Attention

- Our last class will be on Dec. 4.
- I will give a final exam review on Nov. 29.
- Failure to appear for the final exam will result in a grade of "F" in the course.

Memory Hierarchy: Set Associative Cache

Dr. Tao Xie

These slides are adapted from notes by Dr. David Patterson (UCB)

Fundamental Questions

- Q1: Where can a block be placed in the upper level?
  (Block placement)

- Q2: How is a block found if it is in the upper level?
  (Block identification)

- Q3: Which block should be replaced on a miss?
  (Block replacement)

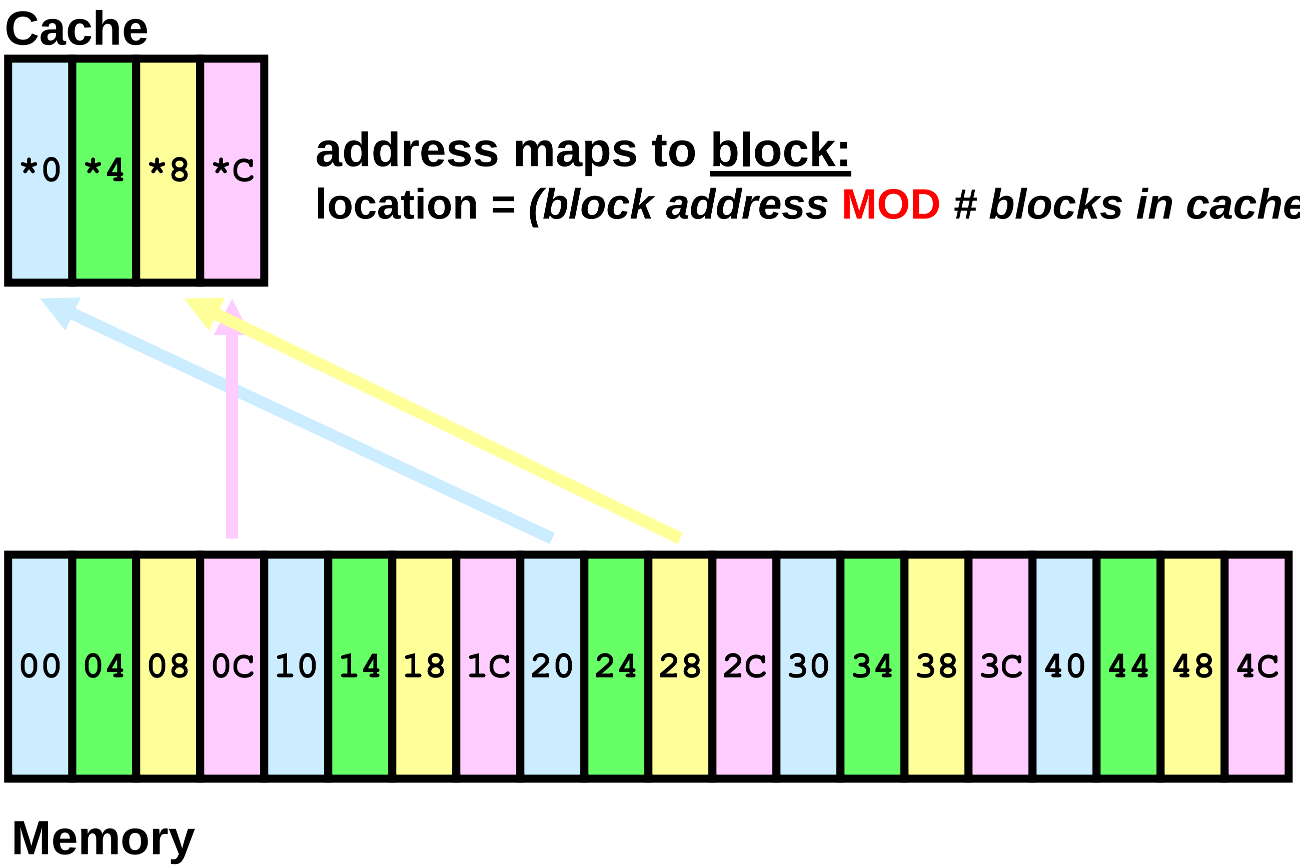
- Q4: What happens on a write?
  (Write strategy)

Q1: Block Placement

- Where can block be placed in cache?
  - In one predetermined place - direct-mapped
    - Use fragment of address to calculate block location in cache
    - Compare cache block with tag to test if block present
  - Anywhere in cache - fully associative
    - Compare tag to every block in cache
  - In a limited set of places - set-associative
    - Use address fragment to calculate set (like direct-mapped)
    - Place in any block in the set
    - Compare tag to every block in set
    - Hybrid of direct mapped and fully associative

Direct Mapped Block Placement

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



Cache

*0	*4	*8	*C

00	04	08	0C	10	14	18	1C	20	24	28	2C	30	34	38	3C	40	44	48	4C

Memory

Example: Accessing A Direct-Mapped Cache

- DM cache contains 4 1-word blocks. Find the # Misses for each cache given this sequence of memory block accesses: 0, 8, 0, 6, 8

DM Memory Access 1: Mapping: 0 modulo 4 = 0

Mem Block	DM Hit/Miss
0	

Block 0	
Block 1	
Block 2	
Block 3	

Example: Accessing A Direct-Mapped Cache

DM cache contains 4 1-word blocks. Find the # Misses for each cache given this sequence of memory block accesses: 0, 8, 0, 6, 8

