

Organização e Arquitetura de Computadores II

Memórias Cache

Introdução

Capítulo 2, 6 do Jean-Loup Baer

Capítulo 5 do Monteiro

Capítulo 4 do Stallings

Capítulo 5 do Hennessy e Patterson

Capítulo 7 do Patterson e Hennessy

Introdução a Caches

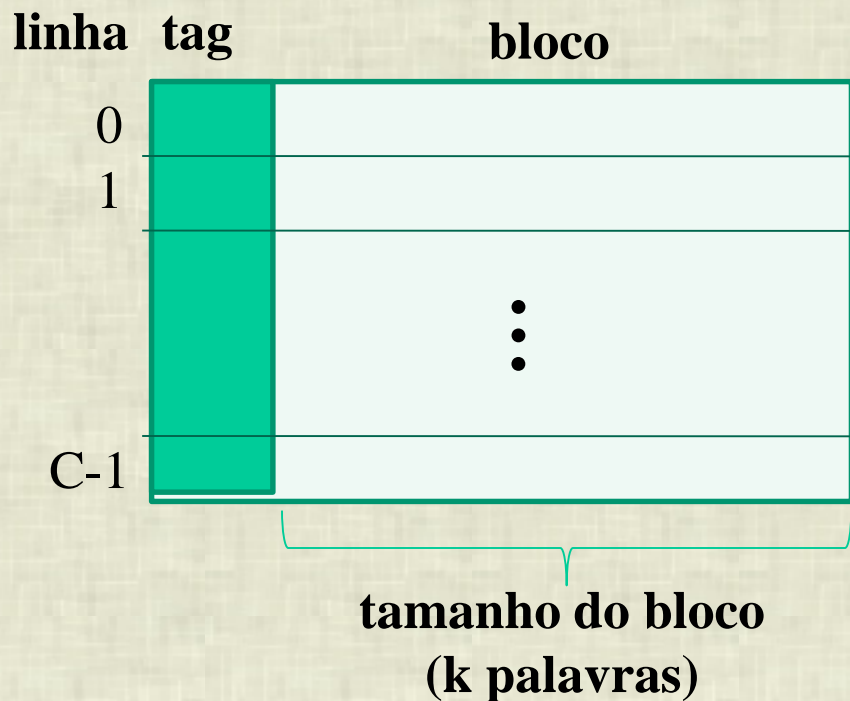
- **Funcionalidade**
 - Área de memória rápida e com informações dinâmicas
- **Cache só pode ter **parte** dos dados do nível mais abaixo**
 - Tamanho menor
- **Problemas**
 - Como identificar se o dado procurado está na cache?
 - Se estiver, como acessar de forma rápida?
 - Se não estiver, como buscar eficientemente de níveis inferiores?
 - Qual dado tirar da cache para colocar o novo dado?
- **Processador não sabe qual memória física tem o dado**
 - Gera apenas endereços e a hierarquia se encarrega de acessar a informação endereçada

Mapeamento de Endereços em Memória Cache

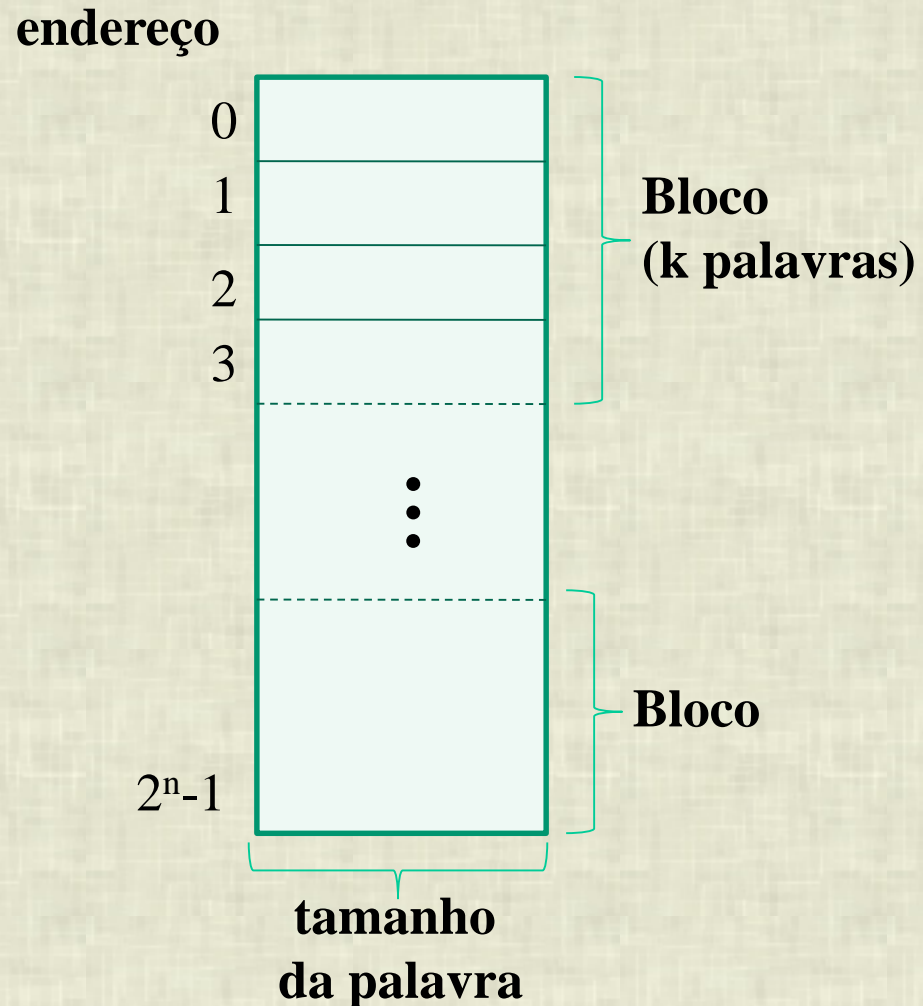
- **Como fazer para pesquisar um dado na cache?**
 - Fazer cache com todos endereços não faz sentido
 - Efetuar varredura sequencial na cache leva muito tempo
- **Solução**
 - Fazer mapeamento de endereços
- **Objetivo**
 - Relacionar informações (dados e instruções) da memória principal com posições da cache
- **Formas de mapeamento de memórias cache**
 - Direto
 - Associativo
 - Conjunto associativo

