Memory (III)

Mapping Functions (II)

Direct Mapping (cont.)

Example
- Assume the cache size is 64 KB. Data are transferred between main memory and the cache in blocks of 4 bytes each. This means that the cache is organized as $16K = 2^{14}$ lines of 4 bytes each. A word is 8-bit long. [Cache size = 64 KB, Block size = 4 B, number of lines $2^{14}$, Word length = 1B]
- The main memory consists of 16 MB, with each byte directly addressable by a 24-bit address ($2^{24} = 16$ M). Thus, for mapping purposes, we can consider main memory to consist of 4 M blocks of 4 bytes each. [Main memory size = 16 MB, address = 24 bits, numbers of blocks = 4M]
- How many bits for w? [2, as the block size is 4 bytes and each word is a byte, so need 2 bit to specify the 4 words.]
- How many bits for r? [14, as the cache has $2^{14}$ number of lines]
- How many bits for tag? [8, as the address is 24-bit long, $24 - 2 - 14 = 8$]
- So, if using direct mapping, the above example would have the address in the following format.

| Tag: 8 bits | Line: 14 bits | Word: 2 bits |

- Summary
  - Address length = $(s + w)$ bits
  - Number of addressable units = $2^{(s+w)}$ words or bytes
  - Block size = line width = $2^w$ words or bytes
  - Number of blocks in main memory = $\frac{2^{(s+w)}}{2^w} = 2^s$
  - Number of lines in cache = $2^r$
  - Size of tag = $(s - r)$ bits

- Pros & Cons
  - Simple
  - Inexpensive
  - Fixed location for given block - if a program accesses 2 blocks that map to the same line repeatedly, cache misses are very high.