DEEC
DEPARTAMENTO DE ENGENHARIA
ELETROTÉCNICA E DE COMPUTADORES
TÉCNICO LISBOA

Arquitectura de Computadores

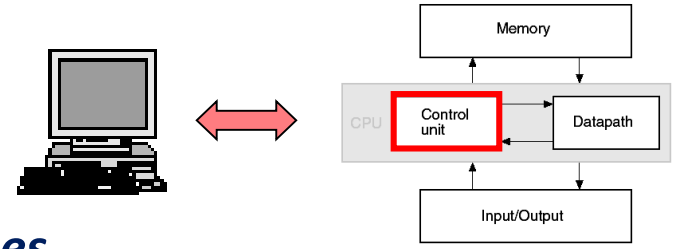
MEEC (2016/17 – 2º Sem.)

Unidade de Memória

Prof. Nuno Horta

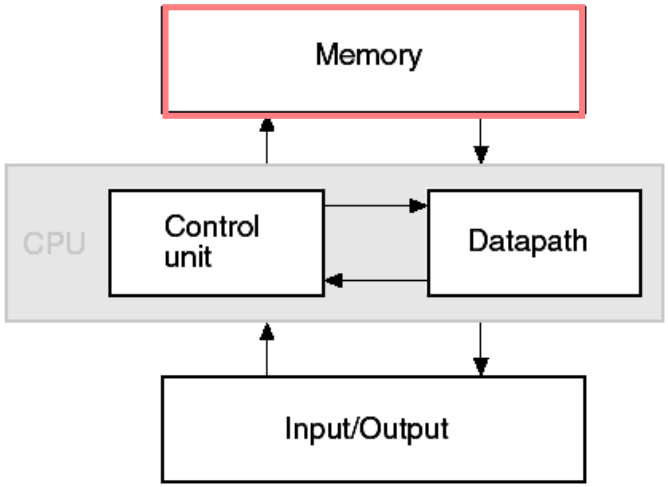
PLANEAMENTO

- *Introdução*
- *Unidade de Processamento*
- *Unidade de Controlo*
- *Arquitectura do Conjunto de Instruções*
- *Unidade Central de Processamento (CPU)*
- *P3 – Processador e Periféricos*
- *Unidades de Entrada/Saída (I/O)*
- *Unidade de Memória*



SUMÁRIO

- *Unidade de Memória*
  - *Hierarquia de Memória*
  - *Memória Cache*
  - *Memória Virtual*
  - *Arrays de Discos Rígidos*

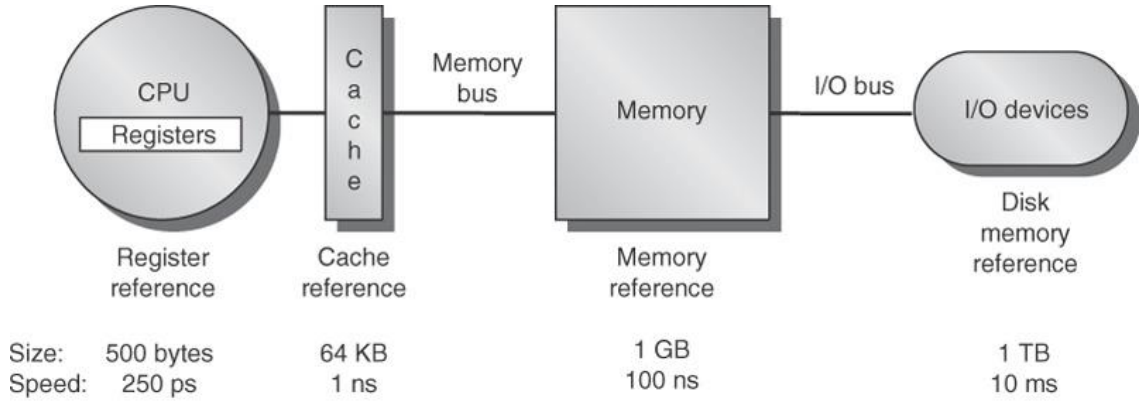


UNIDADE DE MEMÓRIA

Hierarquia de Memória

- **Unidade de Memória (Ideal) –** Idealmente os programadores desejariam uma **memória de dimensão ilimitada e de elevada velocidade** … para além da impossibilidade técnica de uma memória ilimitada acrescem os elevados custos associados às memórias rápidas.

- **Unidade de Memória (Real)** – **Estrutura hierárquica** explorando o Princípio da Localidade … visto que não é necessário aceder a todo o código ou dados de uma forma uniforme, durante a execução existe sim **um acesso a diferentes zonas de código ao longo do tempo**.
  - **Memória mais rápida, mais cara, em menores dimensões utilizada, preferencialmente, durante a execução.**
  - **Memória mais lenta, mais económica, utilizada para armazenamento**

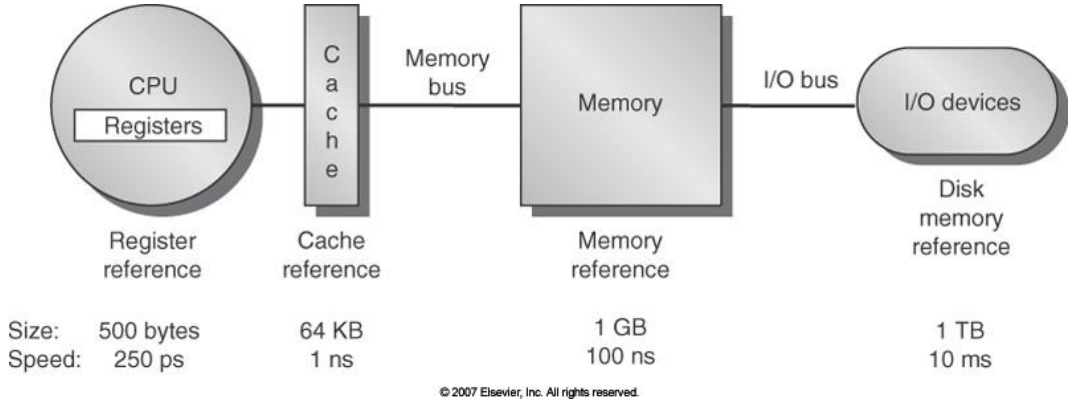


© 2007 Elsevier, Inc. All rights reserved.

UNIDADE DE MEMÓRIA

Hierarquia de Memória

- **Estrutura Hierárquica**



- **Caracterização dos níveis hierárquicos**

Level	1	2	3	4
Name	registers	cache	main memory	disk storage
Typical size	< 1 KB	< 16 MB	< 512 GB	> 1 TB
Implementation technology	custom memory with multiple ports, CMOS	on-chip or off-chip CMOS SRAM	CMOS DRAM	magnetic disk
Access time (ns)	0.25–0.5	0.5–25	50–250	5,000,000
Bandwidth (MB/sec)	50,000–500,000	5000–20,000	2500–10,000	50–500
Managed by	compiler	hardware	operating system	operating system/ operator
Backed by	cache	main memory	disk	CD or tape

UNIDADE DE MEMÓRIA

Hierarquia de Memória

- **Evolução do Desempenho das Memórias DRAM**

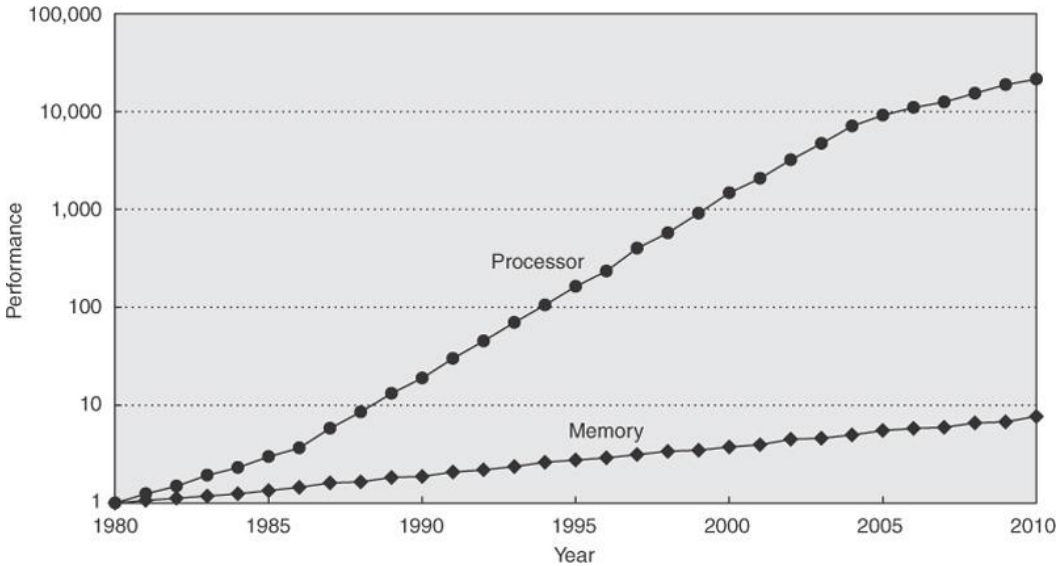
Year introduced	Chip size	$ per GB	Total access time to a new row/column	Column access time to existing row
1980	64 Kbit	$1,500,000	250 ns	150 ns
1983	256 Kbit	$500,000	185 ns	100 ns
1985	1 Mbit	$200,000	135 ns	40 ns
1989	4 Mbit	$50,000	110 ns	40 ns
1992	16 Mbit	$15,000	90 ns	30 ns
1996	64 Mbit	$10,000	60 ns	12 ns
1998	128 Mbit	$4,000	60 ns	10 ns
2000	256 Mbit	$1,000	55 ns	7 ns
2004	512 Mbit	$250	50 ns	5 ns
2007	1 Gbit	$50	40 ns	1.25 ns

FIGURE 5.12 DRAM size increased by multiples of four approximately once every three years until 1996, and thereafter considerably slower. The improvements in access time have been slower but continuous, and cost roughly tracks density improvements, although cost is often affected by other issues, such as availability and demand. The cost per gigabyte is not adjusted for inflation. Copyright © 2009 Elsevier, Inc. All rights reserved.

UNIDADE DE MEMÓRIA

Hierarquia de Memória

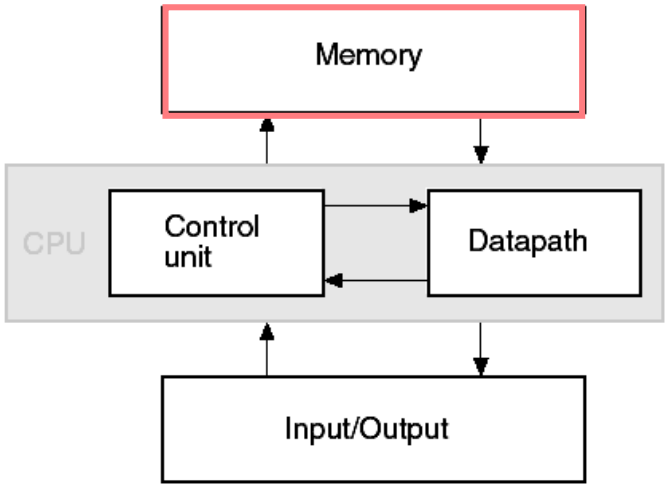
- **Evolução do Desempenho**
  - **Processadores (Melhor evolução do desempenho)**
  - **Memória**
    - Fundamental a pesquisa de novas soluções para diminuir o "gap" entre a evolução do desempenho dos processadores e memórias.