Estrutura Cache/MP

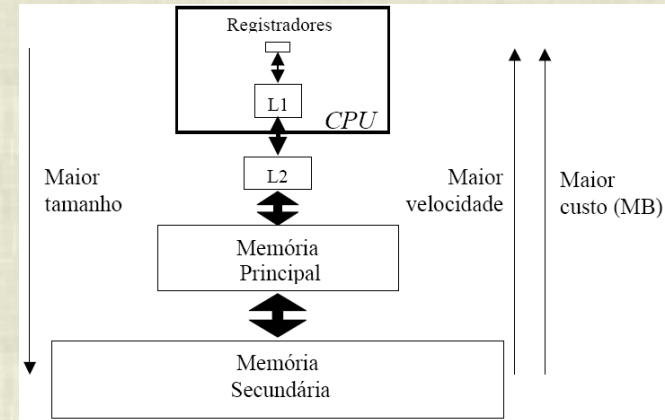
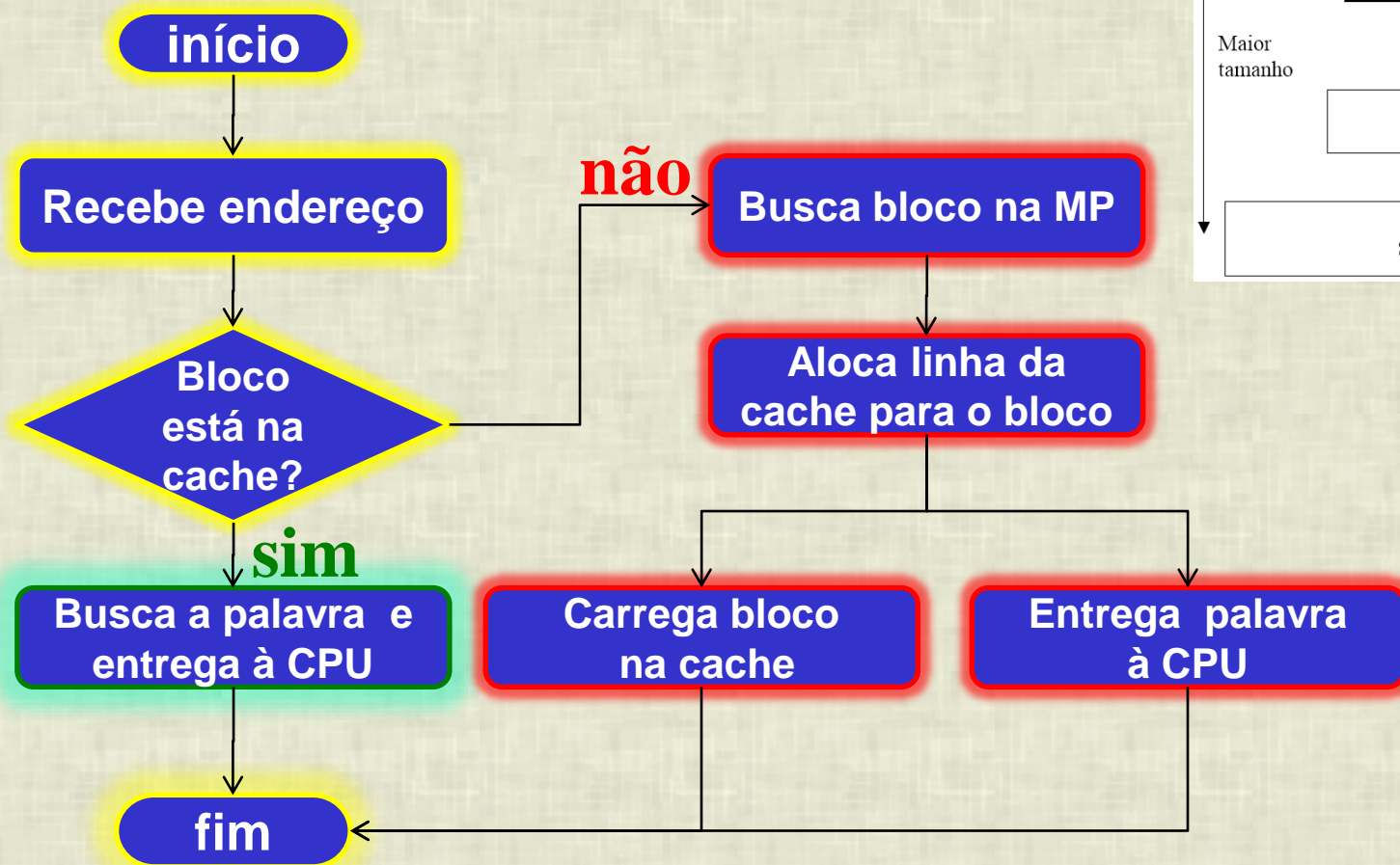
Memória Cache



Memória Principal



Leitura da Cache



*Fig 4.5, Stallings

Índice

1. Mapeamento de Endereços em Memória Cache

1.1 Mapeamento Direto

1.2 Mapeamento Associativo

1.3 Mapeamento Conjunto Associativo

Introdução ao Mapeamento Direto

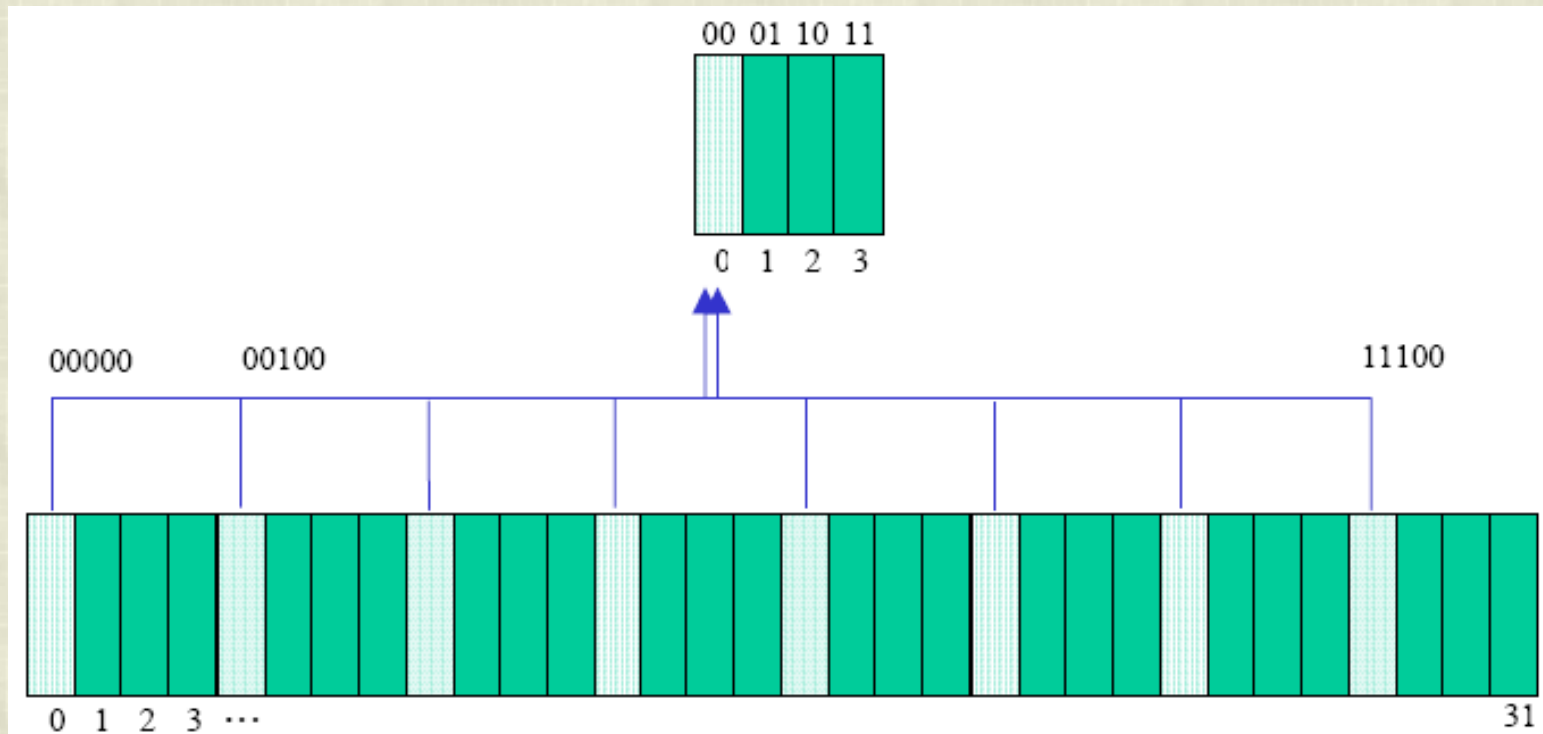
- **Forma mais simples de mapeamento**

- Posição na cache depende do endereço da palavra na memória principal (MP)
 - Cada palavra possui uma posição fixa
- Grupo de palavras mapeado na mesma posição da cache

- **Exemplo**

- Cache de 4 posições e MP de 32 endereços (palavra de 8 bits)
- Cada posição da cache tem 1 de 8 posições da MP
- Endereço obtido pelo resto da divisão inteira do número de posições da cache
 - Mapeamento utilizando os dois bits menos significativos do endereço

Esquema de Mapeamento Direto



- TAG (rótulo) para cada posição da cache → identifica qual das palavras está na cache
- Bit de validade → posição da cache está ocupada ou contém lixo
- Poderiam ser usados os bits mais significativos do endereçamento ao invés dos bits menos significativos? Qual a consequência?

