Memória e Hierarquia de Memória

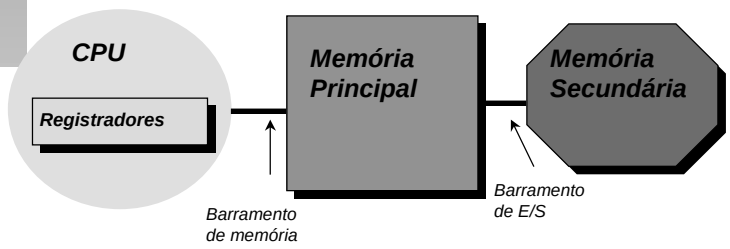
Memória Vs. Armazenamento

Fichário

Pasta

- *O fichário representa o disco rígido, com alta capacidade de armazenamento.*
- *A pasta sobre a mesa representa a memória, de acesso rápido e fácil*
- *Mesa e usuário são a CPU*
- *OBS: Memória é volátil e disco não.*
  - *Faxineira no final do expediente*

Sistema Hierárquico de Memória



Nomenclatura Básica

- *RAM = Random Acces Memory*
- *SRAM = Static RAM*
- *DRAM = Dynamic RAM*
- *VRAM - Video RAM*
- *ROM = Read Only Memory*
- *PROM = Programmable ROM*
- *EPROM = Erasable PROM*
- *EEPROM = Electrically Erasable PROM (apagamento byte a byte)*
- *Flash EPROM = Fast erasable EPROM (apagamento por bloco)*

Tipos Básicos de Memória Semicondutora

Tipo de Memória	*Categoria*	*Apagamento*	*Escrita*	*Volatilidade*
Random-Access Mem. (RAM)	**Read-Write**	**Elétrico byte a byte**	**Elétrica**	**Volátil**
Read-Only Mem. (ROM)	**Read-only**	**Impossível**	**Máscara**	**não-volátil**
Programmable ROM (PROM)			**Elétrica**	
Erasable PROM (EPROM)	**Read-mostly**	**Ultra-violeta**		
Electrically EPROM (EEPROM)		**Elétrico byte a byte**		
Flash EPROM		**Elétrico por bloco**		

RAM Dinâmica vs. Estática

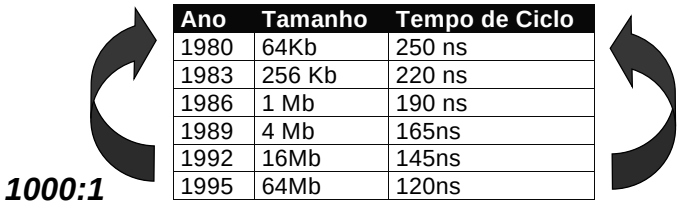
- *DRAM (Dynamic Random Acess Memory)*
  - ***Grande capacidade de integração (baixo custo por bit)***
  - ***Perda de informação após algum tempo: <u>Necessidade de refreshing</u>***
- *SRAM (Static Random Acess Memory)*
  - ***Pequeno tempo de acesso***
  - ***Não existe necessidade de refreshing***
  - ***Alto custo por bit (baixa integração)***

Tecnologias de Memória

Tecnologia	Ano	Tempo de acesso	U$/Mbyte(2001)
SRAM	1997	5-25 ns	175-400
	2001	2,5-12ns	
DRAM	1997	60-120 ns	10-20
	2001	7,5-50ns	1,5
Disco	1997	10-20 ms	0,20 – 0,40
	2001	7-12 ms	0,01

Memórias rápidas são caras
Memórias baratas são lentas

Evolução Tecnológica



Ano	Tamanho	Tempo de Ciclo
1980	64Kb	250 ns
1983	256 Kb	220 ns
1986	1 Mb	190 ns
1989	4 Mb	165ns
1992	16Mb	145ns
1995	64Mb	120ns

1000:1

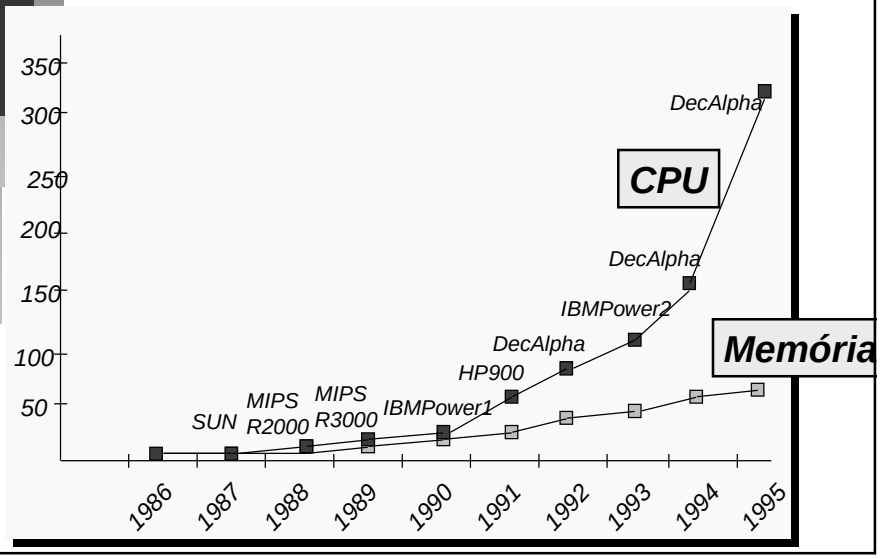
2:1

	Aumento da capacidade	Aumento da velocidade
SRAM	2x em 3 anos	2x em 3 anos
DRAM	4x em 3 anos	2x em 10 anos
Disco	4x em 3 anos	2x em 10 anos