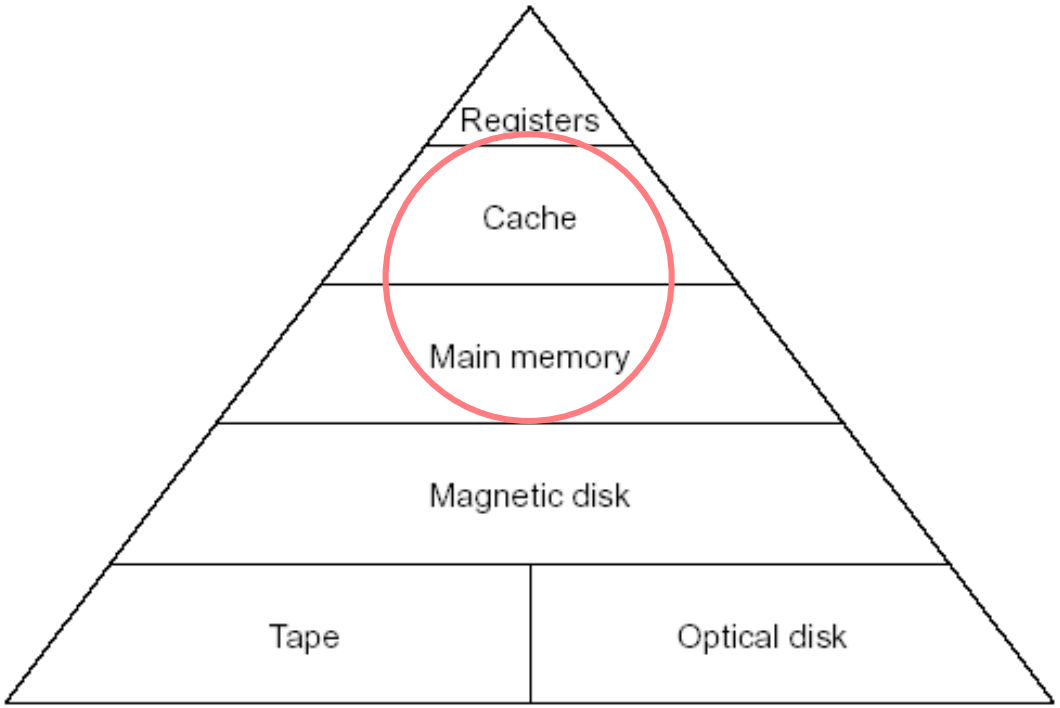
© 2007 Elsevier, Inc. All rights reserved.

SUMÁRIO

- *Unidade de Memória*
  - *Hierarquia de Memória*
  - *Memória Cache*
  - *Memória Virtual*
  - *Arrays de Discos Rígidos*



Memória Cache



UNIDADE DE MEMÓRIA

Memória Cache

Hierarquia de Memória: Estrutura

Memória Cache:

Acessos rápidos,

Custo elevado.

Memória Principal:

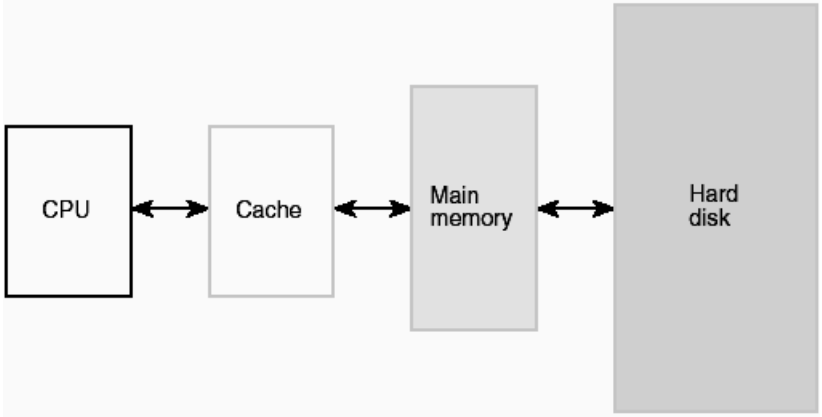
Tempos de Acesso Intermédios,

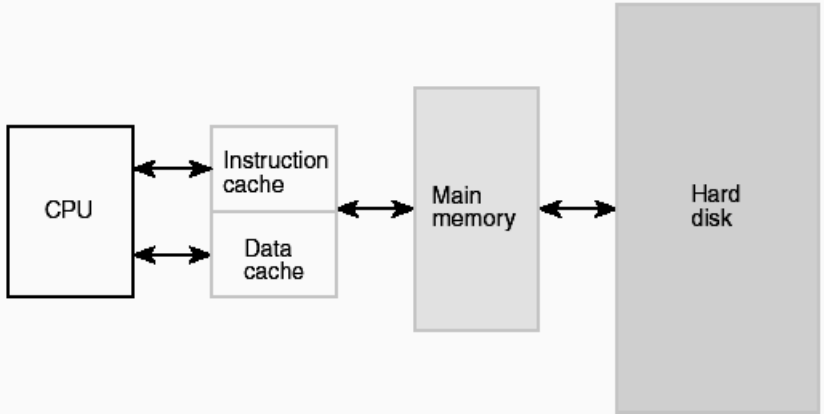
Custo médio.

Memória Secundária:

Acessos lentos,

Custo reduzido.

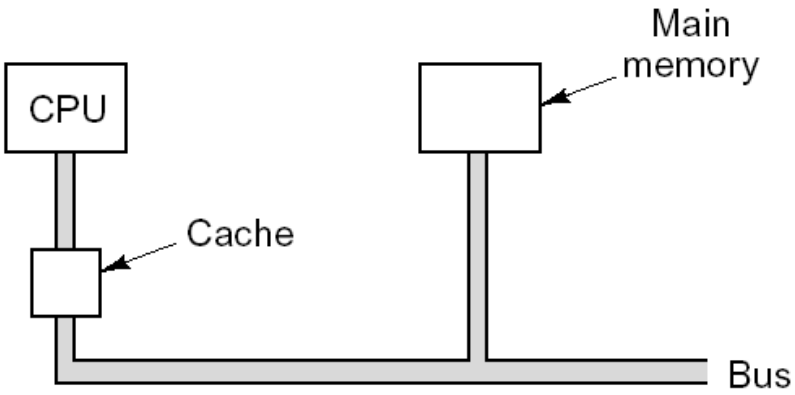




Nota: *Aquisição das Instruções e Operandos deve ser realizada na maioria dos casos da Cache, uma percentagem reduzida da memória principal e só ocasionalmente da memória secundária (disco rígido).*

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.



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 %

(b) c/ Cache 12,5ns; s/ Cache 85ns

UNIDADE DE MEMÓRIA

Memória Cache

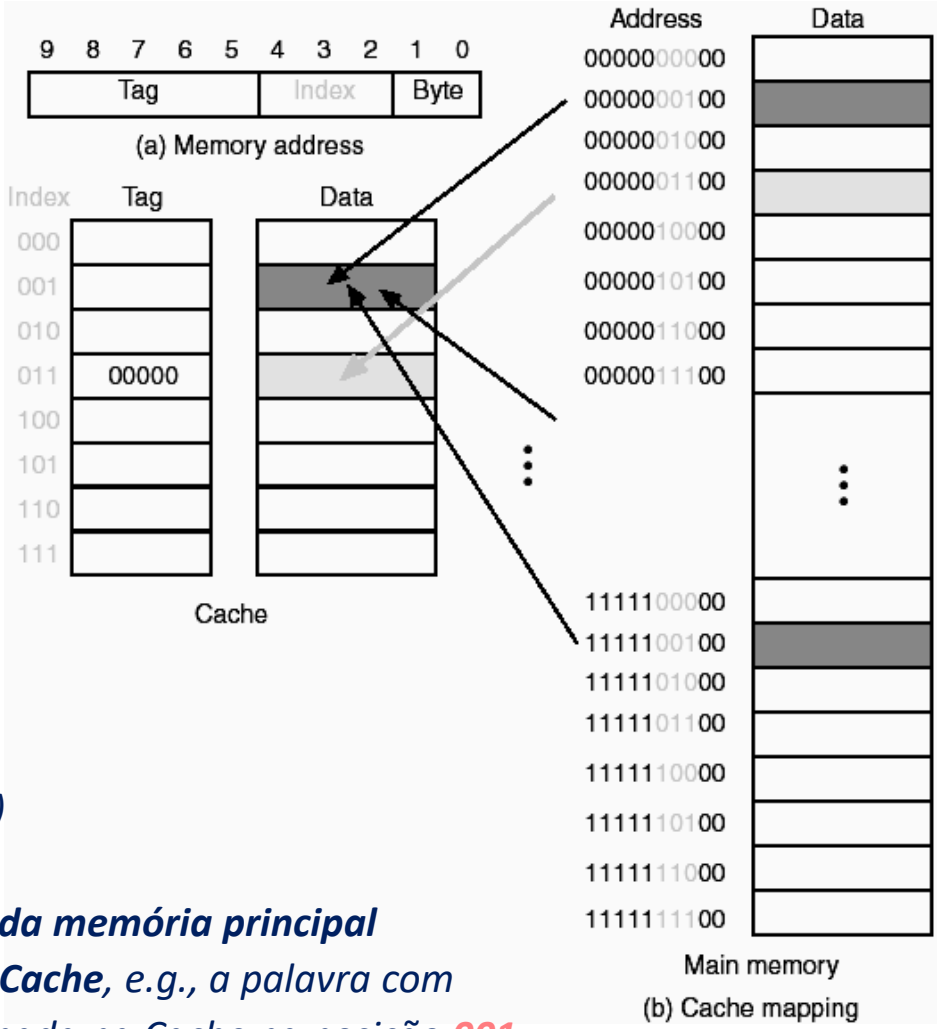
Memória Cache: Mapeamento Directo

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 – U*ma qualquer* ***palavra da memória principal*** *pode estar contida* ***em apenas uma posição da Cache****, e.g., a palavra com endereço* ***10101001 00*** *apenas pode ser armazenada na Cache na posição* ***001****.*

UNIDADE DE MEMÓRIA

Memória Cache

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

Memória Cache: Mapeamento Directo

Exemplo:

ENDEREÇO DE MEM.

9 8 7 6 5	4 3 2	1 0
ETIQUETA	INDEX	BYTE

- "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		[104,105,106,107]
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

Memória Cache: Mapeamento Directo

Exemplo:

ENDEREÇO DE MEM.

9 8 7 6 5	4 3 2	1 0
ETIQUETA	INDEX	BYTE

- "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

UNIDADE DE MEMÓRIA

Memória Cache

Memória Cache: Mapeamento Directo

Exemplo:

ENDEREÇO DE MEM.

9 8 7 6 5	4 3 2	1 0
ETIQUETA	INDEX	BYTE

- "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

UNIDADE DE MEMÓRIA

Memória Cache

Memória Cache: Mapeamento Directo

Exemplo:

ENDEREÇO DE MEM.