Bit de Validade	<u>Tag</u>	<u>Dado</u>
1	001	00110110
0		
0		
1	000	11100011

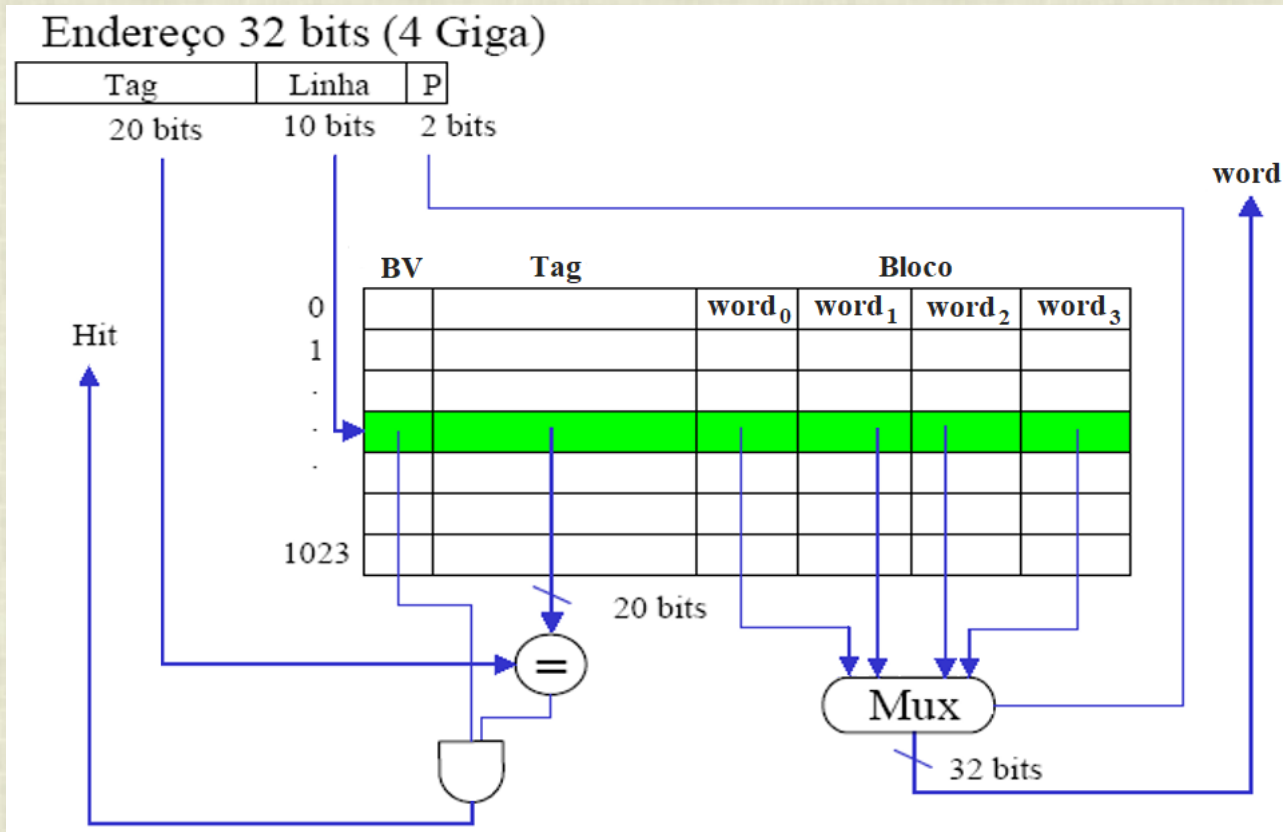
Acesso a Cache com Mapeamento Direto

- **Passos para um acesso**

1. Calcular o resto da divisão inteira do endereço pelo número de posições da cache → Ex. usar os bits menos significativos do endereço
2. Se bit de validade da posição for válido
 - Verificar Tag
 - Senão
 - Acusar miss
 - Ir para 4
3. Se Tag diferente de endereço
 - Acusar miss
 - Ir para 4
 - Senão
 - Ler posição (hit)
 - Ir para 7
4. Buscar dado no nível inferior
5. Colocar na posição
6. Efetuar leitura
7. Fim

Mapeamento Direto – Divisão de bits no registrador de endereçamento

- Exemplo da divisão de blocos em uma cache com 1024 (2^{10}) linhas e bloco com 4 palavras de 32 bits
 - Bit de validade e Tag
 - Transferência de blocos entre níveis de memória



Qual a vantagem de utilizar blocos ao invés de palavras?

Mapeamento Direto - Exercícios

1. Considerando um espaço de endereçamento de 1 Giga. Como fica a divisão de bits para uma cache de 2048 linhas que trabalha com blocos de 8 palavras?

Endereço de 30 bits (1 Giga)		
16 (Tag)	11 (Linha)	3 (Palavra)

2. Quanto tem efetivamente de dados nessa cache, considerando palavras de 32 bits?

Linha da <i>Cache</i>		
1 (validade)	16 (tag)	8*32 (bloco de dados)

(Bit de validade + Tag + Dados) / Dados

$$(1 + 16 + (8*32)) / (8*32)$$

$$273 \text{ bits} / 256 \text{ bits}$$

$$256 / 273 * 100\% = 93.77\%$$

Mapeamento Direto - Conclusões e Questões

- **Vantagens do mapeamento direto**
 - Simplicidade / Velocidade
 - Hardware barato
 - Procura simples (posição fixa)
 - Não existe escolha da vítima (é dada pelo módulo)
- **Desvantagens do mapeamento direto**
 - Pode ter mau aproveitamento das posições da cache (dependendo dos endereços gerados)

Exercícios

1. **(POSCOMP 2011 - 30)** Um sistema de computador possui um mapa de memória de 4 Gbytes, usando endereçamento a byte e uma memória cache com organização de mapeamento direto. A cache tem capacidade de armazenar até 1.024 palavras de 32 bits provenientes do mapa de memória. Assuma que a cache sempre é escrita de forma atômica com quatro bytes vindos de um endereço de memória alinhado em uma fronteira de palavra de 32 bits, e que ela usa 1 bit de validade por linha de cache. Neste caso, as dimensões do rótulo (tag) da cache, do índice e o tamanho da cache são, respectivamente
- a) 12 bits, 18 bits e 54.272 bits
 - b) 14 bits, 18 bits e 56.320 bits
 - c) 20 bits, 10 bits e 54.272 bits
 - d) 20 bits, 12 bits e 54.272 bits
 - e) 22 bits, 10 bits e 56.320 bits