Memória Principal

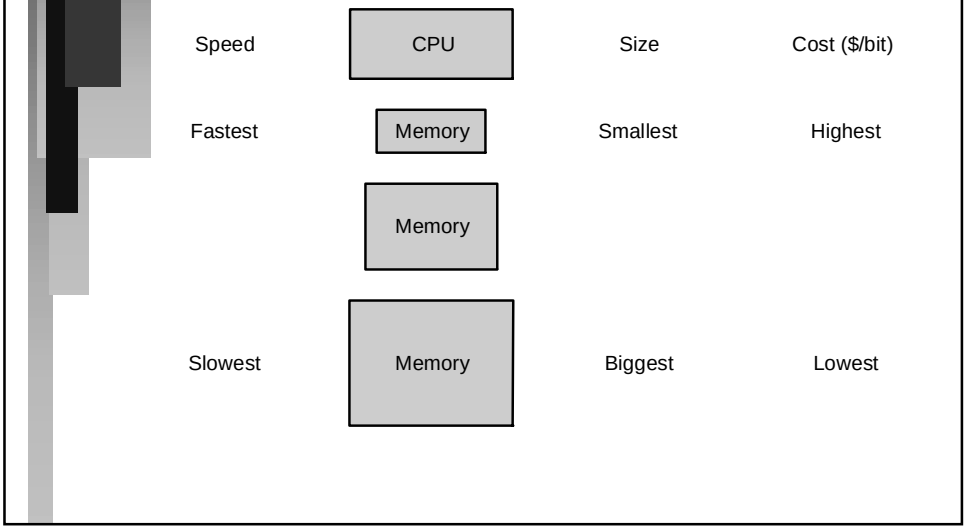
"640K ought to be enough for anybody."
Bill Gates, 1981

Comparação da Performance da Mem. Principal e CPU

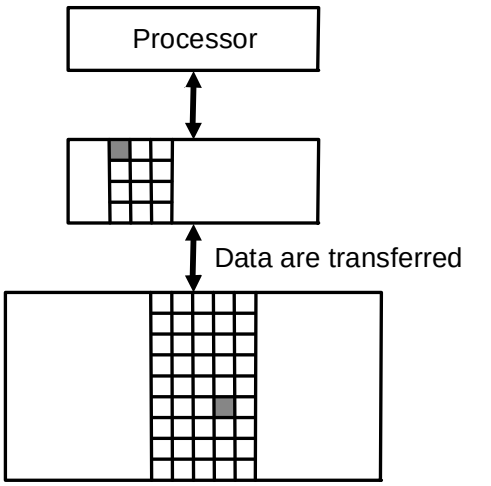


5

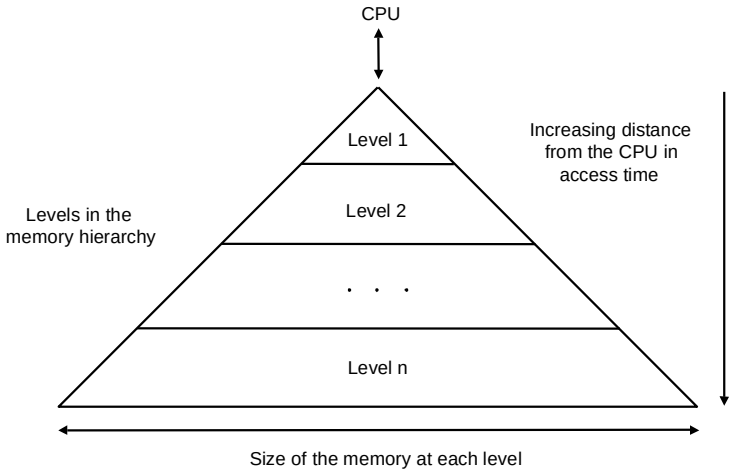
Sistema Hierárquico de Memória



Sistema Hierárquico de Memória

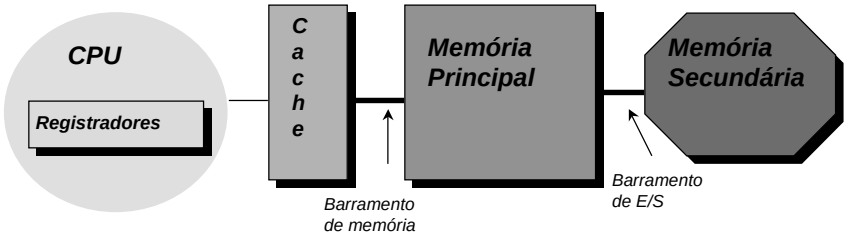


Sistema Hierárquico de Memória



Motivação para hierarquia

- *Principio da localidade + Relação custo/desempenho das tecnologias*
- *Alto desempenho da CPU*

