

CACHE ARCHITECTURE

Mahdi Nazm Bojnordi

Assistant Professor

School of Computing

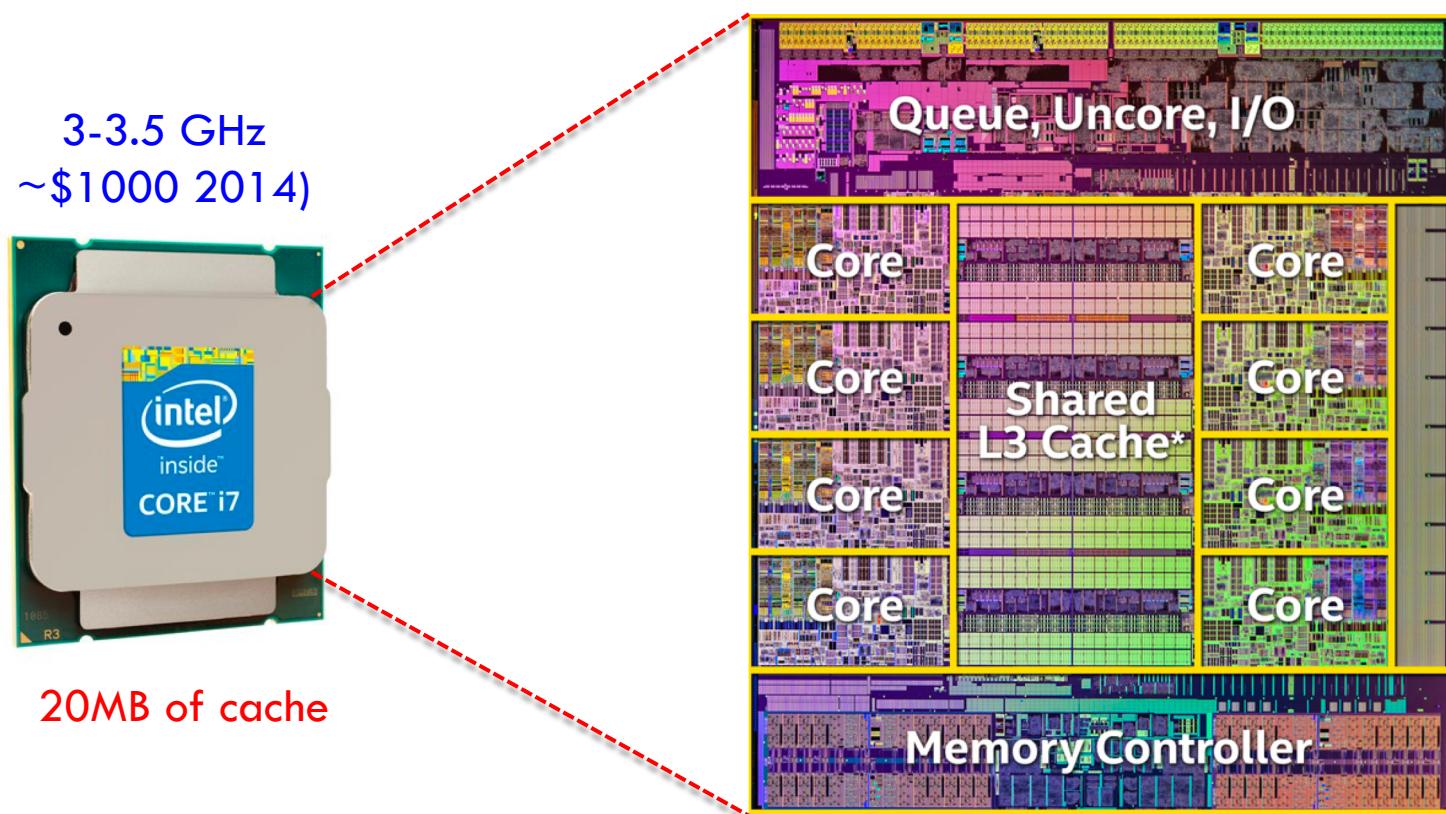
University of Utah

Overview

- Announcement
 - ▣ Mar. 14th: Homework 4 release (**due on Mar. 27th**)
- This lecture
 - ▣ Cache addressing and lookup
 - ▣ Cache optimizations
 - Techniques to improve miss rate
 - Replacement policies
 - Write policies

