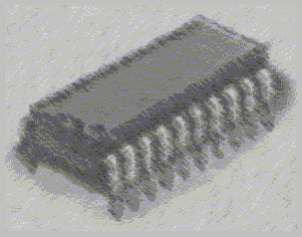
Lesson 4. Memory System

Computers Structure and Organization



Graduated in Computer Sciences
Graduated in Computers Engineering

Automatic Department

Contents

- Memory System characteristics
- Interleaved Memory
- Cache and Virtual Memory principles
- Memory hierarchy
- Cache Memory
  - Mapping function
  - Replacement Algorithm
  - Write policy
  - Line size and number of caches
- Current caches examples

Memory characteristics

- According to von Neumann computer architecture model, memory system is the place where programs and data must be stored to be executed.
- Memory system is a hierarchy of different cost, size and speed memory elements in which the overall system is not quite expensive and it's fast
- To be taken into account when designing a memory system:
  - Storage capacity (maximum addressable space)
  - Velocity (overall system)
  - Cost (memory system)

Main memory (I)

- Memory bandwidth (number of transmitted bits per second) must be increased to improve memory performance.
- Mechanisms to increase main memory bandwidth:
  - Decrease access time / latency
  - Increase word size
  - Interleaved memory

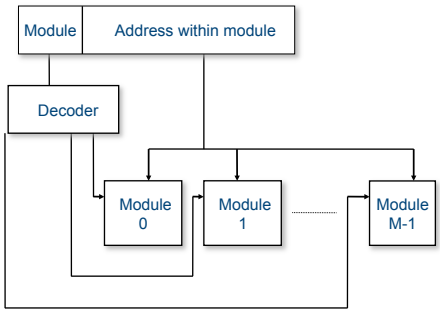
Main memory (II)
Interleaved memory (I)

- Concurrent module accesses to more cheaper and slower memory modules.
- Bandwidth can be multiplied by M, being M the number of modules.
- Memory conflict may appear when different memory accesses point to the same memory module at once.
- Type of interleaving:
  - According to addresses map distribution: higher and lower order
  - According to module access: simple or complex interleaving

Main memory (III)
Interleaved memory (II)

Higher order interleaving:

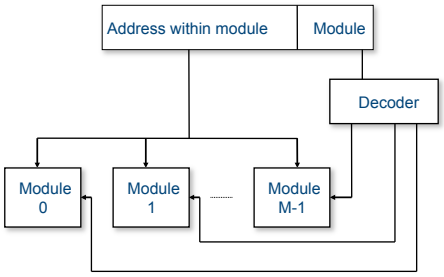
- Memory addresses are distributed among modules. Memory addresses are consecutives inside each module
- **Advantages:**
  - Easier memory expansion
  - High fault tolerant facility when a module failure occurs



Main memory (IV)
Interleaved memory (III)

Lower order interleaving:

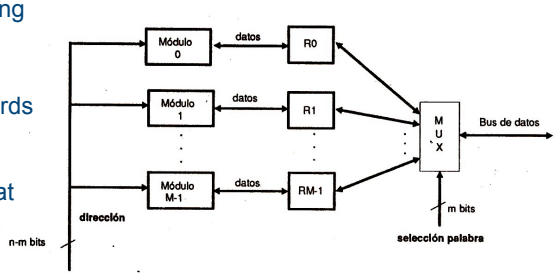
- Consecutive memory addresses are assigned to consecutive modules.
- **Advantage**:
  - Conflict memory accesses decrease when consecutive address accesses are requested.



Main memory (V)
Interleaved memory (IV)

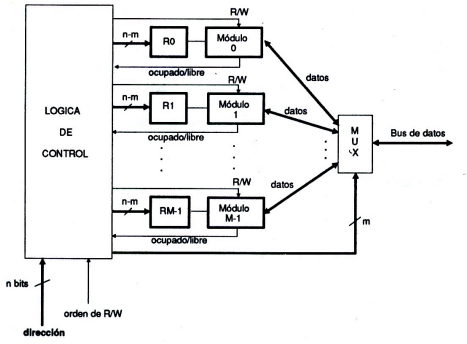
Simple interleaving:

- All modules are accessed with the same memory address.
- Lower Order Interleaving required.
- **Advantage**:
  - 2^m consecutive words are obtained when accessing to all memory modules at once



Main memory (and VI)
Interleaved memory (and V)

- **Complex interleaving:**
- Different memory addresses are attached to each module
- **Advantage**:
  - Accessed memory addresses mustn't be consecutive



Cache and virtual memory principles

- Memory accesses are grouped. Fixed groups of memory refereces are used by the microprocessor on short time periods.