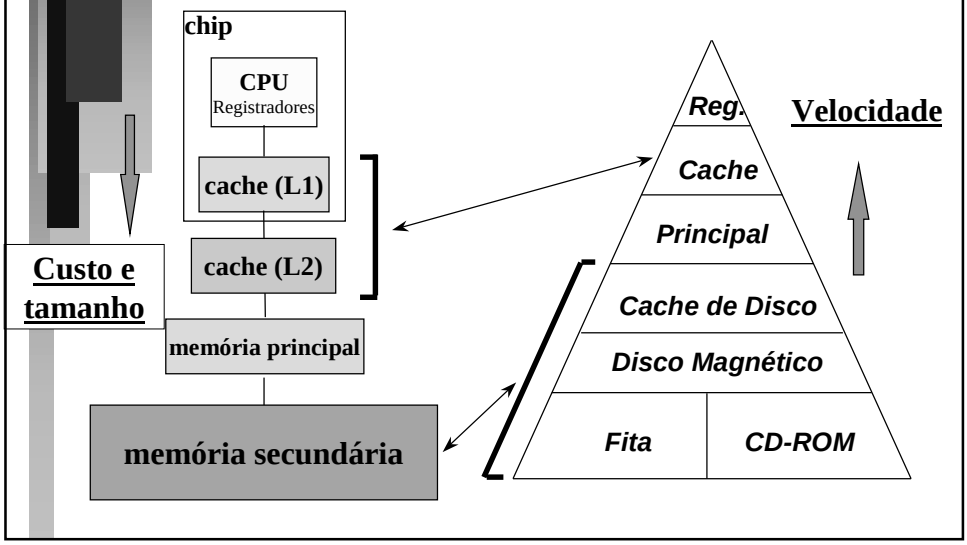


7

Princípio da Localidade

- ***Localidade Temporal***
  - *Num futuro próximo, o programa irá referenciar as instruções e dados referenciados recentemente*
- ***Localidade Espacial***
  - *Num futuro próximo, o programa irá referenciar as instruções e dados que tenham endereços próximos das últimas referências.*

Hierarquia de Memória

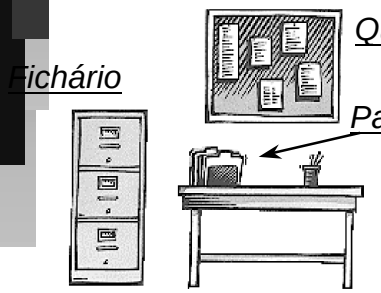


Níveis de memória

Nível	1	2	3	4
Nome	**Registrador**	**Cache**	**Memória Principal**	**Secundária**
Tamanho	**< 1K**	**< 4 M**	**< 4 G**	**> 1 G**
Tecnologia	**BICMOS**	**SRAM**	**DRAM**	**Disco**
Tempo de acesso (ns)	**2-5**	**3-10**	**80-400**	**5.000.000**
Largura de banda(MB/s)	**4000-32.000**	**800-5000**	**400-2000**	**4-32**
Gerência	**Compilador**	**Hardware**	**S.O**	**S.O / usuário**
Copia em	**Cache**	**Memória Principal**	**Disco**	**Fita**

Memória Cache

Memória Principal Vs. Cache



- *O fichário representa o disco rígido.*
- *A pasta sobre a mesa representa a memória principal.*
- *No quadro de avisos se encontram informações que podem ser acessadas de forma muito rápida. O quadro representa a cache.*
- *Mesa e usuário são a CPU*

Cache Simples

- *Unidade de transferência: palavra*

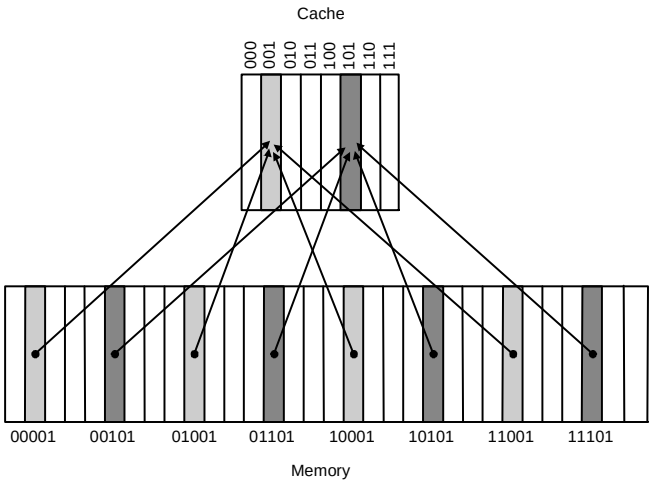
X4
X1
Xn – 2
Xn – 1
X2
X3

a. Before the reference to Xn

X4
X1
Xn – 2
Xn – 1
X2
Xn
X3

b. After the reference to Xn

Mapeamento direto



Endereçando a cache

- *Composição do endereço*
  - *Tag*
  - *Índice*
  - *Endereço de byte*
- *Exemplo:*
  - *Memória: endereço de 32 bits, acesso por palavra(32 bits), endereçamento por byte*
  - *Cache: capacidade para armazenar 64 palavras*