Resposta de Exercícios

1. **(POSCOMP 2011 - 30)** Um sistema de computador possui um mapa de memória de 4 Gbytes, usando endereçamento a byte e uma memória cache com organização de mapeamento direto. A cache tem capacidade de armazenar até 1.024 palavras de 32 bits provenientes do mapa de memória. Assuma que a cache sempre é escrita de forma atômica com quatro bytes vindos de um endereço de memória alinhado em uma fronteira de palavra de 32 bits, e que ela usa 1 bit de validade por linha de cache. Neste caso, as dimensões do rótulo (tag) da cache, do índice e o tamanho da cache são, respectivamente
- a) 12 bits, 18 bits e 54.272 bits
 - b) 14 bits, 18 bits e 56.320 bits
 - c) 20 bits, 10 bits e 54.272 bits**
 - d) 20 bits, 12 bits e 54.272 bits
 - e) 22 bits, 10 bits e 56.320 bits

Exercícios

2. **(Baseado no POSCOMP 2017 - 44)** Considere que um processador tenha 16 bits de endereçamento de byte. Sua cache possui mapeamento direto com uma capacidade de 128 palavras e blocos de 8 palavras (palavra de 16 bits). O endereço se divide em campos para acesso à cache de acordo com a seguinte representação:

Etiqueta	i
Índice	ii
Palavra dentro do bloco	iii
Byte dentro da palavra	iv

O número de bits de cada um dos campos acima, ou seja, os valores de i, ii, iii e iv são, respectivamente:

- a) 12, 6, 5 e 3
- b) 8, 4, 3 e 1
- c) 8, 6, 5 e 3
- d) 12, 4, 3 e 1
- e) 12, 6, 3 e 1

Resposta de Exercícios

2. **(Baseado no POSCOMP 2017 - 44)** Considere que um processador tenha 16 bits de endereçamento de byte. Sua cache possui mapeamento direto com uma capacidade de 128 palavras e blocos de 8 palavras (palavra de 16 bits). O endereço se divide em campos para acesso à cache de acordo com a seguinte representação:

Etiqueta	i
Índice	ii
Palavra dentro do bloco	iii
Byte dentro da palavra	iv

O número de bits de cada um dos campos acima, ou seja, os valores de i, ii, iii e iv são, respectivamente:

- a) 12, 6, 5 e 3
- b) 8, 4, 3 e 1**
- c) 8, 6, 5 e 3
- d) 12, 4, 3 e 1
- e) 12, 6, 3 e 1

Exercícios

3. Considere uma memória principal de 256 bytes, endereçada a byte, e caches de 4 linhas, cada linha contendo um bloco de 8 bytes, além do espaço para colocar os bits de controle (bit de validade e tag). Dadas as sequências de acesso à memória principal tabeladas abaixo, preencha a cache para o caso de mapeamento direto. Diga, também, quantos miss e quantos hits aconteceram

Sequência de acessos (posição representada em hexadecimal)

Acesso	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Posição	16	17	18	19	16	17	18	19	96	97	98	99	1A	1B	C8	C9	CA	1B	1C	1D

Hits =

Miss =

[illegible]

Resposta de Exercícios

Acesso	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Posição	16	17	18	19	16	17	18	19	96	97	98	99	1A	1B	C8	C9	CA	1B	1C	1D

(continuação)

BV	TAG (3 bits)	0	1	2	3	4	5	6	7
0									
0, 1	110	C8	C9	CA	CB	CC	CD	CE	CF
0, 1	000, 100	10,90	11,91	12,92	13,93	14,94	15,95	16,96	17,97
0, 1	000, 100, 000	18,98,18	19,99,19	1A,9A,1A	1B,9B,1B	1C,9C,1C	1D,9D,1D	1E,9E,1E	1F,9F,1F