Principle	Meaning
Temporal locality	If an element is accessed, it will be accessed again in a short time period • Generated data on following operations • Same instruction inside a loop
Spatial locality	If an element is accessed, closer elements to it will be also accessed • Data of an array • Program sequential structure

Memory hierarchy (I)

- **Hit: m**emory access in which datum is found
- **Miss:** memory access in which datum is not found
- **Hit ratio:** hit percentage
- **Hit time:** elapsed time when a hit occurs
- **Failure penalty:** additional time needed to gather a datum when a failure access occurs
- Block size has influence on hit ratio

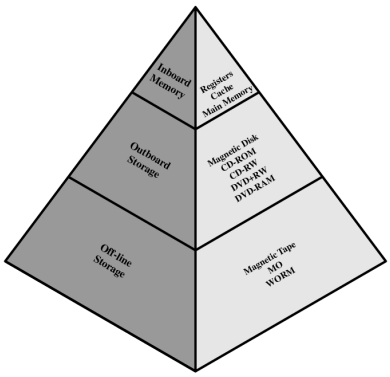


Image from William Stallings. Computer Organization and Architecture. 8th Edition Slides

Memory hierarchy (and II)

How to access to a memory position:

1º Microprocessor requests a memory address

2º Memory address is searched in the fastest memory level (cache)

- If found, requested word is returned to microprocessor
- Otherwise, address is searched on next memory level (main memory)
  - If found, requested word is updated in previous level (cache)
  - Otherwise, address is searched on next memory level (hard disk)

and so on

How to do it:

A mapping function must be defined because requested address may have different addresses in each memory level

Cache memory (I)

- **Aim:**
  - Memory address time access similar to registers access time
- **Cache memory design**:
  - Hit ratio
  - Effective time access
  - Failure penalty
  - Mapping main memory to cache memory addresses

Memoria cache (II)

- A cache memory system parameters:
  - **Mapping function:** addressing main memory and cache addresses
  - **Replacement algorithms:** blocks to be replaced when a new block goes to cache memory and cache is full
  - **Write policy:** when new information is written on main memory
  - Line, block, size
  - **Number of caches:** one or several cache memories

Memoria cache (III)

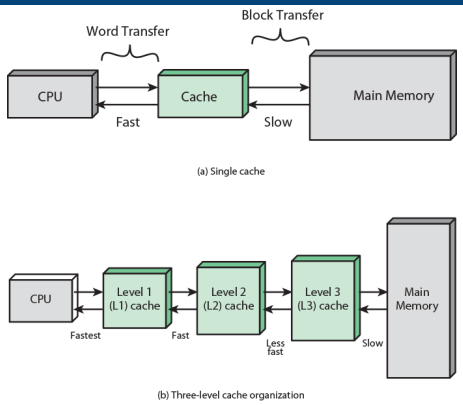


Image from William Stallings. Computer Organization and Architecture. 8th Edition Slides

Memoria cache (IV)

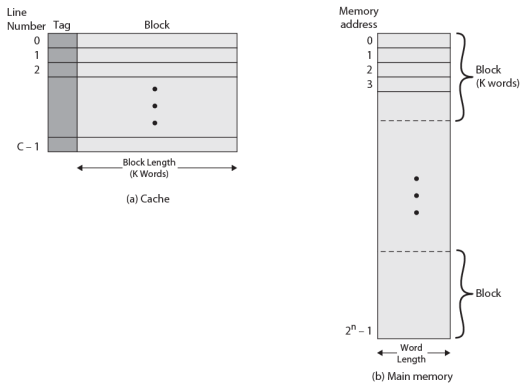
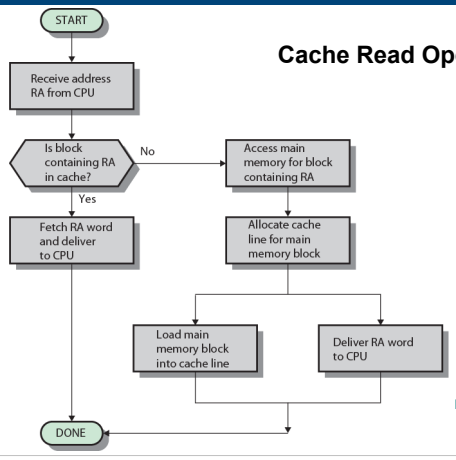


Image from William Stallings. Computer Organization and Architecture. 8th Edition Slides

Memoria cache (V)