11

Acessando a memória cache

Endereço decimal	Endereço binário	Hit ou miss na cache	Bloco na cache
22	10110	Miss	10110 mod 8 =110
26	11010	Miss	11010 mod 8 =010
22	10110	Hit	10110 mod 8 =110
26	11010	Hit	11010 mod 8 =010
16	10000	Miss	10000 mod 8 =000
3	00011	Miss	00011 mod 8 =011
16	10000	Hit	10000 mod 8 =000
18	10010	miss	10010 mod 8 =010

Acessando a memória cache

Índice	V	Tag	Dado
000	N		
001	N		
010	N		
011	N		
100	N		
101	N		
110	N		
111	N		

Estado inicial da cache

Endereço decimal	Endereço binário	Hit ou miss na cache	Bloco na cache
22	10110	Miss	10110 mod 8 =110
26	11010	Miss	11010 mod 8 =010
22	10110	Hit	10110 mod 8 =110
26	11010	Hit	11010 mod 8 =010
16	10000	Miss	10000 mod 8 =000
3	00011	Miss	00011 mod 8 =011
16	10000	Hit	10000 mod 8 =000
18	10010	miss	10010 mod 8 =010

Acessos à memória

12

Acessando a memória cache

Índice	V	Tag	Dado
000	N		
001	N		
010	N		
011	N		
100	N		
101	N		
110	Y	10	Memória(101110)
111	N		

Endereço 10110

Endereço decimal	*Endereço binário*	*Hit ou miss na cache*	*Bloco na cache*
22	10110	Miss	10110 mod 8 =110
26	11010	Miss	11010 mod 8 =010
22	10110	Hit	10110 mod 8 =110
26	11010	Hit	11010 mod 8 =010
16	10000	Miss	10000 mod 8 =000
3	00011	Miss	00011 mod 8 =011
16	10000	Hit	10000 mod 8 =000
18	10010	miss	10010 mod 8 =010

Acessos à memória

Acessando a memória cache

Índice	V	Tag	Dado
000	N		
001	N		
010	Y	11	Memória(11010)
011	N		
100	N		
101	N		
110	Y	10	Memória(101110)
111	N		

Endereço 11010

Endereço decimal	*Endereço binário*	*Hit ou miss na cache*	*Bloco na cache*
22	10110	Miss	10110 mod 8 =110
26	11010	Miss	11010 mod 8 =010
22	10110	Hit	10110 mod 8 =110
26	11010	Hit	11010 mod 8 =010
16	10000	Miss	10000 mod 8 =000
3	00011	Miss	00011 mod 8 =011
16	10000	Hit	10000 mod 8 =000
18	10010	miss	10010 mod 8 =010

Acessos à memória

13

Acessando a memória cache

Índice	V	Tag	Dado
000	Y	10	Memória(10000)
001	N		
010	Y	11	Memória(11010)
011	N		
100	N		
101	N		
110	Y	10	Memória(101110)
111	N		

Endereço 10000

Endereço decimal	*Endereço binário*	*Hit ou miss na cache*	*Bloco na cache*
22	10110	Miss	10110 mod 8 =110
26	11010	Miss	11010 mod 8 =010
22	10110	Hit	10110 mod 8 =110
26	11010	Hit	11010 mod 8 =010
16	10000	Miss	10000 mod 8 =000
3	00011	Miss	00011 mod 8 =011
16	10000	Hit	10000 mod 8 =000
18	10010	miss	10010 mod 8 =010

Acessos à memória

Acessando a memória cache

Índice	V	Tag	Dado
000	Y	10	Memória(10000)
001	N		
010	Y	11	Memória(11010)
011	Y	00	Memória(00011)
100	N		
101	N		
110	Y	10	Memória(101110)
111	N		

Endereço 00011

Endereço decimal	*Endereço binário*	*Hit ou miss na cache*	*Bloco na cache*
22	10110	Miss	10110 mod 8 =110
26	11010	Miss	11010 mod 8 =010
22	10110	Hit	10110 mod 8 =110
26	11010	Hit	11010 mod 8 =010
16	10000	Miss	10000 mod 8 =000
3	00011	Miss	00011 mod 8 =011
16	10000	Hit	10000 mod 8 =000
18	10010	miss	10010 mod 8 =010