9 8 7 6 5	4 3 2	1 0
ETIQUETA	INDEX	BYTE

- "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

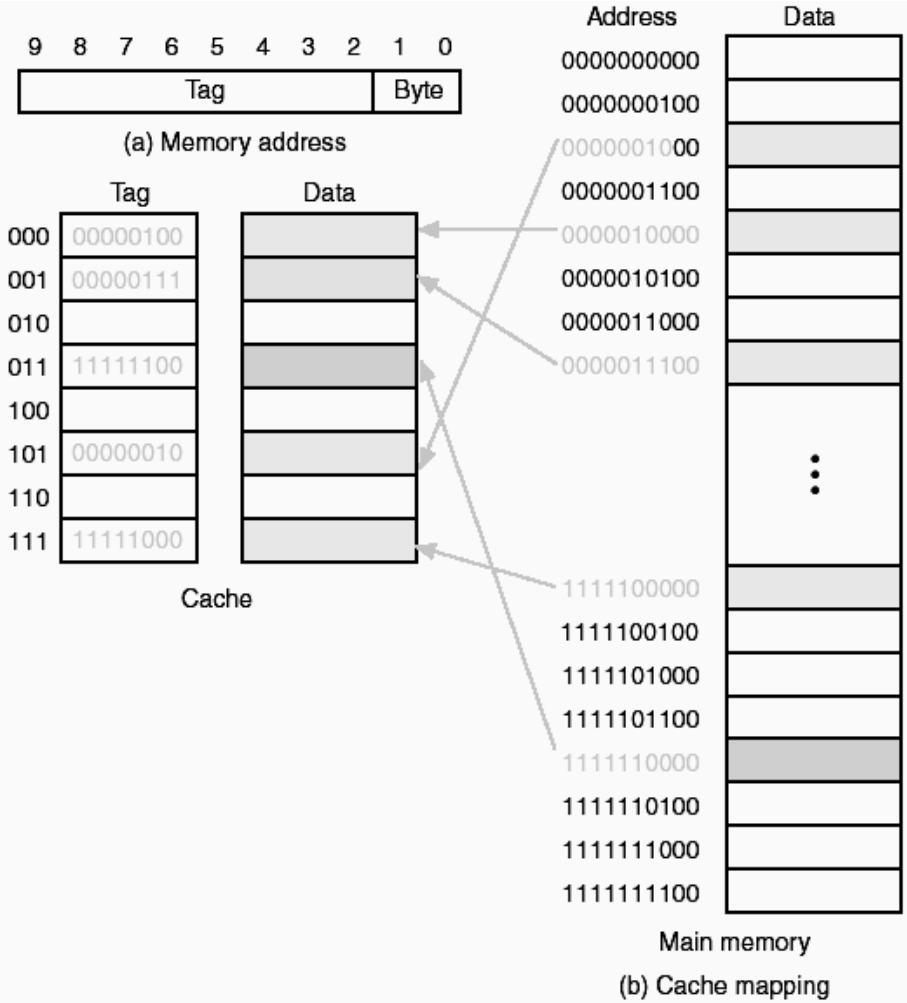
Memória Cache: Mapeamento Associativo

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.



UNIDADE DE MEMÓRIA

Memória Cache

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

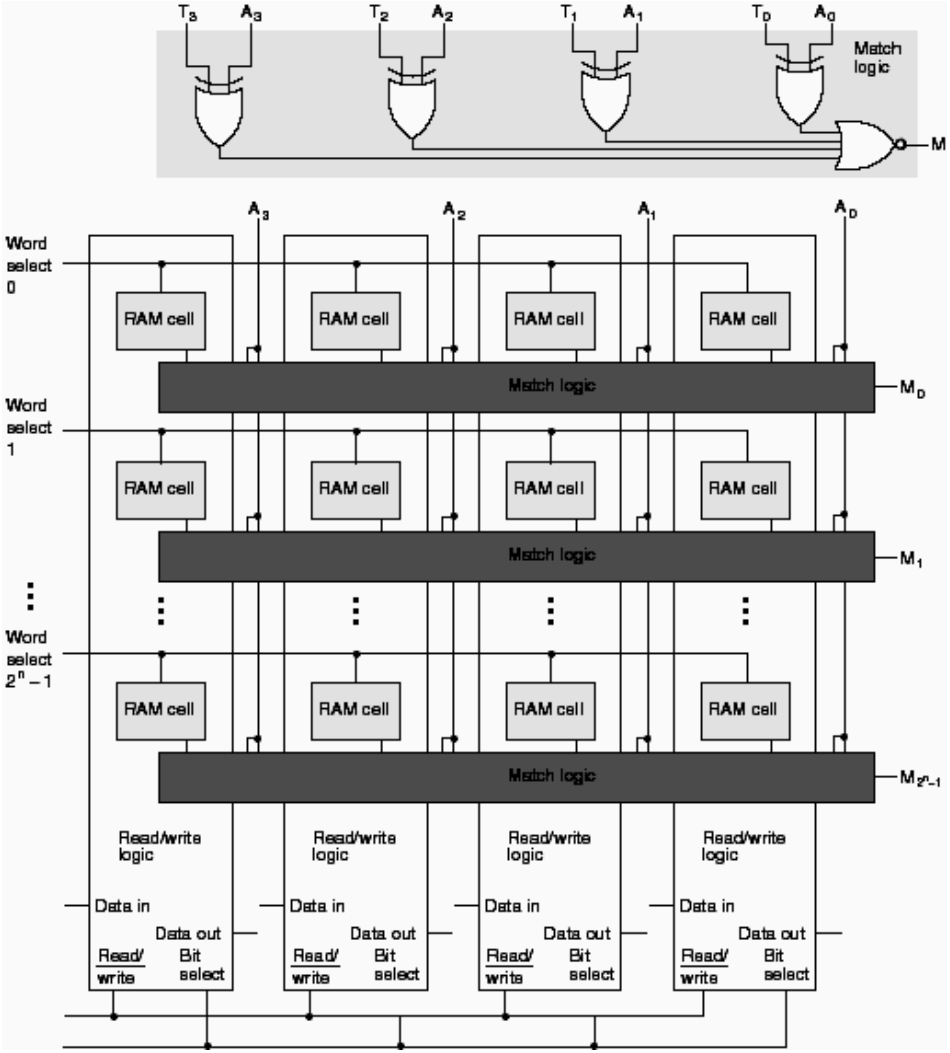
Política de Substituição das TAGs e Dados na Cache:

(1) Aleatória

(2) FIFO – First In First Out

(3) LRU – Least Recent Used

...



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
ETIQUETA	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	H
64	00 0110 0100	H

INDEX	ETIQUETA	DATA
000		
001		
010		
011		
100		
101		
110		
111		

M. CACHE

Memória Cache: Mapeamento Associativo

Exemplo:

ENDEREÇO DE MEM.

9 8 7 6 5 4 3 2	1 0
ETIQUETA	BYTE

- "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	H
64	00 0110 0100	H

INDEX	ETIQUETA	DATA
000		[54,55,56,57]
001		[58,59,5A,5B]
010		[104,105,106,107]
011		[5C,5D,5E,5F]
100		[108,109,10A,10B]
101		[60,61,62,63]
110		[F0,F1,F2,F3]
111		[64,65,66,67]

M. CACHE

UNIDADE DE MEMÓRIA

Memória Cache

Memória Cache: Mapeamento Associativo

Exemplo:

ENDEREÇO DE MEM.

9 8 7 6 5 4 3 2	1 0
ETIQUETA	BYTE

- "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	H
64	00 0110 0100	H

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

M. CACHE

Memória Cache: Mapeamento Associativo

Exemplo:

ENDEREÇO DE MEM.

9 8 7 6 5 4 3 2	1 0
ETIQUETA	BYTE

- "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	H
64	00 0110 0100	H

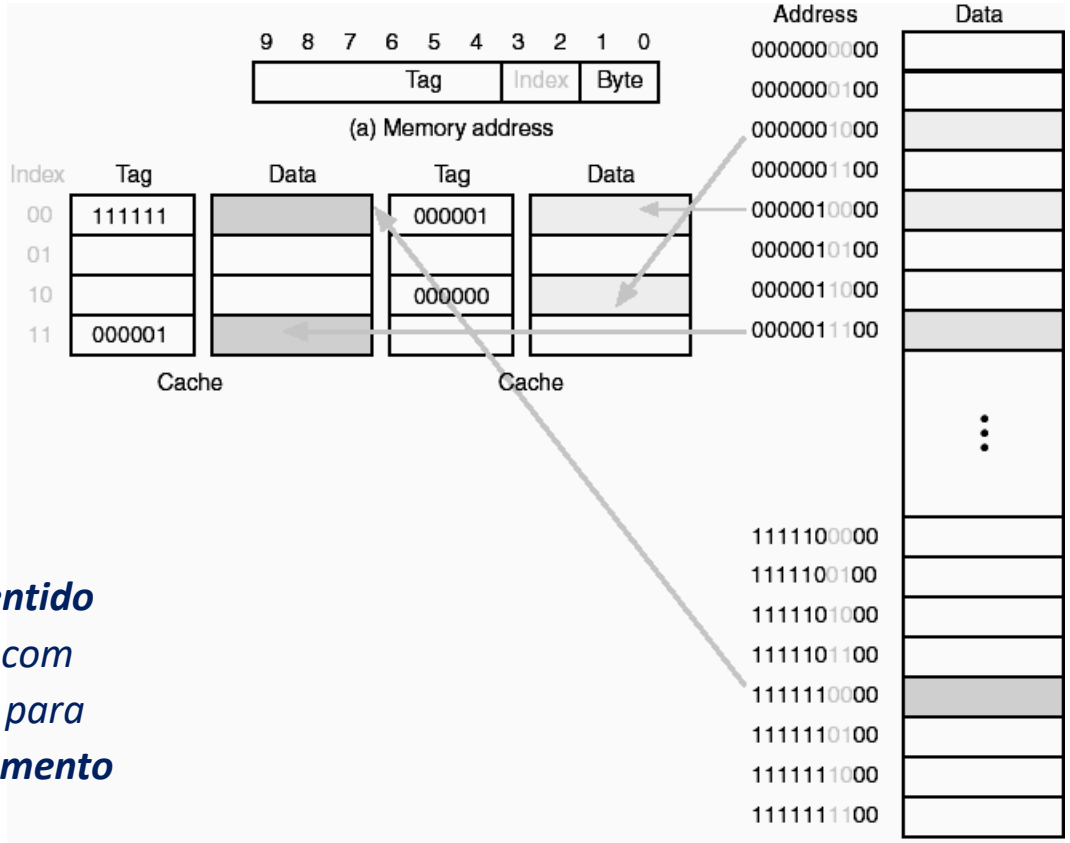
INDEX	ETIQUETA	DATA
000		[54,55,56,57]
001		[58,59,5A,5B]
010		[10C,10D,10E,10F]
011		[5C,5D,5E,5F]
100		[110,111,112,113]
101		[60,61,62,63]
110		[F0,F1,F2,F3]
111		[64,65,66,67]

M. CACHE

UNIDADE DE MEMÓRIA

Memória Cache

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



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

UNIDADE DE MEMÓRIA

Memória Cache

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.