Hits = 14, Miss = 6

Índice

1. Mapeamento de Endereços em Memória Cache

1.1 Mapeamento Direto

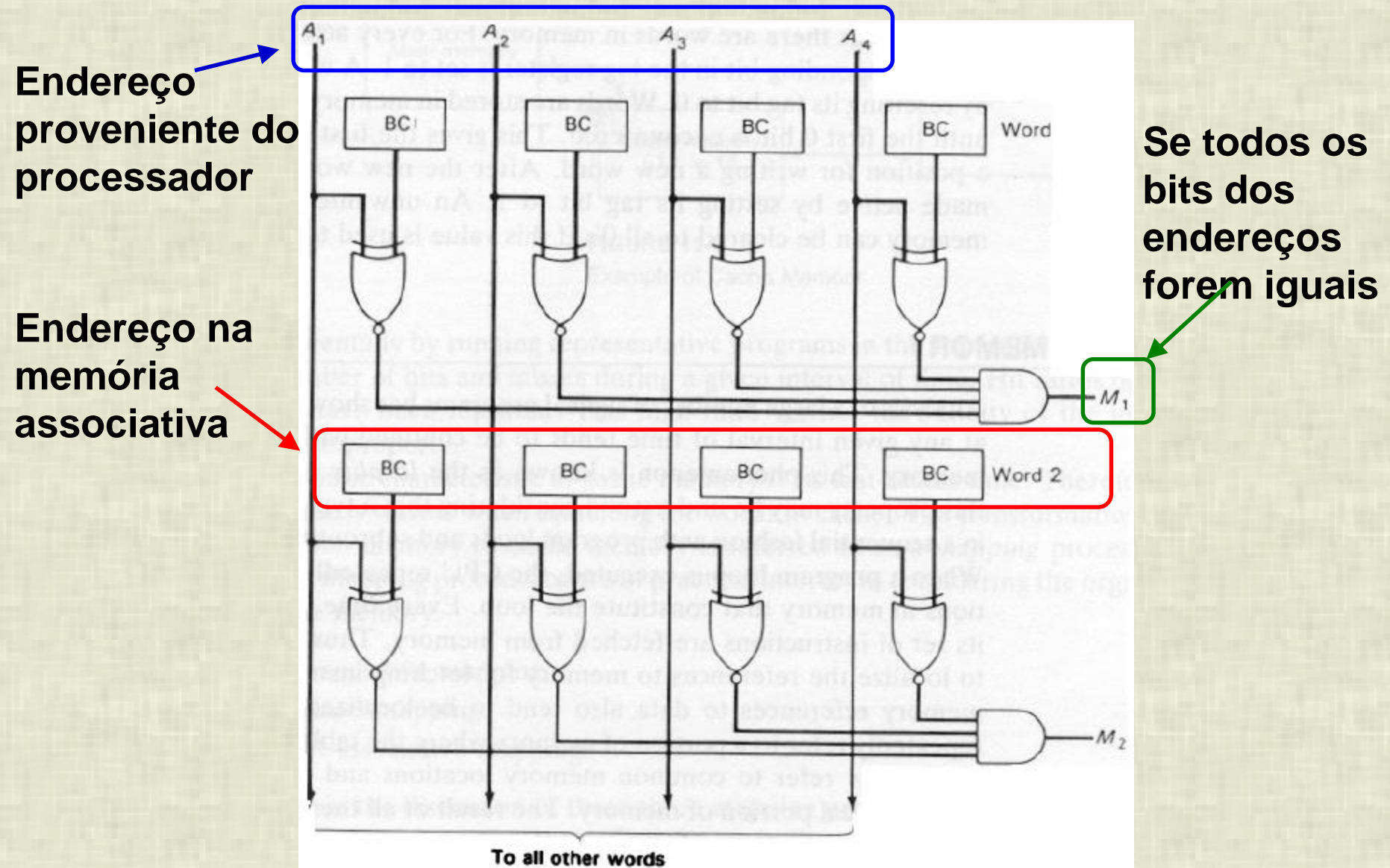
1.2 Mapeamento Associativo

1.3 Mapeamento Conjunto Associativo

Introdução ao Mapeamento Associativo

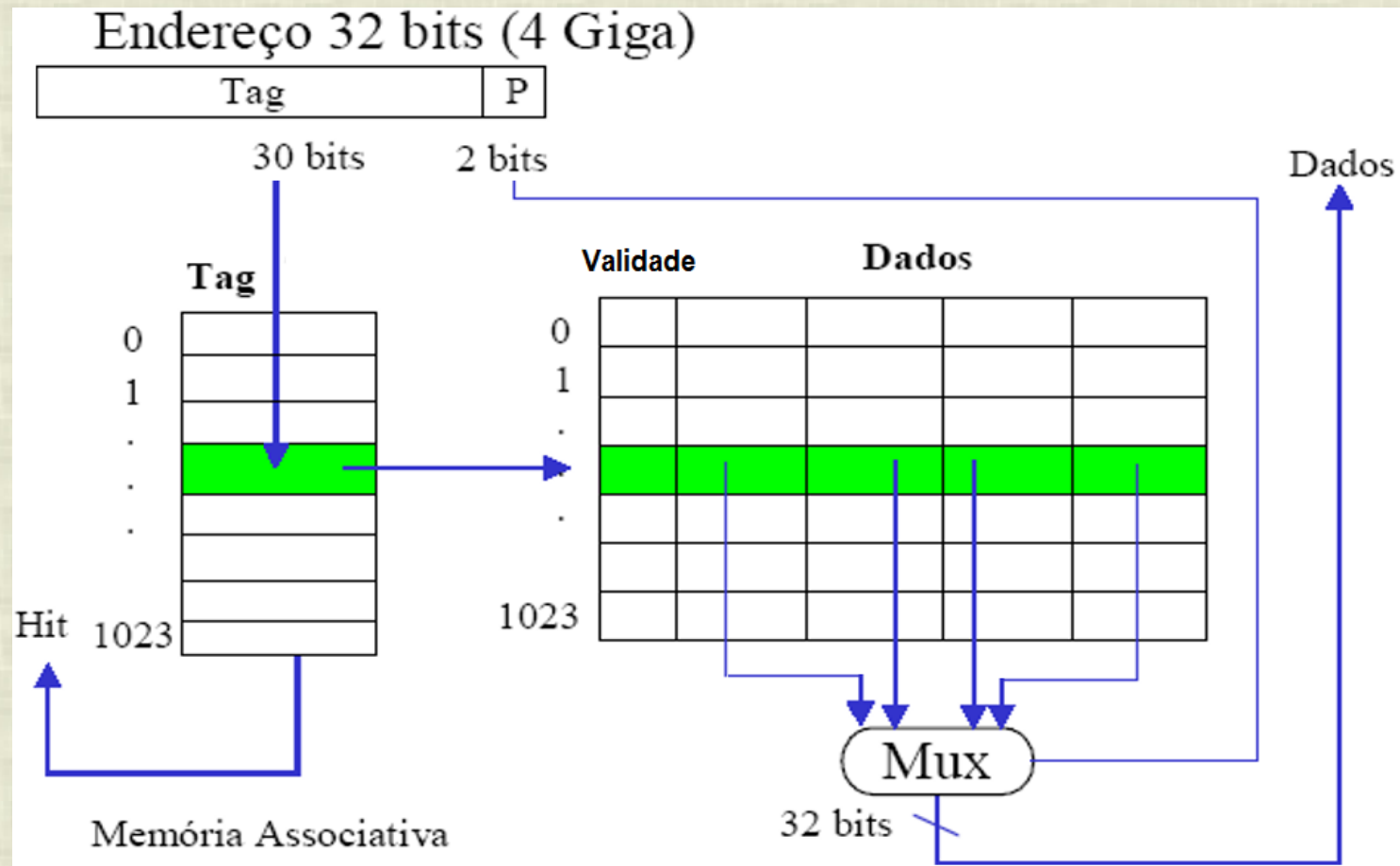
- **Como melhorar o mapeamento Direto, retirando a dependência entre endereço na memória e posição da cache sem comprometer desempenho da procura?**
 - Resposta → Mapeamento associativo
- **Endereço da MP em qualquer posição da cache**
 - Tag não fica mais na cache e sim em memória especial (memória associativa)
- **Consequências**
 - Necessita fazer procura de dado
 - Necessita política de substituição
 - Quando ocorre miss, deve buscar no nível abaixo e, caso a cache esteja com todas posições ocupadas, quem tirar para abrir lugar?
- **Solução para procurar**
 - Procurar em paralelo
 - Usar memória associativa
 - Memória cara e de tamanho limitado

Esquema de Memória para Mapeamento Associativo – Pesquisa em Paralelo



Mapeamento Associativo – Divisão de bits no registrador de endereçamento

- Exemplo da divisão de blocos em uma cache com 1024 (2^{10}) linhas e bloco com 4 palavras de 32 bits
 - Bit de validade e Tag
 - Transferência de blocos entre níveis de memória



Mapeamento Associativo - Exercícios

1. Considerando espaço de endereçamento de 256 Mega palavras. Como ficaria a divisão de bits do endereço para uma **cache** de 2048 linhas e que trabalhe com blocos de 8 palavras?

Endereço de 28 bits (256 Mega)	
25 (Tag)	3 (Palavra)

2. Quanto tem efetivamente de dados nessa cache?

100% (não considerando bit de validade)

3. Qual o tamanho da **memória associativa**?

$$\begin{aligned}\text{Tamanho} &= 2048 * 25 \text{ (tag)} \\ &= 51200 \text{ bits} / 8 \\ &= 6400 \text{ bytes} / 1024 \\ &= 6,25 \text{ Kbytes}\end{aligned}$$

Mapeamento Associativo – Substituição de Dados na Cache

- **Usar políticas**

- **Randômica**

- Escolhe aleatoriamente posição a ser substituída → Simples, mas sujeito a aumentar número de caches miss

- **Contador**

- Um contador baseado em algum tipo de relógio aponta para a próxima posição a ser substituída → simples, mas igualmente à política randômica, está sujeita a aumentar o número de misses

- **LFU (Least Frequent Used)**

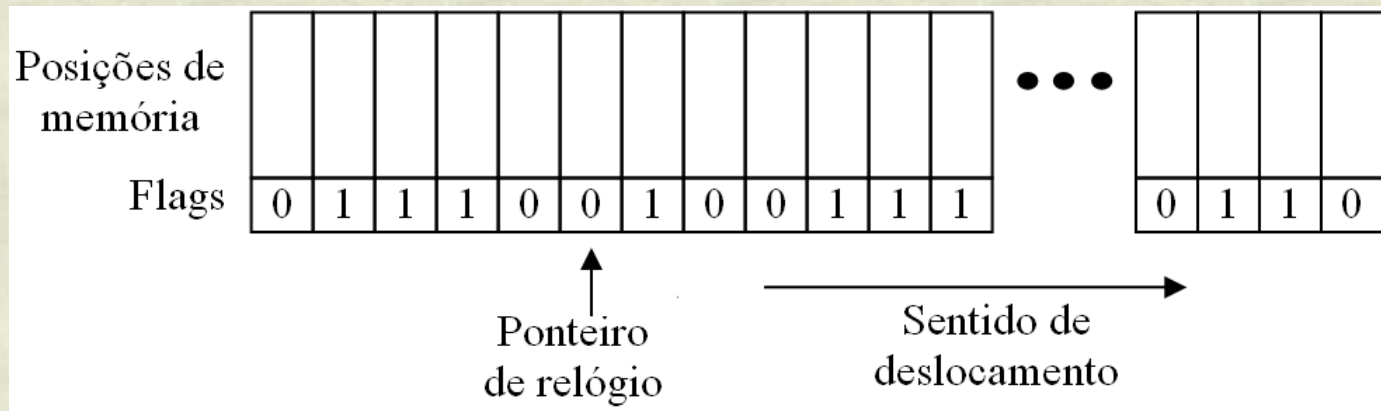
- Posição da cache menos usada é substituída → Escolha precisa mecanismo de contagem para cada acesso

