

Caches

Note Title

tttt .. llll ... oooo

10/24/2008

Given a Direct Mapped 1M Byte Cache (32-bit addresses)
w/ Blocksize = 64 bytes

How many lines? 2^{14} lines 14 bits

How many bits in the tag field 12 bits

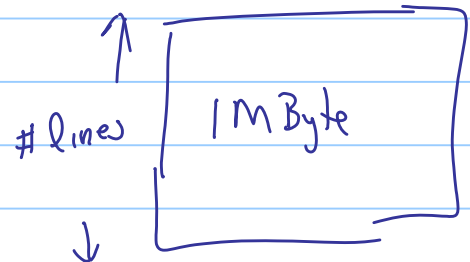
How many bits in offset field? 6 bits

← blocksize →

$$\text{Area} = w \times H$$

$$2^{20} = 2^6 \times H$$

$$H = 2^{14} = 16 \text{ K lines}$$



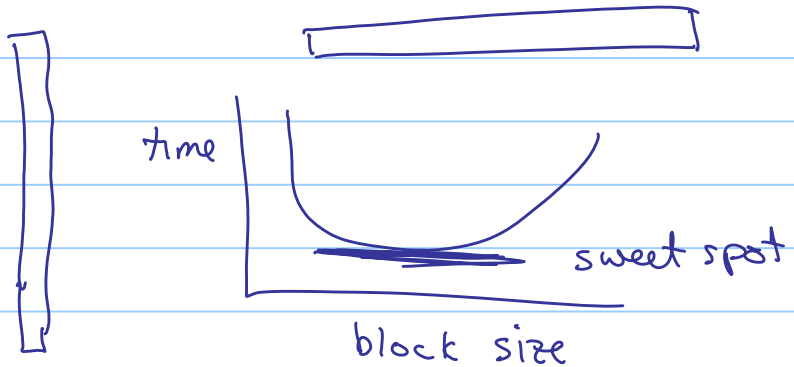
Writing - stores w/ caches

- Design
tradeoff {
- Write through : update the cache block and the memory
 - Write back : only update the cache block
 - "dirty bit" in each cache linewhen replacing the cache entry write back to memory if dirty bit is on.

Block size tradeoff -

Given a fixed sized cache (area)

- look at the extremes:



How do we get to a choice?

What do we measure: Average Access Time

— based on memory access time

cache access time

$$\text{miss rate} = 1 - \text{hit rate}$$

Average access time =

hit rate \times cache access time +

miss rate \times memory access time

$$= \text{cache access time} + \text{miss rate} \times \text{cache miss penalty}$$

Example:

Cache access time: 1 cycle

Cache miss penalty: 20 cycles

Miss rate: 5%

What is the Average access time: $1 \text{ cycle} + .05 * 20 \text{ cycles} = 2 \text{ cycles}$

$$\text{Miss rate: } \frac{\# \text{ misses}}{\# \text{ accesses}} = \frac{\# \text{ misses}}{\# \text{ hits} + \# \text{ misses}}$$

How do misses occur:

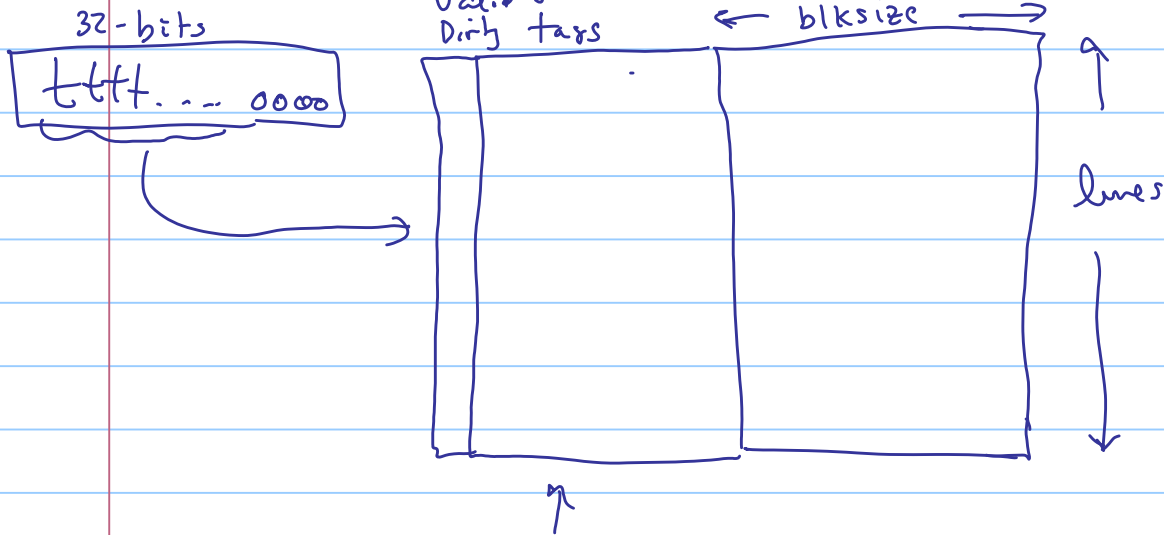
- 1st: compulsory misses - program startup; nothing in cache
- 2nd: conflict misses - a previously loaded block has been evicted by a later-loaded block.

To reduce conflict misses consider what is called a "Fully Associative Cache" -
any ^{memory} block can occupy any line of cache

Q: if no empty lines which block should be evicted?
- "least recently used" heuristic

Design of Fully Associative Cache:

address size - $\log_2 \text{blksize}$



Content-addressable memory = Compare all tags to tag of the word being loaded all at once