Cache Memory: Replacement

This is a brief note on the need for a replacement policy when using cache memory; it would normally have appeared as a slide in the lecture.

• We know main memory is organized as a linear array of individually addressable bytes; each byte has an address that reflects how far it is from the start of memory.

• A computer with $n$-bit addresses (e.g., 32) has $2^n$ addressable locations.

• Suppose we organize these $2^n$ locations into blocks of size $b$ bytes (e.g., 64).

• Then memory has $m = \frac{2^n}{b}$ such blocks.

• Memory is cached one block at a time; the cache contains $c$ total slots, each able to hold $b$ bytes, and $c << m$.

What do you do when all $c$ slots are full and you experience a cache-miss? i.e., you need to load a new block into the cache somewhere, so have to have a way to decide which of the $c$ blocks you will replace. The way you decide that is by using a replacement policy.

Summary of Cache Issues:

1. cache-hit ratio (hence also cache-hit, cache-miss)
2. locality
3. coherency
4. replacement policy