

Memory (IV)

Mapping Functions (cont.)

Example:

- If we use the same example we used in Direct Mapping, what does the address structure look like if using associative mapping?
 - Cache size: 64 KB; block size: 4 bytes; addressable unit: byte
 - Main memory size: 16 MB; address length: 24 bits
 - Num. of lines: $\frac{64KB}{4B} = 16K = 2^{14}$
 - Num. of blocks: $\frac{16MB}{4B} = 4M$
 - 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 the tag? [22, as $24 - 2 = 22$]

Tag: 22 bits	Word: 2 bits
--------------	--------------

- Information we could get from a given address that uses Associative Mapping
 - Address length = (s + w) bits
 - Number of addressable units = $2^{(s+w)}$ words or bytes
 - Block size = line size = 2^w words or bytes
 - Number of block in main memory = $\frac{2^{(s+w)}}{2^w} = 2^s$
 - Number of lines in cache = undetermined
 - Size of tag = s bits
- If we use the same example we used in Direct Mapping, what does the address structure look like if using 2-way set-associative mapping?
 - Cache size: 64 KB; block size: 4 bytes; addressable unit: byte
 - Main memory size: 16 MB; address length: 24 bits
 - Num. of lines: $\frac{64KB}{4B} = 16K = 2^{14}$

Num. of blocks: $\frac{16MB}{4B} = 4M$

Tag: 9 bits	Set: 13 bits	Word: 2 bits
-------------	--------------	--------------

- 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? [13, as it uses 2-way set-associative mapping, the number of lines in a set is 2. So, k = 2. The number of lines in the cache $m = \frac{64KB}{4B} = 16K = 2^{14}$. The number of sets $v = \frac{m}{k} = \frac{2^{14}}{2} = 2^{13}$. So set field needs 13 bits.]
 - How many bits for tag? [9, as the address is 24-bit long, $24 - 2 - 13 = 9$]
- Information we could get from a given address that uses K-way Set Associative Mapping
- Address length = (s + w) bits
 - Number of addressable units = $2^{(s+w)}$ words or bytes
 - Block size = line size = 2^w words or bytes
 - Number of blocks in main memory = $\frac{2^{(s+w)}}{2^w} = 2^s$
 - Number of lines in a set = k
 - Number of sets = $v = 2^r$
 - Number of lines in cache = $k v = k \times 2^r$

Many processor caches in today's designs are either direct mapping or set-associative mapping.

Exercise

- A machine uses 32-bit addresses. It has 1GB (2^{30}) main memory, 64KB (2^{16}) cache with 4K (2^{12}) number of cache lines, and each cache line is 16 bytes (2^4 B). If a word (addressable unit) is 1 byte. How many blocks map to each cache line or set if using direct mapping, 2-way set associative, and 4-way set associative? Hint: figure out the number lines, sets, and blocks first.
- Direct mapping:
 - there are 2^{12} lines in the cache, so all blocks are mapped to one of the 2^{12} lines.
 - each cache line is 16 bytes, so the block size is 2^4 bytes.

- so, the machine has $\frac{2^{30}}{2^4} = 2^{26}$ blocks.
- so, $\frac{2^{26}}{2^{12}} = 2^{14}$ blocks map to one cache line.
- Note: if the address length is fully used, then the machine will have $\frac{2^{32}}{2^4} = 2^{28}$ blocks, and there are $\frac{2^{28}}{2^{12}} = 2^{16}$ blocks map to one cache line.
- 2-way set-associative:
 - 2-way set-associative, so each set in the cache has two lines.
 - the number of sets in the cache is $\frac{2^{12}}{2} = 2^{11}$
 - so, $\frac{2^{26}}{2^{11}} = 2^{15}$ blocks map to one set.
- 4-way set-associative:
 - 4-way set-associative, so each set in the cache has four lines.
 - the number of sets in the cache is $\frac{2^{12}}{2^2} = 2^{10}$
 - so, $\frac{2^{26}}{2^{10}} = 2^{16}$ blocks map to one set.

Replacement Algorithm

- There must be a method for selecting which line in the cache is going to be replaced when there is no room for a new line
- The algorithm is hardware implemented for better speed.
- Direct mapping
 - There is no need for a replacement algorithm
 - Each block only maps to one line, so only need to replace that line.
- Associative & Set Associative Replacement Algorithms
 - Associative & Set-associative mappings need algorithms, as there is no dedicated line for each block.
 - Least Recently Used (LRU) is a widely used replacement algorithm. It replace the block that hasn't been touched in the longest period of time.