Computer Architecture

Chapter 5b. Memory System

Hyuk-Jun Lee, PhD

Dept. of Computer Science and Engineering
Sogang University
Seoul, Korea

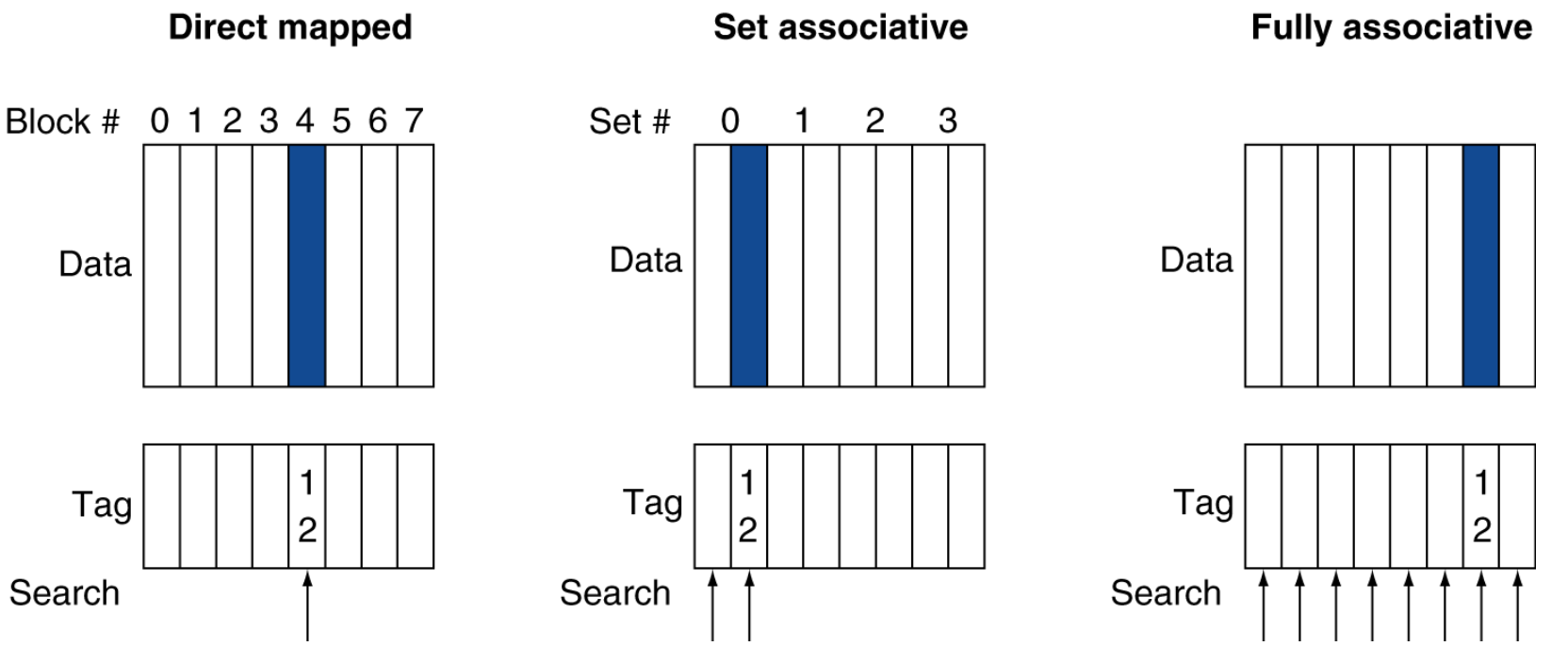
Email: hyukjunl@sogang.ac.kr

Associative Caches

- Fully associative
  - Allow a given block to go in any cache entry
  - Requires all entries to be searched at once
  - Comparator per entry (expensive)
- *n*-way set associative
  - Each set contains *n* entries
  - Block number determines which set
    - (Block number) modulo (#Sets in cache)
  - Search all entries in a given set at once
  - *n* comparators (less expensive)