DM Memory Access 1: Mapping: 0 mod 4 = 0

Mem Block	DM Hit/Miss
0	miss

Block 0	Mem[0]
Block 1	
Block 2	
Block 3	

Set 0 is empty: write Mem[0]

Example: Accessing A Direct-Mapped Cache

- DM cache contains 4 1-word blocks. Find the # Misses for each cache given this sequence of memory block accesses: 0, 8, 0, 6, 8

DM Memory Access 2: Mapping: 8 mod 4 = 0

Mem Block	DM Hit/Miss
0	miss
8	

Block 0	Mem[0]
Block 1	
Block 2	
Block 3	

Example: Accessing A Direct-Mapped Cache

- DM cache contains 4 1-word blocks. Find the # Misses for each cache given this sequence of memory block accesses: 0, 8, 0, 6, 8

DM Memory Access 2: Mapping: 8 mod 4 = 0

Mem Block	DM Hit/Miss
0	miss
8	miss

Block 0	Mem[8]
Block 1	
Block 2	
Block 3	

Set 0 contains Mem[0]. Overwrite with Mem[8]

Example: Accessing A Direct-Mapped Cache

- DM cache contains 4 1-word blocks. Find the # Misses for each cache given this sequence of memory block accesses: 0, 8, 0, 6, 8

DM Memory Access 3: Mapping: 0 mod 4 = 0

Mem Block	DM Hit/Miss
0	miss
8	miss
0	

Block 0	Mem[8]
Block 1	
Block 2	
Block 3	

Example: Accessing A Direct-Mapped Cache

- DM cache contains 4 1-word blocks. Find the # Misses for each cache given this sequence of memory block accesses: 0, 8, 0, 6, 8

DM Memory Access 3: Mapping: 0 mod 4 = 0

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

Block 0	Mem[0]
Block 1	
Block 2	
Block 3	

Set 0 contains Mem[8]. Overwrite with Mem[0]

Example: Accessing A Direct-Mapped Cache

- DM cache contains 4 1-word blocks. Find the # Misses for each cache given this sequence of memory block accesses: 0, 8, 0, 6, 8

DM Memory Access 4: Mapping: 6 mod 4 = 2

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

Block 0	Mem[0]
Block 1	
Block 2	
Block 3	

Example: Accessing A Direct-Mapped Cache

- DM cache contains 4 1-word blocks. Find the # Misses for each cache given this sequence of memory block accesses: 0, 8, 0, 6, 8

DM Memory Access 4: Mapping: 6 mod 4 = 2

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

Block 0	Mem[0]
Block 1	
Block 2	Mem[6]
Block 3	

Set 2 empty. Write Mem[6]

Example: Accessing A Direct-Mapped Cache

- DM cache contains 4 1-word blocks. Find the # Misses for each cache given this sequence of memory block accesses: 0, 8, 0, 6, 8

DM Memory Access 5: Mapping: 8 mod 4 = 0

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

Block 0	Mem[0]
Block 1	
Block 2	Mem[6]
Block 3	

Example: Accessing A Direct-Mapped Cache

- DM cache contains 4 1-word blocks. Find the # Misses for each cache given this sequence of memory block accesses: 0, 8, 0, 6, 8

DM Memory Access 5: Mapping: 8 mod 4 = 0

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

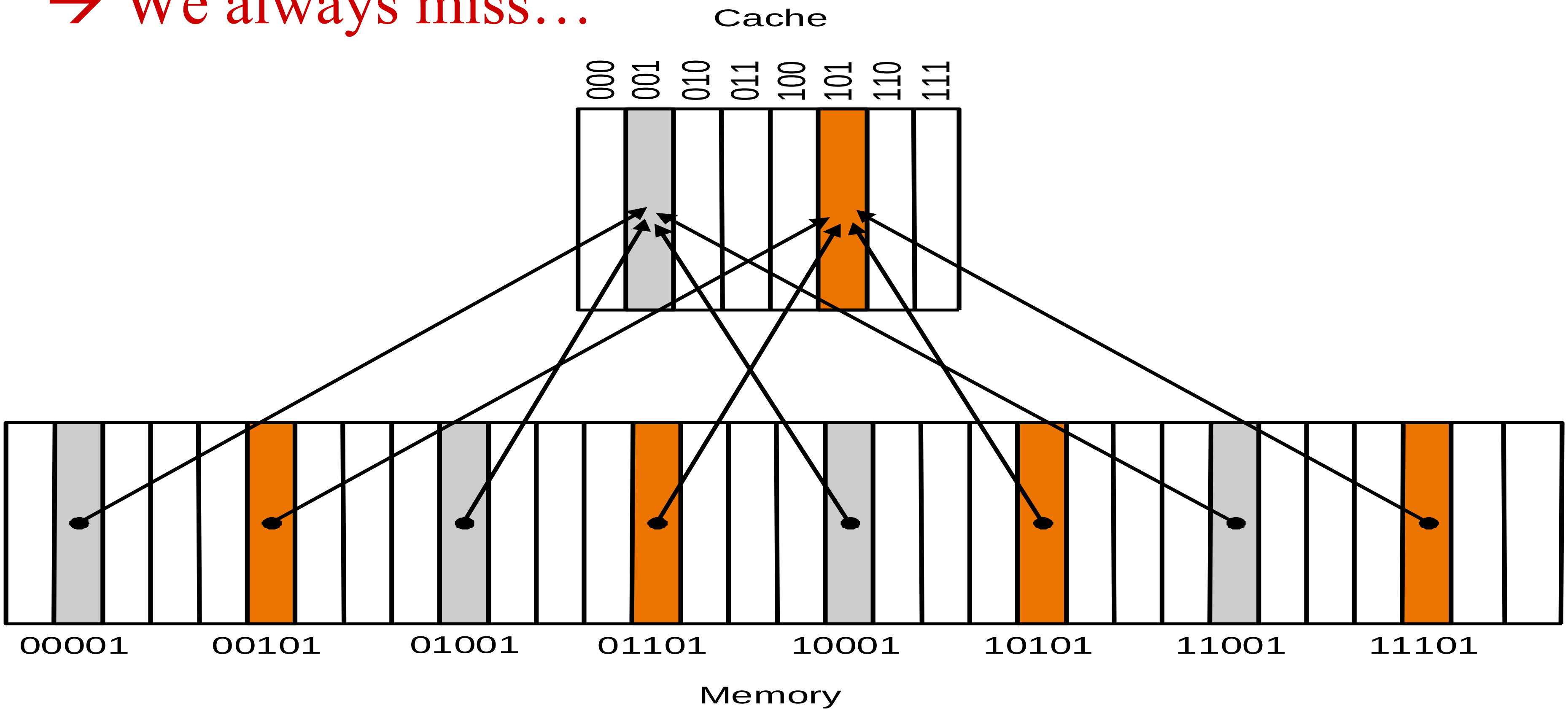
Block 0	Mem[8]
Block 1	
Block 2	Mem[6]
Block 3	