Processor Cache

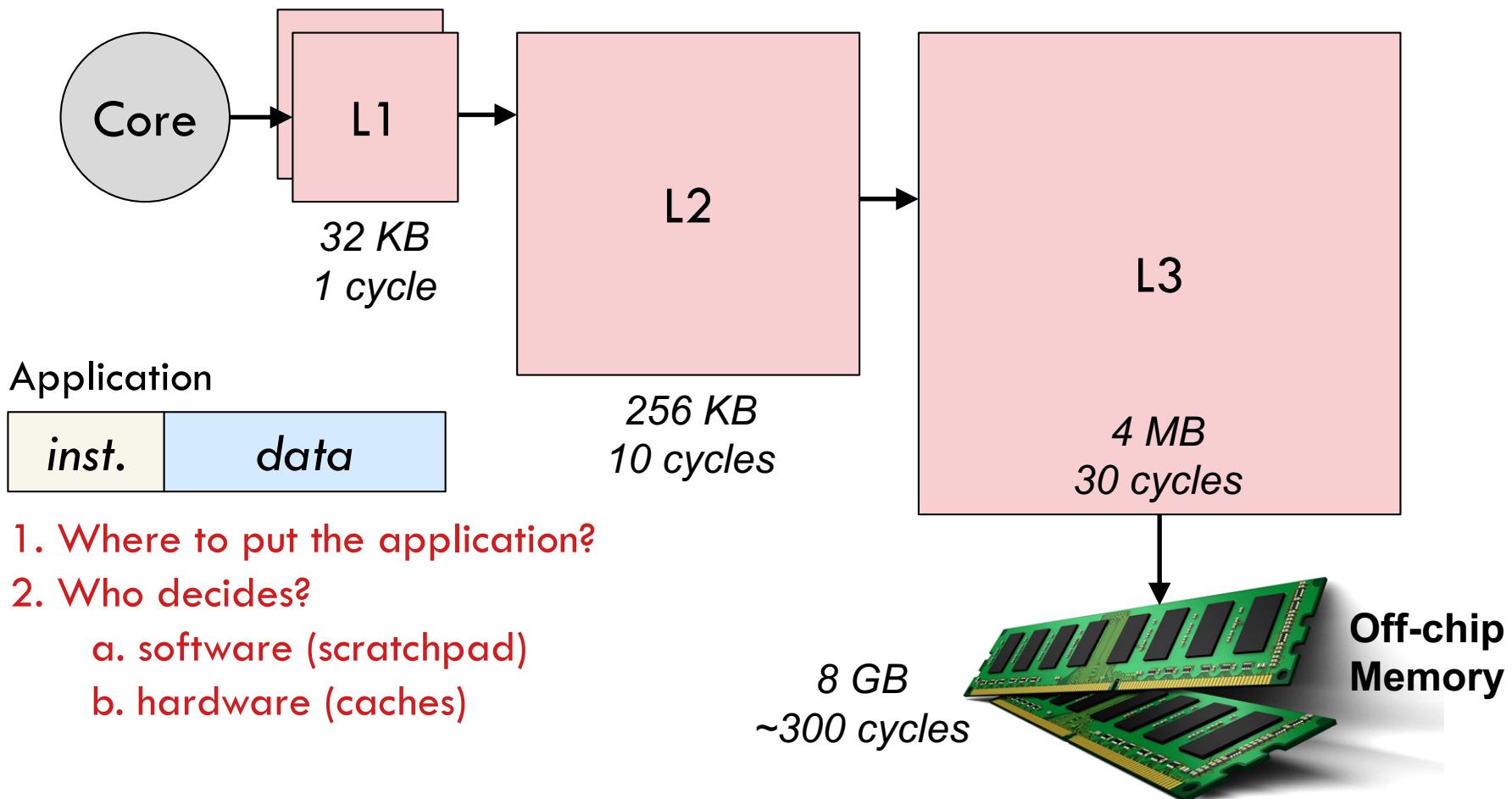
- Occupies a large fraction of die area in modern microprocessors



