



INSTITUTO  
SUPERIOR  
TÉCNICO

---

# *Arquitectura de Computadores*

## *Unidade de Memória*

**Nuno Cavaco Gomes Horta**

**Universidade Técnica de Lisboa / Instituto Superior Técnico**



INSTITUTO  
SUPERIOR  
TÉCNICO

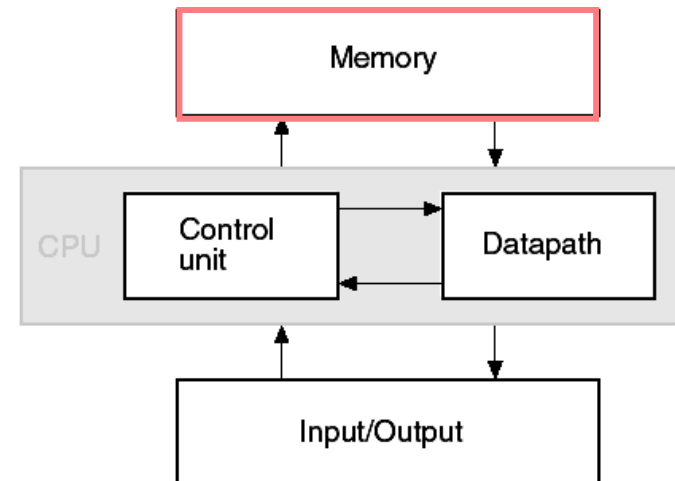
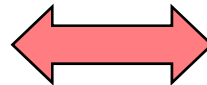
## Sumário

- *Introdução*
- *Unidade de Processamento*
- *Unidade de Controlo*
- *Conjunto de Instruções*
- *Unidade Central de Processamento (CPU)*
- *Unidade de Entrada/Saída (I/O)*
- *Unidade de Memória*

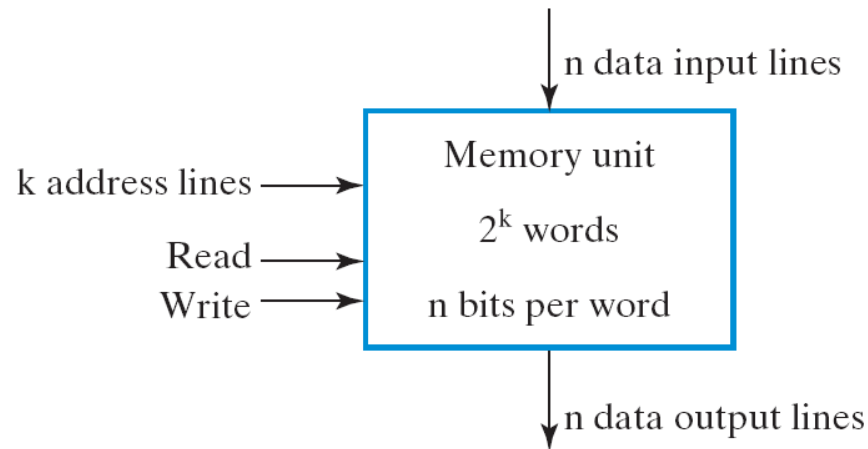


INSTITUTO  
SUPERIOR  
TÉCNICO

### Arquitetura Genérica de um Computador



- **Memória de acesso aleatório**

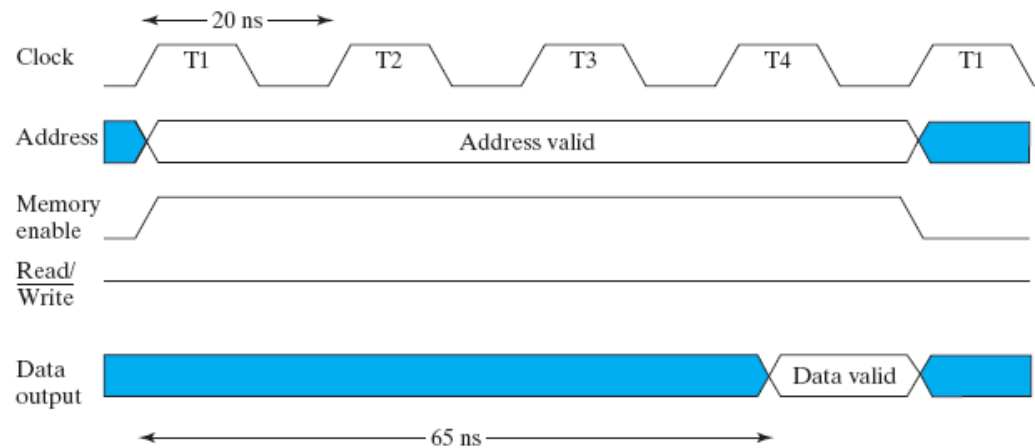
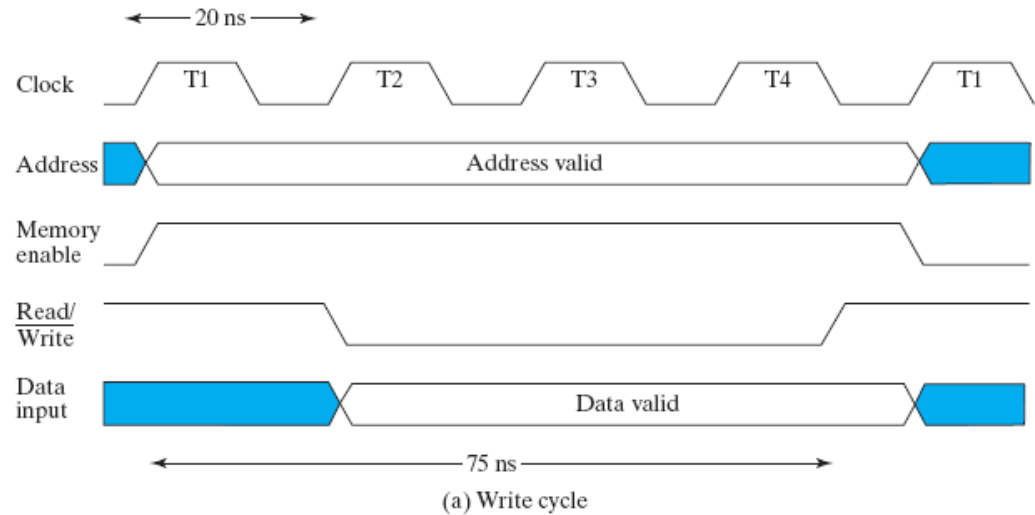


- Consistem numa série de  $2^k$  palavras
- Cada palavras representa um conjunto de bits que é acedido em conjunto
- A cada palavra corresponde um endereço especificado pelas k linhas de endereço da memoria

# Arquitectura de Computadores

## Unidade de Memória

Os tempos de acesso á memória tem de ser arredondados para múltiplos do período de relógio



### *Hierarquia de Memória: Tipos de Memória, RAM – Random Access Memory*

#### **SRAM – Static RAM:**

- Conteúdo de memória mantido desde que o circuito esteja alimentado.
- Memória rápida utilizada para Memória Cache.

#### **DRAM – Dynamic RAM:**

- Conteúdo de memória mantido através de refrescamento.
- Estrutura de array de células com transistor e condensador.
- Grande capacidade mas velocidade menor que SRAM, utilizada para Memória Principal.

#### **SDRAM – Synchronous DRAM:**

- SRAM e DRAM são assíncronas, na DRAM os tempos de operação têm de ser definidos pelo utilizador
- Lê linhas completas da memória e guarda-as numa pequena cache interna (precharge) de forma que os acessos á mesma linha podem ser muito rápidos
- Utiliza vários bancos de memória e uma arquitectura pipeline para permitir esconder o tempo de pre-charge. O endereço da nova linha é fornecido com alguma antecedência ainda durante a leitura da linha anterior.

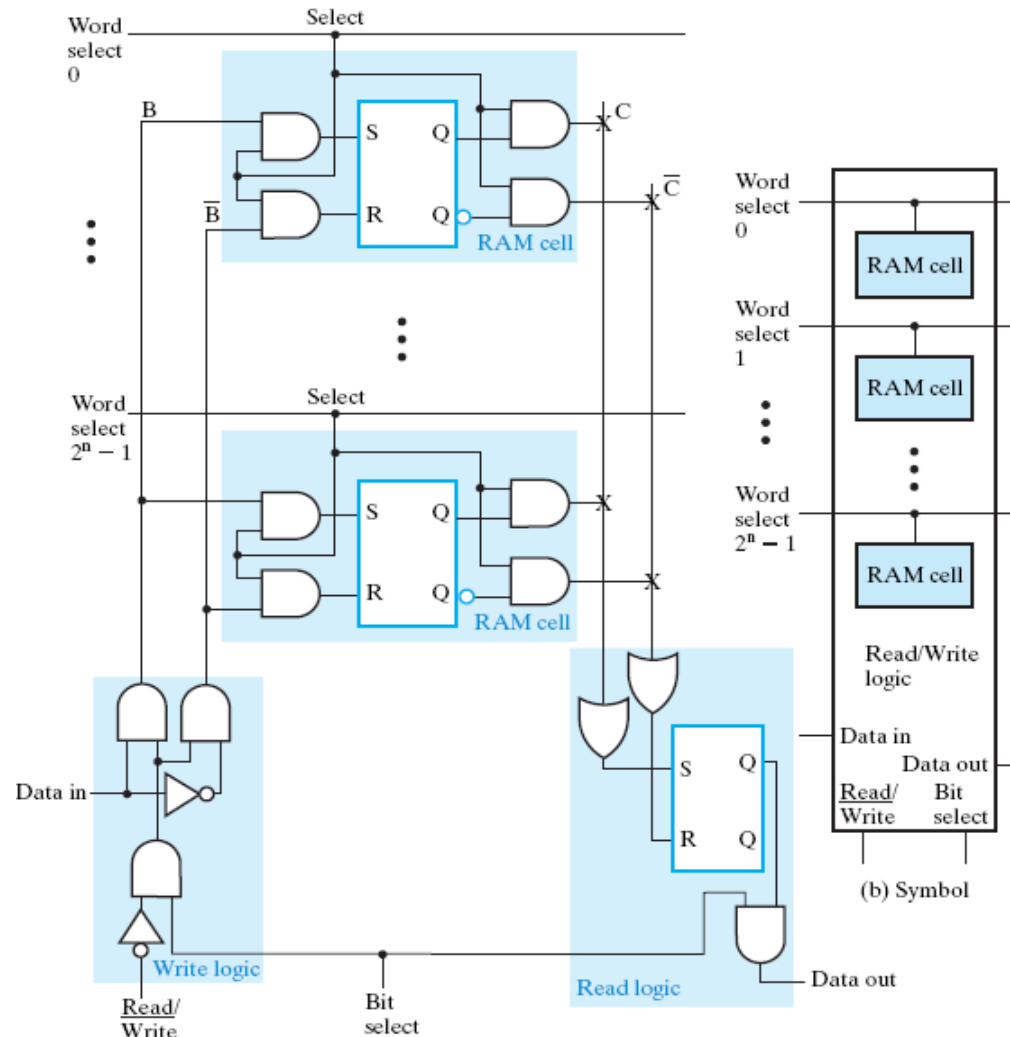


### SRAM chips

#### Bit Slice Model

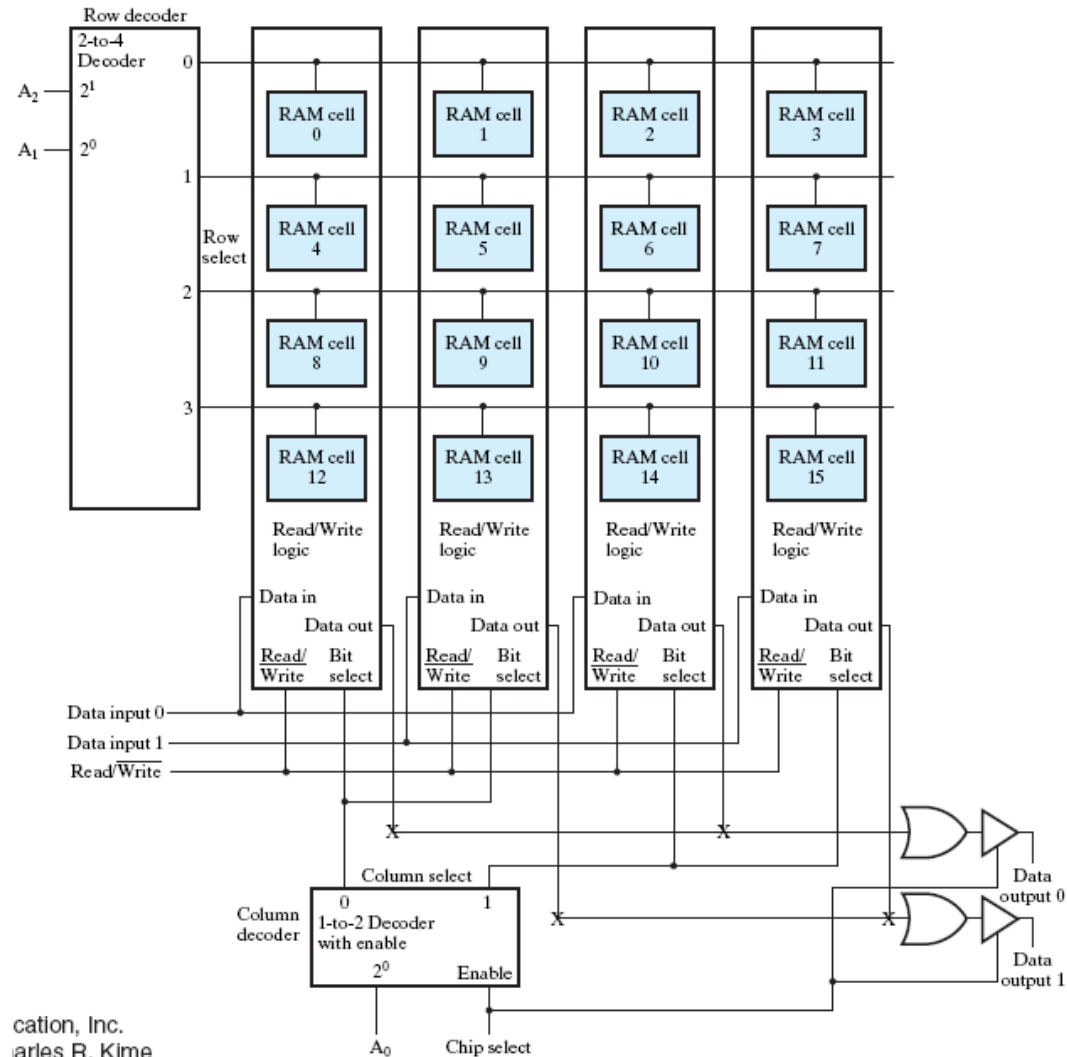
A leitura ou escrita de uma palavra consiste em aceder a  $n$  bit slices simultaneamente.

