Direct Mapping Summary

- Address length = (s + w) bits
- Number of addressable units = 2^{s+w} words or bytes

  Block size = line size - tag size = 2^w words or bytes
- Number of blocks in main memory = $2^{s+w}/2^w = 2^s$
- Number of lines in cache = m = 2^r
- Size of tag = (s - r) bits

Direct Mapping pros & cons

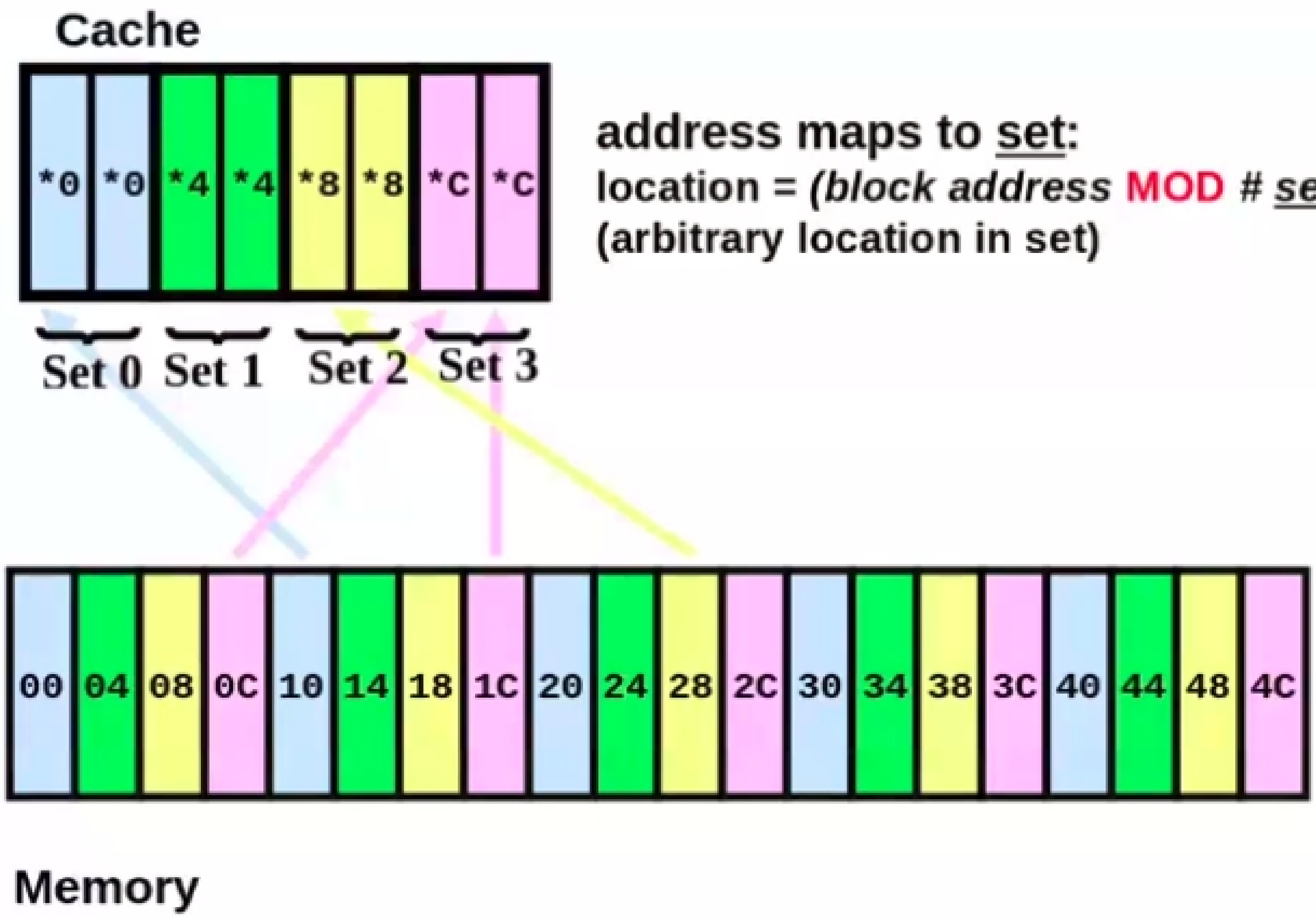
- Simple
- Inexpensive
- Fixed location for given block
  - —If a program accesses 2 blocks that map to the same line repeatedly, cache misses are very high