Set 0 contains Mem[0]. Overwrite with Mem[8]

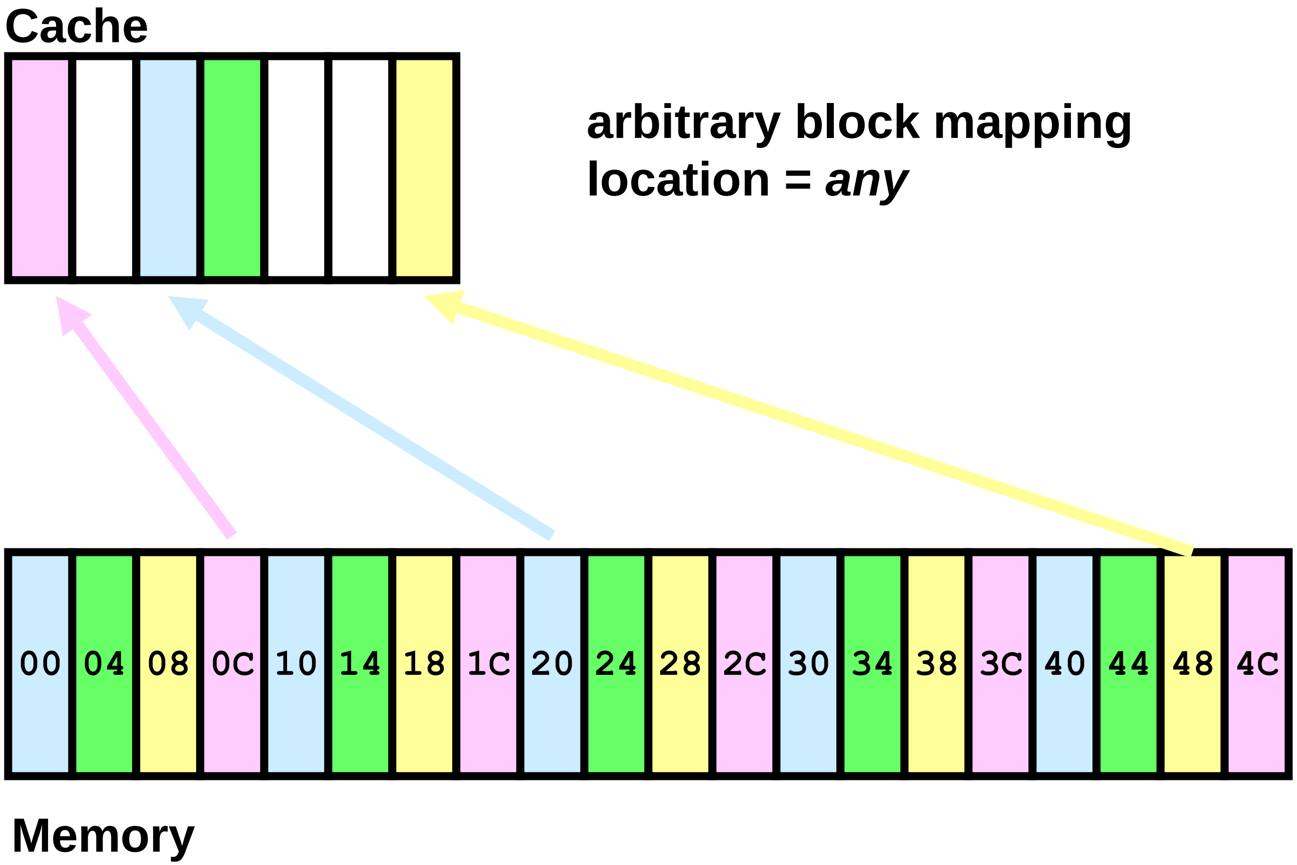
Direct-Mapped Cache with n one-word blocks

- Pros: find data fast
- Con: What if access 00001 and 10001 repeatedly?