- Notar que estes são desenhados como um circuito electrónico, mas podem ser modelados usando um modelo lógico



# Arquitetura de Computadores

## Unidade de Memória

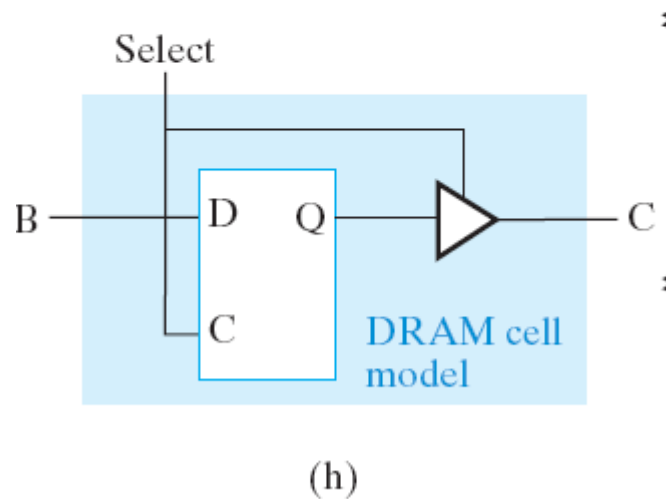
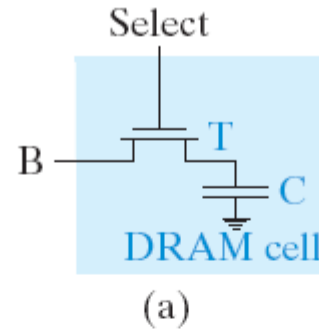


ation, Inc.  
arles R. Kime



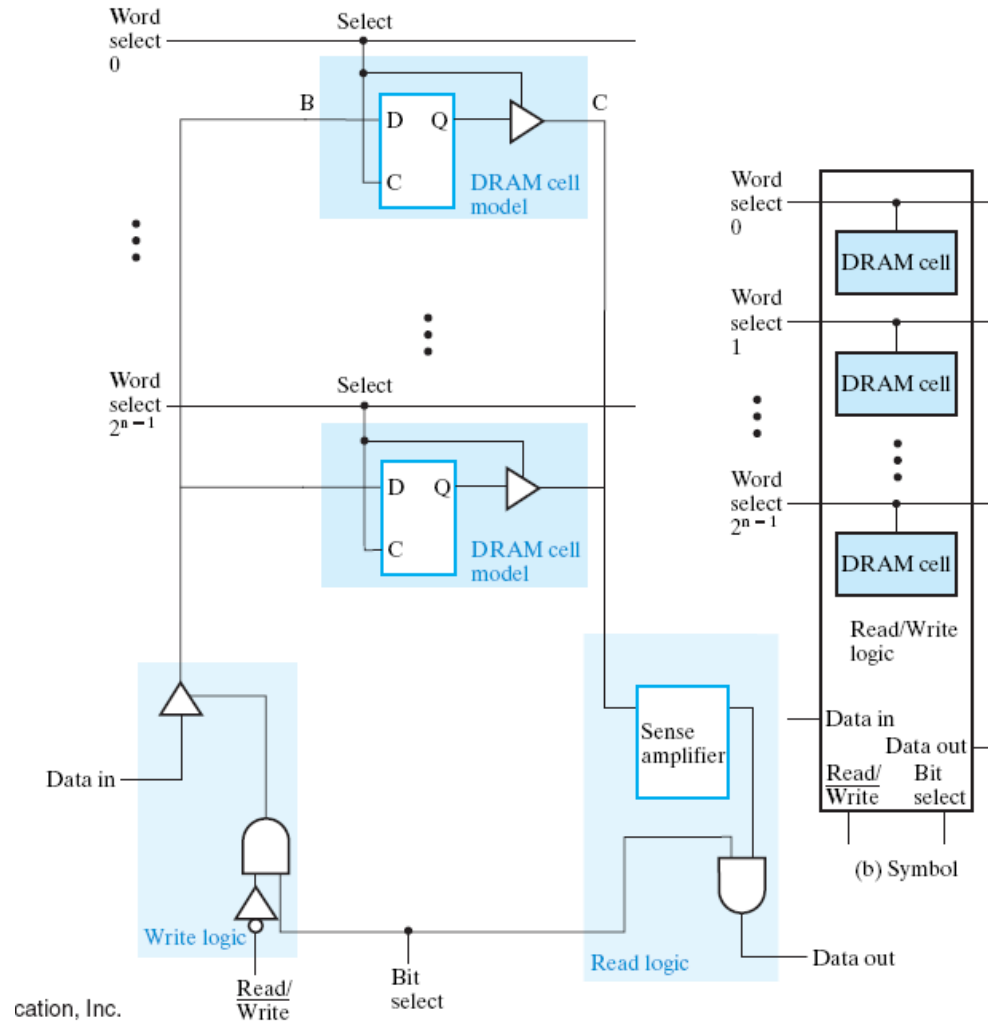
# Arquitetura de Computadores

## Unidade de Memória



# Arquitetura de Computadores

## Unidade de Memória



# Arquitetura de Computadores

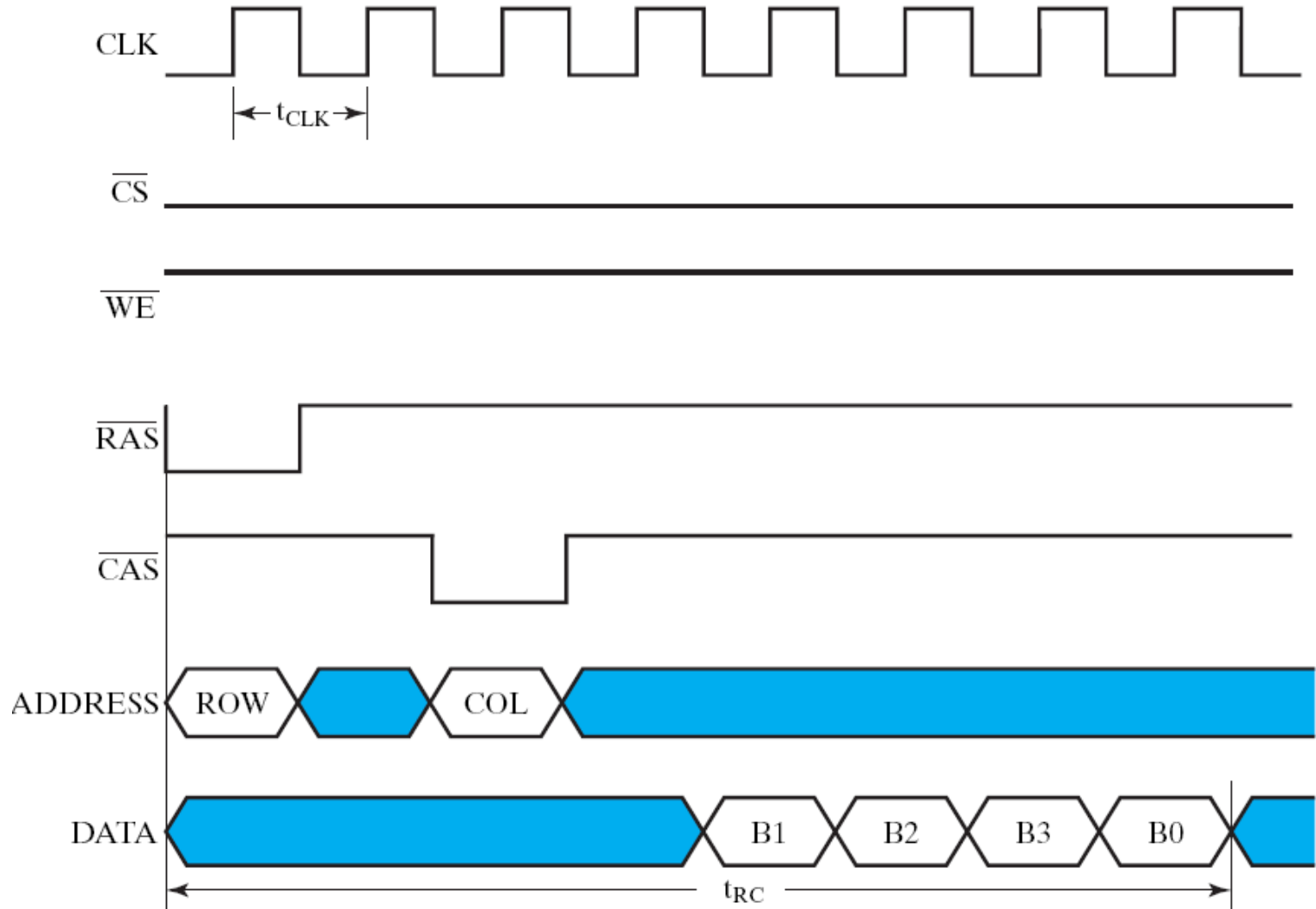
## Unidade de Memória

Type	Abbreviation	Description
Fast Page Mode DRAM	FPM DRAM	Takes advantage of the fact that when a row is accessed all of the values are available to be read out. By changing the column address, data from different addresses can be read out without reapplying the row address and waiting for the delay associated with reading the cells to pass if the row portion of the address match.
Extended Data Output DRAM	EDO DRAM	Extends the length of time that the DRAM holds the data values on its output, permitting the CPU to perform other tasks during the access since it knows the data will still be available.
Synchronous DRAM	SDRAM	Operates with a clock rather than being asynchronous. This permits a tighter interaction between memory and the CPU, since the CPU knows exactly when the data will be available. SDRAM also takes advantage of the row value availability and divides memory into distinct banks permitting overlapped accesses.
Rambus DRAM	RDRAM	A proprietary technology that provides very high memory access rates using a relatively narrow bus.
Error-Correcting Code	ECC	May be applied to most of the DRAM types above to correct bit errors.

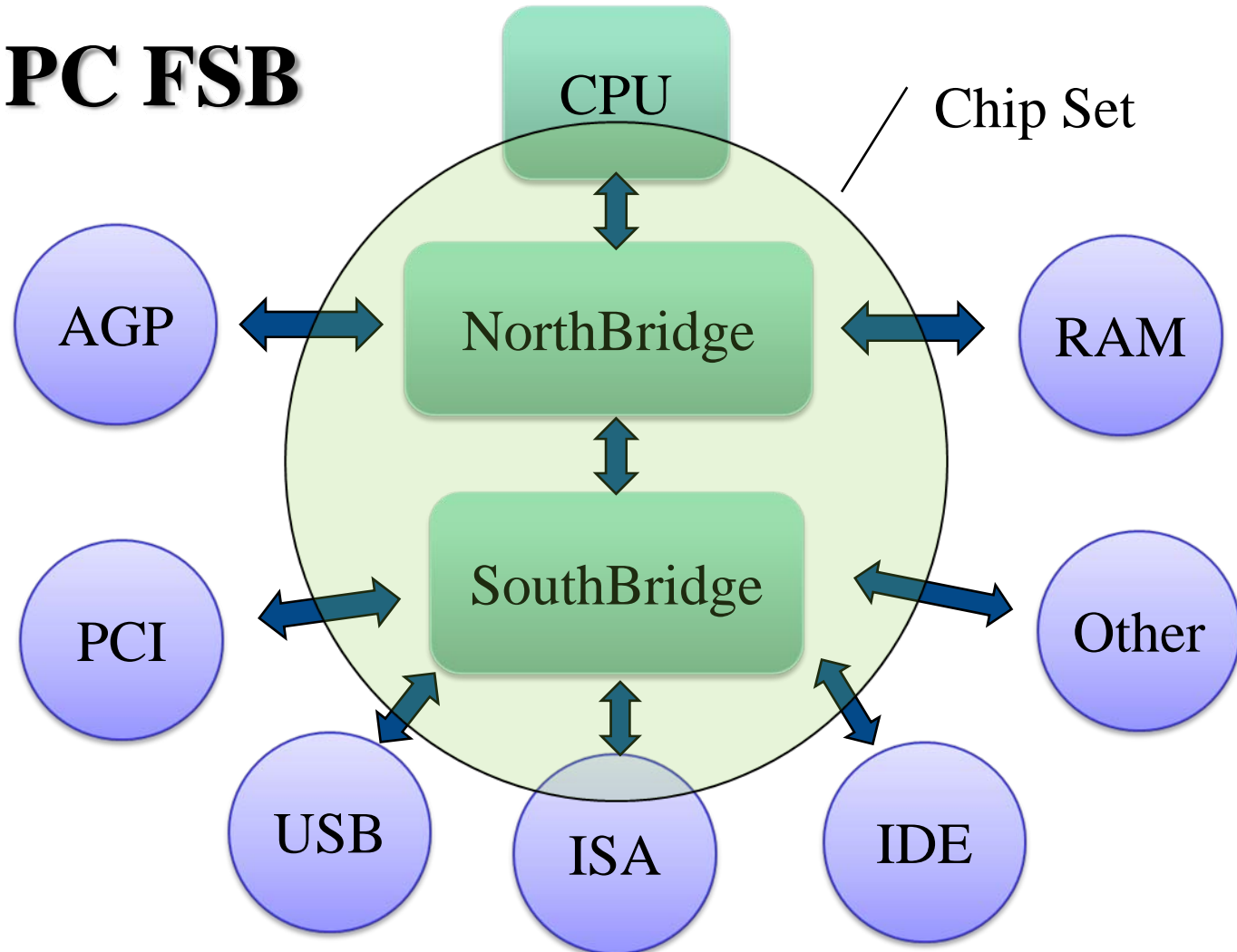


# Arquitetura de Computadores

## Unidade de Memória



### PC FSB



### *Hierarquia de Memória: Tipos de Memória*

#### *ROM – Read Only Memory*

*PROM – Programmable ROM*

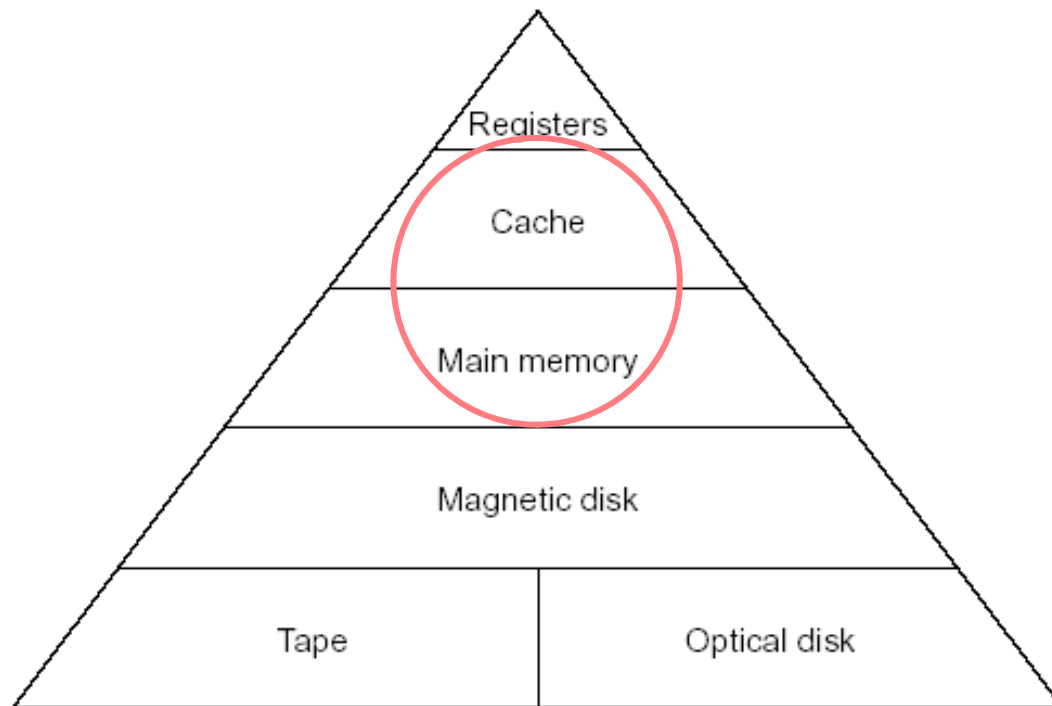
*EPROM – Erasable PROM*

*EEPROM – Electrical Erasable PROM*

*FLASH – EEPROM (apagamento de blocos em vez de bytes)*

Type	Category	Eraseure	Byte alterable	Volatile	Typical use
SRAM	Read/write	Electrical	Yes	Yes	Level 2 cache
DRAM	Read/write	Electrical	Yes	Yes	Main memory
ROM	Read-only	Not possible	No	No	Large volume appliances
PROM	Read-only	Not possible	No	No	Small volume equipment
EPROM	Read-mostly	UV light	No	No	Device prototyping
EEPROM	Read-mostly	Electrical	Yes	No	Device prototyping
Flash	Read/write	Electrical	No	No	Film for digital camera

### Memória Cache



- Memórias típicas (2006)
  - 4GB
  - 10ns de tempo de acesso
- Requisitos de um processador RISC
  - Memória separada para dados e programa
  - Tempos de acesso de 1ns
- Solução: caches