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				
01				
10				
11				

M. CACHE

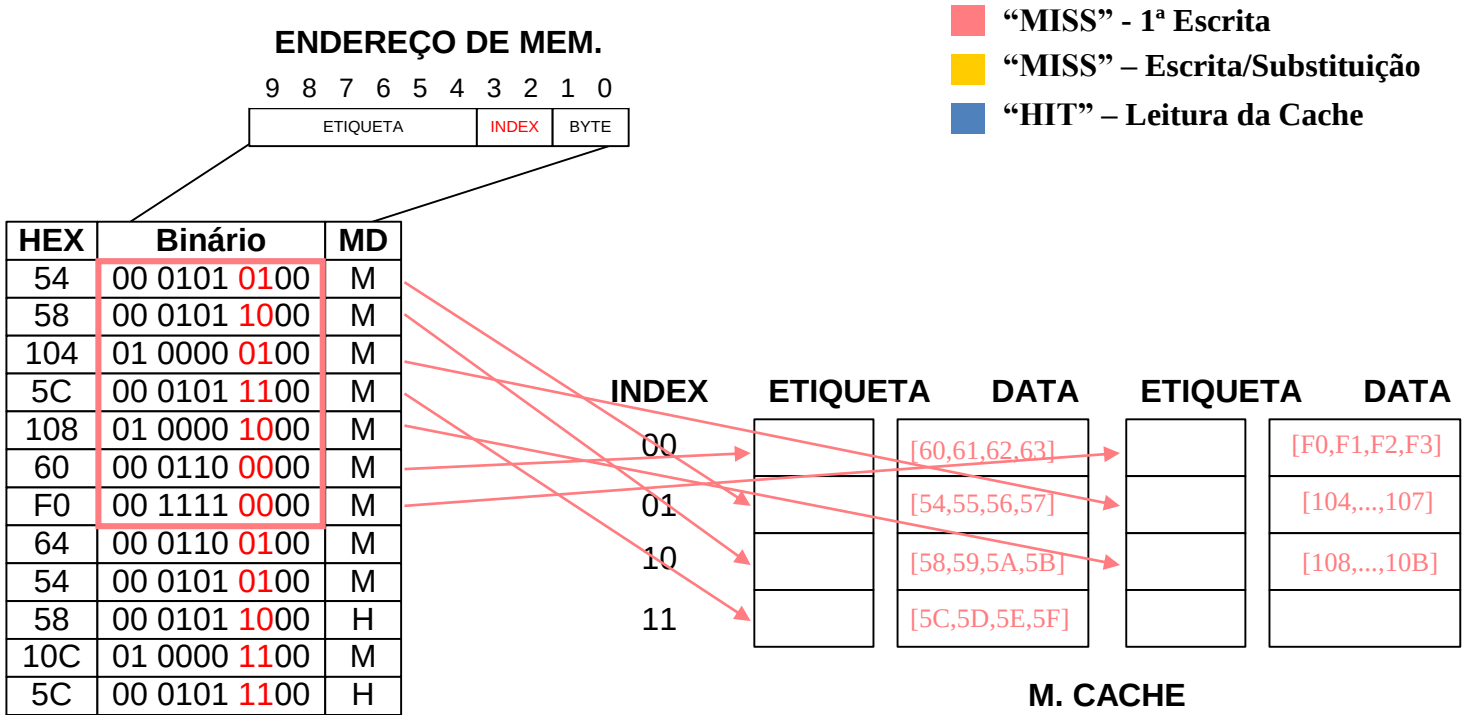
DEEC
DEPARTAMENTO
ELETROTÉCNICA E
TÉCNICO LISBOA

UNIDADE DE MEMÓRIA

Memória Cache

Memória Cache: Mapeamento Associativo-Directo

Exemplo:



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

UNIDADE DE MEMÓRIA

Memória Cache

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

M. CACHE

UNIDADE DE MEMÓRIA

Memória Cache

Memória Cache: Mapeamento Associativo-Directo

Exemplo:

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

ENDEREÇO DE MEM.

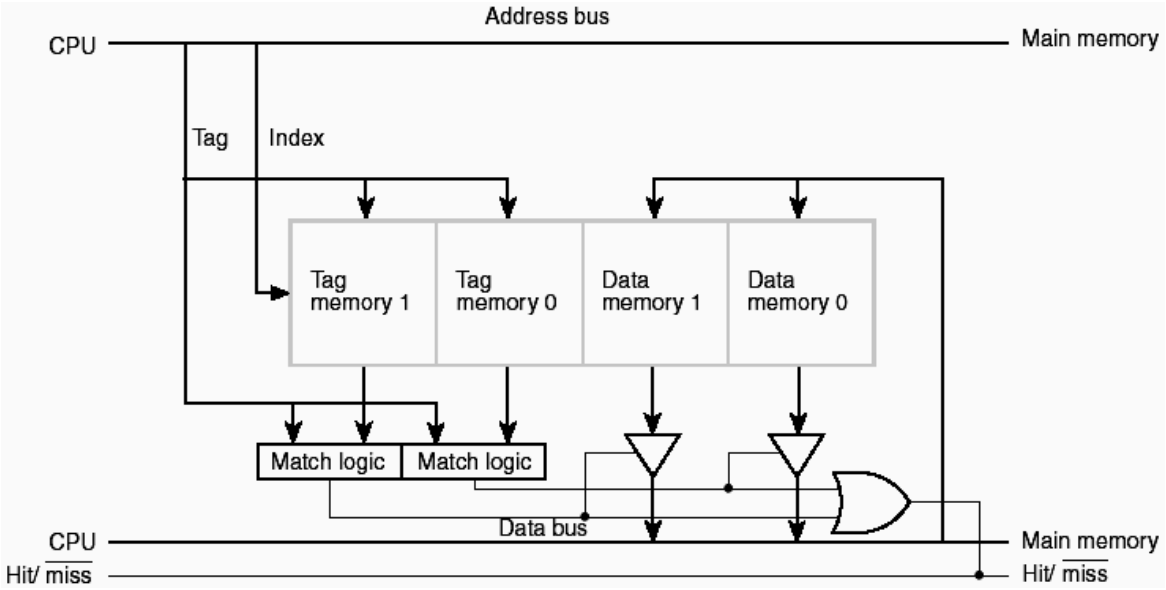
9 8 7 6 5 4	3 2	1 0
ETIQUETA	INDEX	BYTE

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

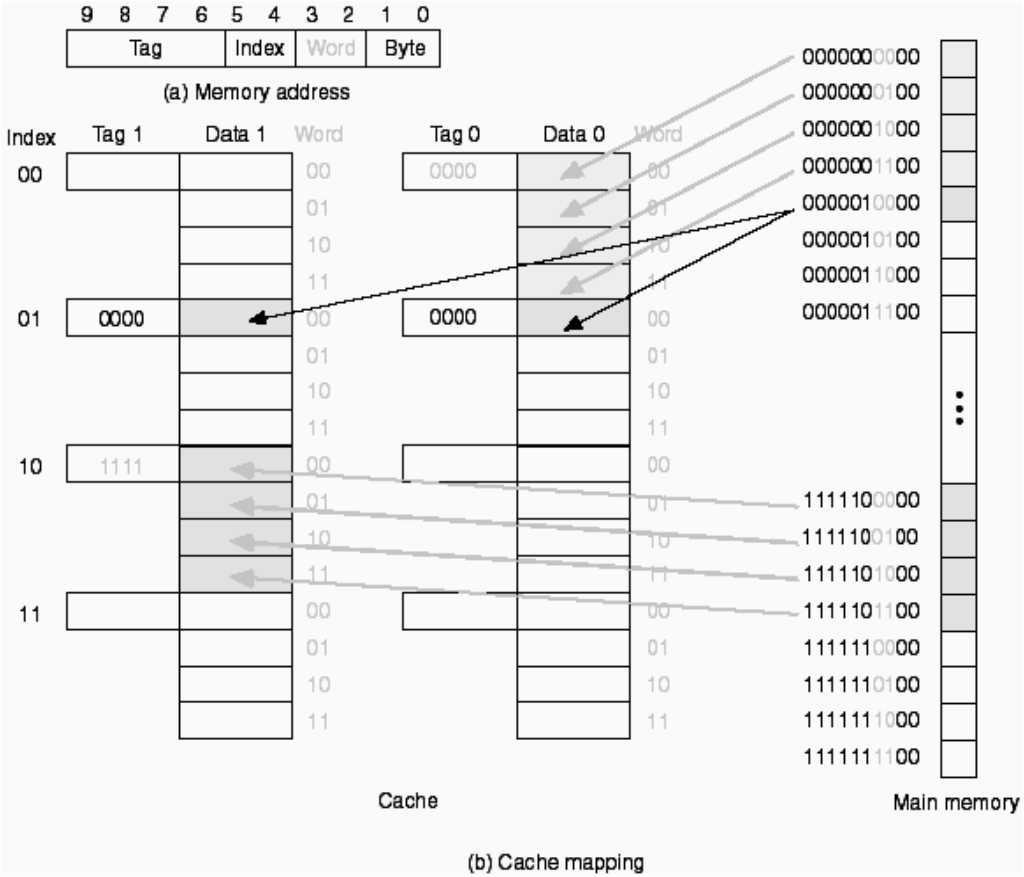
Memória Cache: Mapeamento Associativo-Directo (Set-associative Mapping)



UNIDADE DE MEMÓRIA

Memória Cache

Memória Cache: "Set-associative Mapping" c/ n Palavras por Etiqueta.



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 – *Resultado é escrito sempre na memória principal.*

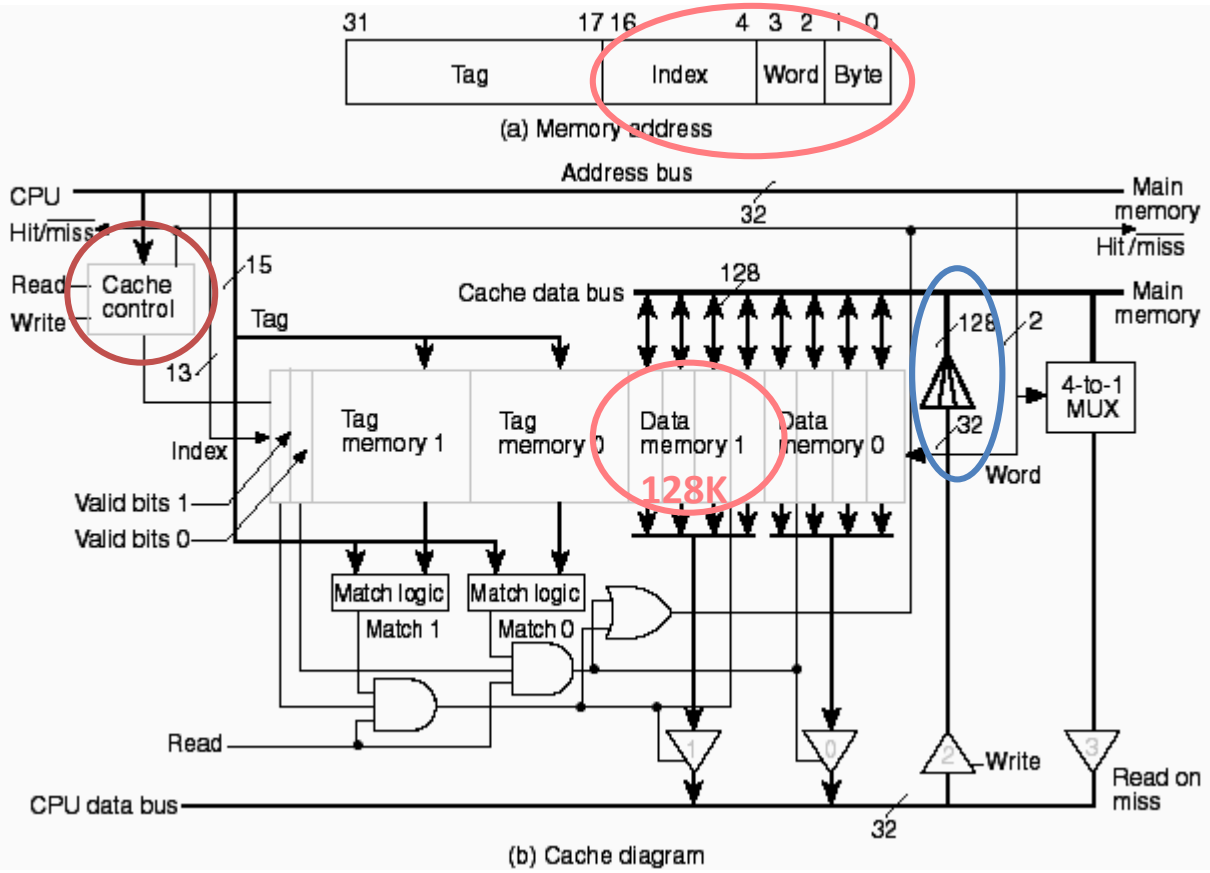
Write Back/ Copy Back – *Resultado escrito na memória cache em caso de um "**cache hit**", no caso de um "**cache miss**" é realizada uma escrita em memória principal **ou**, **Write Allocate**, escrita em memória principal e na memória cache (futuras escritas no mesmo bloco não necessitam de aceder à memória!).*

