Cache memory:

Hierarchy
- L1 → L1-I, L1-D
- L2
- Last Level Cache (LLC)
- Main memory

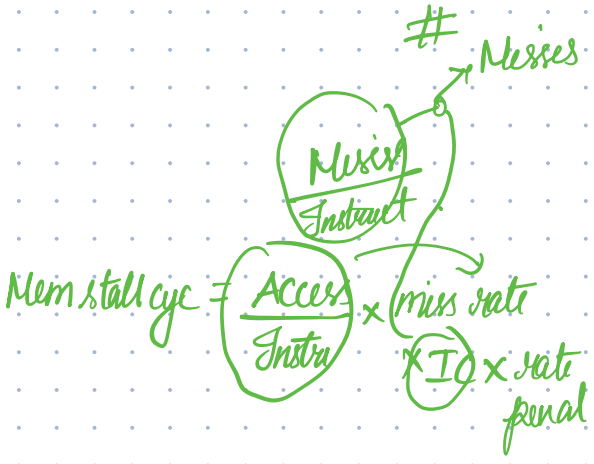
Principle of locality → Temporal locality, Spatial locality

Important terms:
- Core request (word)
- Cache block
- Cache hit / Cache miss
- Cache replacement

CPU:

Instruction execution cycle:

- IF : Instruct Fetch
- ID : Instruct Decode
- Ex : Execution
- MEM : Memory access
- WB : Write back



CPU execution time = (clock cycle + memory stall cycles) × clock cycle time.

Memory stall cycles = number of misses × miss penalty

$= \frac{\text{misses}}{\text{instruct}} \times IC \times \text{miss penalty}$

$= \frac{\text{Access}}{\text{instruct}} \times \text{miss rate} \times IC \times \text{miss penalty}$

$CPI = \frac{\text{clock cycle}}{IC}$

AMAT = Hit time + Miss rate × miss penalty.

CPU time = (CPU cycle + Mem stall) × clock time

Mem stall = $IC \times \frac{\#\text{misses}}{\text{Instruct}} \times \text{penalty}$

#misses = Access × miss rate.

$CPI = \frac{\text{Clock}}{IC}$

AMAT = Hit time + miss rate × penalty

Memory :

Memory Hierarchy:

Cache memory:
1) L1-I, L1-D → L1
2) L2
3) LLC
4) Main memory
5) Secondary memory (Not discuss)

→ Principle of locality

* Temporal locality : If a core request for a data, it will request it again in future.
* Spatial locality : If a core request for a data, it will request nearby data in future.

→ Core will request for word. But we will always fetch a block.

→ The first fetch is always compulsory miss.

32-bit

Cache memory → SRAM technology
→ Multiple words stored in block

Managing data in cache → Block searching
→ Block placement
→ Block replacement

Average Memory Access Time (AMAT) = Hit time + Miss rate x Miss penalty.

Types in cache → Direct mapped cache
→ Fully associative cache
→ Set-associative cache

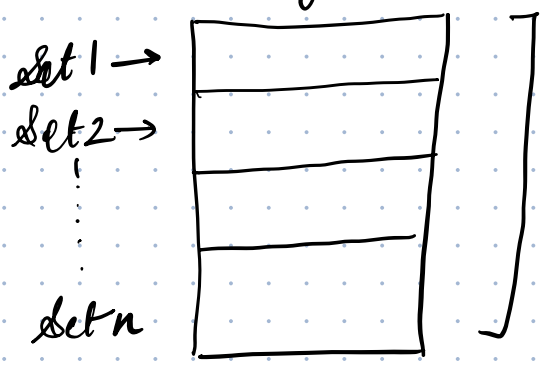
Word
↓
Block
↓
Set

32-bit

→ Based on this we decide which set it is mapping

Direct mapped cache:

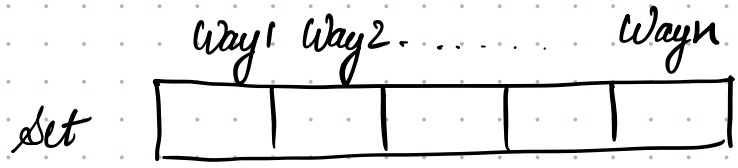
Inside a set, the tag is used to search if the word already exists.