Source: Intel Core i7

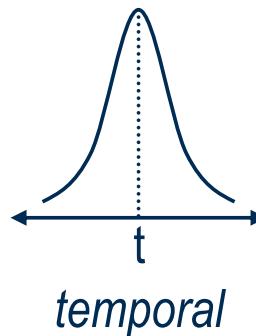
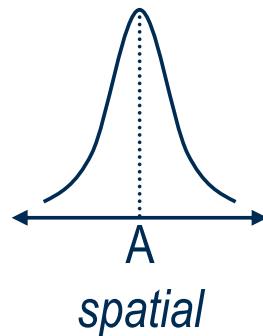
Cache Hierarchy

□ Example three-level cache organization



Principle of Locality

- Memory references exhibit localized accesses
- Types of locality
 - *spatial*: probability of access to $A+\delta$ at time $t+\varepsilon$
highest when $\delta \rightarrow 0$
 - *temporal*: probability of accessing $A+\varepsilon$ at time $t+\delta$
highest when $\delta \rightarrow 0$



```
for (i=0; i<1000; ++i) {  
    sum = sum + a[i];  
}  
temporal   spatial
```

Key idea: store local data in fast cache levels

Cache Terminology

- Block (cache line): unit of data access
- Hit: accessed data found at current level
 - ▣ *hit rate*: fraction of accesses that finds the data
 - ▣ *hit time*: time to access data on a hit
- Miss: accessed data NOT found at current level
 - ▣ miss rate: $1 - \text{hit rate}$
 - ▣ miss penalty: time to get block from lower level

hit time << miss penalty

Cache Performance

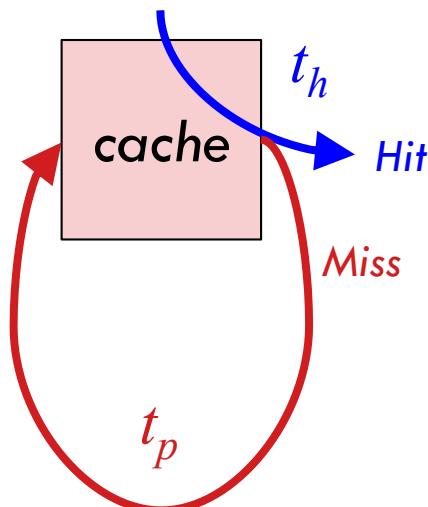
□ Average Memory Access Time (AMAT)

Outcome	Rate	Access Time
Hit	r_h	t_h
Miss	r_m	$t_h + t_p$

$$r_h = 1 - r_m$$

$$AMAT = r_h t_h + r_m (t_h + t_p)$$

$$AMAT = t_h + r_m t_p$$



Example: hit rate is 90%; hit time is 2 cycles; and accessing the lower level takes 200 cycles; find the average memory access time?

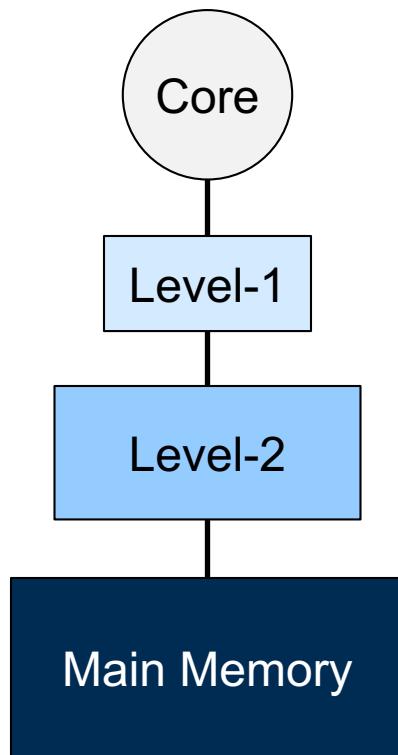
$$AMAT = 2 + 0.1 \times 200 = 22 \text{ cycles}$$

Example Problem

- Assume that the miss rate for instructions is 5%; the miss rate for data is 8%; the data references per instruction is 40%; and the miss penalty is 20 cycles; find performance relative to perfect cache with no misses
 - ▣ $\text{misses/instruction} = 0.05 + 0.08 \times 0.4 = 0.082$
 - ▣ Assuming hit time = 1
 - $\text{AMAT} = 1 + 0.082 \times 20 = 2.64$
 - Relative performance = $1/2.64$