Associative Cache Example

set마다 2가지의 mapping 방법이 있음
2 way set associative cache
% 4해서 값이 나오면 그걸 왼쪽, 오른쪽을 선택해서 저장



12 % 8

20, 28 등등 % 8해서 4가 나오면 계속 cache miss 발생(conflict miss)

cache miss
1. conflict miss - tho have empty blocks still have cache miss cause of conflict
2. compulsory miss(cold miss) - in the beginning we have huge cache miss(empty cache)
3. capacity miss - size of cache is small than working set

Spectrum of Associativity

- For a cache with 8 entries

One-way set associative
(direct mapped)

Block	Tag	Data
0		
1		
2		
3		
4		
5		
6		
7		

Two-way set associative

Set	Tag	Data	Tag	Data
0				
1				
2				
3				

total size is the same but increase associativity

Four-way set associative

Set	Tag	Data	Tag	Data	Tag	Data	Tag	Data
0								
1								

Eight-way set associative (fully associative)

Tag	Data	Tag	Data	Tag	Data	Tag	Data	Tag	Data	Tag	Data	Tag	Data	Tag	Data

Associativity Example

- Compare 4-block caches block offset 4
  - Direct mapped, 2-way set associative, fully associative
  - Block access sequence: 0, 8, 0, 6, 8
- Direct mapped

Block address	Cache index	Hit/miss	Cache content after access			
			0	1	2	3
0	0	miss	**Mem[0]**			
8	0	miss	**Mem[8]**			
0	0	miss	**Mem[0]**			
6	2	miss	Mem[0]		**Mem[6]**	
8	0	miss	**Mem[8]**		Mem[6]	

처음에 비어있고 cache index 0에 block add 0 저장

cache index 0에 mem 0이 저장되어 있어서 miss 나고 mem 8을 저장

Associativity Example

- 2-way set associative

Block address	Cache index	Hit/miss	Cache content after access			
			Set 0		Set 1	
0	0	miss	**Mem[0]**			
8	0	miss	Mem[0]	**Mem[8]**	왼쪽에는 이미 있으니	오른쪽에 저장
0	0	hit	**Mem[0]**	Mem[8]		
6	0	miss	Mem[0]	**Mem[6]**		
8	0	miss	**Mem[8]**	Mem[6]		

파란색 - compulsory miss
red - conflict miss

- Fully associative

fully associative 에서 conflict miss 가 최소화
여기서 miss가 나는 것을 capacity miss라고 측정한다

Block address		Hit/miss	Cache content after access			
0		miss	**Mem[0]**			
8		miss	Mem[0]	**Mem[8]**		
0		hit	**Mem[0]**	Mem[8]		
6		miss	Mem[0]	Mem[8]	**Mem[6]**	
8		hit	Mem[0]	**Mem[8]**	Mem[6]	