→ We always miss…



Fully Associative Block Placement



arbitrary block mapping
location = ***any***

Example: Accessing A Fully-Associative Cache

- Fully-Associative cache contains 4 1-word blocks. Find the # Misses for each cache given this sequence of memory block accesses: 0, 8, 0, 6, 8

FA Memory Access 1:

Mem Block	DM Hit/Miss
0	

Set 0				

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

Example: Accessing A Fully-Associative Cache

- Fully-Associative cache contains 4 1-word blocks. Find the # Misses for each cache given this sequence of memory block accesses: 0, 8, 0, 6, 8

FA Memory Access 1:

Mem Block	DM Hit/Miss
0	miss

Set 0	Mem [0]			

Set 0 is empty: write Mem[0] to Block 0

Example: Accessing A Fully-Associative Cache

- Fully-Associative cache contains 4 1-word blocks. Find the # Misses for each cache given this sequence of memory block accesses: 0, 8, 0, 6, 8

FA Memory Access 2:

Mem Block	DM Hit/Miss
0	miss
8	

Set 0	Mem [0]			

Example: Accessing A Fully-Associative Cache

- Fully-Associative cache contains 4 1-word blocks. Find the # Misses for each cache given this sequence of memory block accesses: 0, 8, 0, 6, 8

FA Memory Access 2:

Mem Block	DM Hit/Miss
0	miss
8	miss

Set 0	Mem [0]	Mem [8]		

Blocks 1-3 are LRU: write Mem[8] to Block 1

Example: Accessing A Fully-Associative Cache

- Fully-Associative cache contains 4 1-word blocks. Find the # Misses for each cache given this sequence of memory block accesses: 0, 8, 0, 6, 8

FA Memory Access 3:

Mem Block	DM Hit/Miss
0	miss
8	miss
0	

Set 0	Mem [0]	Mem [8]		

Example: Accessing A Fully-Associative Cache

- Fully-Associative cache contains 4 1-word blocks. Find the # Misses for each cache given this sequence of memory block accesses: 0, 8, 0, 6, 8

FA Memory Access 3:

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

Set 0	Mem [0]	Mem [8]		

Block 0 contains Mem[0]

Example: Accessing A Fully-Associative Cache

- Fully-Associative cache contains 4 1-word blocks. Find the # Misses for each cache given this sequence of memory block accesses: 0, 8, 0, 6, 8

FA Memory Access 4:

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

Set 0	Mem [0]	Mem [8]		

Example: Accessing A Fully-Associative Cache

- Fully-Associative cache contains 4 1-word blocks. Find the # Misses for each cache given this sequence of memory block accesses: 0, 8, 0, 6, 8

FA Memory Access 4:

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

Set 0	Mem [0]	Mem [8]	Mem [6]	

Blocks 2-3 are LRU : write Mem[6] to Block 2

Example: Accessing A Fully-Associative Cache

- Fully-Associative cache contains 4 1-word blocks. Find the # Misses for each cache given this sequence of memory block accesses: 0, 8, 0, 6, 8

FA Memory Access 5:

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

Set 0	Mem [0]	Mem [8]	Mem [6]	

Example: Accessing A Fully-Associative Cache

- Fully-Associative cache contains 4 1-word blocks. Find the # Misses for each cache given this sequence of memory block accesses: 0, 8, 0, 6, 8

FA Memory Access 5:

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

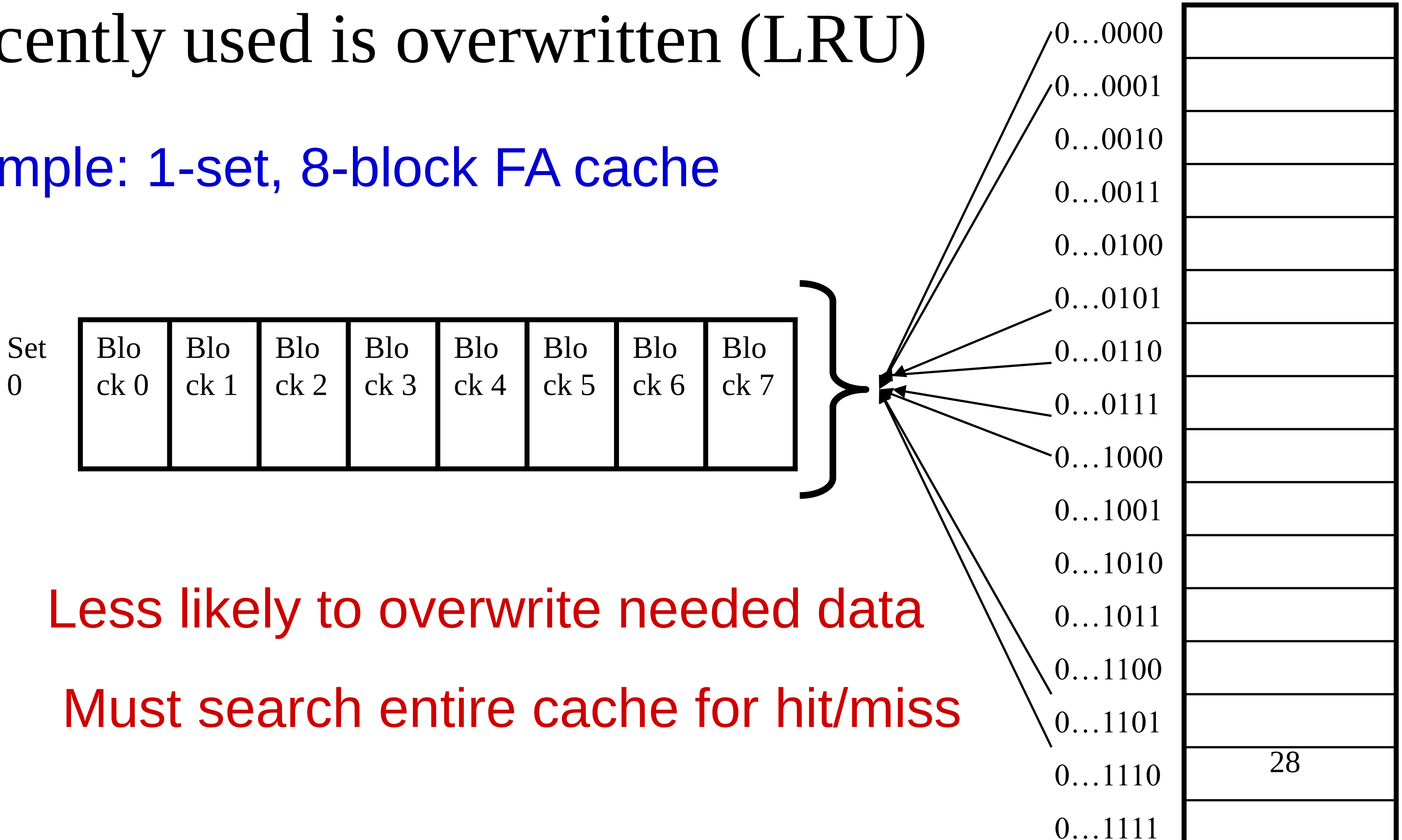
Set 0	Mem [0]	Mem [8]	Mem [6]	

Block 1 contains Mem[8]

Fully-Associative Cache Basics

1 set, n blocks: no mapping restrictions on how blocks are stored in cache: many ways, e.g. least recently used is overwritten (LRU)

Example: 1-set, 8-block FA cache



Set 0

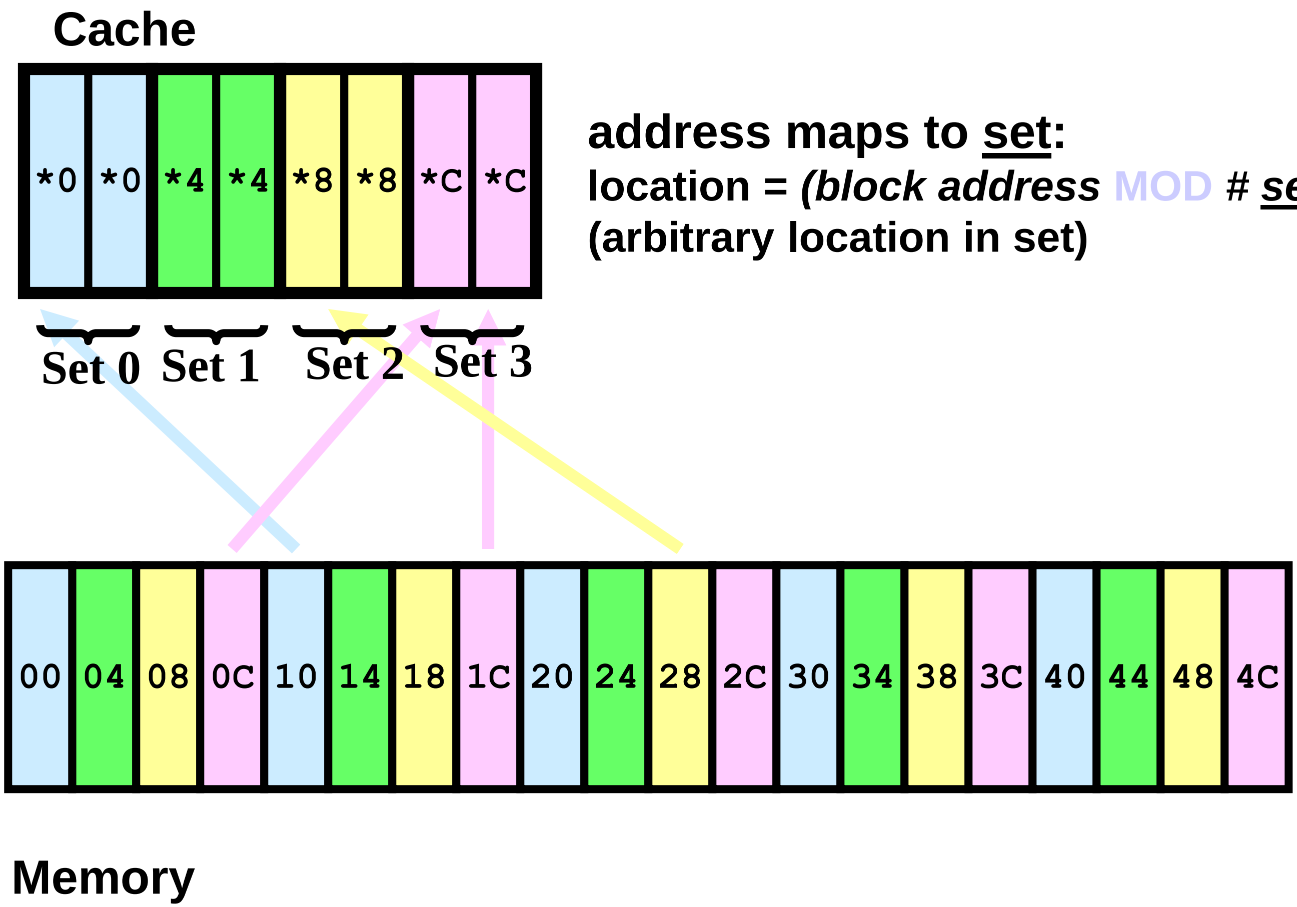
Block 0 | Block 1 | Block 2 | Block 3 | Block 4 | Block 5 | Block 6 | Block 7