- **LRU (Least Recent Used)**

- Posição da cache usada a mais tempo será substituída → Escolha precisa variável de tempo a cada acesso e comparação

Mapeamento Associativo – Algoritmo do Relógio

- Técnica simples para pesquisar endereço de memória cache visitado há mais tempo. Permite a implementação do critério de substituição por tempo
- A técnica, na verdade, não retorna a posição há mais tempo visitada, mas é simples, fornecendo posições visitadas há mais tempo
- Cada posição da cache tem associado um bit (flag) que informa se a mesma foi ou não recentemente visitada. Quando posição de memória é acessada, a flag é escrita
- Um ponteiro (*relógio*) que caminha nos endereços da cache em uma determinada base de tempo, vai apagando as flags na medida em que passa por elas
- Se uma posição de memória não tem a flag escrita é considerada que ela foi acessada há mais tempo e pode ser descartada. Esta informação fica em uma pequena lista no gerente da hierarquia de memória



- O algoritmo pode ser melhorado substituindo flags por contadores. Cada vez que o relógio passa, contador é decrementado, e cada vez que a posição de memória é acessada, contador volta para valor máximo

Acesso a Cache com Mapeamento Associativo

- **Passos para um acesso**

1. Alimentar memória associativa com Tag procurado (comparação)
2. Se Tag está na memória associativa
(Hit) Acessar **memória cache** com **índice** fornecido pela **memória associativa**
Ir para 7
Senão
Acusar miss
3. Se não existir posição livre na cache
Escolher endereço para substituir de acordo com política estabelecida (e.g., LRU)
4. Buscar dado no nível inferior
5. Colocar na posição livre ou escolhida da **memória cache**
6. Cadastrar posição na **memória associativa** para pesquisas futuras
7. Efetuar leitura
8. Fim

Mapeamento Associativo - Conclusões e Questões

- **Vantagem do mapeamento associativo**
 - Melhor distribuição da informação na cache
- **Desvantagens**
 - Memória associativa tem alto custo e tamanho limitado
 - Limita número de linhas da cache
 - Necessita política de substituição
 - Pode ocorrer escolhas inadequadas
 - Gasta tempo
 - Requer hardware extra para controle de qual posição deve ser substituída

Exercícios

(POSCOMP 2013, Questão 48) Sobre memória cache, considere as afirmativas a seguir.

I. No mapeamento associativo, cada bloco da memória principal pode ser carregado em qualquer linha da cache.

II. No mapeamento direto, cada bloco da memória principal é mapeado a apenas uma linha de cache.

III. No mapeamento direto, o acesso repetido a diferentes blocos de memória mapeados na mesma linha de cache resultará em uma alta taxa de acerto.

IV. A técnica de mapeamento associativo é simples e pouco dispendiosa para se implementar.

Assinale a alternativa correta.

a) Somente as afirmativas I e II são corretas.

b) Somente as afirmativas I e IV são corretas.

c) Somente as afirmativas III e IV são corretas.

d) Somente as afirmativas I, II e III são corretas.

e) Somente as afirmativas II, III e IV são corretas.

Resposta de Exercícios

(POSCOMP 2013, Questão 48) Sobre memória cache, considere as afirmativas a seguir.

- I. No mapeamento associativo, cada bloco da memória principal pode ser carregado em qualquer linha da cache.
- II. No mapeamento direto, cada bloco da memória principal é mapeado a apenas uma linha de cache.
- III. No mapeamento direto, o acesso repetido a diferentes blocos de memória mapeados na mesma linha de cache resultará em uma alta taxa de acerto.
- IV. A técnica de mapeamento associativo é simples e pouco dispendiosa para se implementar.

Assinale a alternativa correta.

- ☒ a) Somente as afirmativas I e II são corretas.
- ☐ b) Somente as afirmativas I e IV são corretas.
- ☐ c) Somente as afirmativas III e IV são corretas.
- ☐ d) Somente as afirmativas I, II e III são corretas.
- ☐ e) Somente as afirmativas II, III e IV são corretas.

Exercícios

Considere uma memória principal de 256 bytes, endereçada a byte, e caches de 4 linhas, cada linha contendo um bloco de 8 bytes, além do espaço para colocar os bits de controle (bit de validade e tag), se forem necessários. Dadas as sequências de acesso à memória principal tabeladas abaixo, preencha a caches para o caso de mapeamento associativo. Considere que a política para substituição é LRU. Diga quantos miss e quantos hits aconteceram. Compare a implementação de mapeamento direto com a de mapeamento associativo.

Sequência de acessos (posição representada em hexadecimal)

Acesso	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Posição	16	17	18	19	16	17	18	19	96	97	98	99	1A	1B	C8	C9	CA	1B	1C	1D

MA (5 bits)		BV	0	1	2	3	4	5	6	7

Hits =

Miss =

Resposta de Exercícios

(continuação)

Mapeamento Direto (exercício anterior)

BV	TAG (3 bits)	0	1	2	3	4	5	6	7
0									
0, 1	110	C8	C9	CA	CB	CC	CD	CE	CF
0, 1	000, 100	10,90	11,91	12,92	13,93	14,94	15,95	16,96	17,97
0, 1	000, 100, 000	18,98,18	19,99,19	1A,9A,1A	1B,9B,1B	1C,9C,1C	1D,9D,1D	1E,9E,1E	1F,9F,1F

Hits = 14, Miss = 6

Mapeamento Associativo

MA (5 bits)
00010, 11001
00011
10010
10011

BV	0	1	2	3	4	5	6	7
0,1	10, C8	11, C9	12, CA	13, CB	14, CC	15, CD	16, CE	17, CF
0,1	18	19	1A	1B	1C	1D	1E	1F
0,1	90	91	92	93	94	95	96	97
0,1	98	99	9A	9B	9C	9D	9E	9F