Summary: Cache Performance

- Bridging the processor-memory performance gap



Main memory access time: 300 cycles

Two level cache

- L1: 2 cycles hit time; 60% hit rate
- L2: 20 cycles hit time; 70% hit rate

What is the average mem access time?

$$AMAT = t_{h1} + r_{m1} t_{p1}$$

$$t_{p1} = t_{h2} + r_{m2} t_{p2}$$

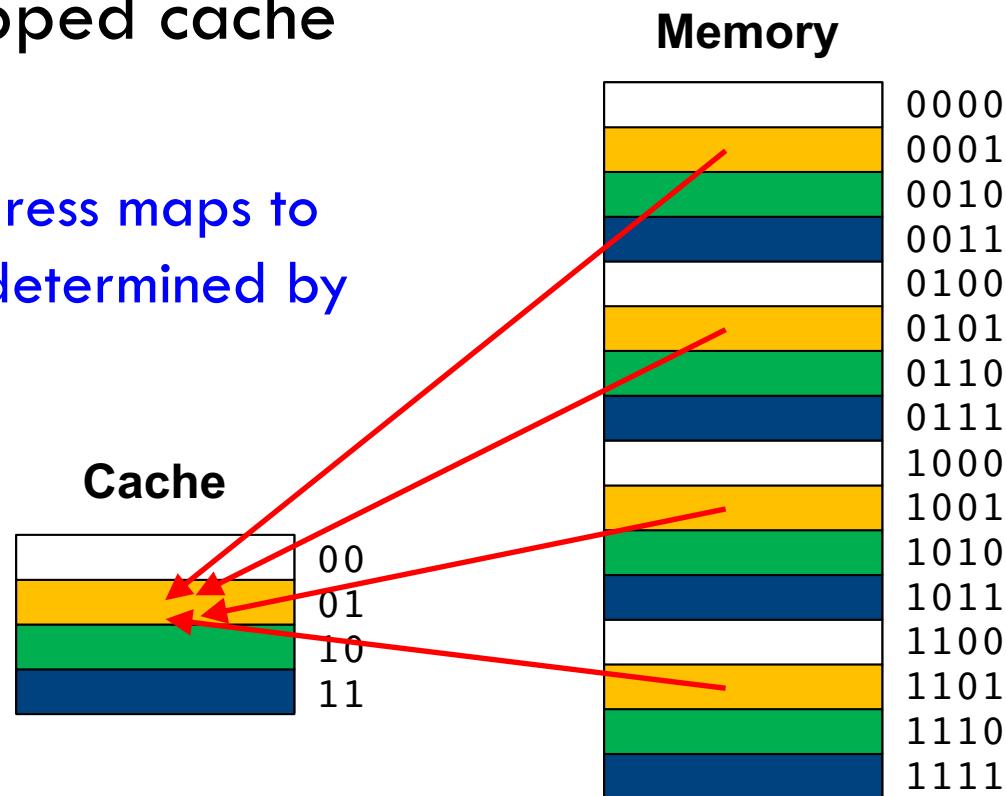
$$AMAT = 46$$

Cache Addressing

- Instead of specifying cache address we specify main memory address
- Simplest: direct-mapped cache