Based on set index we will find which set the word belongs to

Problems:
→ There is only one place to keep the block. Because it's 1 block per set.
→ Conflict is more.

Fully associative cache:

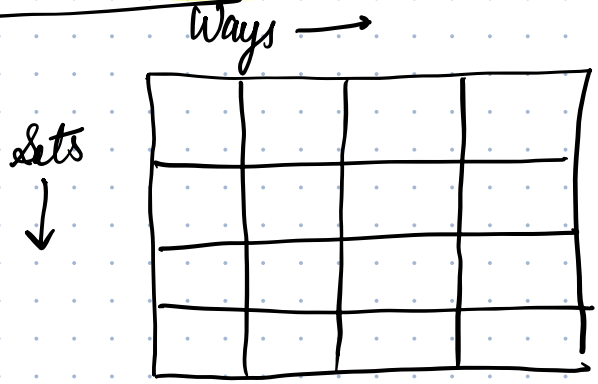


- → Only 1 set
- → But multiple ways
- → No conflict miss
- → High latency

Problems:

→ High latency because we have to search through all the ways by comparing the tags.

Set-associative cache:



→ Best of both worlds

- → comparitively less conflict miss.

Example:

8-way set associative cache:

Size = ways × sets × blocks

Types of misses:

Compulsory miss: First time accessing a block from main memory.

Conflict miss: A block is trying to occupy an existing cache line even if there are empty

Capacity miss: The miss occurs when the cache is full.

Multiprocessor:

→ Multiprogram

→ Multi threaded.

Cache in multi core:

→ Shared cache

→ Private cache

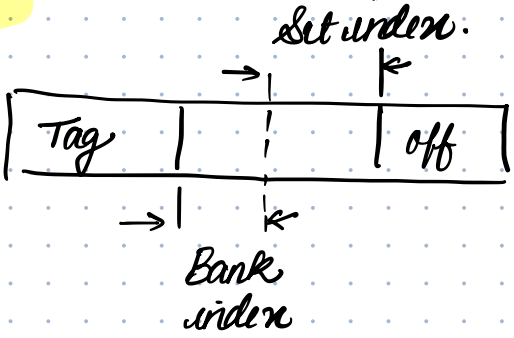
NUCA : Non-Uniform Cache Access :

→ LLC is divided into banks (NUCA). ⇒ We need bank index.

→ Each bank is set-associative.

→ Banks near to the core have less latency

SNUCA : Static NUCA :



→ Easy search
→ More banks

→ If heavy use bank is far, latency would be high
→ No bank migration

→ First find the bank → then set → Then search the ways using tag.

DNUCA : Dynamic NUCA :

→ Allows migration of data inside bank-sets.

→ Introduces bank-sets : Sub-group of banks.

→ Migrate closer to core but within the same bank set.

Problem :

→ Latency → Due to search, would be high because we have search all the banks inside a bank set.

→ Increases network congestion.

NUCA in CMP:

CMP : Multiple cores & cache memory in a single chip → only 1 common LLC

TCMP : Each core has its own small cache. But the L2 is not private.

↓ Tile based CMP

Cache coherence protocols:

→ MSI

→ MESI

CPU execution time = (CPU clock cycle + Memory stall cycles) × clock cycle time

CPU clock cycle = CPI × IC

Memory stall cycle = Number of misses × penalty

$$= \frac{\text{Number of misses}}{\text{Instructions}} \times IC \times \text{penalty}$$

$$= \frac{\text{Memory access}}{\text{Instruction}} \times \text{Miss rate} \times IC \times \text{penalty}.$$