Hits = 15, Miss = 5

Índice

1. Mapeamento de Endereços em Memória Cache

1.1 Mapeamento Direto

1.2 Mapeamento Associativo

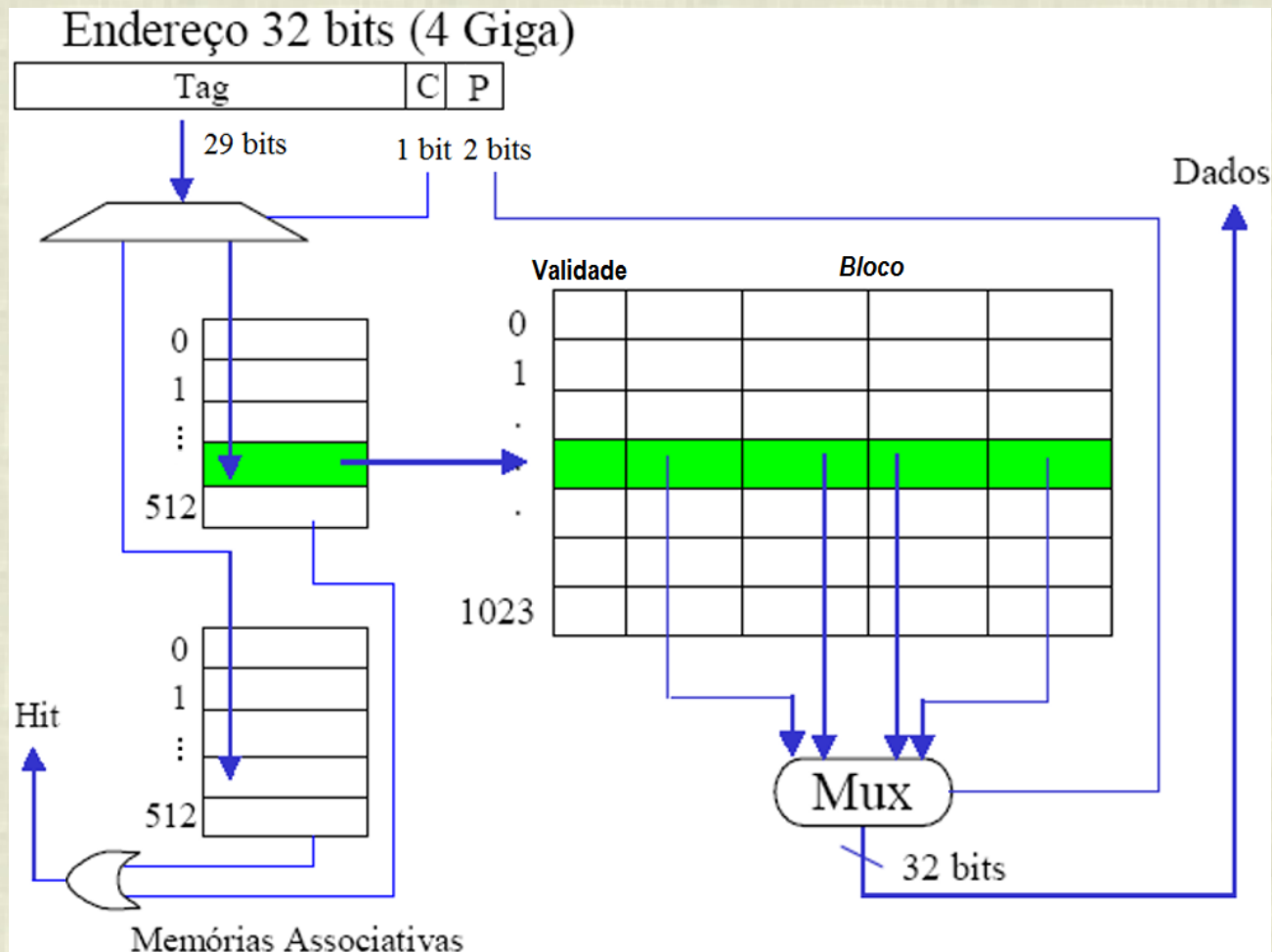
1.3 Mapeamento Conjunto Associativo

Introdução ao Mapeamento Conjunto Associativo

- **Limite de tamanho** da cache devido pesquisa na memória associativa é uma restrição muito forte → **Tendência é aumentar a cache**. Qual seria outra possibilidade?
 - Resposta → Mapeamento Conjunto Associativo
- **Compromisso entre mapeamento direto e totalmente associativo**
- **Cache dividida em S conjuntos de N linhas**
 - Se $S = 1$ → **mapeamento associativo**
 - Se $S =$ número de linhas da cache → **mapeamento direto**
- **Pesquisa dentro do conjunto**
 - Endereço i , da memória principal, pode mapear para endereços no conjunto $i \bmod S$, da cache
- **Necessita de substituição quando ocorre cache miss**
 - Caso conjunto esteja cheio quem tirar?
 - *A substituição deve ser necessariamente dentro do conjunto*

Mapeamento Conjunto Associativo – Divisão de bits no registrador de endereçamento

- Exemplo de cache com 1024 linhas (2^{10}) com blocos com 2 palavras, palavra de 32 bits e 2 conjuntos ($S=2$)



Acesso à Cache com Mapeamento Conjunto Associativo

- **Passos para um acesso**

1. Calcular resto da divisão inteira do endereço pelo número de conjuntos **S**
→ Ex., usar bits menos significativos de endereço
2. Alimentar memória associativa do conjunto com Tag (Comparação)
3. Se Tag está na **memória associativa**
(Hit) Acessar **memória cache** com índice fornecido pela **memória associativa**
Ir para 8
Senão
Acusar miss
4. Se não existir posição livre no conjunto cache
Escolher endereço para substituir de acordo com política estabelecida
5. Buscar dado no nível inferior
6. Colocar dado na posição livre ou escolhida da **cache**
7. Cadastrar posição na **memória associativa** para pesquisas futuras
8. Efetuar leitura
9. Fim

Mapeamento Conjunto Associativo - Exercícios

- Considerando uma cache com 1024 posições (2^{10}) com palavra de 32 bits, um espaço de endereçamento de 4 Gigabytes (2^{32}), dois conjuntos associativos, responda as perguntas que seguem

1. Quanto tem efetivamente de dados nessa cache?

100% (sem considerar o bit de validade)

2. Qual tamanho de cada memória associativa?

Tamanho = 1024 (linhas) / 2 (cj.) * 31 (tag)
= 512 * 31 bits = 15872 bits
= 15872 bits / 8 = 1984 bytes
= 1984 bytes / 1024 = 1,93 Kbytes

3. Como ficaria a mesma cache com 4 conjuntos (S=4)?

Mapeamento Conjunto Associativo - Exercícios

4. Qual tamanho de cada uma das 4 memórias associativas?

$$\begin{aligned}\text{Tamanho} &= 256 * 30 \text{ (tag)} \\ &= 7680 \text{ bits} / 8 \\ &= 960 \text{ bytes} / 1024 \\ &= 0,94 \text{ Kbytes}\end{aligned}$$

5. Como ficaria divisão do endereço de cache de 4 conjuntos com 2048 linhas (**bloco** com 4 palavras de 32 bits)?

Endereço de 32 bits (4 Giga)		
28 (Tag)	2 (Conjunto)	2 (Palavra)

Mapeamento Conjunto Associativo – Vantagens e Desvantagens

- **Vantagem**




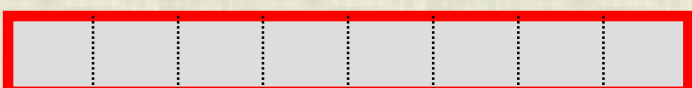
- Aumenta tamanho da cache mantendo tamanho da memória associativa (limitação tecnológica)

- **Desvantagens**

- Memória associativa tem alto custo e tamanho limitado
- Somente faz substituição dentro do conjunto
 - Pode aumentar o número de miss, se comparado com o mapeamento associativo
- Necessita política de substituição
 - Gasta tempo
 - Pode escolher mal
 - Requer hardware extra para controle de qual posição deve ser substituída

Mapeamento Conjunto Associativo – Níveis de Associatividade

- Associatividade da cache é dada pelo número de vias (ways)
- Exemplo: cache de 8 posições pode ter 1 a 8 vias