Set Associative Mapping

Cache is divided into a number of sets

- Each set contains k lines -> k - way associative
- A given block maps to any line in a given set
  - — e.g. Block B can be in any line of set i
- e.g. 2 lines per set
  - — 2 - way associative mapping
  - — A given block can be in one of 2 lines in only one set

Set-Associative Block Placement



address maps to <u>set</u>:
location = *(block address* <span style="color:red">MOD</span> # *<u>sets</u> in cache*)
(arbitrary location in set)

Set Associative Cache Design

Key idea:

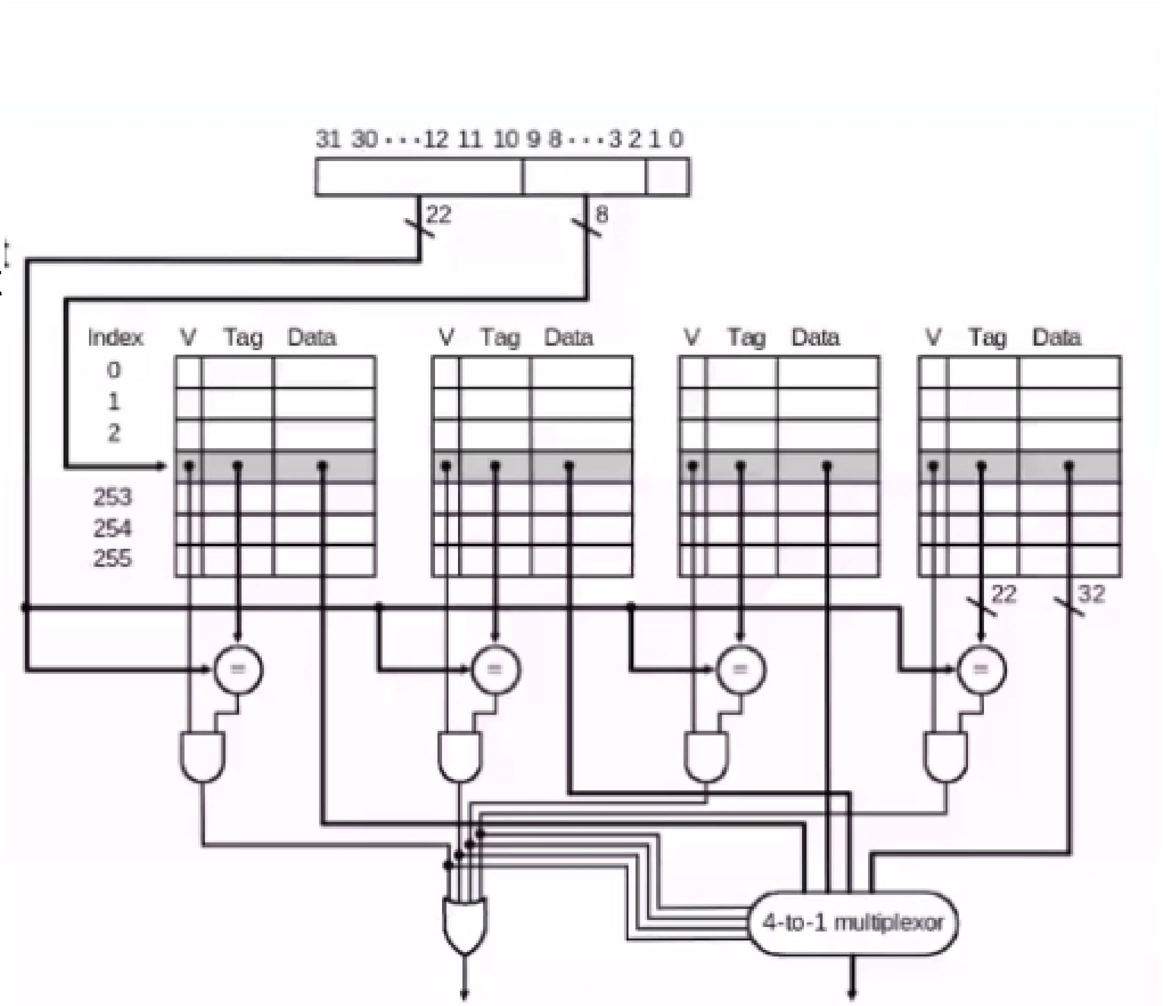
- Divide cache into sets
- Allow block anywhere in a set

Advantages

— Better hit rate

Disadvantage:

— More tag bits
— More hardware
— Higher access time



A Four-Way Set-Associative Cache

Example: Accessing A Set-Associative Cache

- **2-way Set-Associative cache contains 2 sets, 2 one-word blocks each. Find the # Misses for each cache given this sequence of memory block accesses: 0, 8, 0, 6, 8**