0…0000
0…0001
0…0010
0…0011
0…0100
0…0101
0…0110
0…0111
0…1000
0…1001
0…1010
0…1011
0…1100
0…1101
0…1110
0…1111

PRO: Less likely to overwrite needed data

CON: Must search entire cache for hit/miss

Set-Associative Block Placement



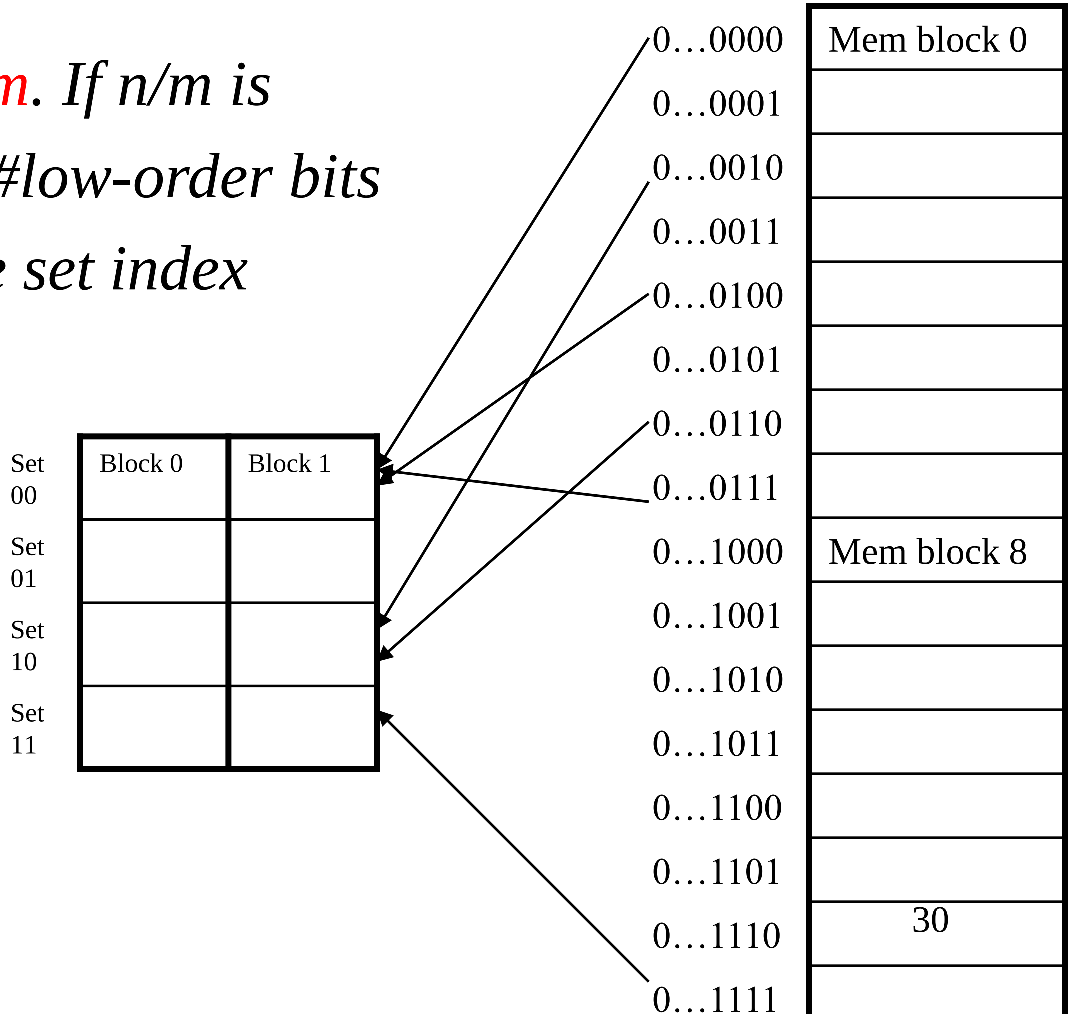
address maps to <u>set</u>:
location = *(block address* MOD # <u>*sets*</u> *in cache*)
(arbitrary location in set)

Set-Associative Cache Basics

n/m sets, m blocks (m-way): blocks are mapped from memory location to a specific set in cache

Mapping: Mem Address % n/m. If n/m is a power of 2, log2(n/m) = #low-order bits of memory address = cache set index

Example: 4 set, 2-way SA cache (ADD mod 4)