Acessos à memória

14

Acessando a memória cache

Índice	V	Tag	Dado
000	Y	10	Memória(10000)
001	N		
010	Y	10	Memória(10010)
011	Y	00	Memória(00011)
100	N		
101	N		
110	Y	10	Memória(101110)
111	N		

Endereço 10010

Endereço decimal	*Endereço binário*	*Hit ou miss na cache*	*Bloco na cache*
22	10110	Miss	10110 mod 8 =110
26	11010	Miss	11010 mod 8 =010
22	10110	Hit	10110 mod 8 =110
26	11010	Hit	11010 mod 8 =010
16	10000	Miss	10000 mod 8 =000
3	00011	Miss	00011 mod 8 =011
16	10000	Hit	10000 mod 8 =000
18	10010	miss	10010 mod 8 =010

Acessos à memória

O que acontece numa falta de cache?

- *Informação deve ser lida da memória*
- *São inseridos ciclos de espera no pipeline até que a informação esteja disponível na cache*
  - *Penalidade*
- *Se o endereço de cache está ocupado, a informação é sobre-escrita*

15

Leitura/Escrita da Cache

- *Leitura:*
  - *Mais frequentes, rápidas e fáceis de implementar*
- *Escrita:*
  - *Mais lentas e complicadas e consistência de dados com a memória principal deve ser mantida (se um bloco da cache foi alterado pela CPU, não pode ser descartado da cache sem garantir que foi copiado para a mem. principal)*

Tipos de acesso à cache

- *Leitura*
- *Escrita*
  - *Dado e tag são atualizados na cache*
  - *Inconsistencia entre memória principal e cache!!*
  - *Como resolver?*

16

Políticas de Escrita e Consistência

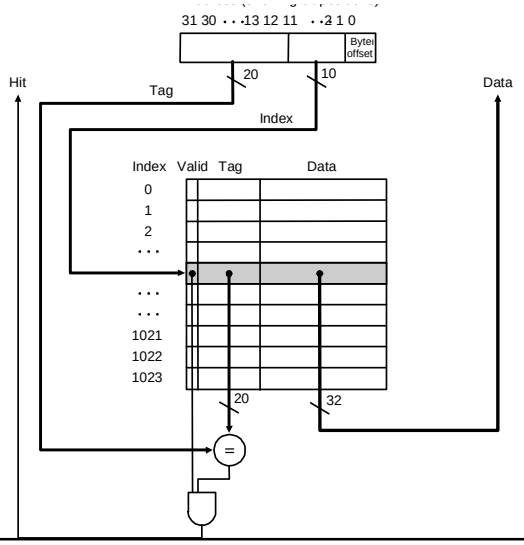
- *Caches do tipo Write through*
  - *Cache e memória são atualizadas simultaneamente*
- *Caches do tipo Write back*
  - *Memória principal é atualizada quando bloco é substituído*
  - *Usa dirty bit para marcar linhas alteradas na cache.*

Memória Cache: escrita

Write through	***Write back***
facilidade de implementação	***redução de acessos à memória***
consistência da memória principal	

- *Para se evitar espera durante escrita:*
  - *Write buffers*

Exemplo: DECStation3100



Tamanho da cache

- *Quantos bits tem uma cache de mapeamento direto com 64K bytes de dados e blocos de uma palavra? Assuma endereços de 32 bits.*

18

Usando a localidade espacial

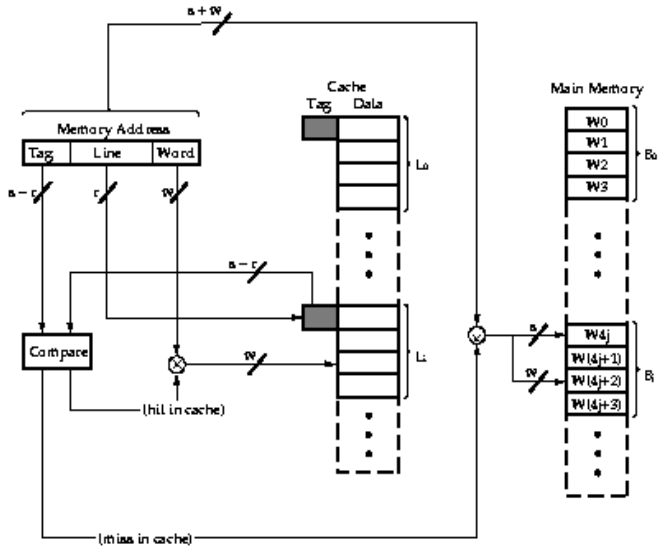
- *Acessando a cache por blocos de palavras*
- *Composição do endereço:*
  - *Tag*
  - *Índice*
  - *Offset de bloco*
  - *Endereço de byte*

