
(B) 80%
(C) 83%
(D) 86%
(E) 98%

B. LRU is an effective cache replacement strategy primarily because programs

(A) exhibit locality of reference
(B) usually have small working sets
(C) read data much more frequently than write data

C. If increasing the block size of a cache improves performance it is primarily because programs

(A) exhibit locality of reference
(B) usually have small working sets
(C) read data much more frequently than write data

Problem 3. A student has miswired the address lines going to the memory of an unpipelined BETA. The wires in question carry a 30-bit word address to the memory subsystem, and the hapless student has in fact reversed the order of all 30 address bits. Much to his surprise, the machine continues to work perfectly.

A. Explain why the miswiring doesn't affect the operation of the machine.

B. The student now replaces the memory in his miswired BETA with a supposedly higher performance unit that contains both a fast fully associative cache and the same memory as before. The reversed wiring still exists between the BETA and this new unit. To his surprise, the new unit does not significantly improve the performance of his machine. In desperation, the student then fixes the reversal of his address lines and the machine's performance improves tremendously. Explain why this happens.

Problem 4. The following questions ask you to evaluate alternative cache designs using patterns of memory references taken from running programs. **Each of the caches under consideration has a total capacity of 8 (4-byte) words**, with one word stored in each cache line. The cache designs under consideration are:

DM: a direct-mapped cache.

S2: a 2-way set-associative cache with a least-recently-used replacement policy.

FA: a fully-associative cache with a least-recently-used replacement policy.

The questions below present a sequence of addresses for memory reads. **You should assume the sequences repeat from the start whenever you see "...".** Keep in mind that byte addressing is used; addresses of consecutive words in memory differ by 4. Each question asks which cache(s) give the best hit rate for the sequence. Answer by considering the steady-state hit rate, i.e., the percentage of memory references that hit in the cache after the sequence has been repeated many times.

A. Which cache(s) have the best hit rate for the sequence 0, 16, 4, 36, ...

B. Which cache(s) have the best hit rate for the sequence 0, 4, 8, 12, 16, 20, 24, 28, 32, ...

C. Which cache(s) have the best hit rate for the sequence 0, 4, 8, 12, 16, 20, 24, 28, 32, 28, 24, 20, 16, 12, 8, 4, ...

D. Which cache(s) have the best hit rate for the sequence 0, 4, 8, 12, 32, 36, 40, 44, 16, ..

E. Assume that a cache access takes 1 cycle and a memory access takes 4 cycles. If a memory access is initiated only after the cache has missed, what is the maximum miss rate we can tolerate before use of the cache actually slows down accesses?