### Hierarquia de Memória: Estrutura

#### Memória Cache:

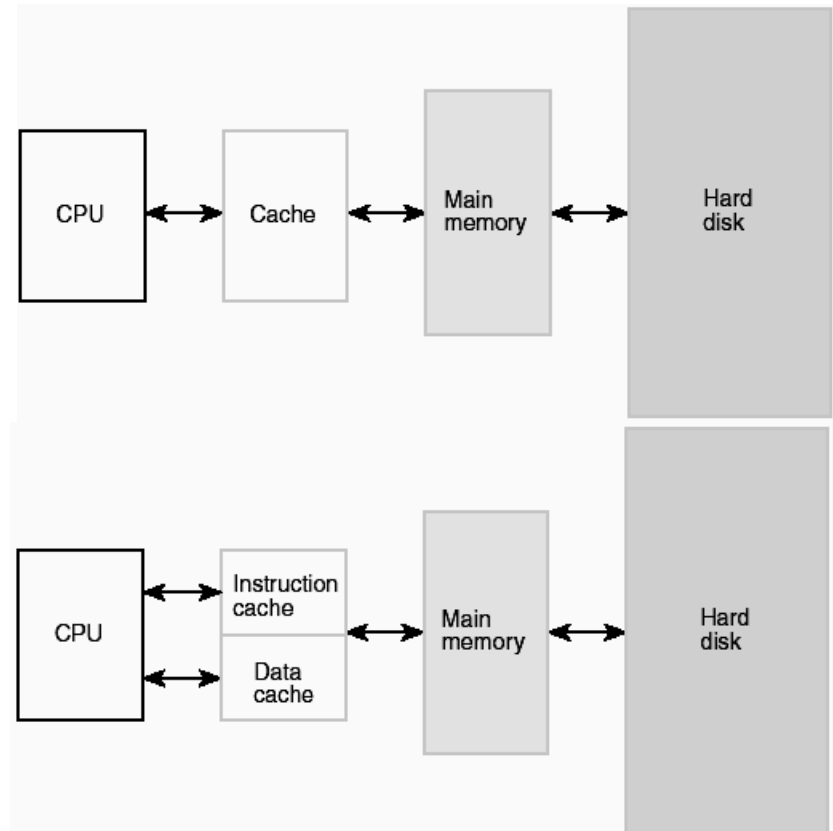
*Acessos rápidos,  
Custo elevado.*

#### Memória Principal:

*Tempos de Acesso Intermédios,  
Custo médio.*

#### Memória Secundária:

*Acessos lentos,  
Custo reduzido.*



*Nota: Aquisição das Instruções e Operandos deve ser realizada na maioria dos casos da Cache, uma percentagem reduzida da memória principal e só ocasionalmente da memória secundária (disco rígido).*

- **Com caches**

- com tempos de acesso de 2ns
- hit rate de 95%
- Tempo médio de acesso é de
  - $0.92 \times 2 + 0.05 \times 10 = \mathbf{2.4ns}$

- **Com memoria virtual**

- 4.999995% da memória principal
- $50 \times 10^{-9}$  de acessos ao disco
- $13ms = 1.3 \times 10^{-7}ns$ 
  - Tempo médio de acesso
    - $0.95 \times 2 + 0.04999995 \times 10 + 50 \times 10^{-9} \times 1.3 \times 10^{-7} = \mathbf{3.05ns}$

- **Localidade temporal e espacial**
- Localidade temporal
  - Relativa aos tempos em que instruções e operandos são acedidos
- Localidade espacial
  - Relativa as posições me memória em que instruções e operandos são acedidos
- **Um programa típico associa localidade espacial com localidade temporal**
- Exemplo: um ciclo de 8 instruções que é executado 100 vezes, acede apenas a 8 posições na memória de programa
  - Os primeiros acessos são efectuados à memoria enquanto que os últimos acessos provêm da cache mais rápida.



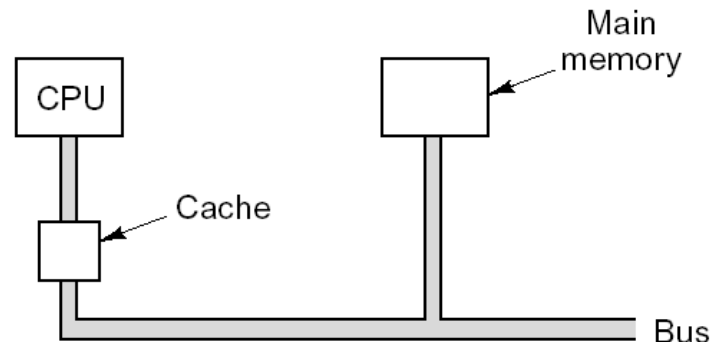
- **Localidade temporal e espacial**
- Exemplos
  - Memória de programa
    - Um cliço de 8 instruções que é executado 100 vezes faz 800 acessos a memória de programa mas só acede a 8 posições distintas
  - Memória de dados
    - O processamento de um vector de números faz múltiplos acessos a esse vector
    - Os endereços seguintes aos acedidos são mais prováveis de ser acedidos que os restantes, durante o progresso do cálculo



- **Localidade temporal e espacial**
- A memória física funciona como uma cache para acessos ao disco
- Localidade espacial e temporal significa que depois um primeiro acesso ao disco os dados serão copiados para memória, e os acessos seguintes passam a poder ser efectuados à memória principal ou à cache.

### *Hierarquia de Memória: Memória Cache*

*A Memória Cache é implementada com memórias rápidas e utilizada pela CPU para armazenar as palavras de memória utilizadas com maior frequência com o objectivo de tornar mais rápido todo o processamento.*



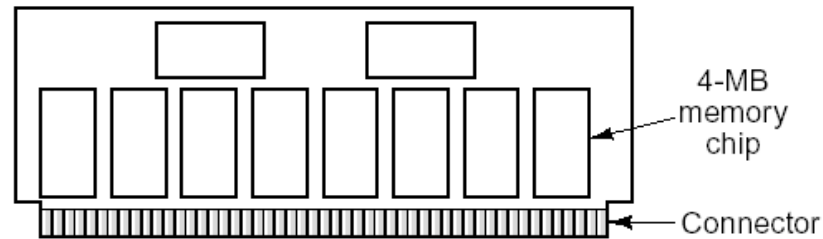
# Arquitetura de Computadores

## Unidade de Memória

### *Hierarquia de Memória: Memória Principal*

*SIMM – Single Inline Memory Module*

*DIMM – Dual Inline Memory Module*



**SIMM de 32 MB e 2  
Circuitos de Controlo**

# Arquitectura de Computadores

## Unidade de Memória

### Memória Cache:

**Exemplo:** Na execução da subrotina apresentada:

(a) Determine a “Hit Rate”, considerando  $DELAYVALUE = 50$ .

(b) Determine o tempo de acesso efectivo à memória com e sem Cache, assumindo que a mem. Cache tem um tempo de acesso de 8ns e a mem. Principal de 85ns.

```
0029 Delay:          PUSH    R1
002A                MOV     R1, DELAYVALUE
002C DelayLoop:     DEC     R1
002D                BR.NZ  DelayLoop
002E                POP    R1
002F                RET
```

### Solução:

(a) 94,2 %



INSTITUTO  
SUPERIOR  
TÉCNICO





### Memória Cache: Mapeamento Directo

#### Exemplo:

Palavras de 32 Bits

Mem. Cache: 8 Palavras

Mem. Principal: 1KB

(256 Palavras)

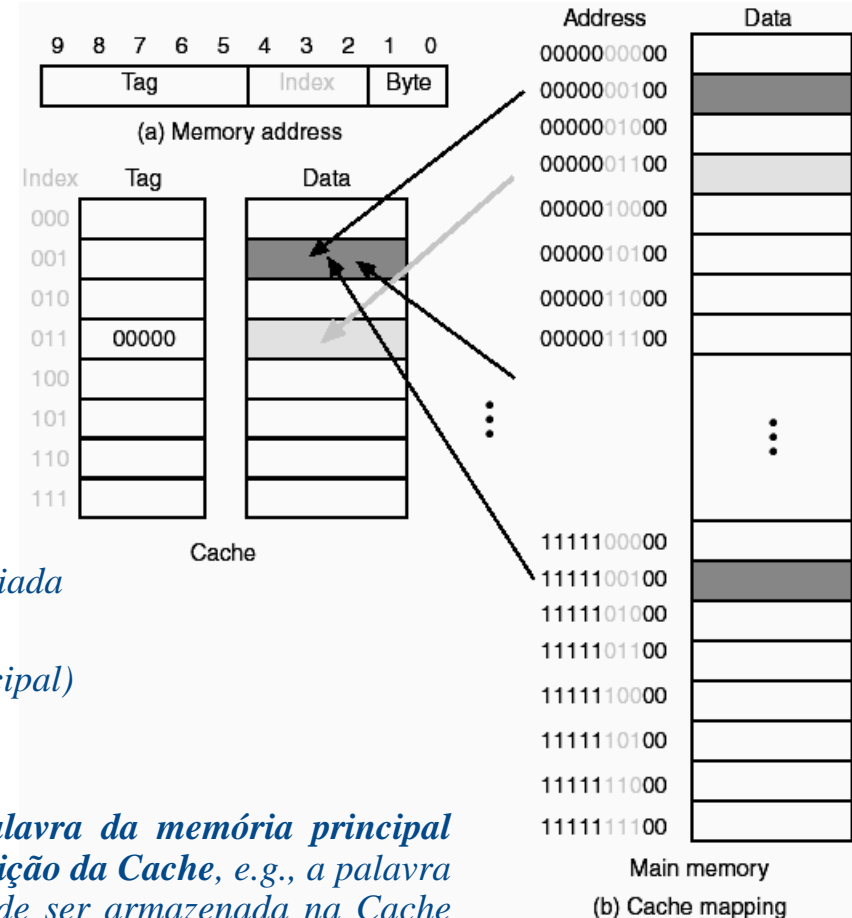
Endereço de Memória: 10 bits

Endereço da Cache: 3 bits

Etiqueta na Cache: 5 bits

A etiqueta (Tag), guardada na Cache, associada ao endereço na Cache, permite determinar a localização da palavra em memória (principal)

**Mapeamento Directo** – Uma qualquer palavra da memória principal pode estar contida **em apenas uma posição da Cache**, e.g., a palavra com endereço **1010100100** apenas pode ser armazenada na Cache na posição **001**.



# Arquitectura de Computadores

## Unidade de Memória

### Memória Cache: Mapeamento Directo

**Exemplo:** Para a sequência de endereços apresentada determine a ocorrência de “miss”s e “hit”s no acesso à memória Cache.

#### ENDEREÇO DE MEM.

9 8 7 6 5 4 3 2 1 0

ETIQUETA	INDEX	BYTE
----------	-------	------

HEX	Binário	MD
54	00 0101 0100	M
58	00 0101 1000	M
104	01 0000 0100	M
5C	00 0101 1100	M
108	01 0000 1000	M
60	00 0110 0000	M
F0	00 1111 0000	M
64	00 0110 0100	M
54	00 0101 0100	H
58	00 0101 1000	H
10C	01 0000 1100	M
5C	00 0101 1100	H
110	01 0001 0000	M
60	00 0110 0000	H
F0	00 1111 0000	M
64	00 0110 0100	H

INDEX	ETIQUETA	DATA
000		
001		
010		
011		
100		
101		
110		
111		

M. CACHE



INSTITUTO  
SUPERIOR  
TÉCNICO

# Arquitectura de Computadores

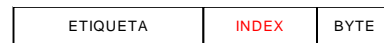
## Unidade de Memória

### Memória Cache: Mapeamento Directo

Exemplo:

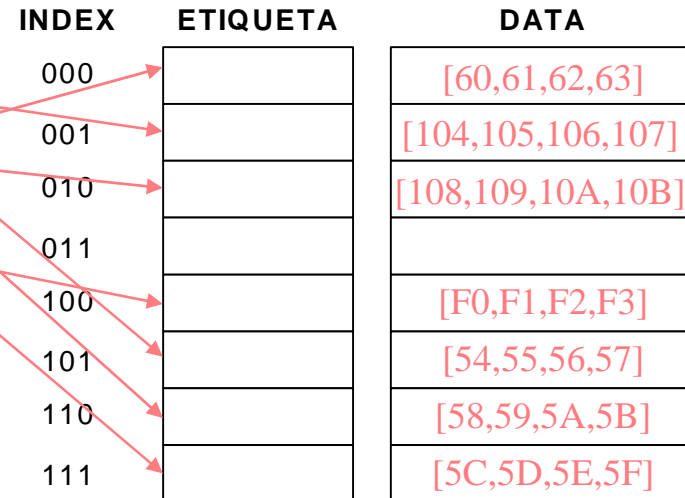
ENDEREÇO DE MEM.

9 8 7 6 5 4 3 2 1 0



- “MISS” - 1ª Escrita
- “MISS” – Escrita/Substituição
- “HIT” – Leitura da Cache

HEX	Binário	MD
54	00 0101 0100	M
58	00 0101 1000	M
104	01 0000 0100	M
5C	00 0101 1100	M
108	01 0000 1000	M
60	00 0110 0000	M
F0	00 1111 0000	M
64	00 0110 0100	M
54	00 0101 0100	H
58	00 0101 1000	H
10C	01 0000 1100	M
5C	00 0101 1100	H
110	01 0001 0000	M
60	00 0110 0000	H
F0	00 1111 0000	M
64	00 0110 0100	H



M. CACHE



INSTITUTO  
SUPERIOR  
TÉCNICO

# Arquitectura de Computadores

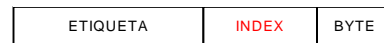
## Unidade de Memória

### Memória Cache: Mapeamento Directo

Exemplo:

ENDEREÇO DE MEM.

9 8 7 6 5 4 3 2 1 0



- “MISS” - 1ª Escrita
- “MISS” – Escrita/Substituição
- “HIT” – Leitura da Cache

HEX	Binário	MD
54	00 0101 0100	M
58	00 0101 1000	M
104	01 0000 0100	M
5C	00 0101 1100	M
108	01 0000 1000	M
60	00 0110 0000	M
F0	00 1111 0000	M
64	00 0110 0100	M
54	00 0101 0100	H
58	00 0101 1000	H
10C	01 0000 1100	M
5C	00 0101 1100	H
110	01 0001 0000	M
60	00 0110 0000	H
F0	00 1111 0000	M
64	00 0110 0100	H

INDEX	ETIQUETA	DATA
000		[60,61,62,63]
001		[64,65,66,67]
010		[108,109,10A,10B]
011		
100		[F0,F1,F2,F3]
101		[54,55,56,57]
110		[58,59,5A,5B]
111		[5C,5D,5E,5F]

M. CACHE

# Arquitectura de Computadores

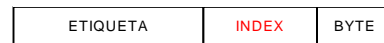
## Unidade de Memória

### Memória Cache: Mapeamento Directo

Exemplo:

ENDEREÇO DE MEM.

9 8 7 6 5 4 3 2 1 0



- “MISS” - 1ª Escrita
- “MISS” – Escrita/Substituição
- “HIT” – Leitura da Cache

HEX	Binário	MD
54	00 0101 0100	M
58	00 0101 1000	M
104	01 0000 0100	M
5C	00 0101 1100	M
108	01 0000 1000	M
60	00 0110 0000	M
F0	00 1111 0000	M
64	00 0110 0100	M
54	00 0101 0100	H
58	00 0101 1000	H
10C	01 0000 1100	M
5C	00 0101 1100	H
110	01 0001 0000	M
60	00 0110 0000	H
F0	00 1111 0000	M
64	00 0110 0100	H

INDEX	ETIQUETA	DATA
000		[60,61,62,63]
001		[64,65,66,67]
010		[108,109,10A,10B]
011		
100		[F0,F1,F2,F3]
101		[54,55,56,57]
110		[58,59,5A,5B]
111		[5C,5D,5E,5F]

M. CACHE



INSTITUTO  
SUPERIOR  
TÉCNICO

# Arquitectura de Computadores

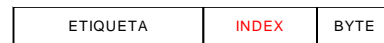
## Unidade de Memória

### Memória Cache: Mapeamento Directo

Exemplo:

ENDEREÇO DE MEM.