Usando a localidade espacial

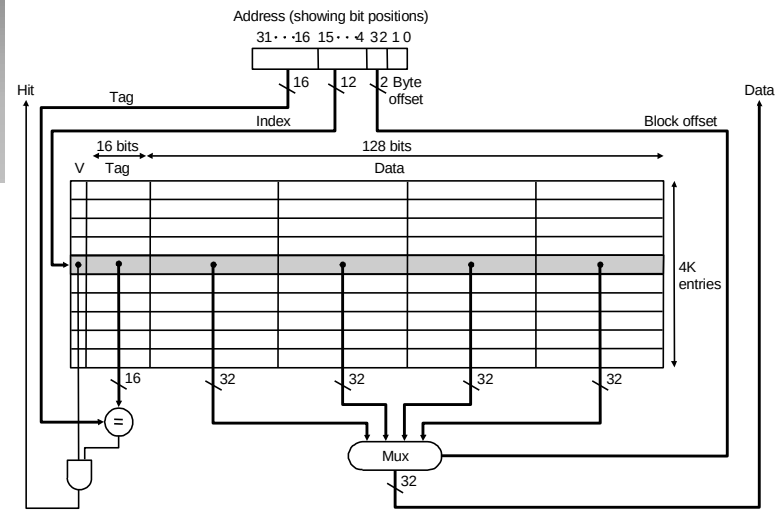
- *Exemplo:*
  - *Memória: endereço de 32 bits, acesso por palavra(32 bits), endereçamento por byte*
  - *Cache: capacidade para armazenar 64 blocos de 4 palavras cada*

19

Mapeamento Direto



Mapeamento Direto-multiword

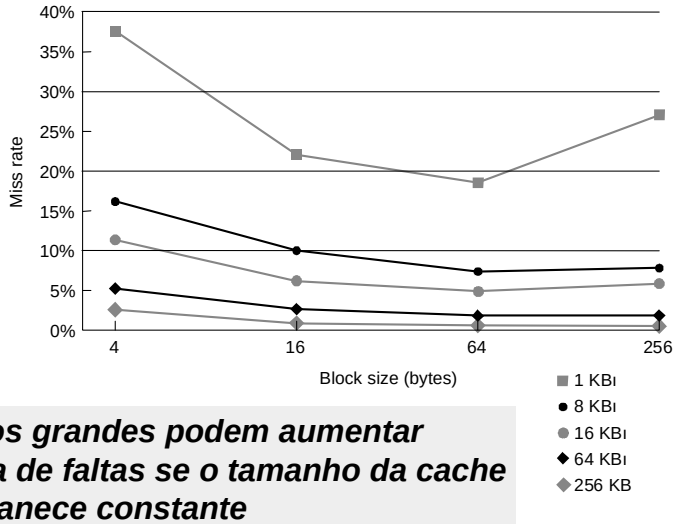


20

Usando a localidade espacial

- *O que acontece durante uma falta em acesso de leitura ou escrita?*
  - *Todo o bloco tem que ser carregado na cache*
  - *A escrita da palavra acontece*
  - *Cache write-through:*
    - *Todo o bloco é atualizado na memória*

Tamanho do bloco e taxa de faltas



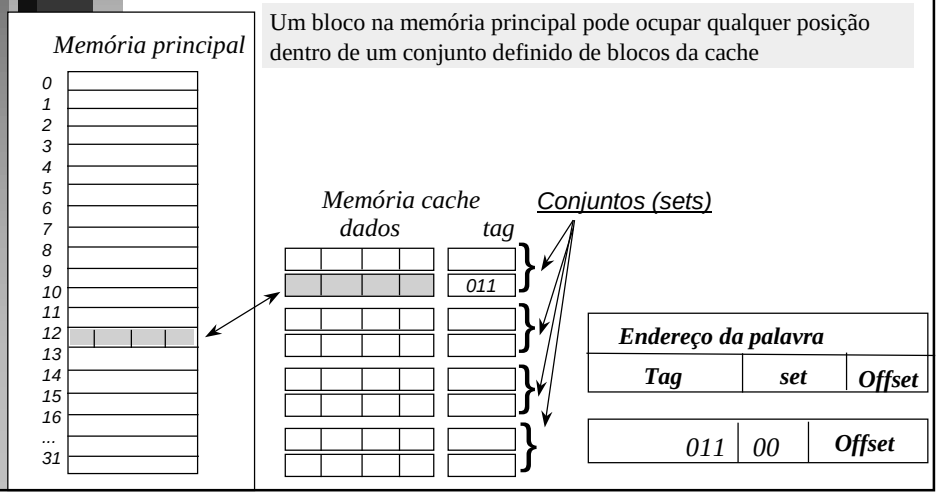
Blocos grandes podem aumentar a taxa de faltas se o tamanho da cache Permanece constante