UNIDADE DE MEMÓRIA

Memória Cache

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

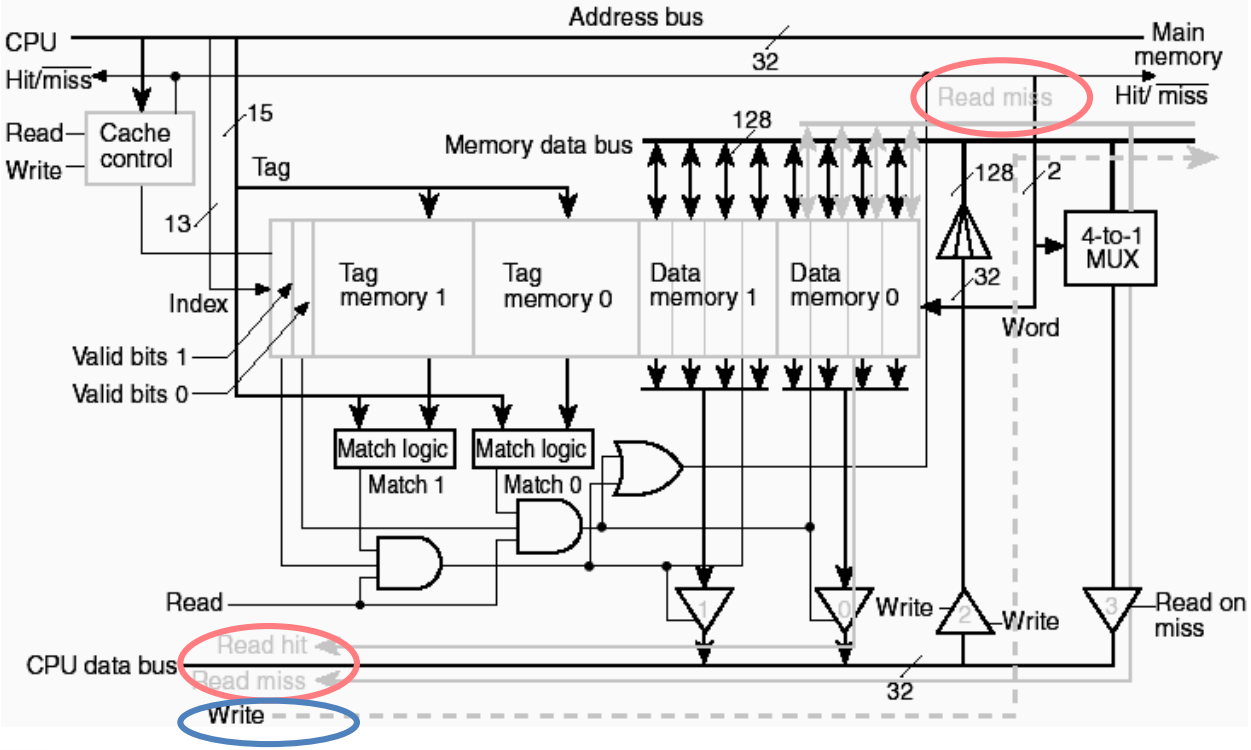
Mem. Cache de 256K



UNIDADE DE MEMÓRIA Memória Cache

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

Mem. Cache de 256K



UNIDADE DE MEMÓRIA

Memória Cache

Memória Cache:

Cache de Instruções e Dados

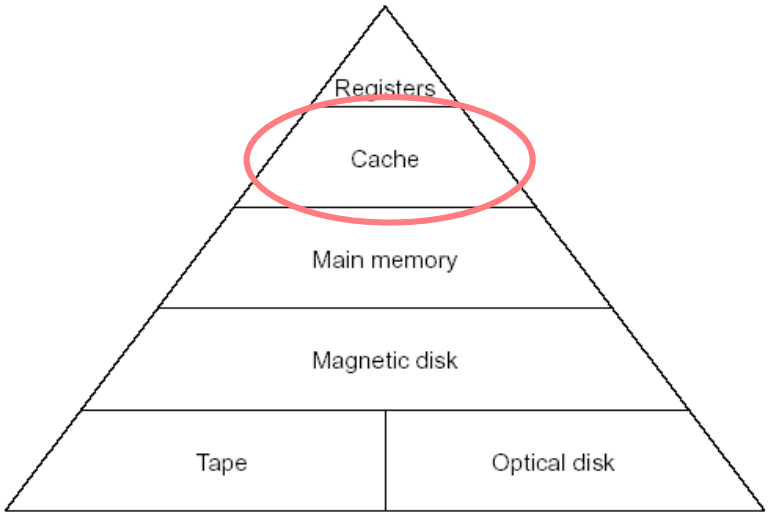
vs

Cache de Instruções e Cache de Dados

Cache Multinível: (Maior velocidade, controlo mais complexo)

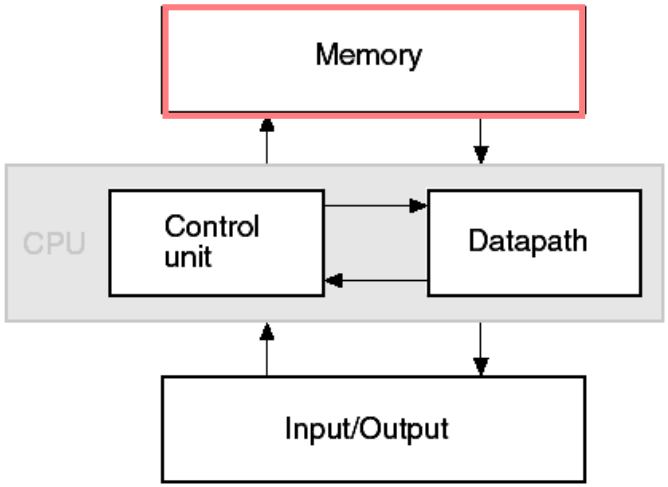
L1 (cache interna à CPU)

L2 (cache externa)

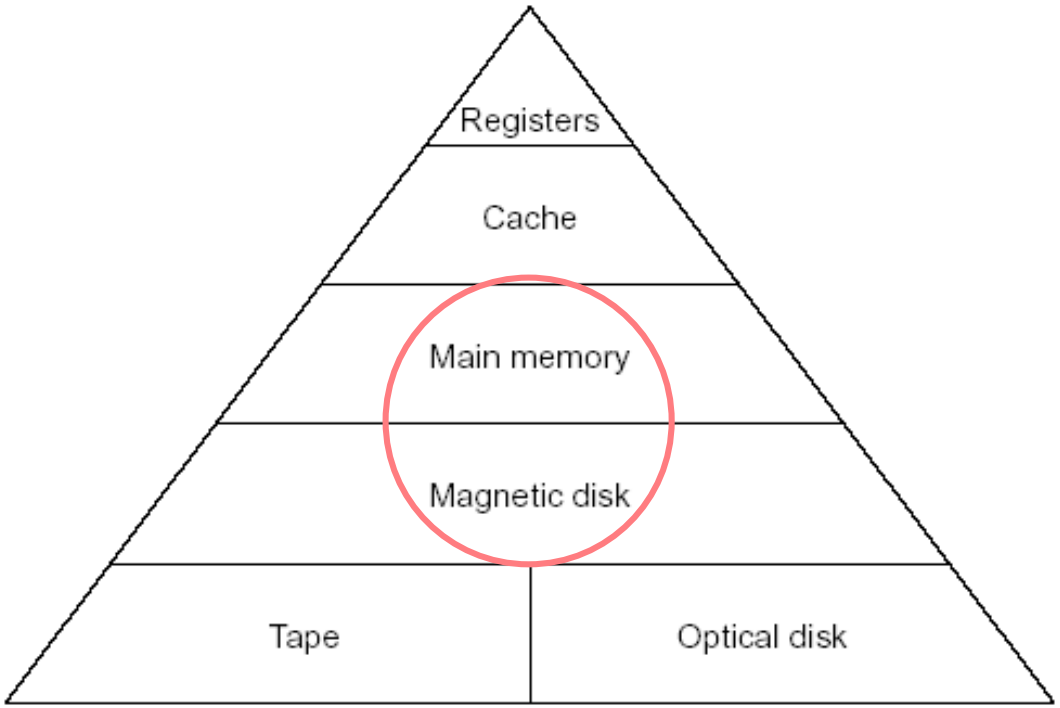


SUMÁRIO

- *Unidade de Memória*
  - *Hierarquia de Memória*
  - *Memória Cache*
  - *Memória Virtual*
  - *Arrays de Discos Rígidos*



Memória Virtual



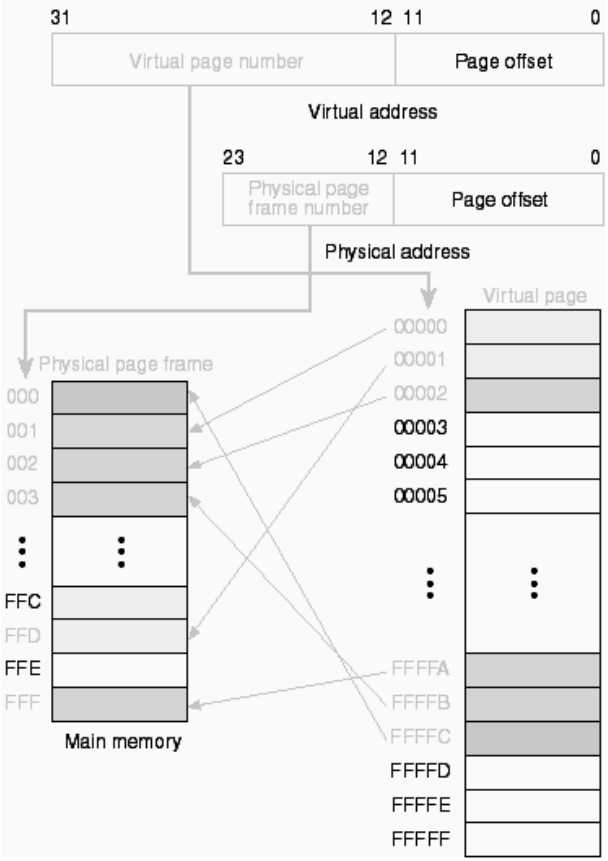
UNIDADE DE MEMÓRIA

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.*



UNIDADE DE MEMÓRIA

Memória Virtual

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:

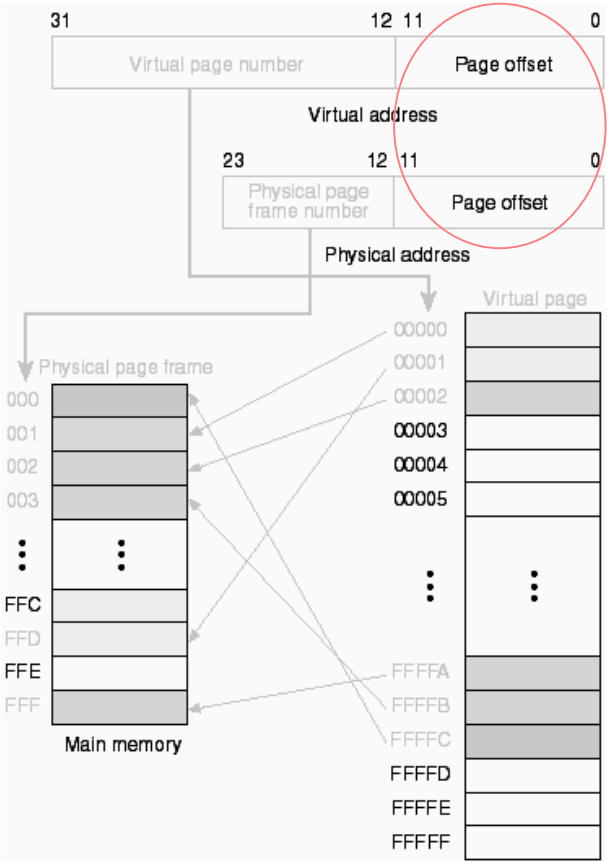
Espaço de Endereçamento Virtual: 32 Bits (2^{32})

Page: 2^{12} = 4K bytes (1K words de 32 bits)

Nº Páginas Virtuais: 2^{20} =(2^{32} /2^{12})

Memória Principal: 16 MBytes

Page Frames: 2^{12} =(2^{24} /2^{12})



UNIDADE DE MEMÓRIA

Memória Virtual

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:

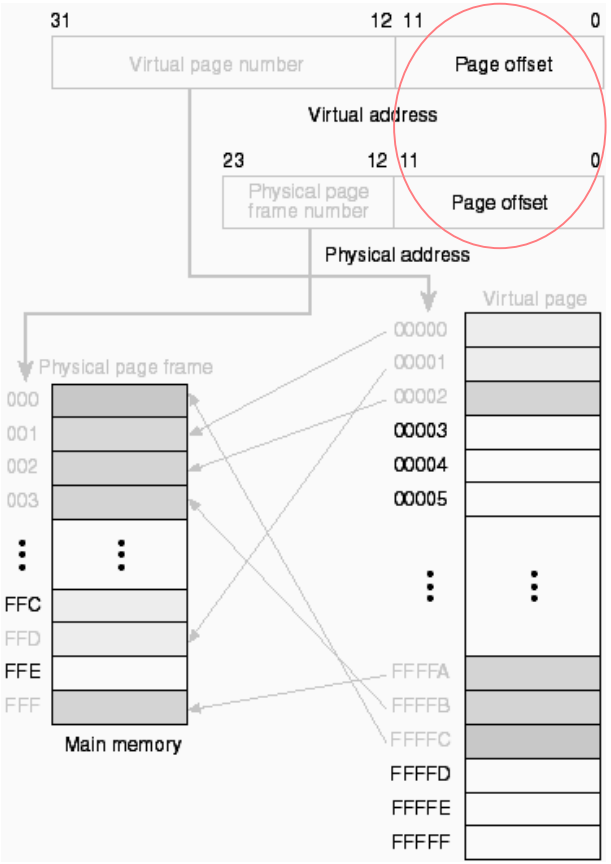
Espaço de Endereçamento Virtual: 32 Bits (2^{32}) – 4 GBytes

Page: 2^{12} = 4K bytes (1K words de 32 bits)

Nº Páginas Virtuais: 2^{20} =(2^{32} /2^{12})

Memória Principal: 1 GBytes

Page Frames: 2^{18} =(2^{30} /2^{12})



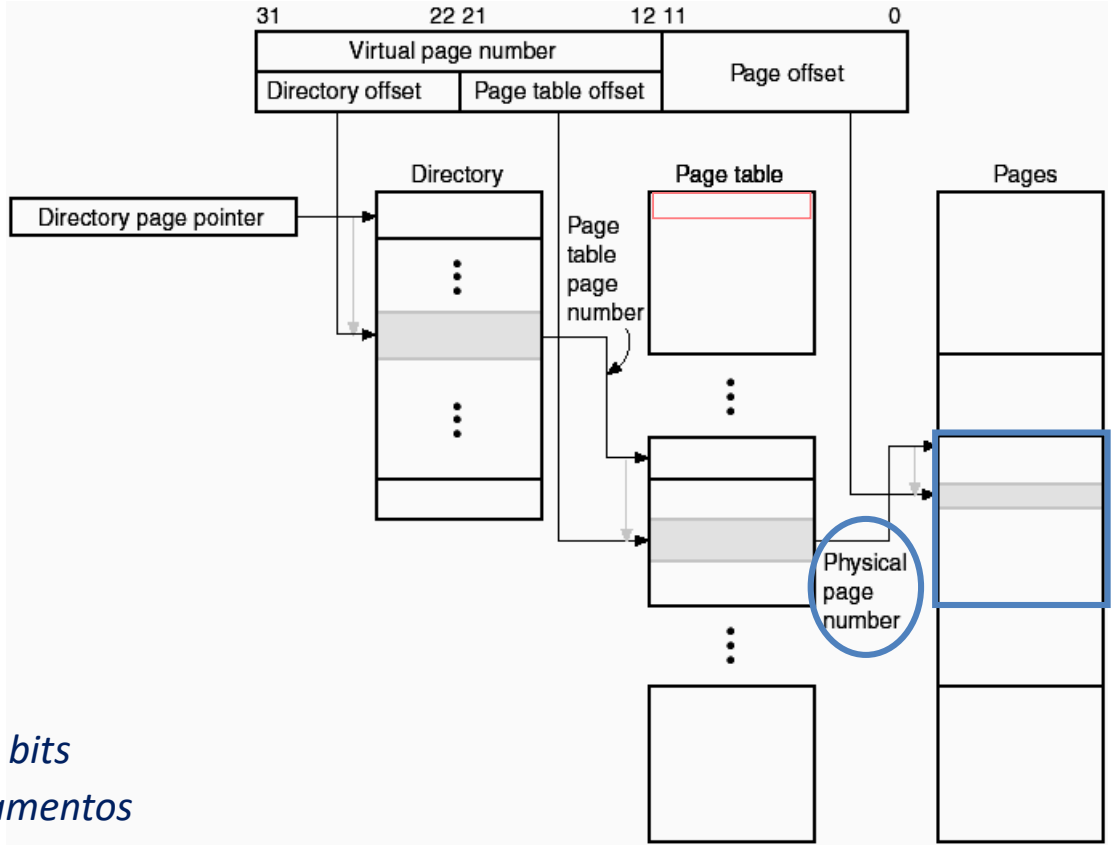
UNIDADE DE MEMÓRIA

Memória Virtual

Memória Virtual: Paginação

Page Tables: *Mapeamento de Páginas Virtuais em Memória e em HD. Cada PT tem* ***(2^{10})*** *apontadores para páginas de mem.*

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} = 4KB X 2^{10}) mapeado com recurso a uma página (**page table**).*

UNIDADE DE MEMÓRIA

Memória Virtual

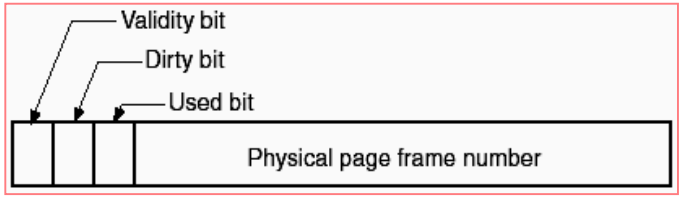
Memória Virtual: Paginação

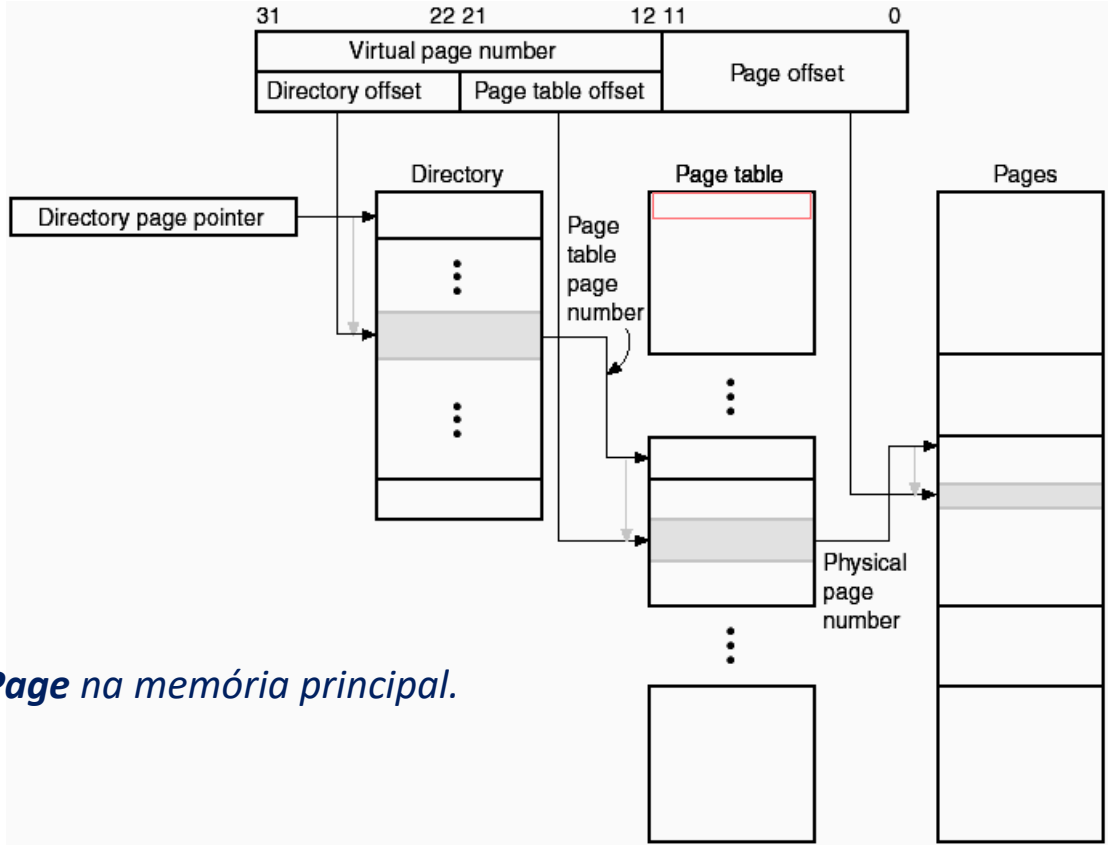
Validaty Bit: *Valida* ***Page Frame*** *em memória.*

Dirty Bit: *Indica se a página em memória foi escrita desde o seu carregamento. Implica atualização 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 (!!?)