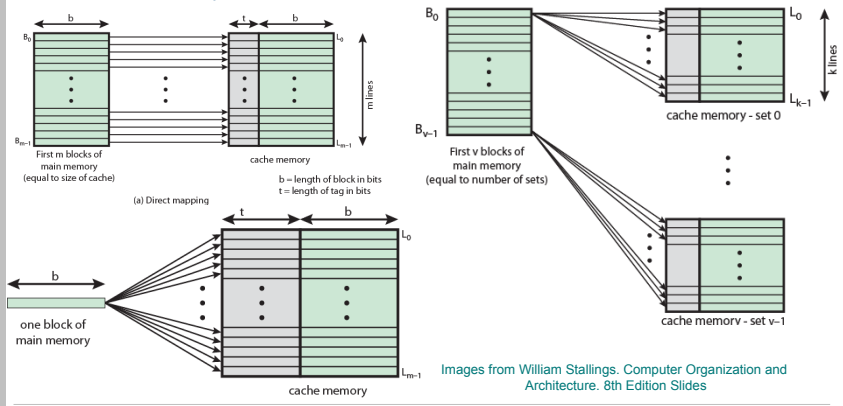


Cache Read Operation - Flowchart

Image from William Stallings. Computer Organization and Architecture. 8th Edition Slides

Cache memory (VI)
Mapping function (I)

- To map memory blocks into cache lines



Images from William Stallings. Computer Organization and Architecture. 8th Edition Slides

Cache memory (VII)
Mapping function (II)

Direct mapping

- Each main memory I block will be in cache (**i mod k)** block, where **k** is the number of cache blocks
- A cache address is compound of:
  - Tag
  - Cache block number
  - Position inside block

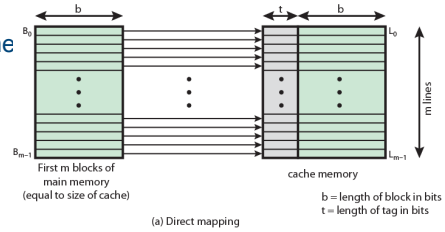


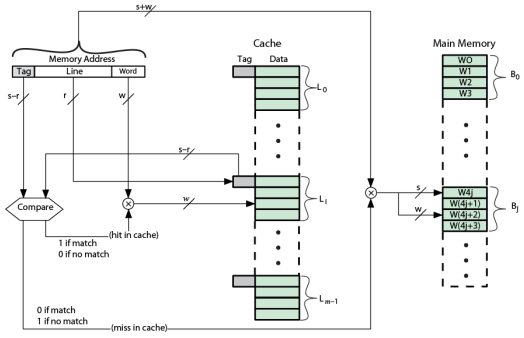
Image from William Stallings. Computer Organization and Architecture. 8th Edition Slides

Cache memory (VIII)
Mapping function (III)

Direct mapping

Advantage:
Tag and word inside block concurrent access

Drawback:
Miss ratio increases when accessing memory addresses belonging to the same cache block

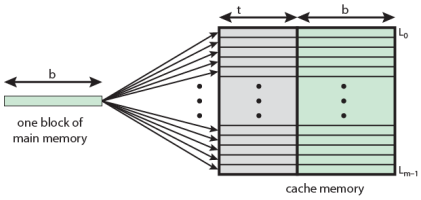


Images from William Stallings. Computer Organization and Architecture. 8th Edition Slides

Cache memory (IX)
Mapping function (IV)

Associative mapping

- Each main memory block can be allocated in whatever cache block
- A cache address is formed of:
  - Tag
  - Position inside block



Advantage:

- Flexibility in replacement policies

Drawback:

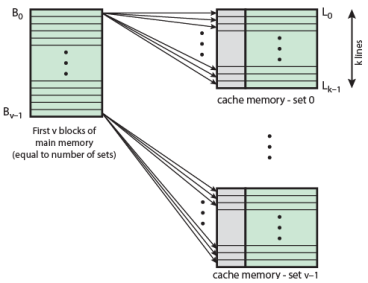
- Comparing cost

Images from William Stallings. Computer Organization and Architecture. 8th Edition Slides

Cache memory (X)
Mapping function (V)

Set-associative mapping

- Cache memory is splitted in C set of B blocks
- Following mapping function is applied:
  - Direct mapping in set level
  - Associative mapping in block level
- A i main memory block can be allocated in whatever (i mod C) block of the cache C set
- A cache address is compound of:
  - Tag
  - Set
  - Position inside block

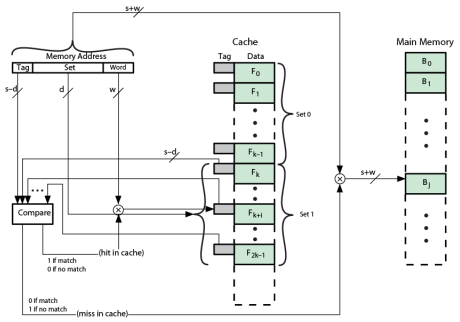


Images from William Stallings. Computer Organization and Architecture. 8th Edition Slides

Cache memory (XI)
Mapping function (VI)

Set-associative mapping

Advantage:
Decrease comparing cost and has a similar direct mapping performance



Images from William Stallings. Computer Organization and Architecture. 8th Edition Slides

