CS2100 - L14 - Caches (Set Associative & Fully Associative)

Week 8+9

14.1 - Cache performance

14.2 - Types of cache misses

14.3 - Set associative cache

14.4 - Fully associative cache

14.5 - Block replacement policy

14.6 - Cache framework

14.1 - Cache performance

Memory Access Time: Formula

Average Access Time

= Hit rate x Hit Time + (1-Hit rate) x Miss penalty

- **Hit**: Data is in cache (e.g., **X**)
  - **Hit rate**: Fraction of memory accesses that hit
  - **Hit time**: Time to access cache
- **Miss**: Data is not in cache (e.g., **Y**)
  - **Miss rate** = 1 – Hit rate
  - **Miss penalty**: Time to replace cache block + deliver data

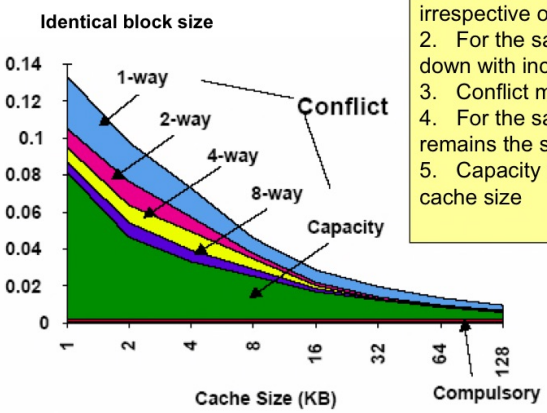
- Hit time < Miss penalty

Cache Performance

block size remains the same

Observations:

1. Cold/compulsory miss remains the same irrespective of cache size/associativity
2. For the same cache size, conflict miss goes down with increasing associativity
3. Conflict miss is 0 for FA caches
4. For the same cache size, capacity miss remains the same irrespective of associativity
5. Capacity miss decreases with increasing cache size



not a math formula

Total Miss = Cold miss + Conflict miss + Capacity miss

Capacity miss (FA) = Total miss (FA) – Cold miss (FA), when Conflict Miss➔0

well defined for FA

14.2 - Types of cache misses

Cache Misses: **Classifications**

Compulsory / Cold Miss

- First time a **memory block** is accessed
- Cold fact of life: Not much can be done
- **Solution:** Increase cache block size

Conflict Miss

- Two or more distinct memory blocks map to the same cache block
- Big problem in direct-mapped caches
- **Solution 1**: Increase cache size
  - Inherent restriction on cache size due to SRAM technology
- **Solution 2**: **Set-Associative caches** (coming next ..)

Capacity Miss

- Due to limited cache size
- Will be further clarified in "fully associative caches" later

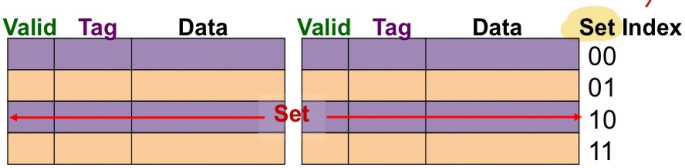
14.3 - Set associative cache

Set Associative (SA) Cache

- **N-way Set Associative Cache**
  - A memory block can be placed in a fixed number of locations (`N > 1`) in the cache
- **Key Idea:**
  - Cache consists of a number of sets:
    - Each set contains **N** cache blocks
  - Each memory block maps to a unique cache set
  - Within the set, a memory block can be placed in **any** element of the set

SA Cache Structure

Functionally the same as DM



2-way Set Associative Cache

Set-Associative Cache: **Mapping**

Memory Address

Cache Block size = 2^N bytes

Cache Set Index
= (BlockNumber) modulo (NumberOfCacheSets)

Cache Block size = 2^N bytes
Number of cache sets = 2^M
Offset = **N bits**
Set Index = **M bits**
Tag = **32 – (N + M) bits**

Observation:
It is essentially unchanged from the direct-mapping formula

Advantage of Associativity (3/3)

Rule of Thumb:
A direct-mapped cache of size **N** has about the same miss rate as a 2-way set associative cache of size **N / 2**

14.4 - Fully associative cache

Fully Associative (FA) Cache

- **Fully Associative Cache**
  - A memory block can be placed in any location in the cache

- **Key Idea:**
  - Memory block placement is no longer restricted by cache index / cache set index

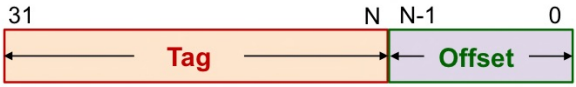
++ Can be placed in any location, **BUT**

--- Need to search all cache blocks for memory access → lots of hardware needed

Fully Associative Cache: Mapping

Memory Address

Cache Block size = 2^N bytes



Cache Block size = 2^N bytes
Number of cache blocks = 2^M
Offset = **N bits**
Tag = **32 – N bits**

Observation:
The block number serves as the tag in FA cache

14.5 - Block replacement policy

Block Replacement Policy (3/3)

least Recently Used

- Drawback for LRU

  Difficult to shuffle blocks within set

  - Hard to keep track if there are many choices

- Other replacement policies:
  - First in first out (FIFO)
    - Second chance variant (gauges whether block has been accessed after being placed)
  - Random replacement (RR)
  - Least frequently used (LFU)

Cache Organizations: Summary

One-way set associative (direct mapped)

N=1

Block	Tag	Data
0		
1		
2		
3		
4		
5		
6		
7		

Two-way set associative

N=2

Set	Tag	Data	Tag	Data
0				
1				
2				
3				

Four-way set associative

N=4

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

14.6 - Cache framework

Cache Framework (1/2)

Block Placement: Where can a block be placed in cache?

Direct Mapped:	N-way Set-Associative:	Fully Associative:
• Only one block defined by index	• Any one of the **N** blocks within the set defined by index	• Any cache block

Block Identification: How is a block found if it is in the cache?

Direct Mapped:	N-way Set Associative:	Fully Associative:
• Tag match with only one block	• Tag match for all the blocks within the set	• Tag match for all the blocks within the cache

Cache Framework (2/2)

Block Replacement: Which block should be replaced on a cache miss?

Direct Mapped:	n-way Set-Associative:	Fully Associative:
• No Choice	• Based on replacement policy	• Based on replacement policy

Write Strategy: What happens on a write?

Write Policy: Write-through vs. write-back

Write Miss Policy: Write allocate vs. write no allocate

L13 - DM Cache