9 8 7 6 5 4 3 2 1 0



- “MISS” - 1ª Escrita
- “MISS” – Escrita/Substituição
- “HIT” – Leitura da Cache

HEX	Binário	MD
54	00 0101 0100	M
58	00 0101 1000	M
104	01 0000 0100	M
5C	00 0101 1100	M
108	01 0000 1000	M
60	00 0110 0000	M
F0	00 1111 0000	M
64	00 0110 0100	M
54	00 0101 0100	H
58	00 0101 1000	H
10C	01 0000 1100	M
5C	00 0101 1100	H
110	01 0001 0000	M
60	00 0110 0000	H
F0	00 1111 0000	M
64	00 0110 0100	H

INDEX	ETIQUETA	DATA
000		[60,61,62,63]
001		[64,65,66,67]
010		[108,109,10A,10B]
011		[10C,10D,10E,10F]
100		[F0,F1,F2,F3]
101		[54,55,56,57]
110		[58,59,5A,5B]
111		[5C,5D,5E,5F]

M. CACHE



INSTITUTO SUPERIOR TÉCNICO



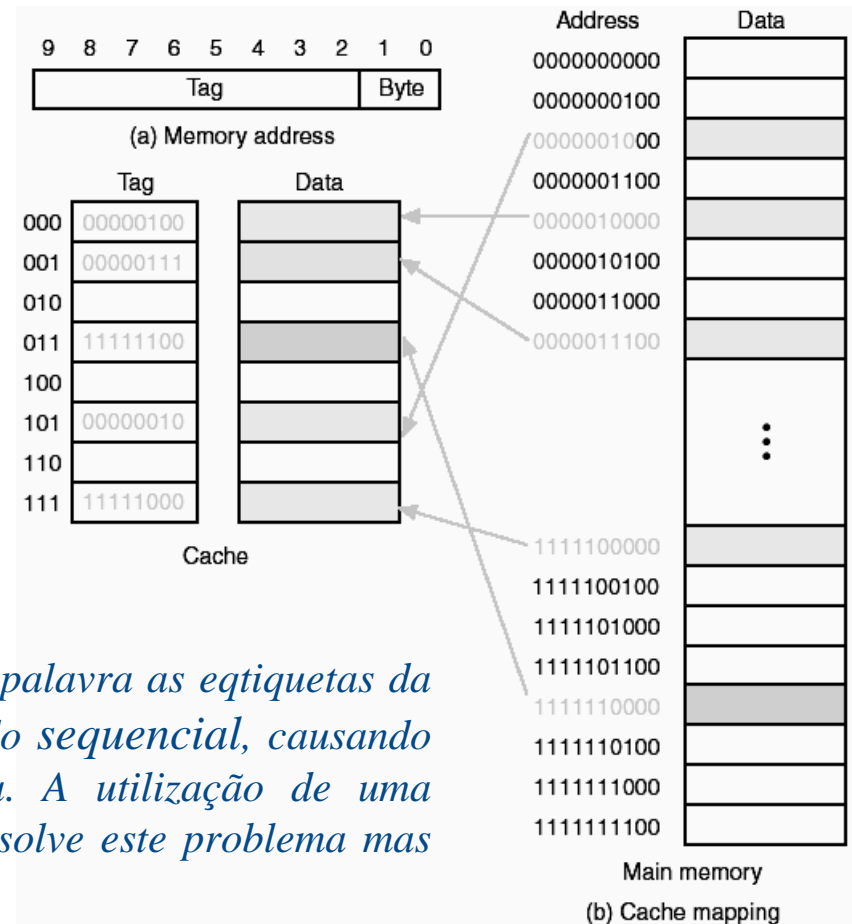
### Memória Cache: Mapeamento Associativo

#### Mapeamento Associativo –

*Uma qualquer palavra da memória principal pode estar contida em qualquer posição da Cache.*

*Neste caso a etiqueta (Tag) deve ser de 8 em vez de 5 bits.*

*Caso seja necessário localizar uma palavra as etiquetas da Cache são pesquisadas em modo sequencial, causando uma degradação da eficiência. A utilização de uma memória associativa permite resolver este problema mas aumenta muito o consumo.*



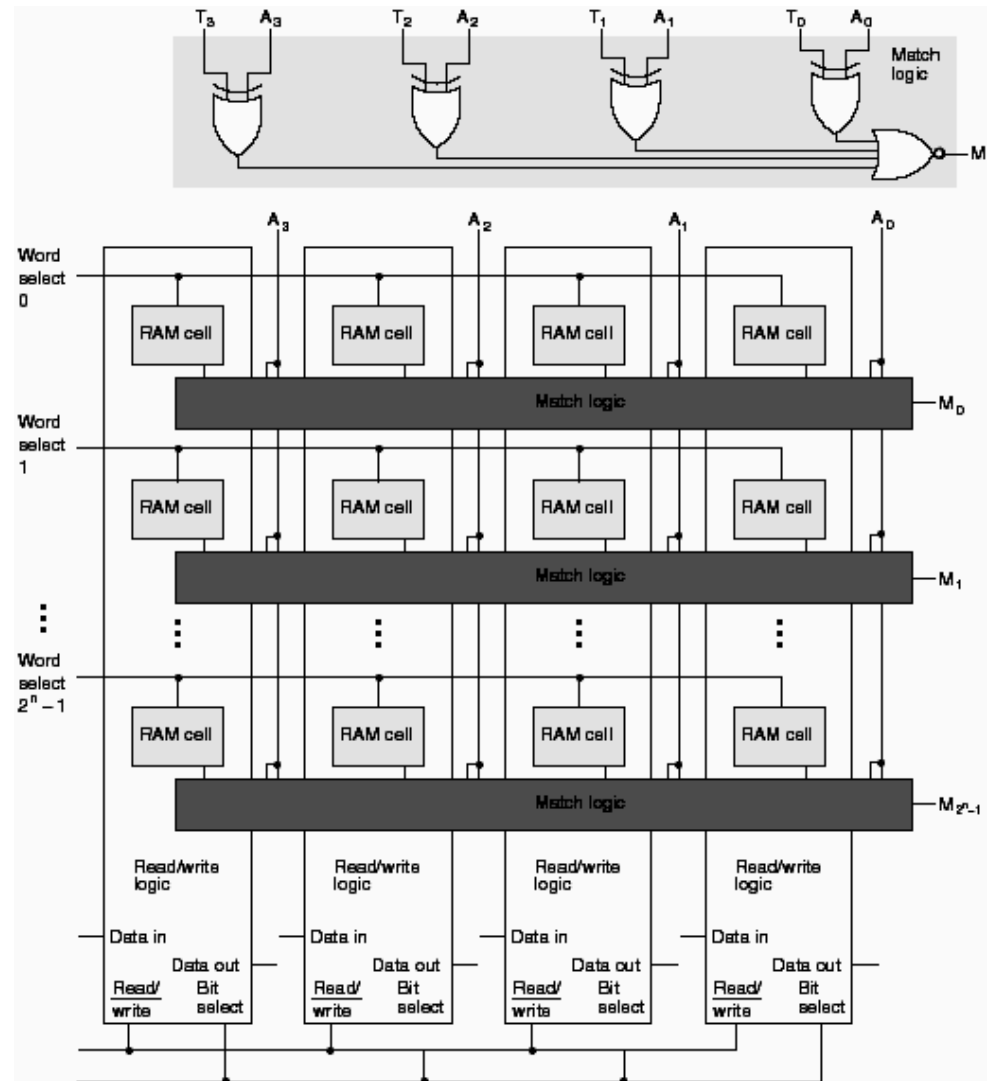
# Arquitectura de Computadores

## Unidade de Memória

*Memória Cache: Memória Associativa p/ Etiquetas de 4 bits (TAG Memory)*

*Política de Substituição das TAGs e Dados na Cache:*

- (1) *Aleatória*
- (2) *FIFO – First In First Out*  
*(basta ter um índice que percorre a memória)*
- (3) *LRU – Least Recent Used*
- ... *(soluções simplificadas, por exemplo com um used bit)*





### Memória Cache: Mapeamento Associativo

**Exemplo:** Para a sequência de endereços apresentada determine a ocorrência de “miss”s e “hit”s no acesso à memória Cache.

ENDEREÇO DE MEM.

9 8 7 6 5 4 3 2 1 0



HEX	Binário	MD
54	00 0101 0100	M
58	00 0101 1000	M
104	01 0000 0100	M
5C	00 0101 1100	M
108	01 0000 1000	M
60	00 0110 0000	M
F0	00 1111 0000	M
64	00 0110 0100	M
54	00 0101 0100	H
58	00 0101 1000	H
10C	01 0000 1100	M
5C	00 0101 1100	H
110	01 0001 0000	M
60	00 0110 0000	H
F0	00 1111 0000	H
64	00 0110 0100	H

INDEX	ETIQUETA	DATA
000		
001		
010		
011		
100		
101		
110		
111		

M. CACHE

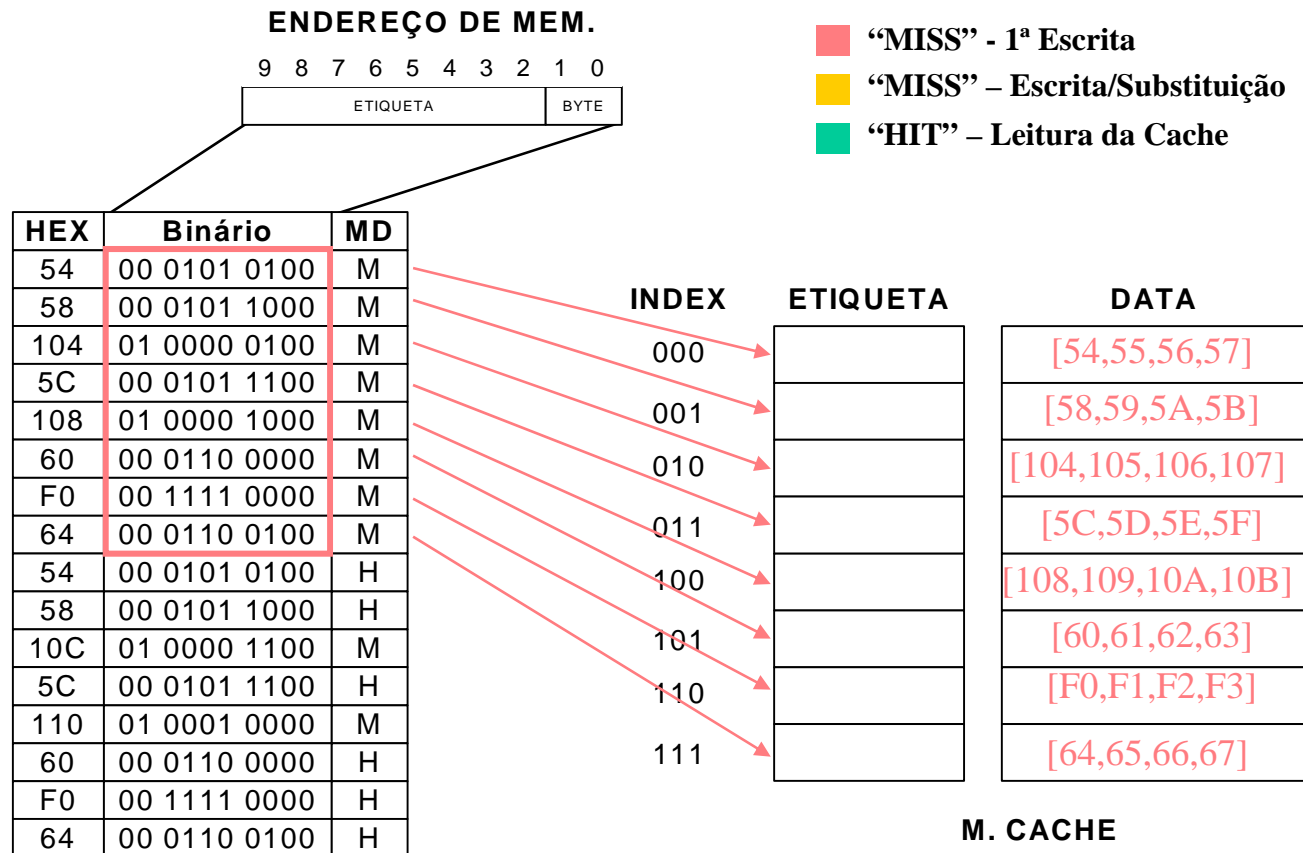


# Arquitectura de Computadores

## Unidade de Memória

### Memória Cache: Mapeamento Associativo

*Exemplo:*

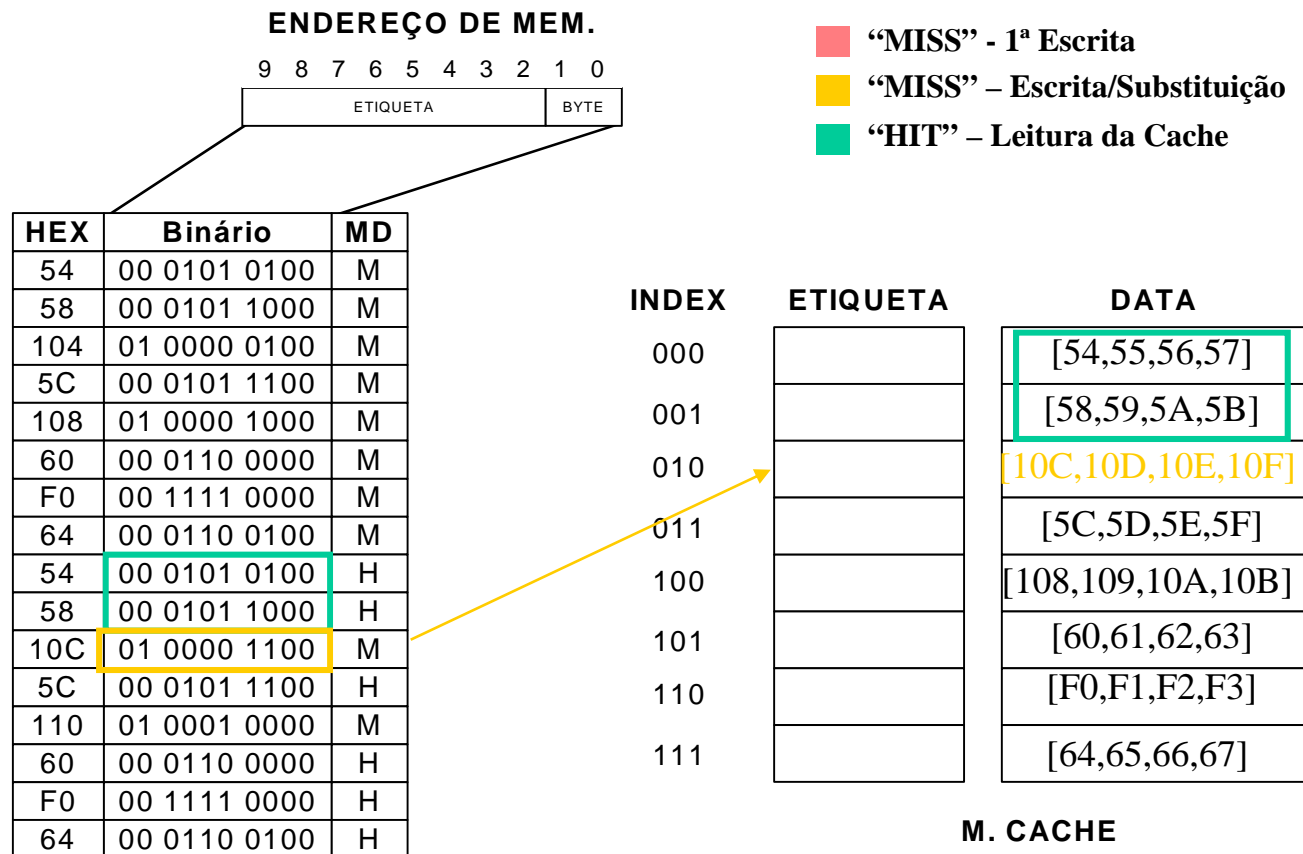


# Arquitetura de Computadores

## Unidade de Memória

### Memória Cache: Mapeamento Associativo

Exemplo:

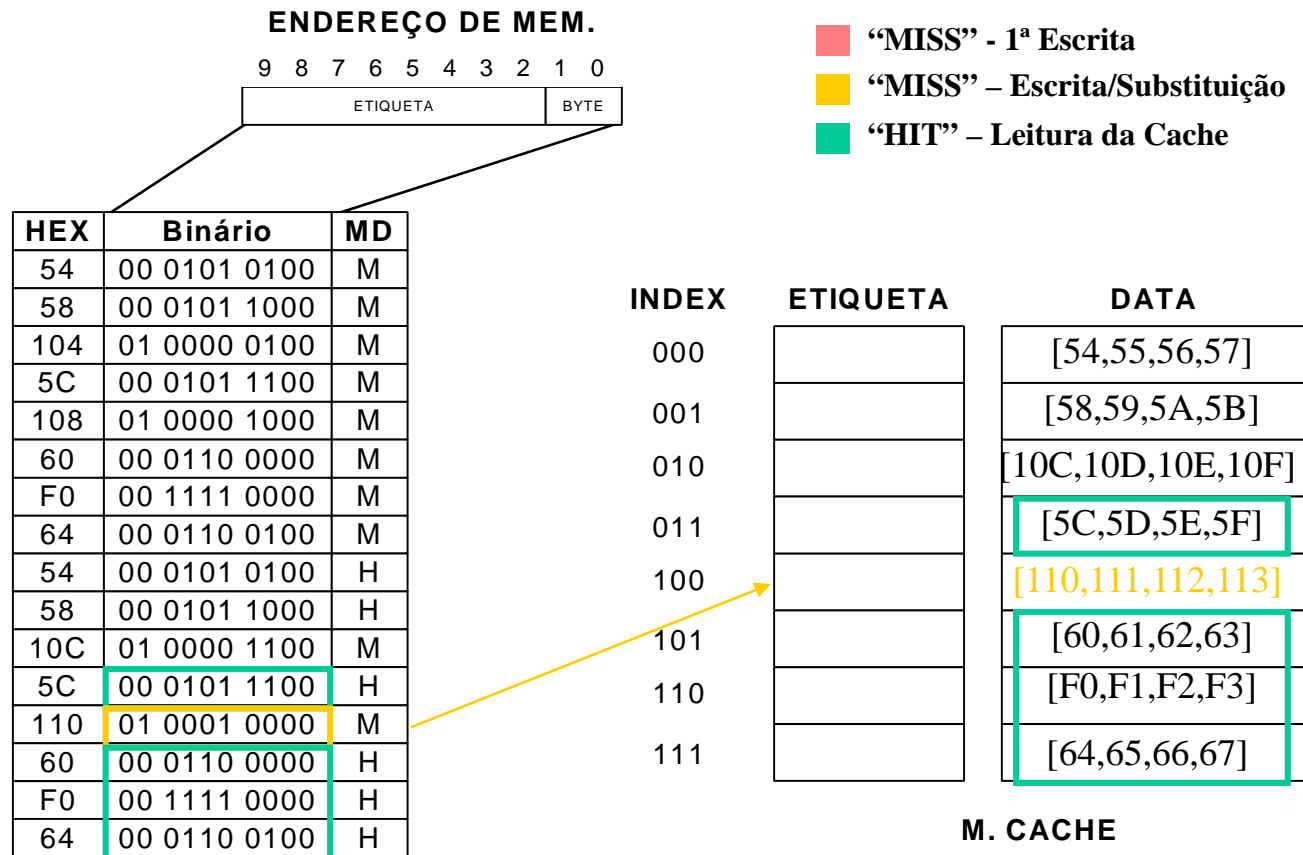


# Arquitectura de Computadores

## Unidade de Memória

### Memória Cache: Mapeamento Associativo

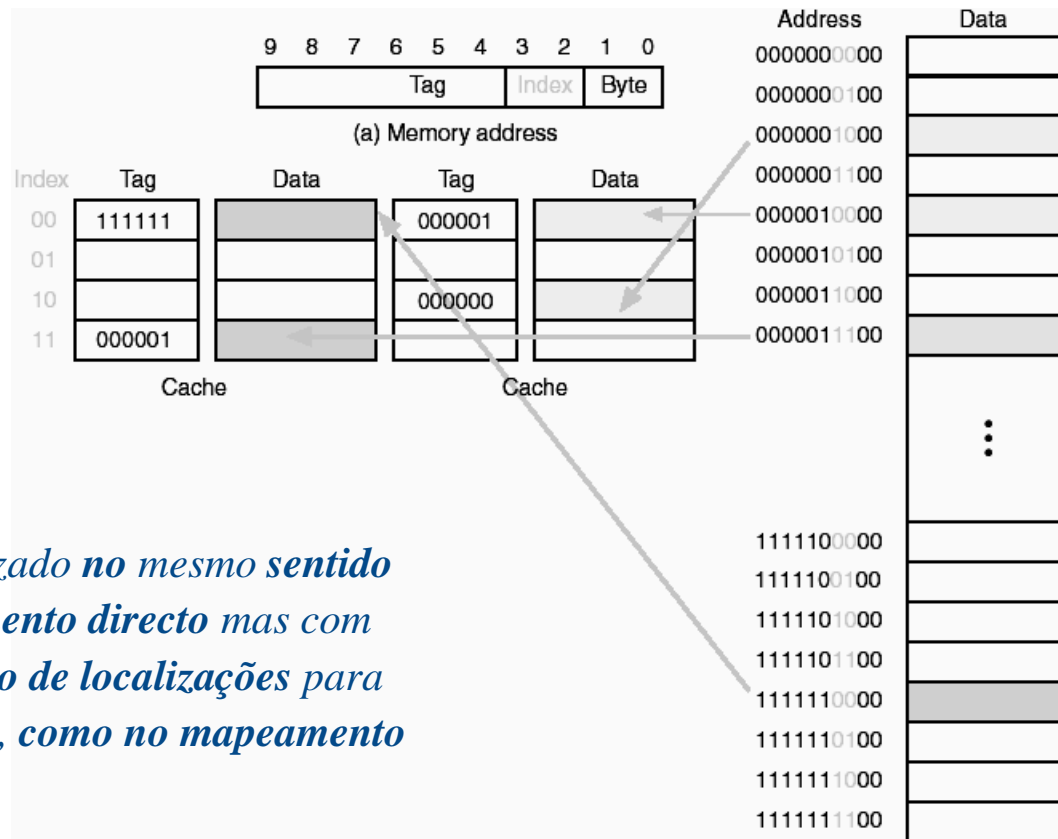
Exemplo:



# Arquitectura de Computadores

## Unidade de Memória

### Memória Cache: Mapeamento Associativo-Directo (Set-associative Mapping)



*Índice utilizado no mesmo sentido do mapeamento directo mas com um conjunto de localizações para cada índice, como no mapeamento associativo.*



INSTITUTO  
SUPERIOR  
TÉCNICO

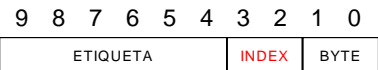
# Arquitetura de Computadores

## Unidade de Memória

### Memória Cache: Mapeamento Associativo-Directo

**Exemplo:** Para a sequência de endereços apresentada determine a ocorrência de “miss”s e “hit”s no acesso à memória Cache.

ENDEREÇO DE MEM.



HEX	Binário	MD
54	00 0101 0100	M
58	00 0101 1000	M
104	01 0000 0100	M
5C	00 0101 1100	M
108	01 0000 1000	M
60	00 0110 0000	M
F0	00 1111 0000	M
64	00 0110 0100	M
54	00 0101 0100	M
58	00 0101 1000	H
10C	01 0000 1100	M
5C	00 0101 1100	H
110	01 0001 0000	M
60	00 0110 0000	M
F0	00 1111 0000	M
64	00 0110 0100	H

INDEX	ETIQUETA	DATA	ETIQUETA	DATA
00				
01				
10				
11				

M. CACHE



INSTITUTO  
SUPERIOR  
TÉCNICO

# Arquitectura de Computadores

## Unidade de Memória

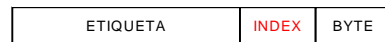
### Memória Cache: Mapeamento Associativo-Directo

Exemplo:

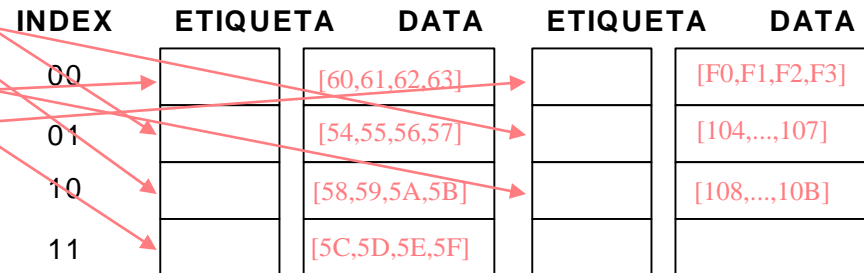
- “MISS” - 1ª Escrita
- “MISS” – Escrita/Substituição
- “HIT” – Leitura da Cache

ENDEREÇO DE MEM.

9 8 7 6 5 4 3 2 1 0



HEX	Binário	MD
54	00 0101 0100	M
58	00 0101 1000	M
104	01 0000 0100	M
5C	00 0101 1100	M
108	01 0000 1000	M
60	00 0110 0000	M
F0	00 1111 0000	M
64	00 0110 0100	M
54	00 0101 0100	M
58	00 0101 1000	H
10C	01 0000 1100	M
5C	00 0101 1100	H
110	01 0001 0000	M
60	00 0110 0000	M
F0	00 1111 0000	M
64	00 0110 0100	H



M. CACHE

# Arquitectura de Computadores

## Unidade de Memória

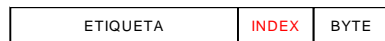
### Memória Cache: Mapeamento Associativo-Directo

Exemplo:

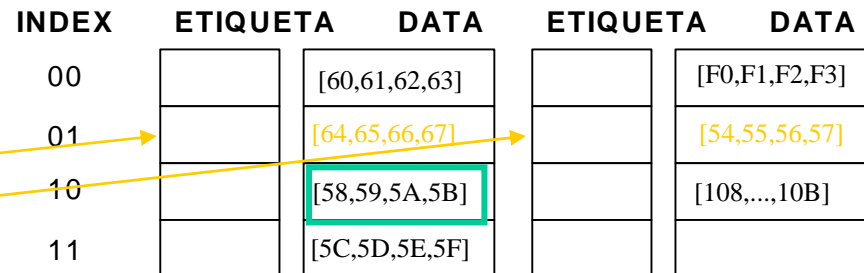
- “MISS” - 1ª Escrita
- “MISS” – Escrita/Substituição
- “HIT” – Leitura da Cache

ENDEREÇO DE MEM.

9 8 7 6 5 4 3 2 1 0



HEX	Binário	MD
54	00 0101 0100	M
58	00 0101 1000	M
104	01 0000 0100	M
5C	00 0101 1100	M
108	01 0000 1000	M
60	00 0110 0000	M
F0	00 1111 0000	M
64	00 0110 0100	M
54	00 0101 0100	M
58	00 0101 1000	H
10C	01 0000 1100	M
5C	00 0101 1100	H
110	01 0001 0000	M
60	00 0110 0000	M
F0	00 1111 0000	M
64	00 0110 0100	H



M. CACHE



# Arquitectura de Computadores

## Unidade de Memória

### Memória Cache: Mapeamento Associativo-Directo

Exemplo:

- “MISS” - 1ª Escrita
- “MISS” – Escrita/Substituição
- “HIT” – Leitura da Cache

ENDEREÇO DE MEM.

9 8 7 6 5 4 3 2 1 0

ETIQUETA INDEX BYTE

HEX	Binário	MD
54	00 0101 0100	M
58	00 0101 1000	M
104	01 0000 0100	M
5C	00 0101 1100	M
108	01 0000 1000	M
60	00 0110 0000	M
F0	00 1111 0000	M
64	00 0110 0100	M
54	00 0101 0100	M
58	00 0101 1000	H
10C	01 0000 1100	M
5C	00 0101 1100	H
110	01 0001 0000	M
60	00 0110 0000	M
F0	00 1111 0000	M
64	00 0110 0100	H

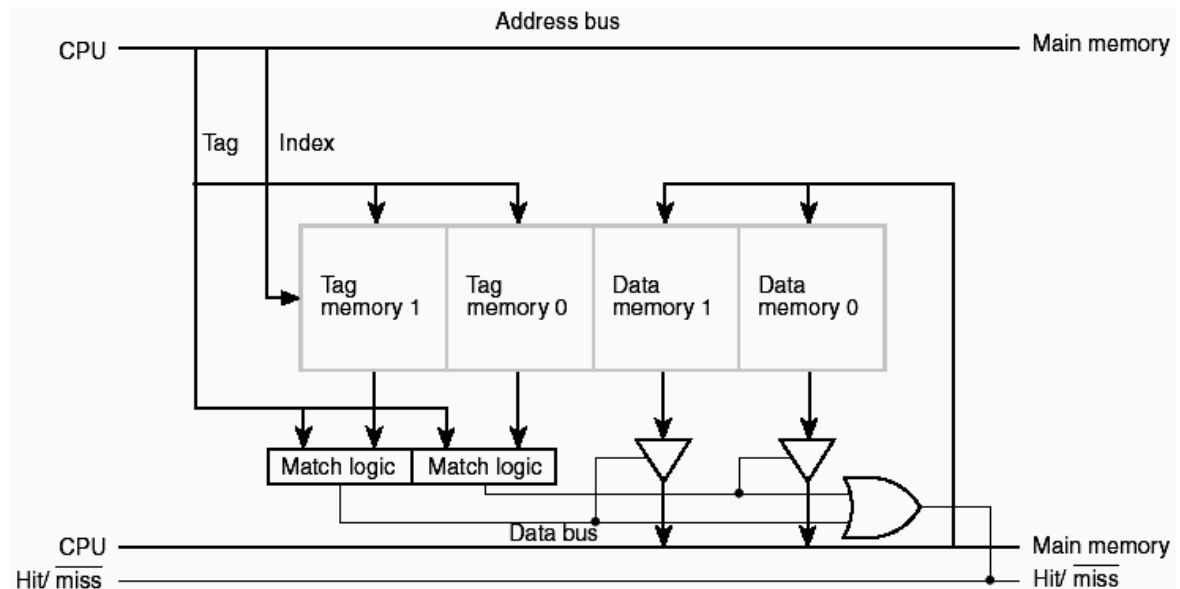
INDEX	ETIQUETA	DATA	ETIQUETA	DATA
00		[F0,F1,F2,F3]		[60,61,62,63]
01		[64,65,66,67]		[54,55,56,57]
10		[58,59,5A,5B]		[108,....,10B]
11		[5C,5D,5E,5F]		[10C,....,10F]

M. CACHE



INSTITUTO  
SUPERIOR  
TÉCNICO

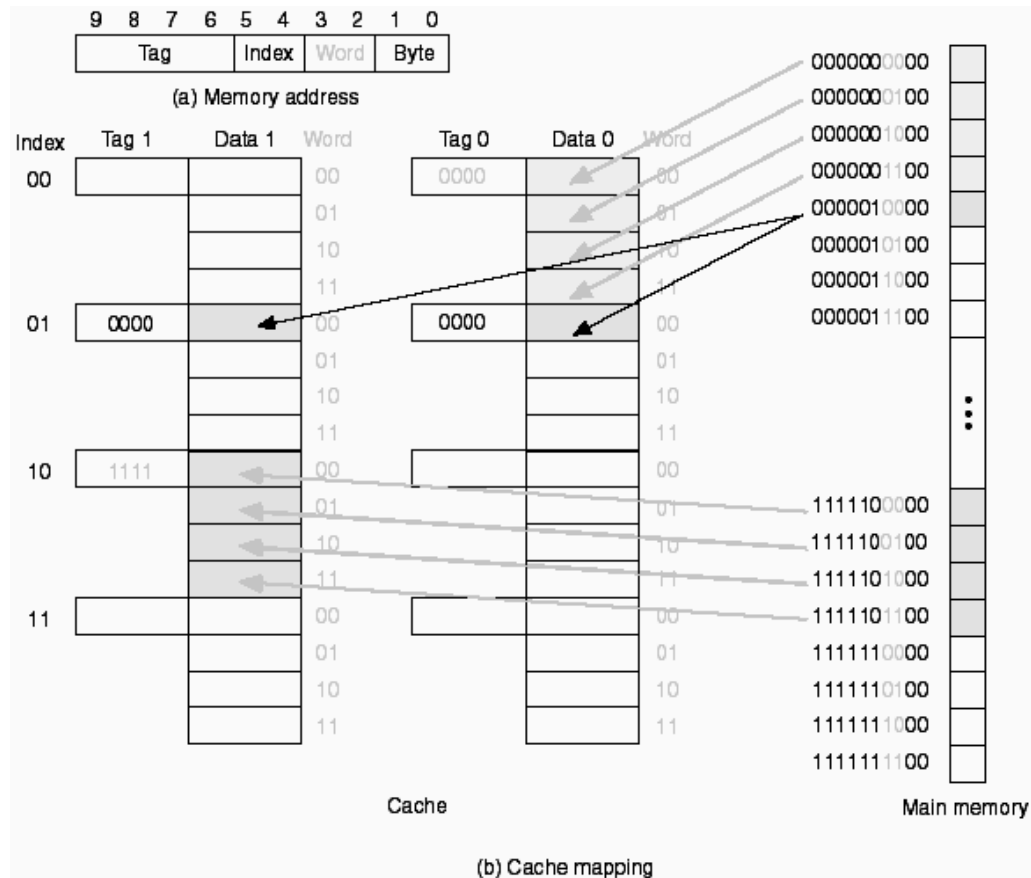
### Memória Cache: Mapeamento Associativo-Directo (Set-associative Mapping)



# Arquitetura de Computadores

## Unidade de Memória

*Memória Cache: “Set-associative Mapping” c/ n Palavras por Etiqueta.*



# Arquitetura de Computadores

## Unidade de Memória

---

### *Memória Cache: Carregamento e Métodos de Escrita*

#### *Aquisição de Dados da Cache p/ CPU:*

*Considera-se um bit adicional para validação por cada etiqueta para assegurar que não são carregados dados inválidos, em particular na fase inicial.*

#### *Memória de Escrita: (p/ Resultados das Operações)*

*Memória Principal*

*Memória Cache*

*Memória Principal e Cache*

#### *Métodos de Escrita:*

*Write Through*

*Write Back/ Copy Back*



INSTITUTO  
SUPERIOR  
TÉCNICO

### Métodos de Escrita:

**Write Through** – *Resultado* é escrito sempre na *memória principal*.

**Write Buffer** – escritas são armazenadas num write buffer que guarda o pedido até que a operação de escrita se concretize.