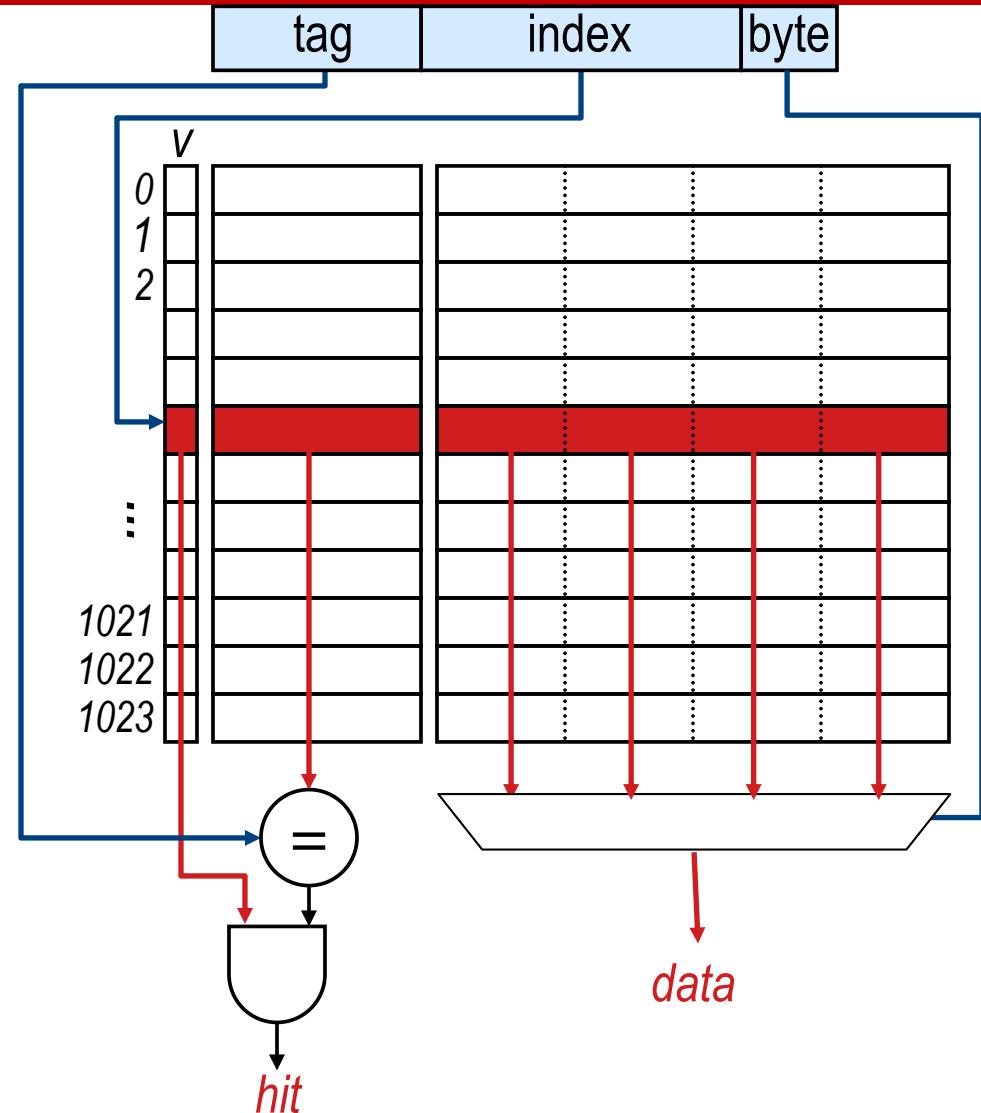
Note: each memory address maps to a single cache location determined by modulo hashing

How to exactly specify which blocks are in the cache?



Direct-Mapped Lookup

- Byte offset: to select the requested byte
- Tag: to maintain the address
- Valid flag (v): whether content is meaningful
- Data and tag are always accessed



Example Problem

- Find the size of tag, index, and offset bits for an 8MB, direct-mapped L3 cache with 64B cache blocks. Assume that the processor can address up to 4GB of main memory.

Example Problem

- Find the size of tag, index, and offset bits for an 8MB, direct-mapped L3 cache with 64B cache blocks. Assume that the processor can address up to 4GB of main memory.
- $4\text{GB} = 2^{32} \text{ B} \rightarrow \text{address bits} = 32$
- $64\text{B} = 2^6 \text{ B} \rightarrow \text{byte offset bits} = 6$
- $8\text{MB}/64\text{B} = 2^{17} \rightarrow \text{index bits} = 17$
- tag bits = $32 - 6 - 17 = 9$

Cache Optimizations

- How to improve cache performance?