Cache memory (XII)
Mapping function (VII)

- **Direct mapping example**

Main memory: 1Mbyte. Cache memory 1Kbyte
Block size: 8 bytes

$$\text{Number of blocks of main memory} = \frac{\text{Main memory}}{\text{Block size}} = \frac{2^{20}}{2^3} = 2^{17}\ \text{blocks}$$

$$\text{Number of blocks of cache memory} = \frac{\text{Cache Memory capacity}}{\text{Block size}} = \frac{2^{10}}{2^3} = 2^7\ \text{blocks}$$

$$\frac{2^{17}}{2^7} = 2^{10}\ \text{main memory blocks per cache block}$$

10	7	3
Tag	Block	Word

Cache memory (XIII)
Mapping function (VIII)

- **Associative mapping**

Main memory: 1Mbyte. Cache memory 1Kbyte
Block size: 8 bytes

$$\text{Number of main memory blocks} = \frac{\text{Main Memory capacity}}{\text{Block size}} = \frac{2^{20}}{2^3} = 2^{17}\ \text{blocks}$$

17	3
Tag	Word

Cache memory (XIV)
Mapping function (and IX)

- **Set-associative mapping**

Main memory: 1Mbyte. Cache memory 1Kbyte
Block size: 8 bytes. Set size: 2 blocks

$$\text{Number of main memory blocks} = \frac{\text{Main Memory capacity}}{\text{Block size}} = \frac{2^{20}}{2^3} = 2^{17}\ \text{blocks}$$

$$\text{Number of cache blocks} = \frac{\text{Cache Memory capacity}}{\text{Block size}} = \frac{2^{10}}{2^3} = 2^7\ \text{blocks}$$

$$\text{Number of cache sets} = \frac{\text{Number of cache blocks}}{\text{Set size}} = \frac{2^7}{2^1} = 2^6\ \text{sets}$$

$$\frac{2^{17}}{2^6} = 2^{11}\ \text{main memory blocks per cache set}$$

11	6	3
Tag	Set	Word

Cache memory (XV)
Replacement policy

- Specifies cache block to be replaced when cache memory is full.
- Las más utilizadas son:
  - **Random**
  - **LRU Least-Recently Used:** replace block which has had fewest hits
  - **FIFO - First In First Out:** replace block that has been in cache longest
- Direct mapping hasn't replacement policy

Cache memory (XVI)
Writes policy (I)

- Specifies how and when main memory update occures when writting on cache

Write through: All writes go to main memory as well as cache

Advantages: easy implementation

Drawbacks: Lots of traffic and slows down writes

Copy back: Updates initially made in cache only. If block is to be replaced, write to main memory only if update bit is set

Advantages: high speed writting. No much traffic
Drawbacks: Update bit for cache slot is set when update occurs → more complexity in implementation

Cache memory (XVII)
Writes policy (and II)

- How to act when a write failure occurs:
- **With allocation:** gather searched block from main memory and write on cache only.
- Related to copy back
- **Without allocation:** write on main memory only.
- Related to write through

Cache memory (XVIII)
More ideas to be taken into account (I)

Tamaño de la memoria cache y sus bloques

- Increased block size will increase hit ratio at first (principleo of locality) Hit ratio will decreases as block becomes bigger.
- Larger blocks reduce the number of blocks that fit in cache.
- From 8 to 64 bytes is a reasonable size

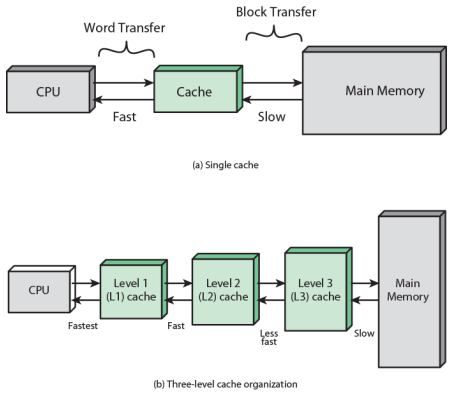
Unified or split caches

- One cache only for data and instructions or two different caches, one for data and one for instructions

Cache de uno y dos niveles

- High logic density enables caches on chip
- Common to use both on and off chip cache
- L1 ob chip, L2 off chip

Cache memory (and XIX)
More ideas to be taken into account (&II)



Images from William Stallings. Computer Organization and Architecture. 8th Edition Slides

Cache Memory System Examples
Pentium Pro and PowerPC 604

Characteristic	Pentium Pro	PowerPC 604
Cache organization	Split caches	Split caches
Cache size	8 KB both caches	16 KB both caches
Associative	4 set-associative	4 set-associative
Replacement	LRU	LRU
Block size	32 bytes	32 bytes
Writes policy	Copy Back	Copy Back and Write through