**Write Back/ Copy Back** – *Resultado* escrito na *memória cache* em caso de um “**cache hit**”

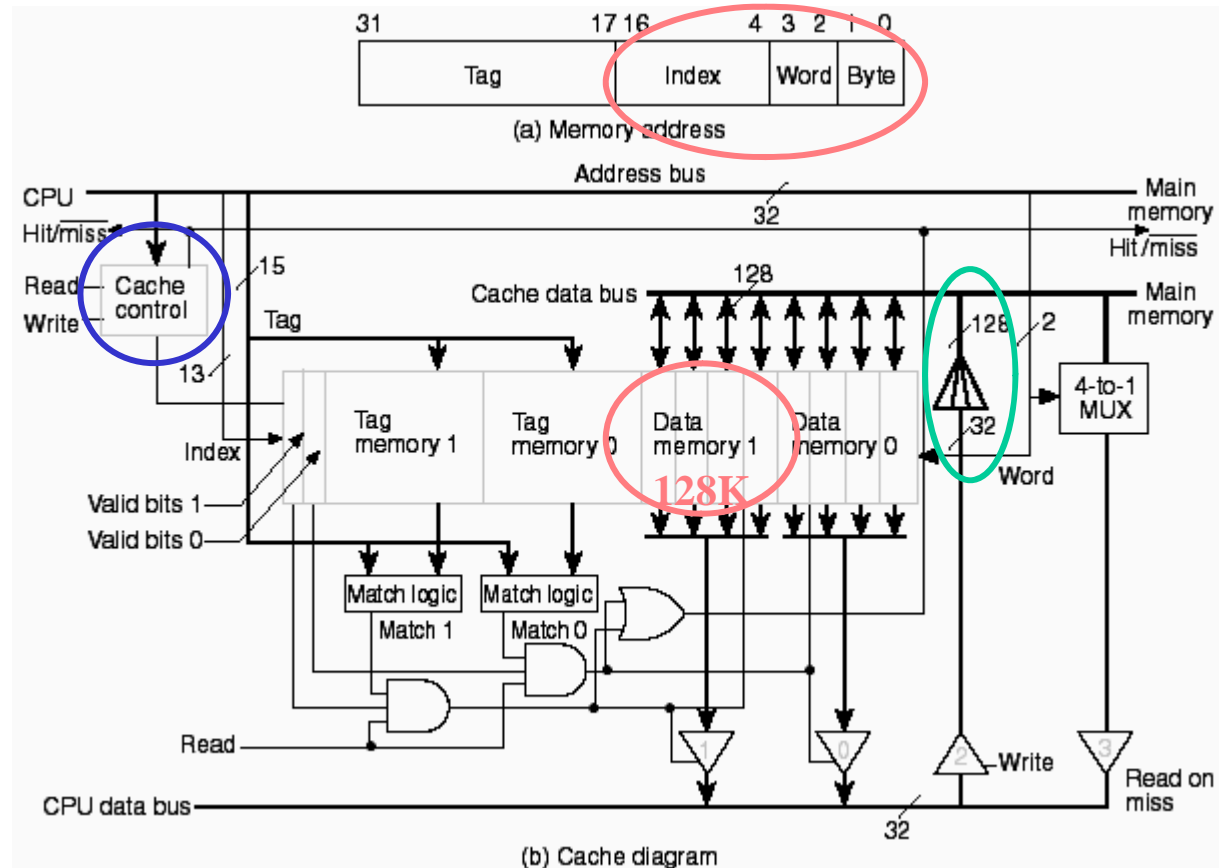
- **Write Allocate** : no caso de um “**cache miss**” é realizada uma escrita em *memória principal* ou,
- **No Write Allocate** : escrita em *memória principal* e na *memória cache* (futuras escritas no mesmo bloco não necessitam de aceder à memória!).
- **Desvantagens** : aumenta a miss penalty, já que um miss passa a implicar uma escrita para a memória principal. Implica snooping em sistemas multiprocessador.

# Arquitectura de Computadores

## Unidade de Memória

Memória Cache: *Exemplo* (Diagrama de Blocos para “Set-associative Mapping” c/ n Palavras por Etiqueta)

Mem. Cache de 256K



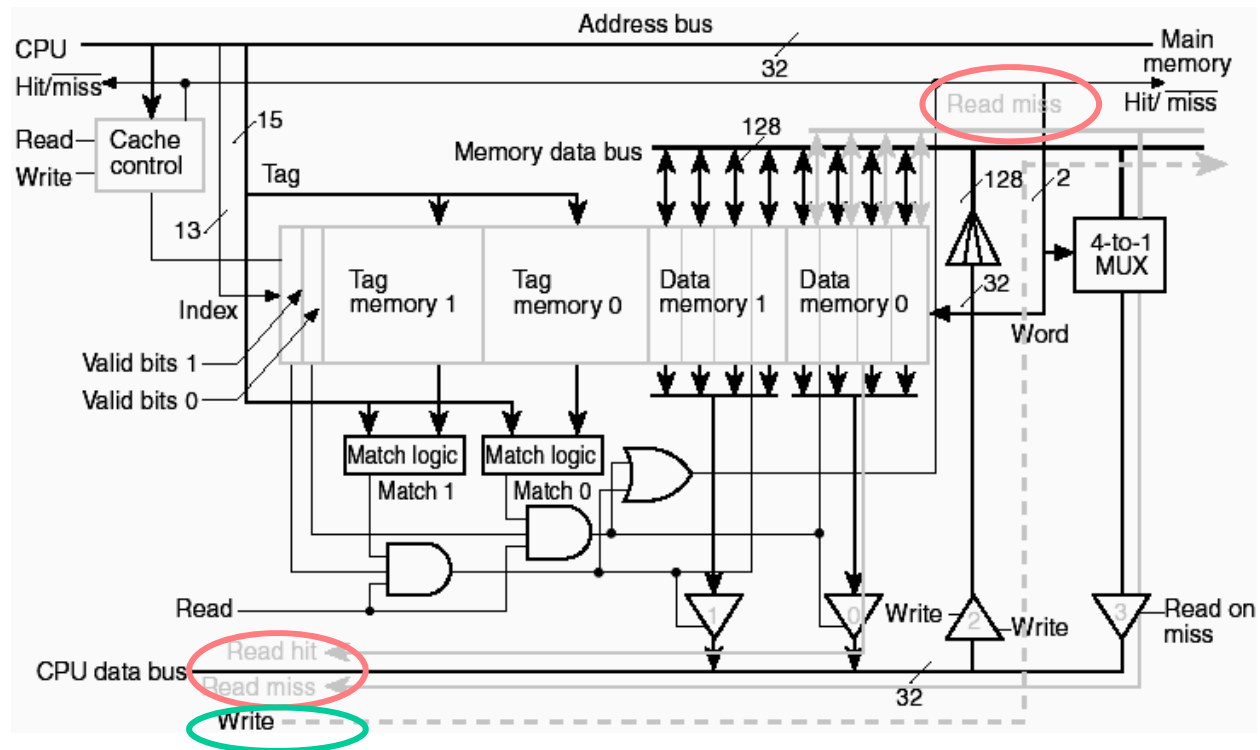
INSTITUTO  
SUPERIOR  
TÉCNICO

# Arquitectura de Computadores

## Unidade de Memória

*Memória Cache: Exemplo (Operações de Escrita e Leitura para “Set-associative Mapping” c/ n Palavras por Etiqueta)*

*Mem. Cache de 256K*



# Arquitectura de Computadores

## Unidade de Memória

### Memória Cache:

*Cache de Instruções e Dados*

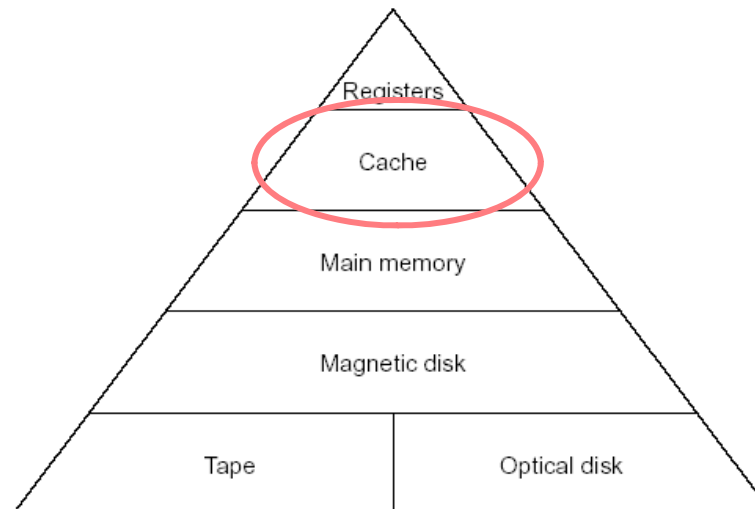
vs

*Cache de Instruções e Cache de Dados*

*Cache Multinível: (Maior velocidade, controlo mais complexo)*

*L1 (cache interna à CPU)*

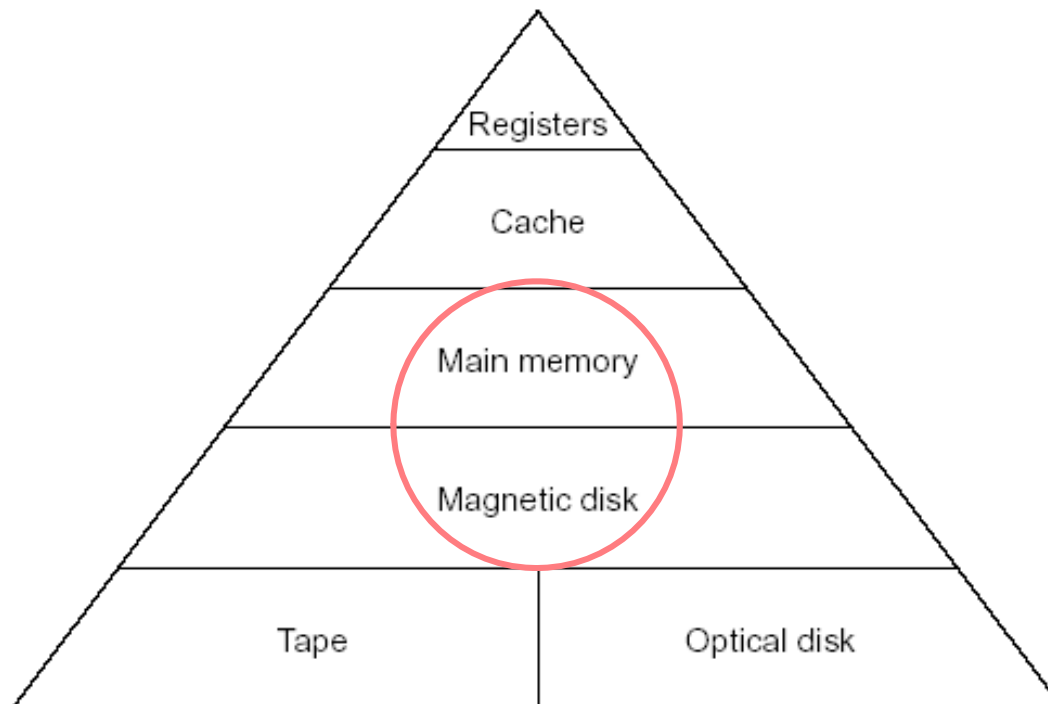
*L2 (cache externa)*



INSTITUTO  
SUPERIOR  
TÉCNICO



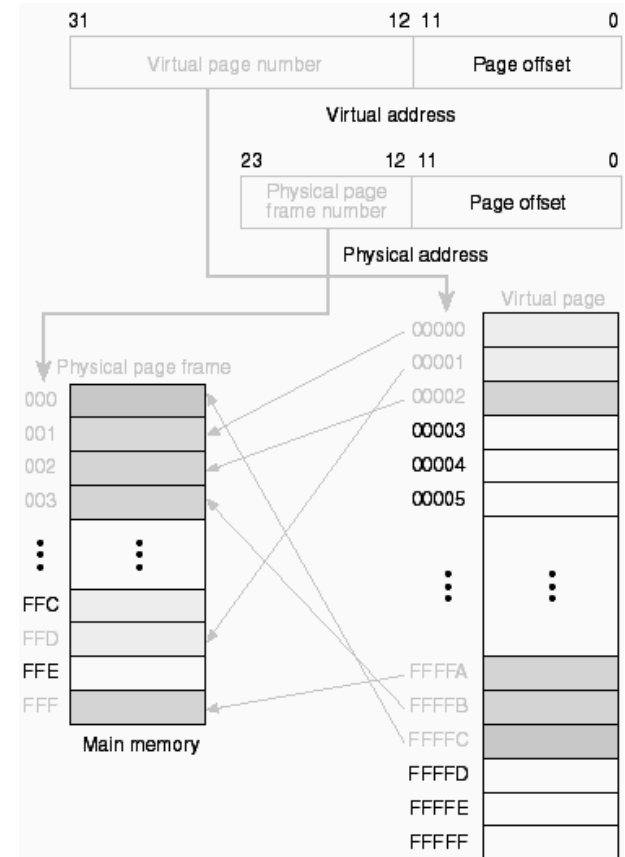
### Memória Virtual



### Memória Virtual: Endereços Físicos, Endereços Virtuais e Mapeamento

**Memória Cache:** Solução que permite *aumentar a velocidade dos acessos* à memória sem necessitar de uma única memória rápida e de elevada capacidade, o que seria uma solução demasiado dispendiosa.

**Memória Virtual:** Solução para *aumentar a dimensão da memória Principal* com recurso ao disco rígido.





### Memória Virtual: Paginação

**Virtual Page:** Blocos de endereços, semelhante a linhas na Cache mas de maiores dimensões.

**Physical Page Frame:** O espaço de endereçamento em memória está dividido em “Page Frames” de dimensão igual às “Virtual Pages”.