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
0	

Set 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 1: Mapping: 0 mod 2 = 0

Mem Block	DM Hit/Miss
0	miss

Set 0	Mem[0]	
Set 1		

Set 0 is empty: write Mem[0] to Block 0

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 2: Mapping: 8 mod 2 = 0

Mem Block	DM Hit/Miss
0	miss
8	

Set 0	Mem[0]	
Set 1		

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 2: Mapping: 8 mod 2 = 0

Mem Block	DM Hit/Miss
0	miss
8	miss

Set 0	Mem[0]	Mem[8]
Set 1		

Set 0, Block 1 is LRU: write Mem[8]

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 3: Mapping: 0 mod 2 = 0

Mem Block	DM Hit/Miss
0	miss
8	miss
0	

Set 0	Mem[0]	Mem[8]
Set 1		

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 3: Mapping: 0 mod 2 = 0

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

Set 0	Mem[0]	Mem[8]
Set 1		

Set 0, Block 0 contains Mem[0]

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	

Set 0	Mem[0]	Mem[8]
Set 1		

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	

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

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

Set 0	Mem[8]	Mem[6]
Set 1		

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

Set-Associative Cache Basics

n/m sets, m blocks (m-way): blocks are mapped from memory location to a specific set in cache

Mapping: Mem Address % n/m. If n/m is a power of 2, log2(n/m) = #low-order bits of memory address = cache set index

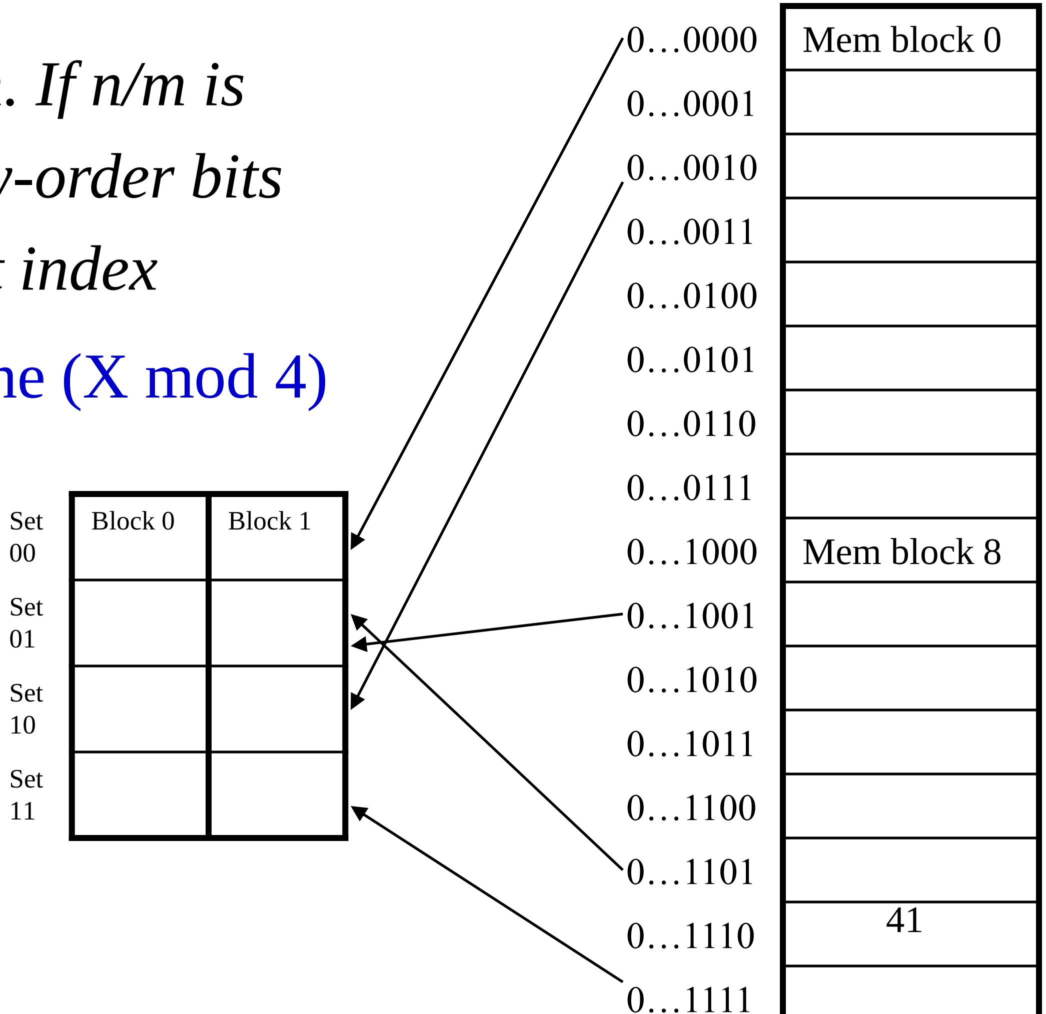
Example: 4 set, 2-way SA cache (X mod 4)

PRO:

Easier to find but won’t always overwrite

CON:

Must search set for hit/miss



Set 00
Set 01
Set 10
Set 11

Block 0 | Block 1

0…0000 Mem block 0
0…0001
0…0010
0…0011
0…0100
0…0101
0…0110
0…0111
0…1000 Mem block 8
0…1001
0…1010
0…1011
0…1100
0…1101
0…1110
0…1111