21

Reduzindo a taxa de faltas

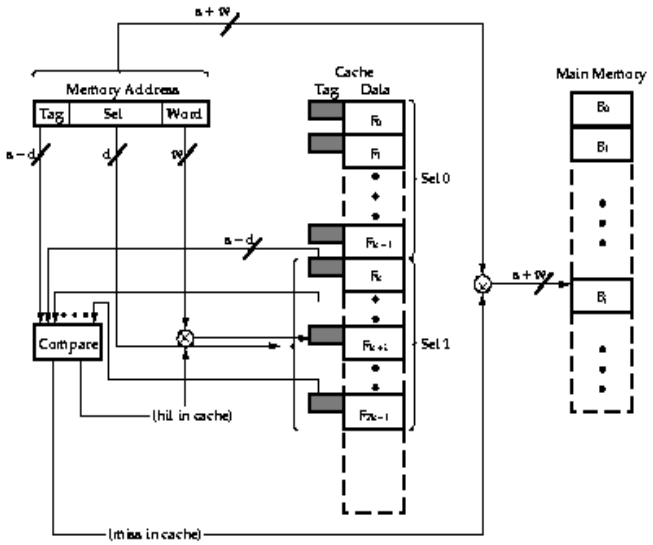
- *Estratégias para posicionamento dos blocos:*
  - *Mapeamento direto: cada bloco possui posição única na cache*
  - *Associativa por conjunto: cada bloco pode ser colocado em algumas posições na cache*
  - *Completamente Associativa: cada bloco pode ser colocado em qualquer posição da cache*

Mapeamento Associativo por Conjunto

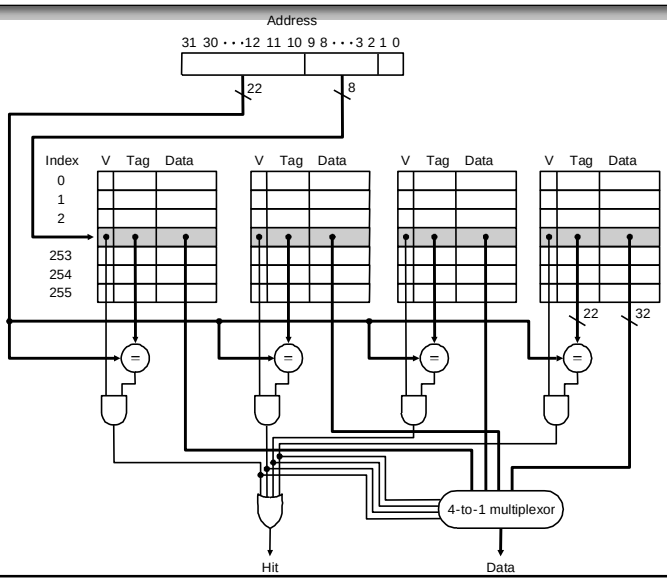


22

Mapeamento Associativo por Conjunto

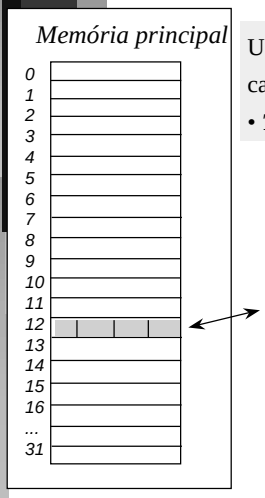


Mapeamento Associativo por Conjunto



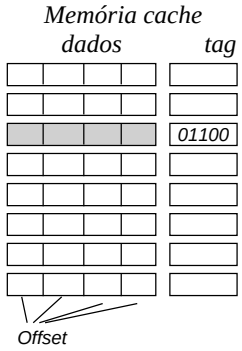
23

Mapeamento Associativo



Um bloco na memória principal pode ocupar qualquer posição na cache

- ***Tag ⇒ armazena na cache o end. Do bloco na mem. principal***

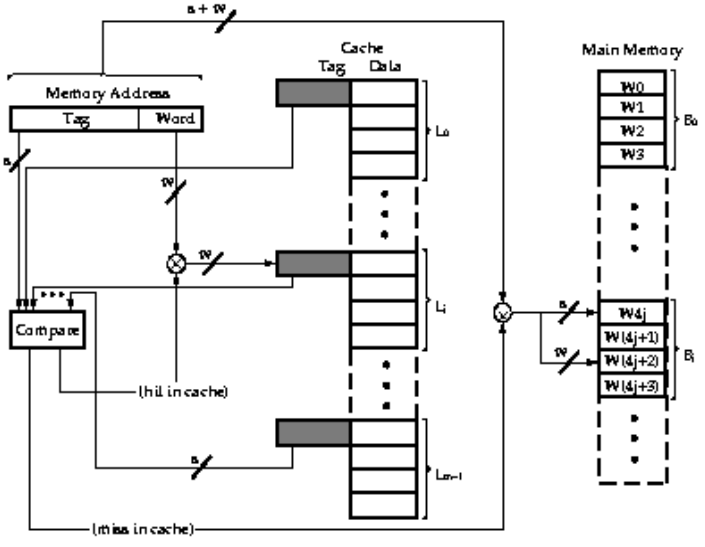


Exemplo: tag = 12 (01100_2)