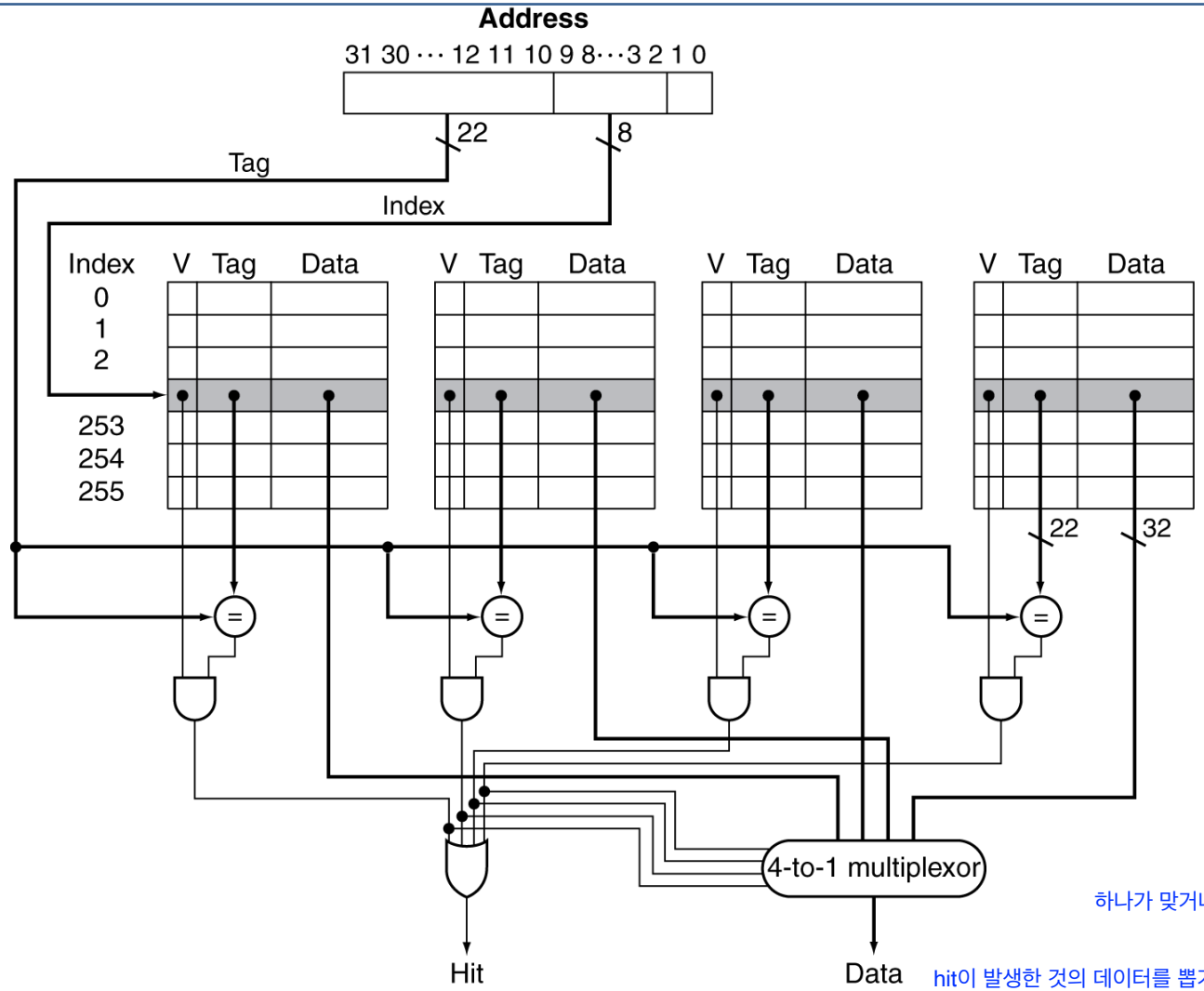
How Much Associativity

- Increased associativity decreases miss rate
  - But with diminishing returns
- Simulation of a system with 64KB D-cache, 16-word blocks, SPEC2000
  - 1-way: 10.3%
  - 2-way: 8.6%
  - 4-way: 8.3%
  - 8-way: 8.1%

associativity를 높이는데도 cost가 들기 때문에 너무 높은게 좋은게 아니다

direct map cache - 256 entry in single way. 4 way. cache size 1024 block
index bit = 10bits
근데 8칸이라서 2bit가 tag bit이 된다4

Set Associative Cache Organization



4개를 병렬로 읽고
tag bit을 주어진 tag와 비교
하나가 맞거나(hit) 아무것도 안맞는다(miss)

hit이 발생한 것의 데이터를 뽑기 위해

Replacement Policy

- Direct mapped: no choice
- Set associative
  - Prefer non-valid entry, if there is one
  - Otherwise, choose among entries in the set
- Least-recently used (LRU)
  - Choose the one unused for the longest time
    - Simple for 2-way, manageable for 4-way, too hard beyond that
- Random
  - Gives approximately the same performance as LRU for high associativity

l1 miss -> l2 miss -> l3 miss -> main memory

Multilevel Caches

- Primary cache attached to CPU
  - Small, but fast
- Level-2 cache services misses from primary cache
  - Larger, slower, but still faster than main memory
- Main memory services L-2 cache misses
- Some high-end systems include L-3 cache

l3 cache가 없을 때