Associativity Considerations

- DM and FA are special cases of SA cache
  - Set-Associative: n/m sets; m blocks/set
  - Direct-Mapped: m=1
  - Fully-Associative: m=n
- Advantage of Associativity: as associativity increases, miss rate decreases (because more blocks per set that we're less likely to overwrite)
- Disadvantage of Associativity: as associativity increases, hit time increases (because we have to search more blocks – more HW required)
- Block Replacement: LRU or random. Random is easier to implement and often not much worse

Q2: Block Identification

- Every cache block has an address tag that identifies its location in memory
- Hit when tag and address of desired word match (comparison by hardware)
- Q: What happens when a cache block is empty?
  A: Mark this condition with a valid bit

Valid	Tag	Data
1	0x00001C0	0xff083c2d

Q2: Block Identification

- Tag on each block
  - No need to check index or block offset
- Increasing associativity shrinks index, expands tag

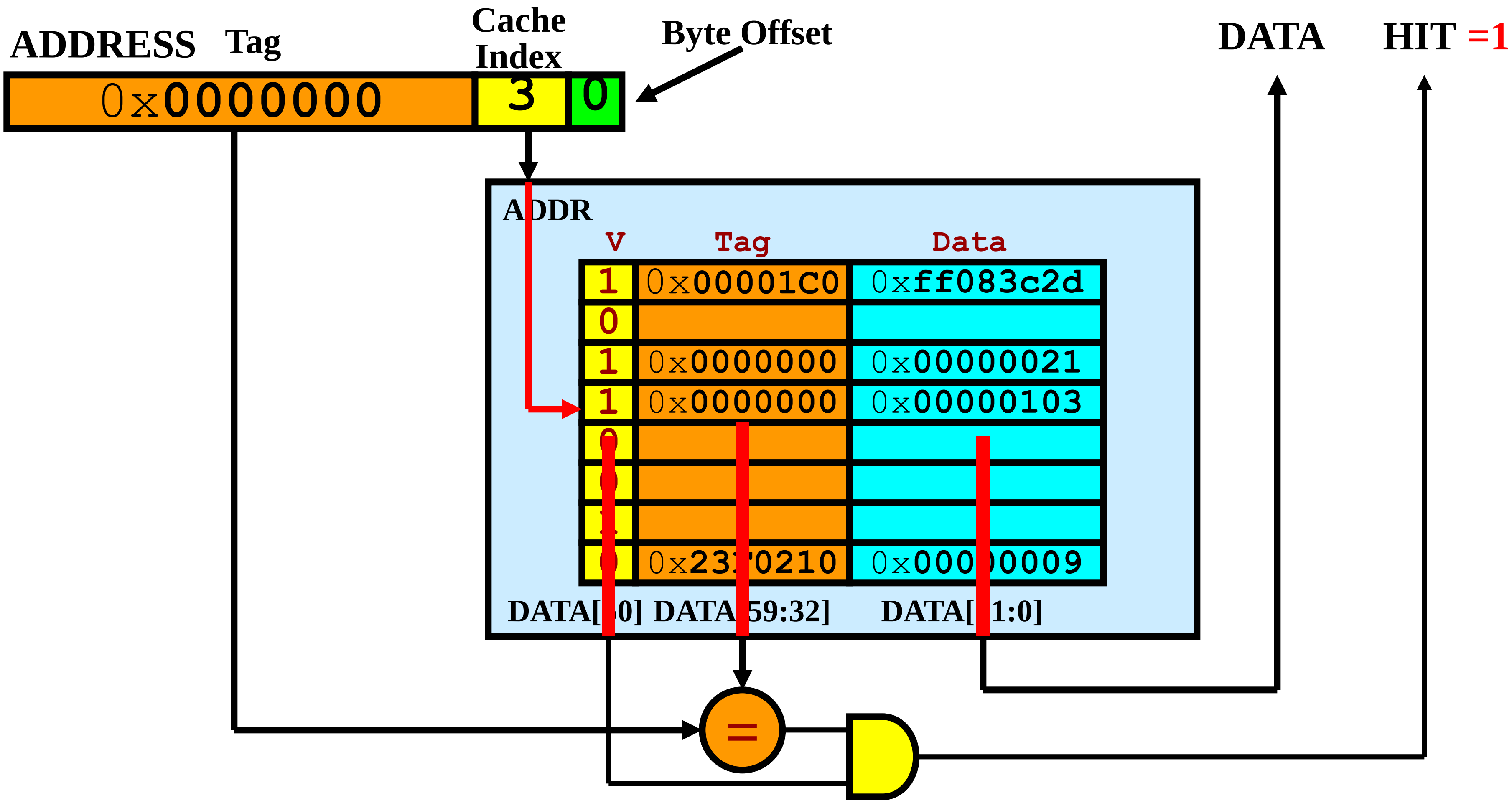
Block Address		Block Offset
Tag	Index	

Fully Associative: No index

Direct Mapped: Large index

An address is divided into two parts. The block address can be further divided into the tag field and the index field. The block offset field selects the desired data from the block, the index field selects the set, and the tag field is compared against it for a hit.

Direct-Mapped Cache Design



V	Tag	Data
1	0x00001C0	0xff083c2d
0		
1	0x0000000	0x00000021
1	0x0000000	0x00000103
0		
0		
1		
0	0x2370210	0x00000009