# Caches Workout

**Discussion 07 - 25 November 2019**
**CS 250P - Computer Systems Architecture**
**TA - Aftab Hussain**

---

**Problem 1.** The L1 cache is of relatively small size (e.g. ~32kB). Can you explain how the L1 cache could still have a high hit rate (or low MPKI - misses per kilo instructions) ?

**Problem 2.** Why does it take more time to access the L2 cache than it takes to access the L1 cache?

**Problem 3.** Say a processor executes a 1000 instructions, of which 200 instructions are loads and stores. Say 5% of these 200 instructions miss the L1 cache. What is the MPKI of the L1 cache?

**Problem 4.** Explain inclusive, exclusive, and non-inclusive cache hierarchies.

**Problem 5.** Let's say in any given 64-byte cache, we want to store data as 8-byte words.
How many such words can we store in this cache?
How many bits do we need to uniquely identify (and thereby access) each of these words?
How many bits do we need to access a byte within each word?

**Problem 6.** Let's say in any given 64-byte cache, we want to store data as 16-byte words.
How many such words can we store in this cache?
How many bits do we need to uniquely identify (and thereby access) each of these words?
How many bits do we need to access a byte within each word?

**Problem 7.** *Cache layout terminology.* 40-bit addresses are used to access data in the cache. Say our 64-byte cache can store 8-byte words. (The cache is direct mapped, single set associative).

(a) What is each entry of the cache, which holds an 8-byte word called? Write all the terms that may be used interchangeably.

The last 6 bits of the 40 bit address are used to access data in the cache. Say these last 6 bits are, 101011

(b) What do the first 3 bits of these 6 bits identify? What are these bits called?

(c) What do the last 3 bits identify? What are they called?

**Problem 8.** Following on from the previous problem, there may be multiple 40-bit addresses that have the same last 6 bits. I.e., the first 34-bits of those addresses may be different. How do we distinguish between those addresses? What are those bits called?

**Problem 9.** What is a direct-mapped cache?

**Problem 10.** Explain a cache design structure that allows us to refer to multiple entries in the cache, using a single 40-bit address. Mention what cache performance metric is affected by such a design and how.

**Problem 11.** A *set* is the same as a *cache line* in the design in Problem 7. In Problem 10, we use a different cache design. How does the definition of a *set* change in these two designs?

**Problem 12.** Let us say you have a 32kB, 4-way set-associative data cache array. The size of each cache line is 32 bytes.

(a) How many sets do we have?

Problem 10 tells us about three segments of the address bits. Assume we use 40-bit addresses.

(b) How many bits are used for each of these segments for the cache design in this problem?

The first segment of those bits are used to access what is known as a tag array.

(c) How big is the tag array?

**Problem 13.** Give an example where a write-allocate policy for caches would be useful.

**Problem 14.** Explain the performance differences between a serial tag/data access strategy and a parallel tag/data access strategy.

**Problem 15.** What is a victim cache?