SA Memory Access 1: Mapping: 0 mod 2 = 0

Mem Block	**DM Hit/Miss**	**Set 0**
0		**Set 1**

SA Block Replacement Rule: replace least recently used block in set

Example: Accessing A Set-Associative Cache

- **2-way Set-Associative cache contains 2 sets, 2 one-word blocks each. Find the # Misses for each cache given this sequence of memory block accesses: 0, 8, 0, 6, 8**

SA Memory Access 4: Mapping: 6 mod 2 = 0

Mem Block	DM Hit/Miss
0	miss
8	miss
0	hit
6	miss

Set 0	Mem[0]	Mem[6]
Set 1		

Set 0, Block 1 is LRU: overwrite with Mem[6]

Example: Accessing A Set-Associative Cache

- **2-way Set-Associative cache contains 2 sets, 2 one-word blocks each. Find the # Misses for each cache given this sequence of memory block accesses: 0, 8, 0, 6, 8**

SA Memory Access 5: Mapping: 8 mod 2 = 0

Mem Block	DM Hit/Miss
0	miss
8	miss
0	hit
6	miss
8	miss

Mem[8] **Mem[6]**

Set 1

Set 0, Block 0 is LRU: overwrite with Mem[8]

Accessing a Set-Associative Cache

Example

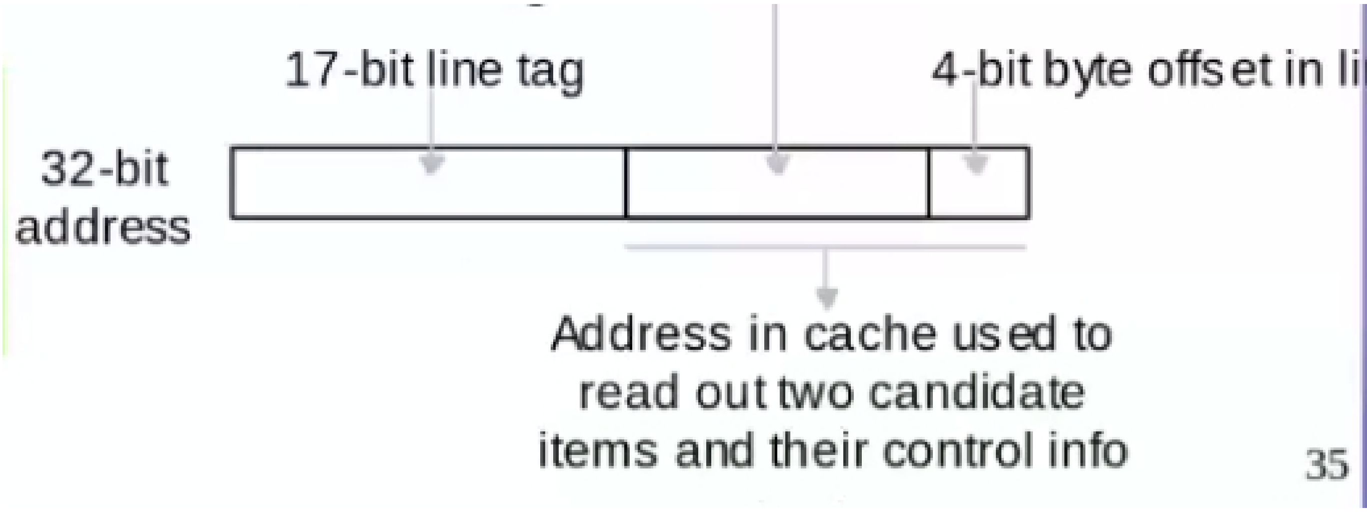
Show cache addressing scheme for a byte-addressable memory with 32-bit addresses. Cache line width 2” = 16 B. Set size 2” = 2 lines. Cache size 2^L = 4096 lines (64 KB).

Solution

Byte offset in line is $\log_2 16$ = 4 b. Cache set index is ($\log_2 4096/2$)= 11b. This leaves 32 — 11— 4 = 17 b for the tag. 11-bit set index in cache

Components of the 32- bit address

In an example two-way set-associative cache.


17-bit line tag
4-bit byte offset in line
32-bit address
Address in cache used to read out two candidate items and their control info

Associative Memory

- read: specify tag field value and word select
- checks all lines — finds matching tag
  - return contents of data field @ selected word
- access time independent of location or previous access
- write to data field @ tag value + word select
  - what if no words with matching tag?