Endereço da palavra	
Tag	*Offset*

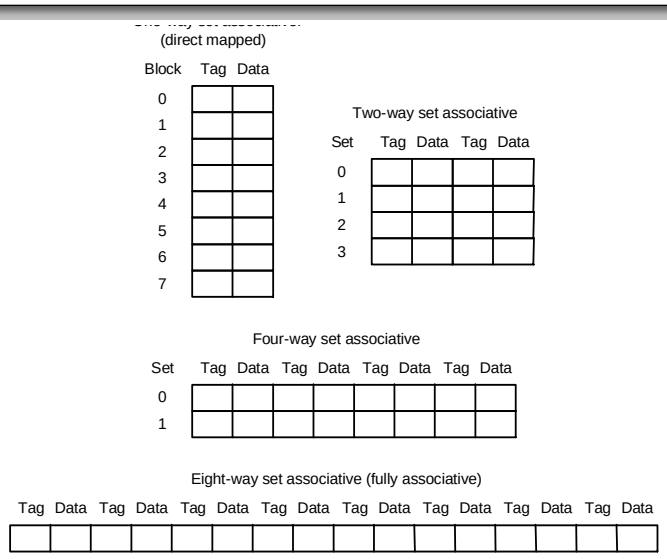
000000...01100	*Offset*

Mapeamento Associativo



24

Grau de associatividade



Comparação de Métodos de Mapeamento

- *Mapeamento direto*
  - ***Simples e Barata***
  - ***Lenta***
  - ***Mais faltas***
- *Associativa*
  - ***Rápida***
  - ***Menos falta***
  - ***Cara (comparação do endereço em paralelo)***
- *Associativa por conjunto: combinação das anteriores*
  - *Se NCC = NBC* ⇒ Ass. por conjunto = Mapeamento Direto
  - *Se NCC = 1* ⇒ Ass. por conjunto = Associativa

NBC = núm. blocos da cache
NCC = núm. conjuntos da cache

25

Políticas de Substituição de Páginas

- *Randômica:*
  - *Simples e fácil de implementar*
- *FIFO (First-In-First-Out)*
- *LFU (Least-Frequently Used)*
- *LRU (least-recently used)*
  - *Menor taxa de faltas*

	Associatividade					
	2 way		4-way		8-way	
Size	LRU	random	LRU	random	LRU	random
16 KB	5.18	5.69	4.67	5.29	4.39	4.96
64 KB	1.88	2.01	1.54	1.66	1.39	1.53
256KB	1.15	1.17	1.13	1.13	1.12	1.12

Caches separadas

- *Cache de dados e cache de instruções*
  - *Vantagens:*
    - *Melhor capacidade de otimizações*
    - *Evita hazard estrutural*
  - *Desvantagens:*
    - *maior taxa de falta*

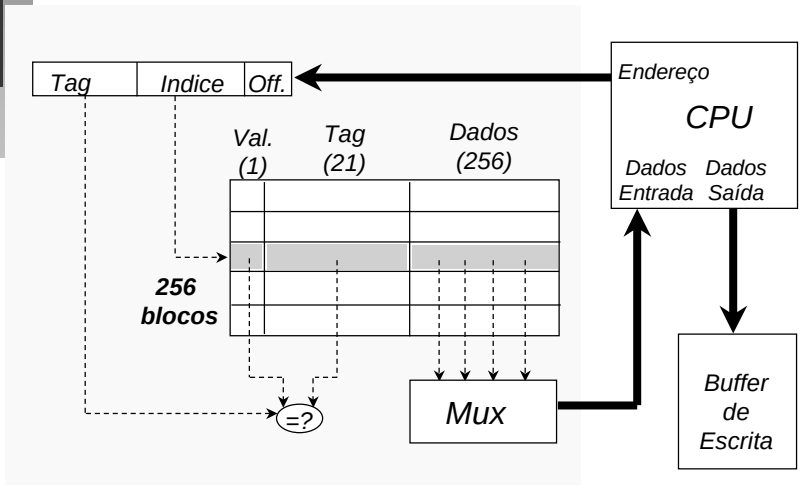
Programa	Miss rate (instr.)	Miss rate (dado)	Miss rate (sep.)	Miss rate (única)
Gcc	6.1%	2.1%	5.4%	4.8%
Spice	1.2%	1.3%	1.2%	

26

Exemplo: Alpha AXP 21064

- *Cache separadas de dados e de instruções*
- *Características:*
  - *Tamanho:8192 bytes*
  - *Blocos de 32 bits*
  - *Mapeamento direto*
  - *Write through*
  - *Four buffer write-buffer*

Alpha AXP 21064- Cache Dados



27

Exercício

Considere referências aos seguintes endereços de memória: 1,4,8,5,20,17,19,56, 9,11, 4,43,5,6,9, 17. Calcule o número de faltas para uma cache de 16 palavras com blocos de 1 palavra e mostre o estado final da cache. Compare os resultados para as seguintes organizações:

(a) - mapeamento direto
(b) - two-way set associativa,
(c) -completamente associativa.

Suponha que a cache está inicialmente vazia e quando necessário use como política de substituição o algoritmo LRU.

28