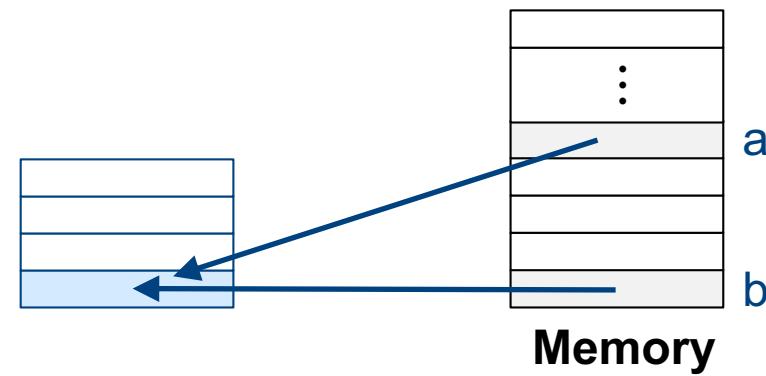
$$AMAT = t_h + r_m t_p$$

- Reduce hit time (t_h)
 - ▣ Memory technology, critical access path
- Improve hit rate ($1 - r_m$)
 - ▣ Size, associativity, placement/replacement policies
- Reduce miss penalty (t_p)
 - ▣ Multi level caches, data prefetching

Set Associative Caches

- Improve cache hit rate by allowing a memory location to be placed in more than one cache block
 - ▣ N-way set associative cache
 - ▣ Fully associative
- For fixed capacity, higher associativity typically leads to higher hit rates
 - ▣ more places to simultaneously map cache lines
 - ▣ 8-way SA close to FA in practice

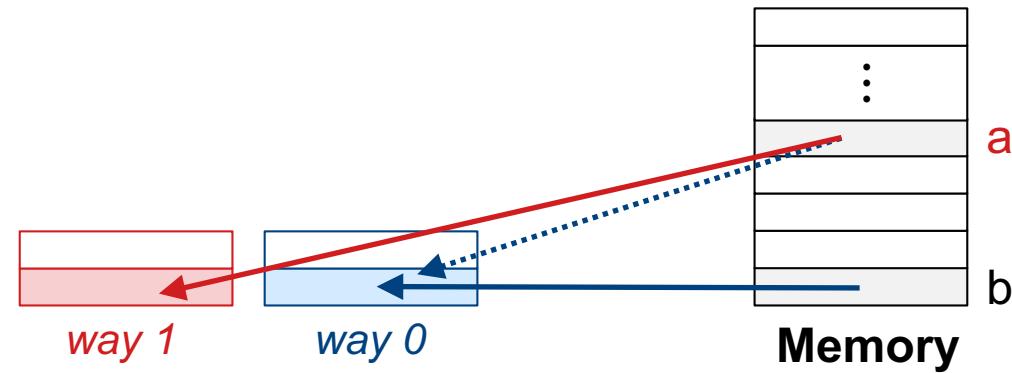
```
for (i=0; i<10000; i++) {  
    a++;  
    b++;  
}
```