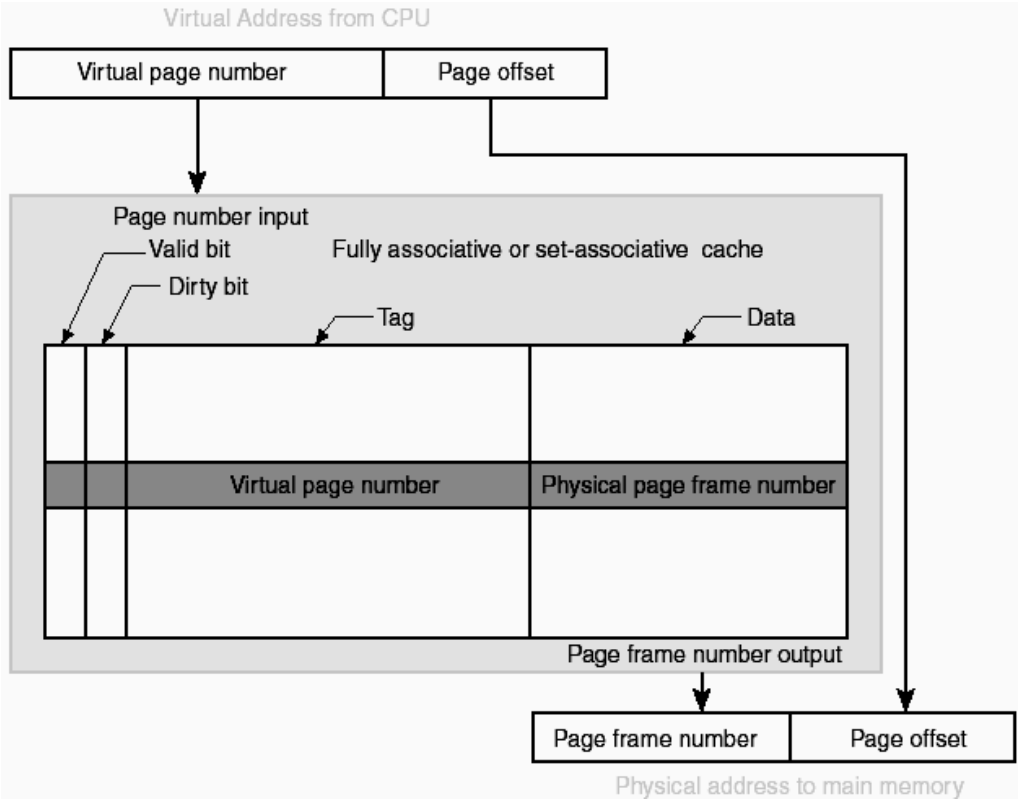
(1) Diretório

(2) Página

(3) Instrução

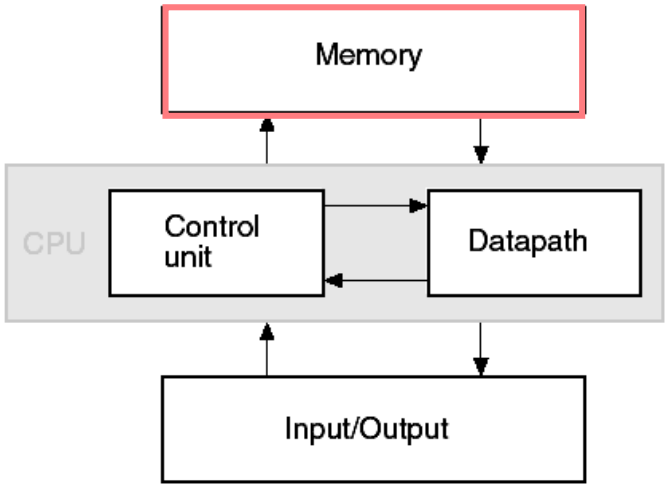
Introdução de nova Cache

para traduzir diretamente os endereços virtuais em endereços físicos. Guarda a localização das páginas mais recentemente acedidas.



SUMÁRIO

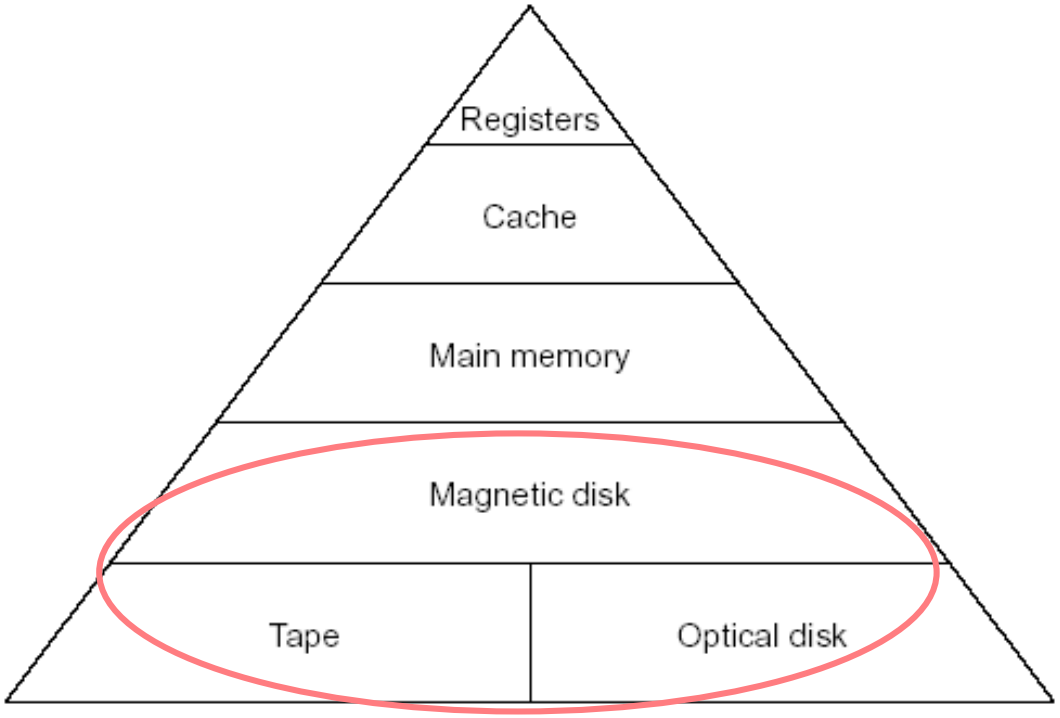
- *Unidade de Memória*
  - *Hierarquia de Memória*
  - *Memória Cache*
  - *Memória Virtual*
  - *Arrays de Discos Rígidos*



UNIDADE DE MEMÓRIA

Arrays de Discos Rígidos

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

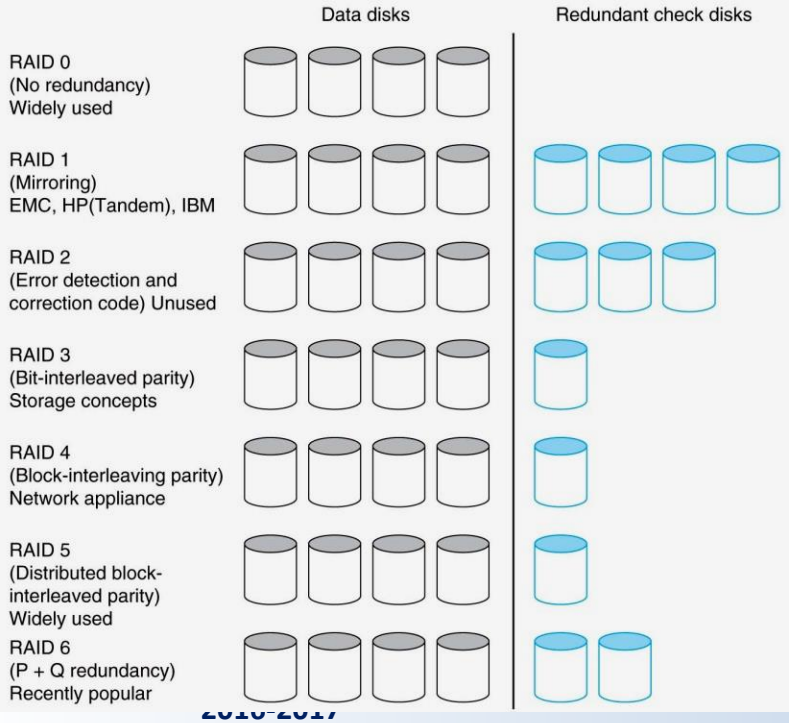


UNIDADE DE MEMÓRIA

Arrays de Discos Rígidos

- ***Arrays* de Discos –** Uma solução que permite aumentar a performance e a fiabilidade.
  - **Aumento do *throughput* devido ao aumento do número de drives e portanto braços de leitura.**
  - **A introdução de redundância no *array* de discos permite aumentar a fiabilidade de um conjunto de discos quando comparado com o único disco.**

- ***RAID – Redundant Array of Inexpensive (or Independent) Disks***



UNIDADE DE MEMÓRIA

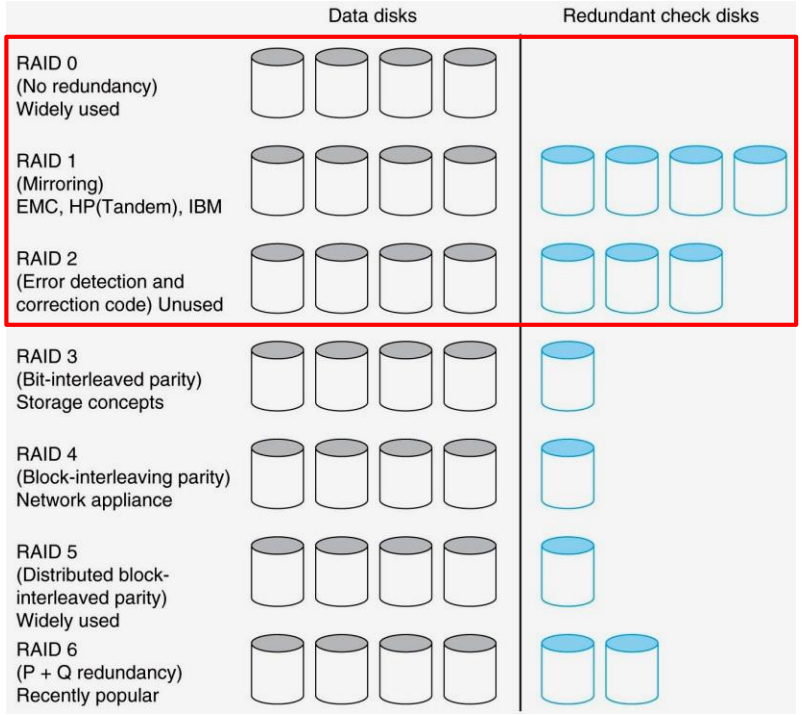
Arrays de Discos Rígidos

- ***RAID – Redundant Array of Inexpensive (or Independent) Disks***

- ***RAID 0 –*** Não tem redundância, utilizado como referência nas medidas de *performance* e fiabilidade para os restantes niveis.

