Problem 1. (25 points) In Lecture 11, we introduced CPI, the average Cycles Per Instruction, as a clear measurement of cache performance impact on the system. In the following problem, we assume that it takes:

- 1 cycle to fetch data from the 1st level cache,
- 3 cycles to fetch data from the 2nd level cache, and
- 15 cycles to fetch data from main memory.

(a) Suppose we have only the 1st level cache. This cache has a hit ratio of 90% for both data and instruction inquiries. To complete one execution, we need to first fetch the instruction, and then the data needed, with no overlap at all. Calculate the CPI.

(b) Then we add the 2nd level cache. In order to keep the new CPI value within 4 cycles per instruction, what should be the hit ratio of the 2nd level cache?

(c) Now let's consider using the "split caches" that we discussed in Lecture 11. Suppose we do not have the 2nd level cache. Instead, we split the 1st level cache evenly into an Instruction cache and a Data cache. According to "the 30% rule" at the end of Lecture 10, what will be the hit ratio of each separate cache now?

(d) What will be the new CPI for the system described in (c)? Assume that the two caches use different buses (thus they can be accessed simultaneously), but there is only one main memory bus.

(e) Compare the result of (d) with those of (a) and (b). Are they different from what you expected? Try to explain. Then list the main advantage, and at least two disadvantages of using split caches.

Problem 2. (20 points) As we have seen in class, in order to perform TLB lookup at the same time a set-associative cache is beginning to be searched, some restrictions on the length of the displacement field are necessary. Assume we desire to do TLB lookup concurrently with cache search, and answer the following questions.

(a) If both the (main-memory) page size and the (cache) line size are held fixed, how does an increase in cache size (the number of lines in the cache) affect the number of lines required in each set (the "set size")? Justify your answer.

(b) If the cache contains 16K words, pages are 1K words long, and lines contain 16 words, what range of set sizes will allow simultaneous TLB and cache access?

(c) Does the required set size depend on the line size? (For example, if you redid part (b) assuming a line size of 32 words, or 64 words, etc., would your answer change?)
(d) Suppose now that the set size and line size are held fixed. How does an increase in cache size affect the required page size? Justify your answer.

**Problem 3.** *(25 points)* Set-associative and sectored caches are two compromises between direct mapping and full association. A *cyclic reference string* of order *n* is a sequence of references to blocks

\[0, 1, 2, \ldots, n-2, n-1, 0, 1, 2, \ldots, n-2, n-1, 0, 1, \ldots\]

Assume that LRU replacement is used in both kinds of caches, and that

- there are *f* lines in each cache,
- there are *b* blocks per set in the set-associative cache,
- there are *s* sectors in the sectored cache.
- there are *n* distinct blocks in the cyclic reference string.

(a) Suppose \(b = s\). Which kind of cache has a higher hit ratio? Does the answer depend on *f* or *n*? How?

(b) Suppose \(b \neq s\). Is the answer still the same as in part (a)? Why?

**Problem 4.** *(30 points)* Given the MSI state transition diagram (see CS&G page 295) and the MESI state transition diagram (see CS&G page 301):

(a) Enumerate the triggers for state changes

(b) Enumerate all of the MSI rules (including those not shown in the diagram). Give them in this format: \([I]\) : \([P]Rd/[B]usRd\) \(\rightarrow [S]\)

(c) Enumerate all of the MESI rules (including those not shown in the diagram)

(d) In MESI, what state transition(s) never occur?

(e) What is the difference between Flush and Flush’?

(f) Indicate the mapping of MSI rules to the corresponding MESI rules

(g) Given this mapping, indicate the specific transitions where triggers affect MESI differently than MSI

(h) Indicate the reasoning why these transition behaviors are different.