EE 4683/5683: Computer Architecture

Homework #4

Instructions:

Graduate Students: Answer all questions.

Undergraduate Students: The last question is bonus question.

Problem 1

A program repeatedly performs a three-step process: it reads in a 4-KB block of data from disk, does some processing on that data, and then writes out the result as another 4-KB elsewhere on the disk. Each block is contiguous and randomly located on a single track on the disk. The disk drive rotates at 7200 RPM, has an average seek time of 8 ms, and has a transfer rate of 20 MB/sec. The controller overhead is 2 ms. No other program is using the disk or processor, and there is no overlapping of disk operation with processing. The processing step takes 20 million clock cycles, and the clock rate is 400 MHz.

What is the overall speed of the system in blocks processed per second?

Problem 2

Solve problem 5.1

Problem 3

Solve problem 5.2

Problem 4

Solve problem 5.20 (a, b and c)

Problem 5 (*For Graduate students, Undergrads get BONUS if solution is correct!)*

(a) A uniprocessor system uses a separate instruction and data caches with the hit ratios *hi* and *hd*, respectively. The access time from the processor to either cache is *c* clock cycles, and the block is transfer time between the caches and main memory is *b* block cycles.
Among all memory references made by the CPU, *fi* is the percentage of the references to instructions. Among blocks replaced in the data cache, *fdir* is the percentage of dirty blocks. Assuming a write-back policy, determine the effective memory access time in terms of *hi, hd, c, b, fi, fdir* for this memory system.

(b) The processor memory system described in part (a) is used to construct a bus-based shared memory multiprocessor. Assume that the hit ratio and access time remain the same as in part (a). However, the effective memory-access time will be different because every processor must now handle cache invalidations in addition to reads and writes.
Let finv be the fraction of data references that cause invalidation signals to be sent to other caches. The processor sending the invalidation signal requires i clock cycles to complete the invalidation process. Other processors are not involved in the invalidation

process. Assuming a write-back policy again, determine the effective memory-access time for this multiprocessor.