- ***RAID 1 –*** Implementa a redundância de uma forma bastante simples, pois existem sempre 2 cópias dos dados, apresenta a desvantagem de ter um custo elevado.

- ***RAID 2 –*** Utiliza esquema de detecção e correcção utilizado em memórias (**não utilizado**)



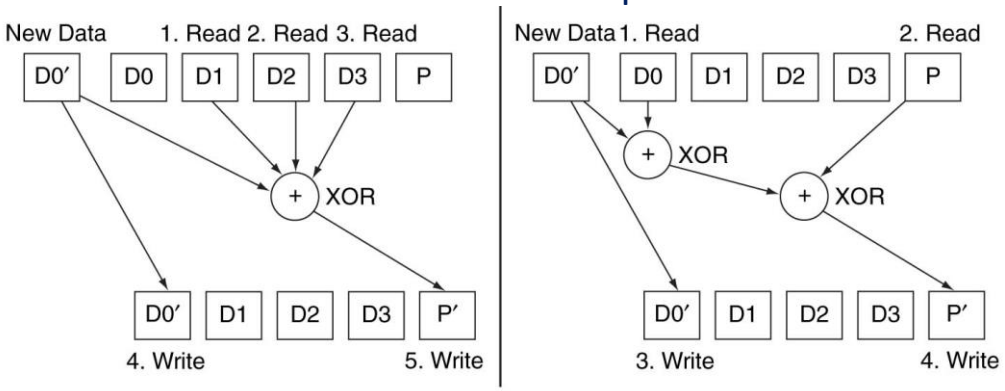
UNIDADE DE MEMÓRIA

Arrays de Discos Rígidos

- ***RAID – Redundant Array of Inexpensive (or Independent) Disks***

- ***RAID 3 –*** Utiliza um disco para Paridade, sendo esta determinada com base no dado novo e nos dados dos restantes discos. Leituras e escritas de grandes quantidades de dados.

- ***RAID 4 –*** Utiliza um disco para Paridade, determinada com base no valor antigo do disco que se vai alterar, no novo dado e no valor anterior da paridade. **Permite acessos independentes de leitura a vários discos**, possui controlo mais complexo.

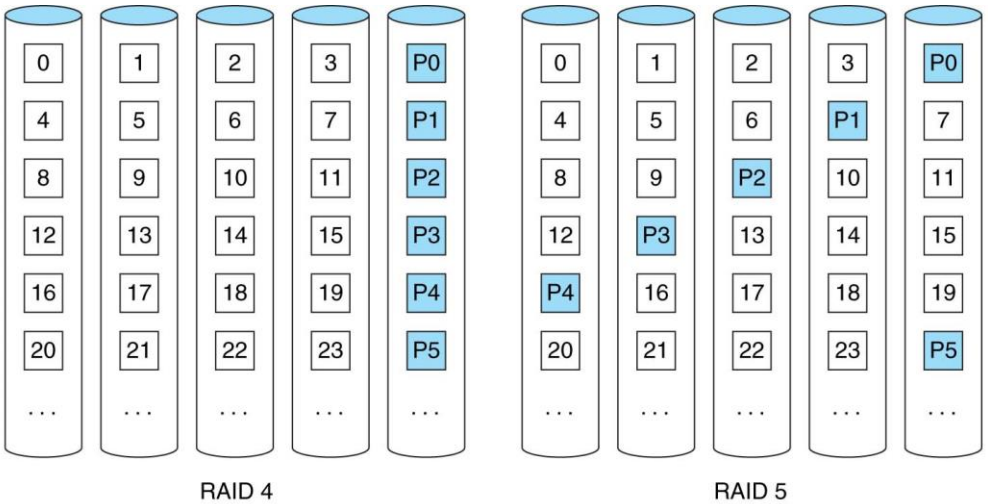


UNIDADE DE MEMÓRIA

Arrays de Discos Rígidos

- ***RAID – Redundant Array of Inexpensive (or Independent) Disks***

- ***RAID 5 –*** Distribui a informação de paridade ao longo dos vários discos do *array*. Deste modo, como a informação de paridade não está num só disco **podemos ter escritas simultâneas.**



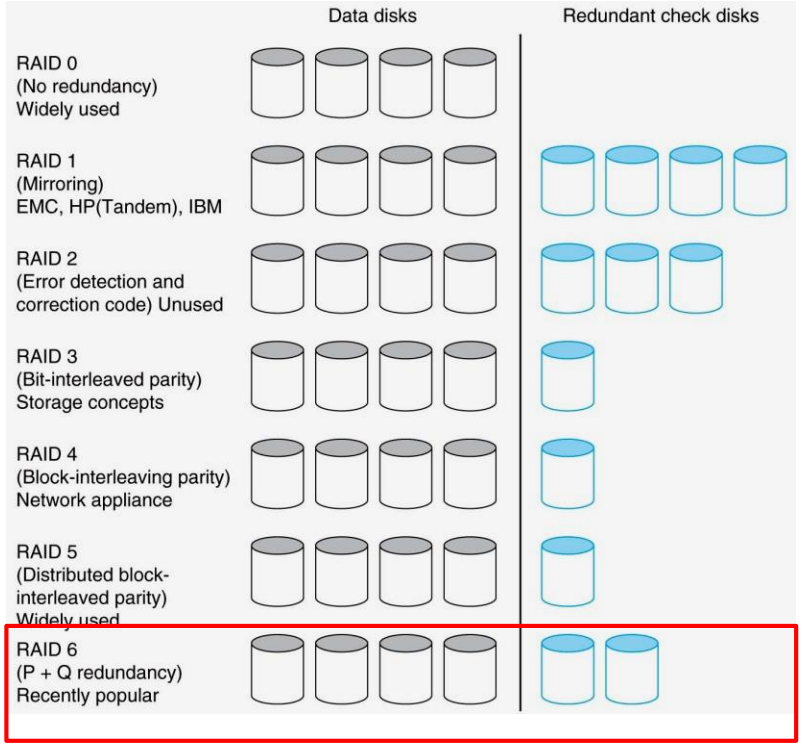
UNIDADE DE MEMÓRIA

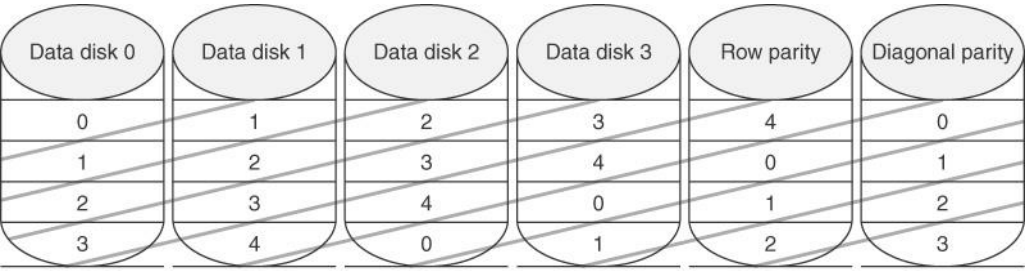
Arrays de Discos Rígidos

- ***RAID – Redundant Array of Inexpensive (or Independent) Disks***

- ***RAID 6 –*** Os esquemas de RAID de 1 a 5 consideram apenas a possibilidade de um disco falhar. Neste esquema considera-se a possibilidade de 2 falhas.

  RAID-DP (Row-Diagonal Parity) – Utiliza 2 discos adicionais e determina a paridade por linha e por diagonal.





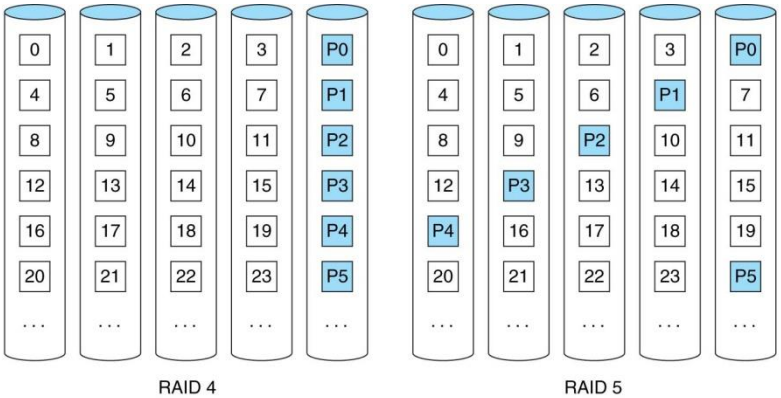
© 2007 Elsevier, Inc. All rights reserved.

UNIDADE DE MEMÓRIA

Arrays de Discos Rígidos

- ***RAID – Redundant Array of Inexpensive (or Independent) Disks***

- **Exemplo:** Determine uma sequência de acessos ao sistema de armazenamento de modo a tornar evidente as vantagens em termos de acessos do RAID 5 ao RAID 4 e de este ao RAID 3.



- **Exemplo:** Considere uma falha simultânea nos discos 1 e 3, em RAID 6. Descreva como podem ser recuperados os dados.

UNIDADE DE MEMÓRIA

Arrays de Discos Rígidos

- ***RAID – Redundant Array of Inexpensive (or Independent) Disks***

RAID level		Disk failures tolerated, check space overhead for 8 data disks	Pros	Cons	Company products
0	Nonredundant striped	0 failures, 0 check disks	No space overhead	No protection	Widely used
1	Mirrored	1 failure, 8 check disks	No parity calculation; fast recovery; small writes faster than higher RAIDs; fast reads	Highest check storage overhead	EMC, HP (Tandem), IBM
2	Memory-style ECC	1 failure, 4 check disks	Doesn't rely on failed disk to self-diagnose	~ Log 2 check storage overhead	Not used
3	Bit-interleaved parity	1 failure, 1 check disk	Low check overhead; high bandwidth for large reads or writes	No support for small, random reads or writes	Storage Concepts
4	Block-interleaved parity	1 failure, 1 check disk	Low check overhead; more bandwidth for small reads	Parity disk is small write bottleneck	Network Appliance
5	Block-interleaved distributed parity	1 failure, 1 check disk	Low check overhead; more bandwidth for small reads and writes	Small writes → 4 disk accesses	Widely used
6	Row-diagonal parity, EVEN-ODD	2 failures, 2 check disks	Protects against 2 disk failures	Small writes → 6 disk accesses; 2X check overhead	Network Appliance

UNIDADE DE MEMÓRIA Bibliografia

- **Bibliografia**

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

[2] G. Arroz, J. Monteiro, A. Oliveira, “Arquitectura de Computadores: dos Sistemas Digitais aos Microprocessadores”, IST Press, 2007.

- **Outras Referências**

[3] J. Hennessy, D. Patterson, “Computer Architecture – A Quantitative Approach”, Morgan Kaufmann, 2007.

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