Set Associative Caches

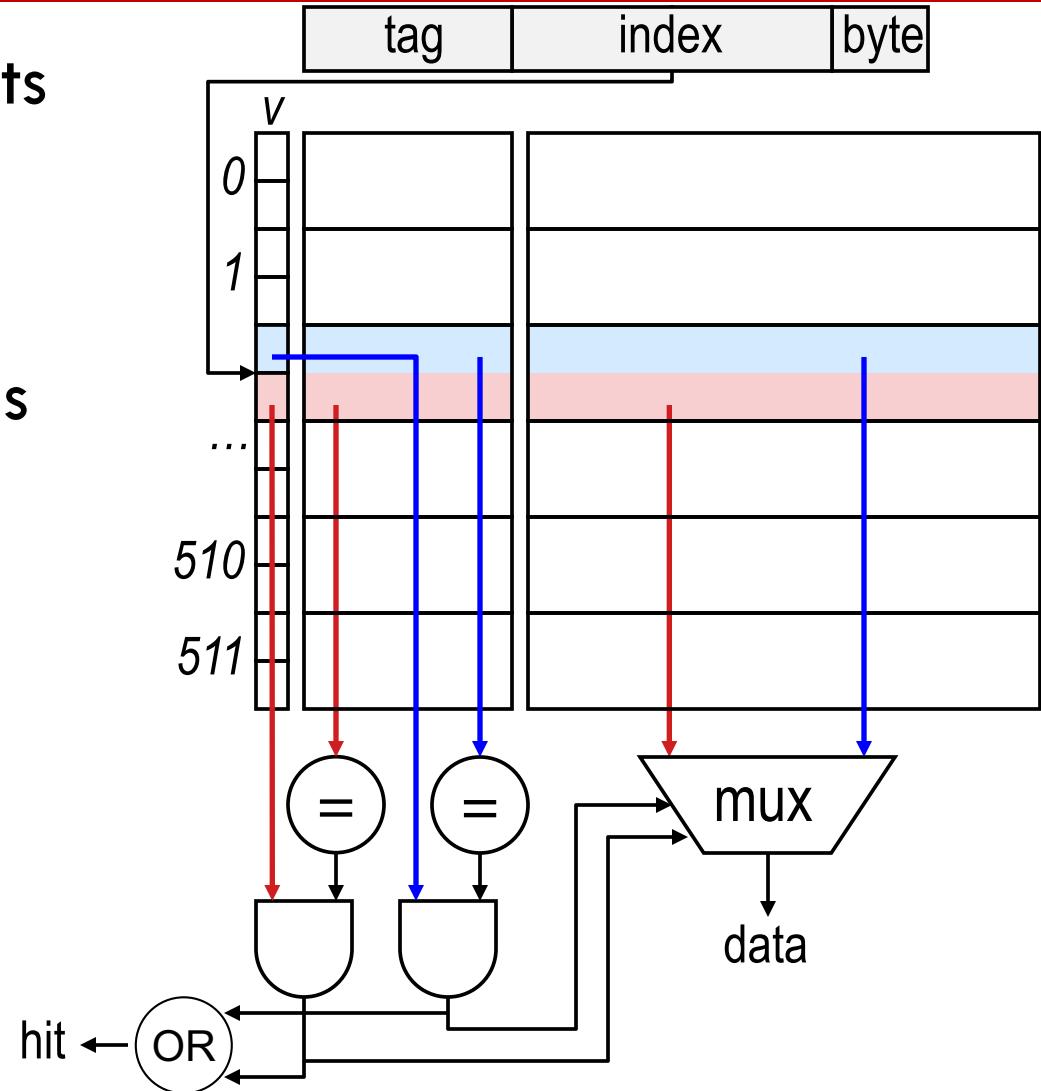
- Improve cache hit rate by allowing a memory location to be placed in more than one cache block
 - ▣ N-way set associative cache
 - ▣ Fully associative
- For fixed capacity, higher associativity typically leads to higher hit rates
 - ▣ more places to simultaneously map cache lines
 - ▣ 8-way SA close to FA in practice

```
for (i=0; i<10000; i++) {  
    a++;  
    b++;  
}
```



n-Way Set Associative Lookup

- Index into cache sets
- Multiple tag comparisons
- Multiple data reads
- Special cases
 - ▣ Direct mapped
 - Single block sets
 - ▣ Fully associative
 - Single set cache



Example Problem

- Find the size of tag, index, and offset bits for an 4MB, 4-way set associative cache with 32B cache blocks. Assume that the processor can address up to 4GB of main memory.

Example Problem

- Find the size of tag, index, and offset bits for an 4MB, 4-way set associative cache with 32B cache blocks. Assume that the processor can address up to 4GB of main memory.
- $4\text{GB} = 2^{32} \text{ B} \rightarrow \text{address bits} = 32$
- $32\text{B} = 2^5 \text{ B} \rightarrow \text{byte offset bits} = 5$
- $4\text{MB}/(4 \times 32\text{B}) = 2^{15} \rightarrow \text{index bits} = 15$
- tag bits = $32 - 5 - 15 = 12$

Cache Miss Classifications

- Start by measuring miss rate with an ideal cache
 - ▣ 1. ideal is fully associative and infinite capacity
 - ▣ 2. then reduce capacity to size of interest
 - ▣ 3. then reduce associativity to degree of interest

1. Cold (compulsory)

- ❑ Cold start: first access to block
- ❑ How to improve
 - large blocks
 - prefetching

2. Capacity

- ❑ Cache is smaller than the program data
- ❑ How to improve
 - large cache

3. Conflict

- ❑ Set size is smaller than mapped mem. locations
- ❑ How to improve
 - large cache
 - more assoc.