#### Exemplo:

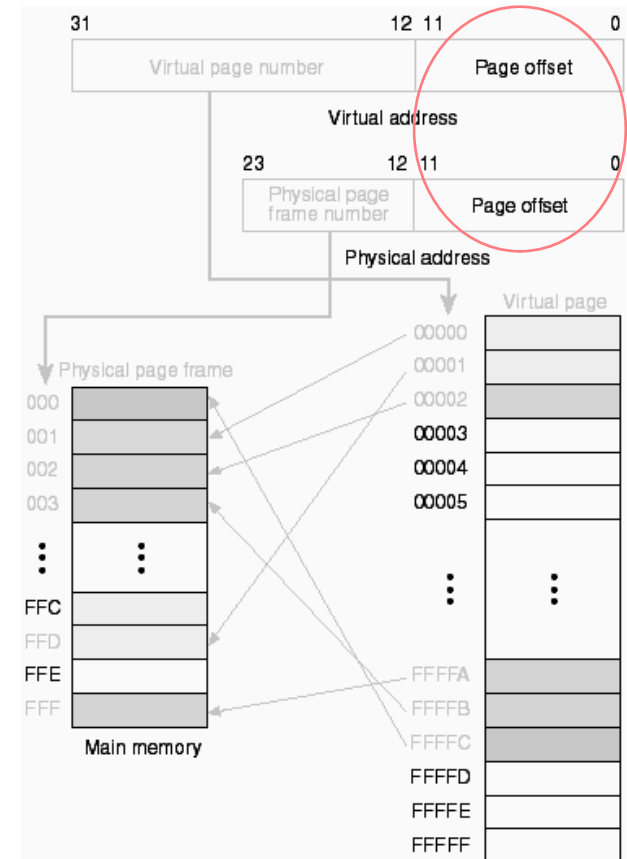
Espaço de Endereçamento Virtual: 32 Bits

Page: 4K bytes (1K words de 32 bits)

Nº Páginas Virtuais:  $2^{20}$

Memória Principal: 16 Mbytes

Page Frames:  $2^{12}$



# Arquitectura de Computadores

## Unidade de Memória

### Memória Virtual: Paginação

**Page Table:** Mapeamento de Páginas Virtuais em Memória e em HD.

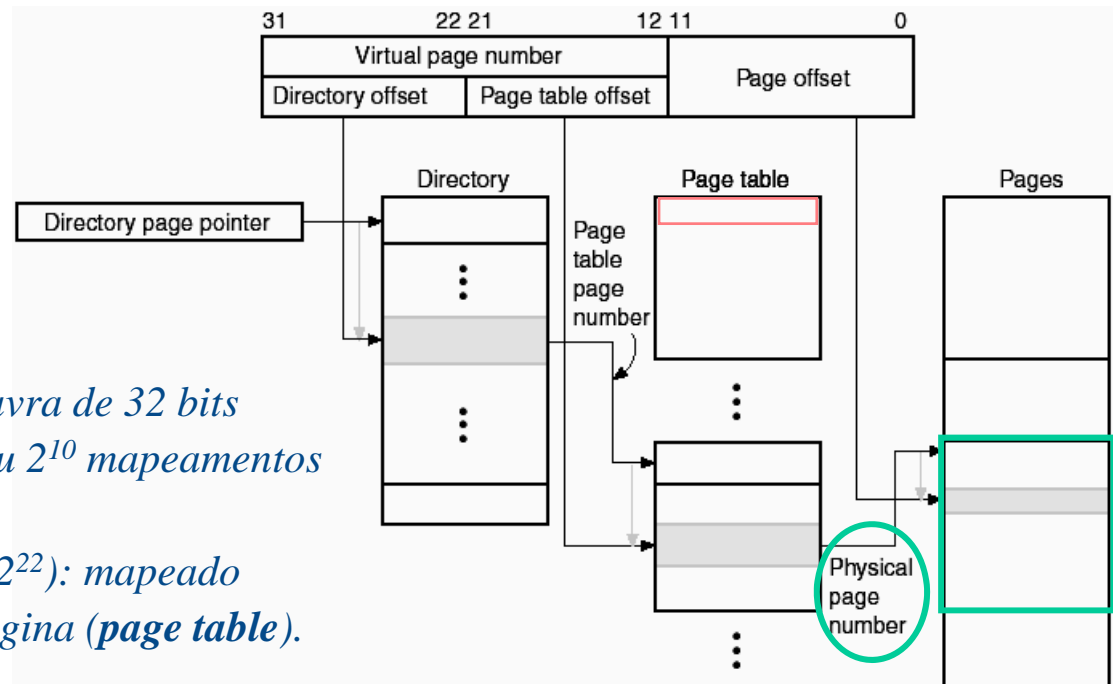
**Directory Page:** Guarda informação sobre as páginas (*page tables*) de um programa.

#### Exemplo:

1 Mapeamento: 1 palavra de 32 bits

1 Página (4KB): 1K ou  $2^{10}$  mapeamentos

1 Programa de 4MB ( $2^{22}$ ): mapeado com recurso a uma página (*page table*).



# Arquitectura de Computadores

## Unidade de Memória



INSTITUTO  
SUPERIOR  
TÉCNICO

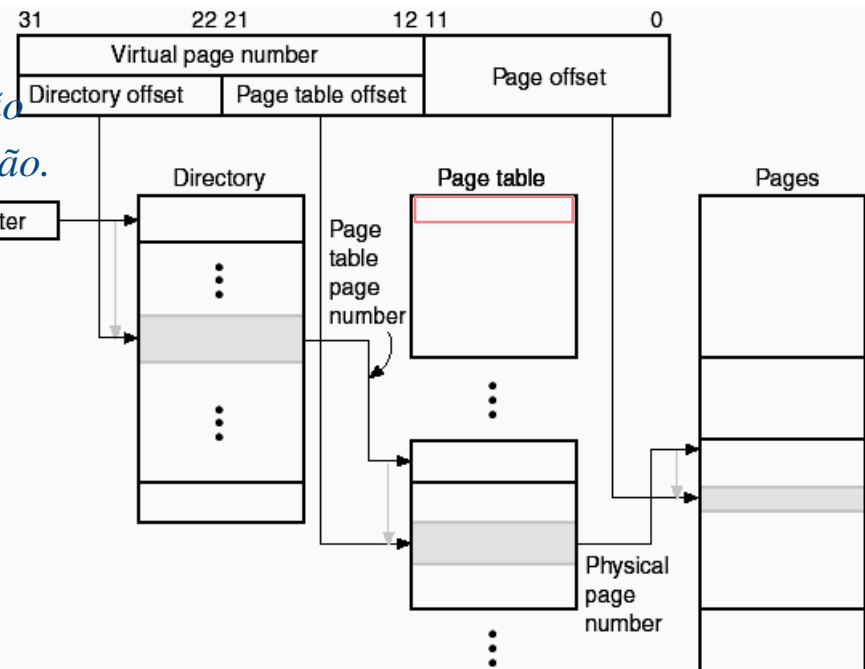
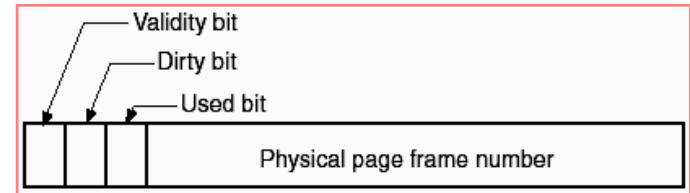
### Memória Virtual: Paginação

**Validity Bit:** Valida *Page Frame* em memória.

**Dirty Bit:** Indica se a página em memória foi escrita desde o seu carregamento. Implica actualização do HD a quando de uma substituição.

**Used Bit:** Indica se a página foi utilizada recentemente para efeito do algoritmo de substituição de páginas.

**Directory Page Pointer:** Indica a localização de uma *Directory Page* na memória principal.



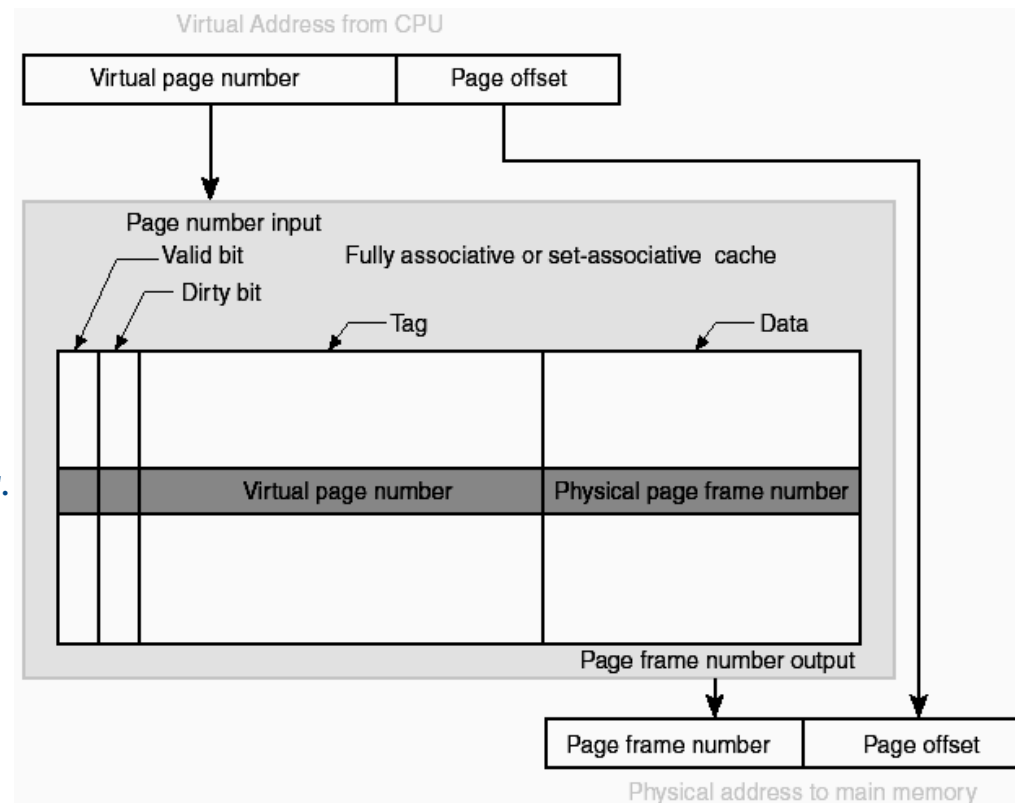
### Memória Virtual: Paginação

#### Aquisição de Instrução ou Operando:

Em 3 acessos à mem. principal (!!?)

- (1) Directório
- (2) Página
- (3) Instrução

*Introdução de nova Cache para traduzir directamente os endereços viruais em endereços físicos. Guarda a localização das páginas mais recentemente acedidas.*

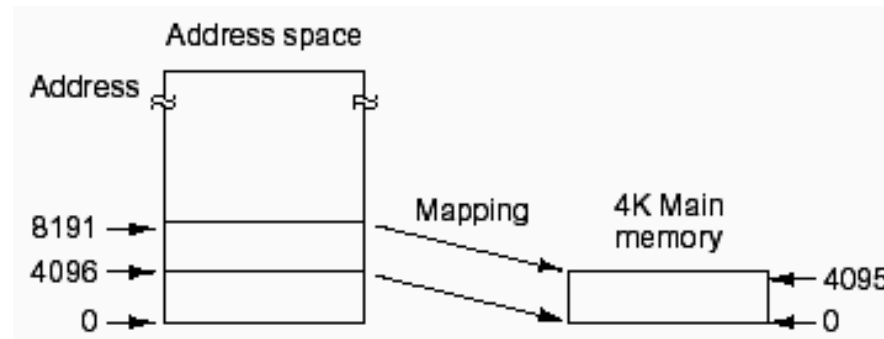


# Arquitectura de Computadores

## Unidade de Memória

### Memória Virtual:

*Exemplo: Mapeamento de um bloco de Memória Virtual em Memória Principal.*



*Num computador com memória virtual deverá ser executada a seguinte sequência de passos:*

- (1) Guardar o conteúdo de memória principal no disco.*
- (2) Localização das palavras 4096 a 8191 em disco.*
- (3) Carregamento das palavras 4096 a 8191 para memória principal.*
- (4) Actualização do mapa de endereços i.e. as palavras 4096 a*



INSTITUTO  
SUPERIOR  
TÉCNICO

# Arquitetura de Computadores

## Unidade de Memória

### Memória Virtual:

### Exemplo: Paginação (Virtual Page vs Physical Page Frame)

Page	Virtual addresses	Page frame	Physical addresses
15	81440 – 85535		
14	57344 – 81439		
13	53248 – 57343		
12	49152 – 53247		
11	45056 – 49151		
10	40960 – 45055		
9	36864 – 40959		
8	32768 – 36863		
7	28672 – 32767		
6	24576 – 28671		
5	20480 – 24575		
4	16384 – 20479		
3	12288 – 16383		
2	8192 – 12287		
1	4096 – 8191		
0	0 – 4095		

Page frame	Physical addresses
7	28672 – 32767
6	24576 – 28671
5	20480 – 24575
4	16384 – 20479
3	12288 – 16383
2	8192 – 12287
1	4096 – 8191
0	0 – 4095

Bottom 32K of main memory

Page frame Physical addresses



INSTITUTO SUPERIOR TÉCNICO

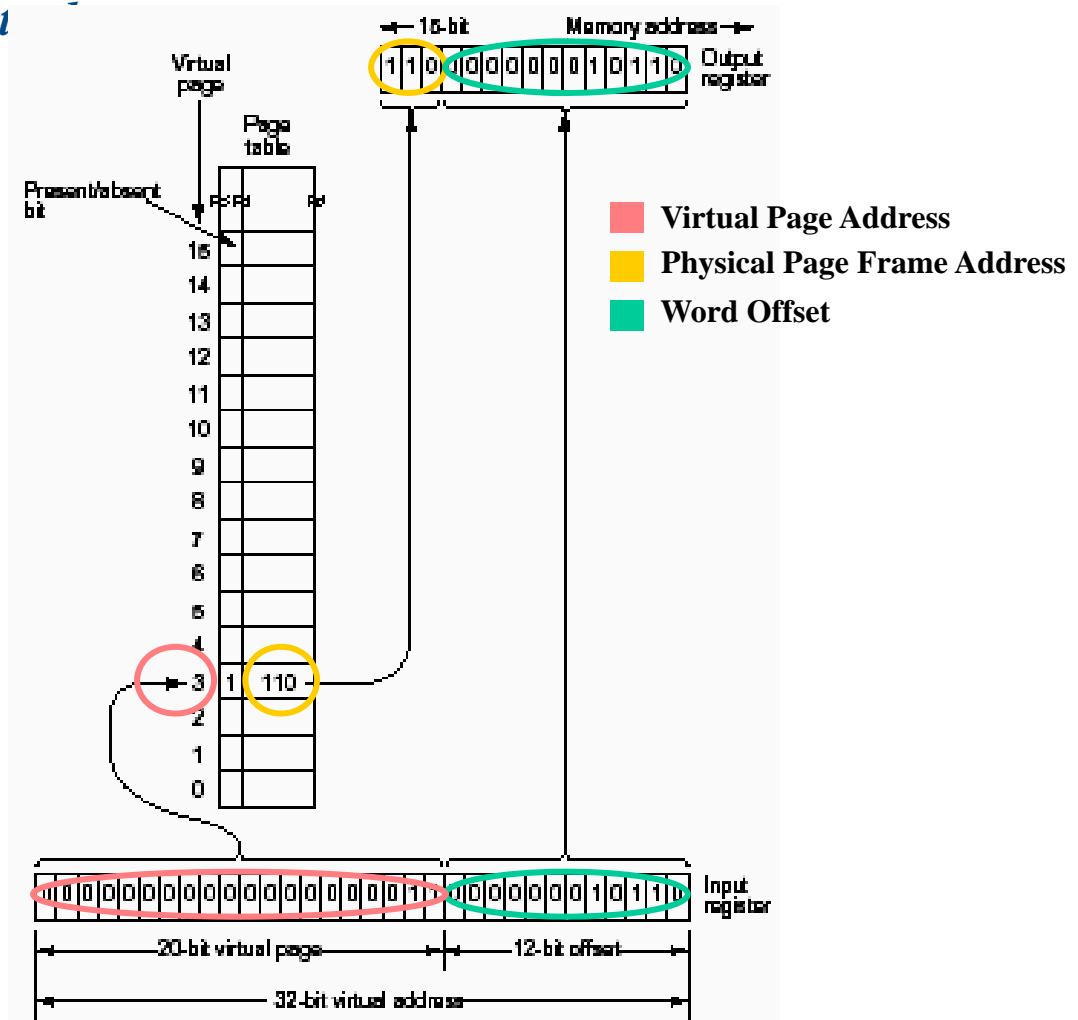


# Arquitectura de Computadores

## Unidade de Memória

### Memória Virtual:

**Exemplo:** Determinação do End. de Memória Principal tendo por base End. Virt.

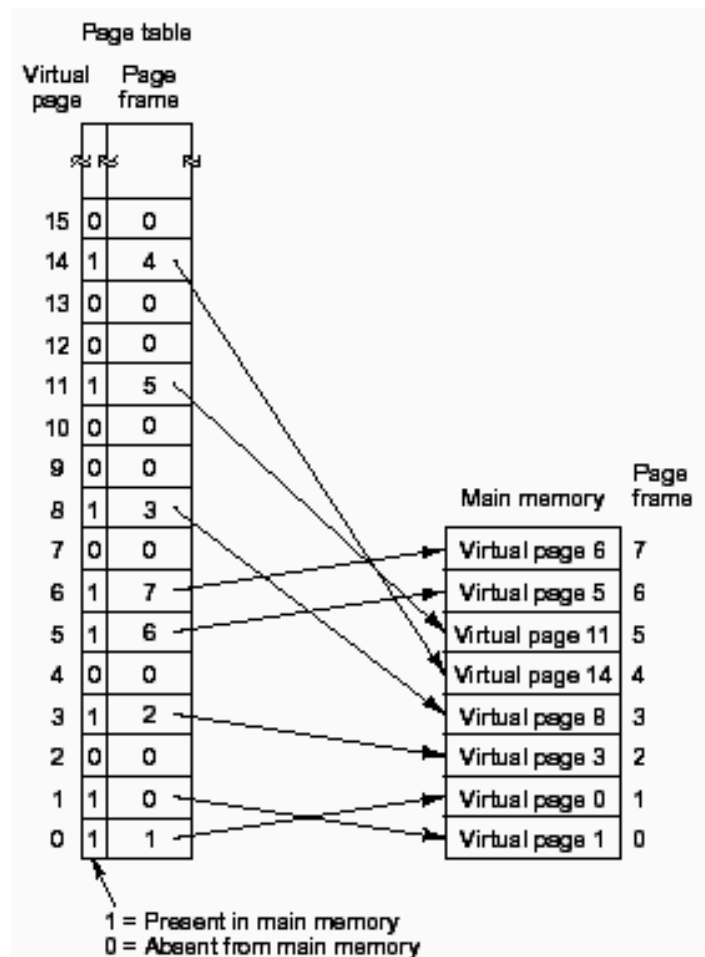


# Arquitectura de Computadores

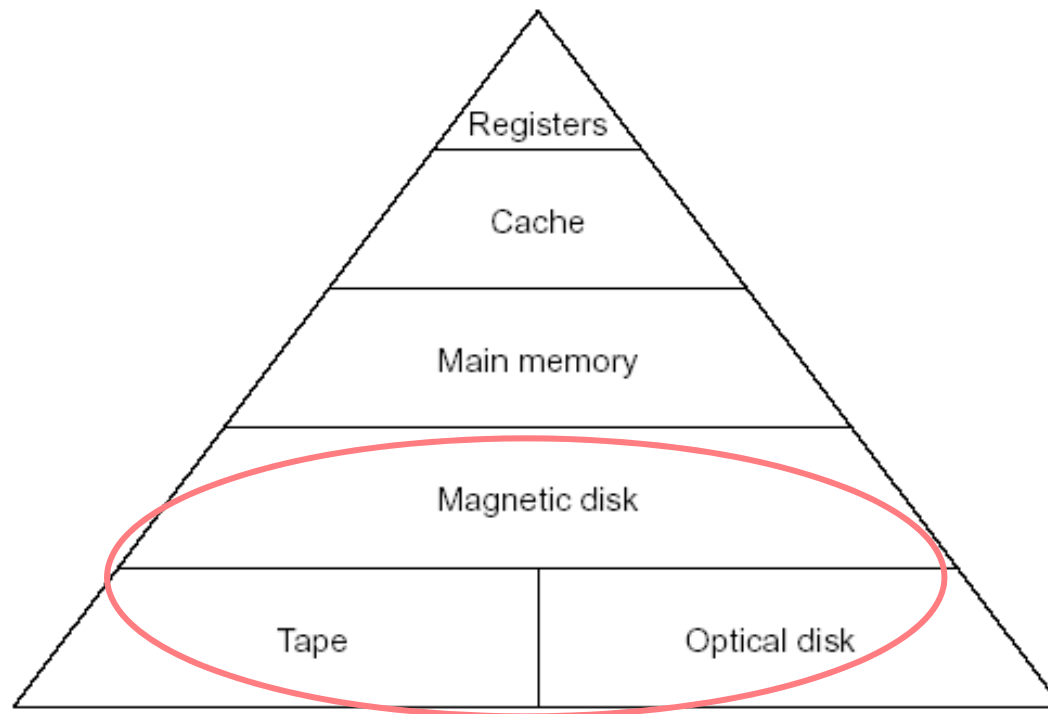
## Unidade de Memória

### Memória Virtual:

### Exemplo: Mapeamento de Páginas Virtuais em Páginas da Memória Principal.

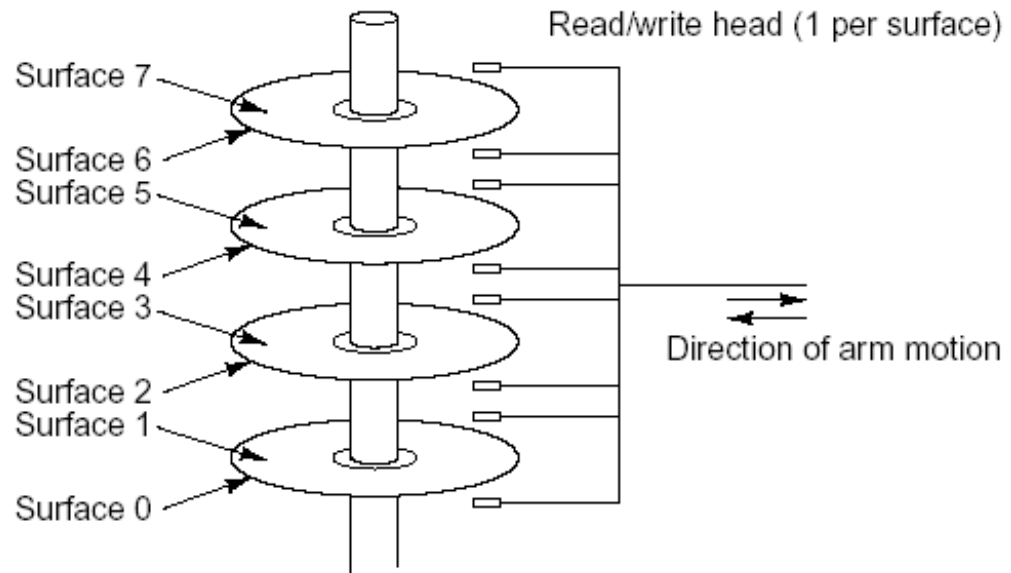


### *Memória Secundária (cont.): CDs, DVDs, etc.*



### *Hierarquia de Memória: Memória Secundária*

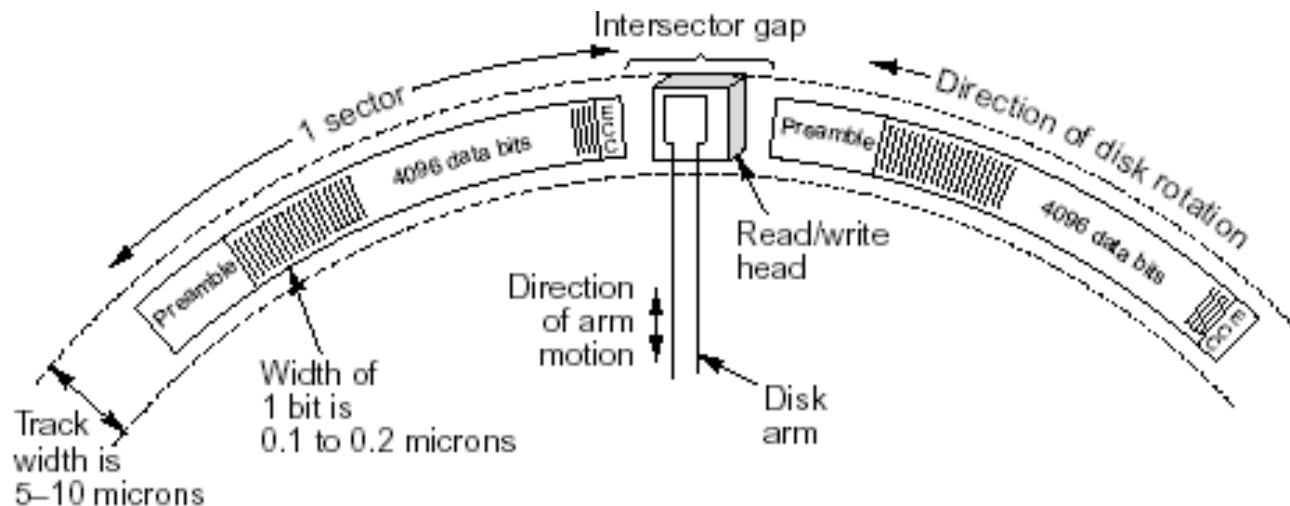
### *Disco Rígido ou Disco Magnético*



**Disco Rígido com 4 pratos.**

### Hierarquia de Memória: Memória Secundária

### Disco Rígido ou Disco Magnético



**Ilustração parcial de uma pista (Track) de disco com 2 sectores.**

### *Hierarquia de Memória: Memória Secundária*

#### *Discos IDE – Integrated Drive Electronics*

- *Drives e Controlador na mesma unidade.*
- *Interface IBM PC Bus*

#### *Discos SCSI – Small Computer System Interface*

- *Diferentes interfaces e ritmos de transferência mais elevados.*

Name	Data bits	Bus MHz	MB/sec
SCSI-1	8	5	5
Fast SCSI	8	10	10
Wide Fast SCSI	16	10	20
Ultra SCSI	8	20	20
Wide Ultra SCSI	16	20	40
Ultra2 SCSI	8	40	40
Wide Ultra2 SCSI	16	40	80

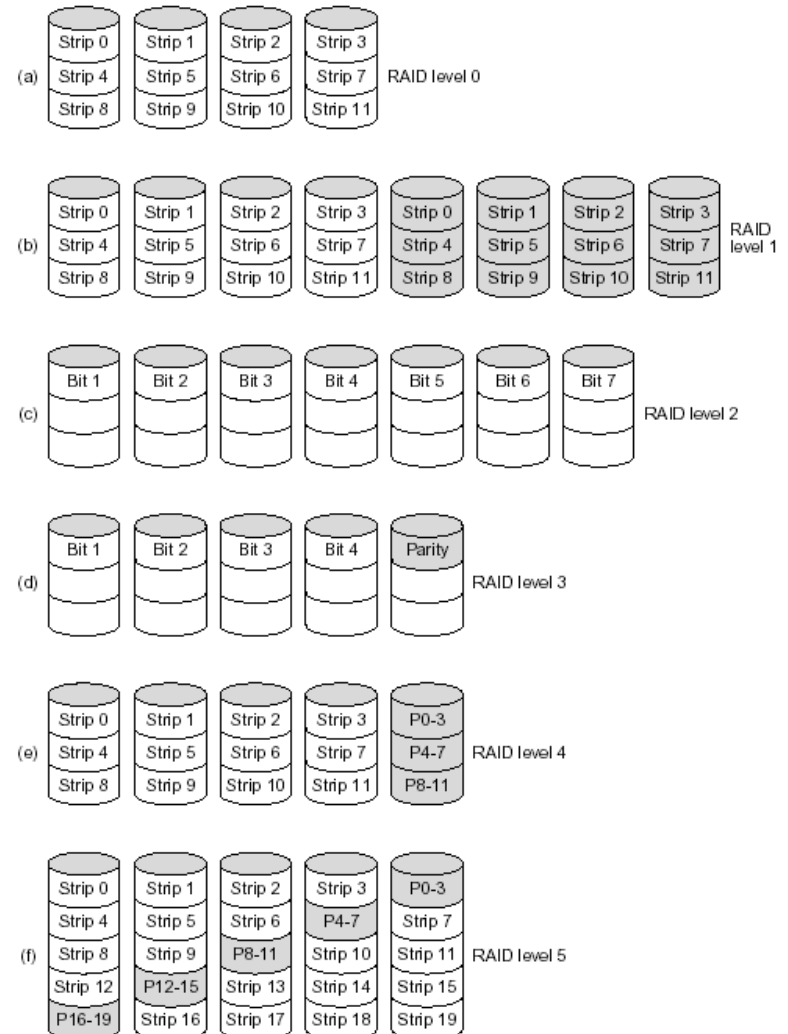
# Arquitectura de Computadores

## Unidade de Memória

### Hierarquia de Memória: Memória Secundária

**RAID – Redundant Array of Independent Disks.**

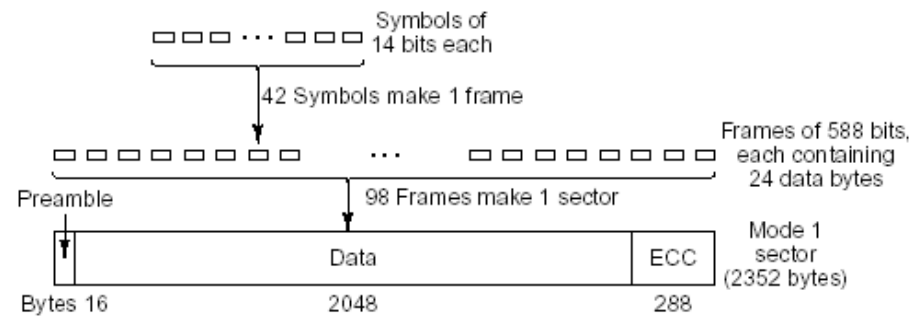
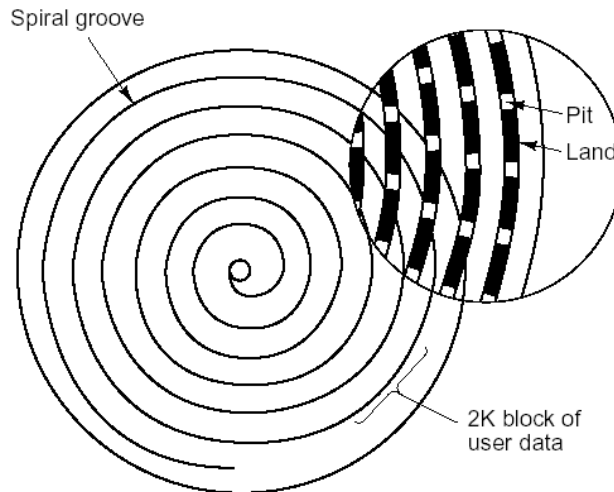
**SLED – Single Larger Expensive Disk**





### Hierarquia de Memória: Memória Secundária

### CD-ROM – Compact Disc – Read Only Memory



### Estrutura de gravação num CD-ROM.

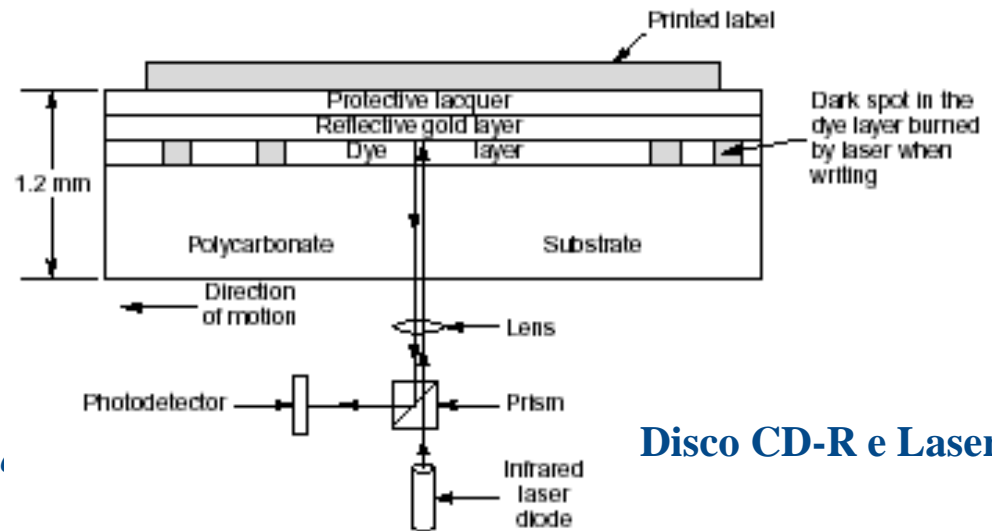
### Organização de dados num CD-ROM





### Hierarquia de Memória: Memória Secundária

#### CD – R (Gravável)



Disco CD-R e Laser

#### CD – RW (Regravável)

#### DVD – Digital Versatile Disk

*Single-Sided e Camada Única (4.7 GB)*

*Single-Sided e Camada Dupla (8.5 GB)*

*Double-Sided e Camada Única (9.4 GB)*

*Double-Sided e Camada Dupla (17 GB)*



INSTITUTO  
SUPERIOR  
TÉCNICO

## **BIBLIOGRAFIA**

**[1] M. Morris Mano, Charles R. Kime, “Logic and Computer Design Fundamentals”, Prentice-Hall International, Inc. (Capítulo 12)**

**[2] A. S. Tanenbaum, “Structured Computer Organization”, Prentice-Hall International, Inc. (Capítulo 6)**