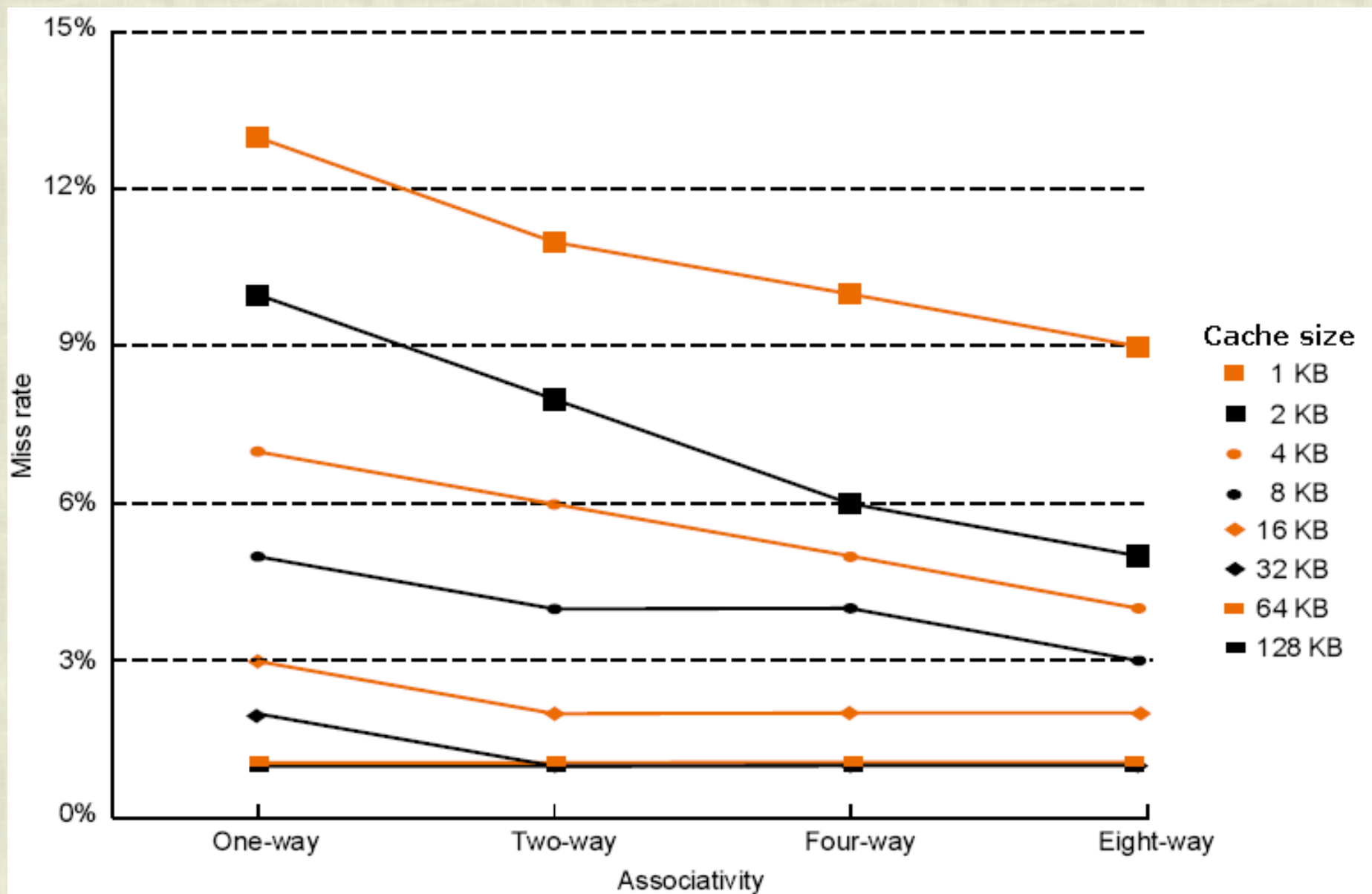
Vias (Ways)	Mapeamento	Esquema
1	Direto	
2	Conjunto-associativo (4 cj)	
4	Conjunto-associativo (2 cj)	
8	Associativo	

- Quantos conjuntos possui cache L1 4-Way de 64 Kbytes e com processador Ultra Sparc III (assumir bloco de 32 palavras de 64 bits)?

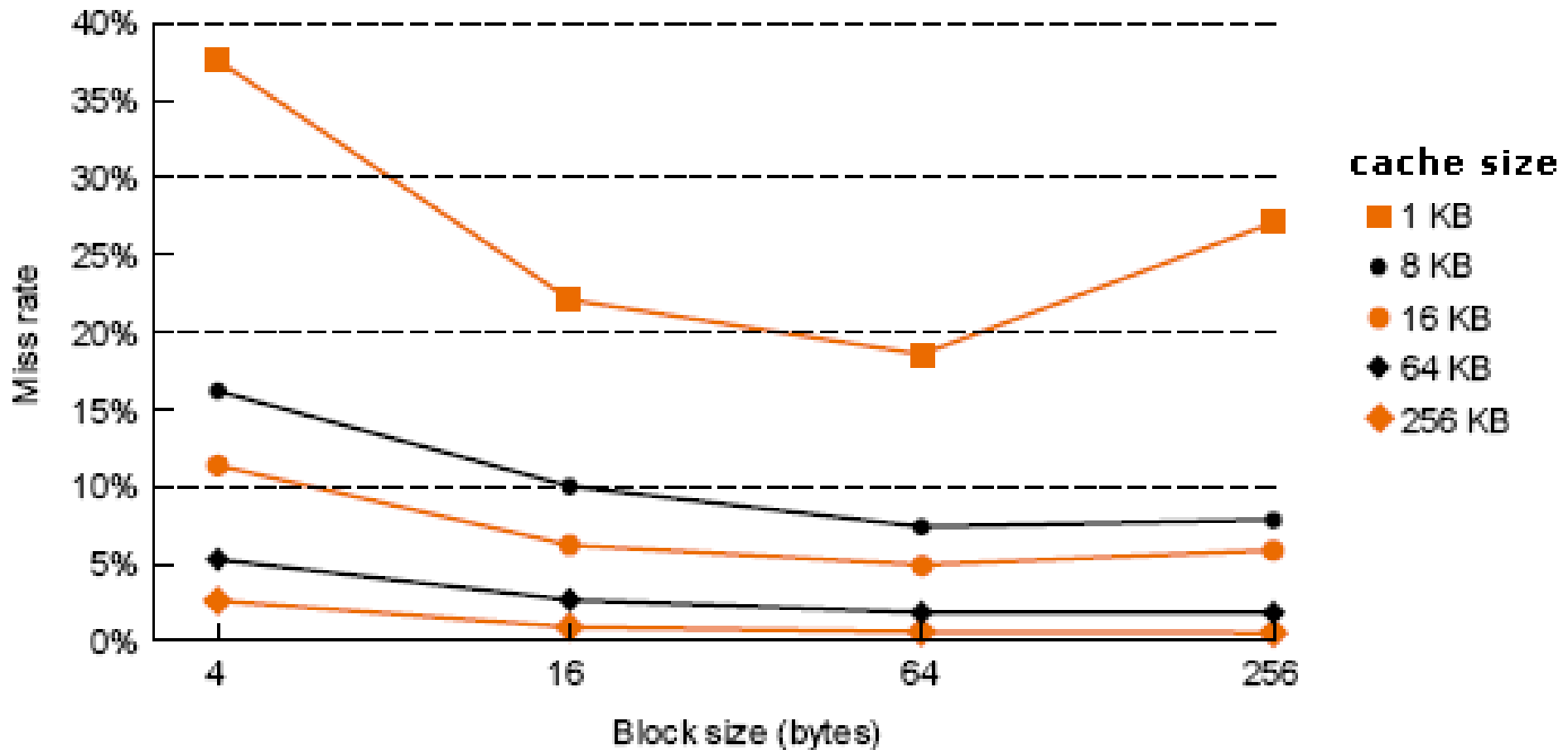
Tamanho da linha da cache = $32 * 64 \text{ b} = 2048 \text{ b} = 256 \text{ B} = 0,25 \text{ KB}$

$64 \text{ KB} / 0,25 \text{ KB} = 256 \text{ linhas} / 4 \text{ (4-Way} \rightarrow 4 \text{ linhas por cj)} = 64 \text{ cj}$

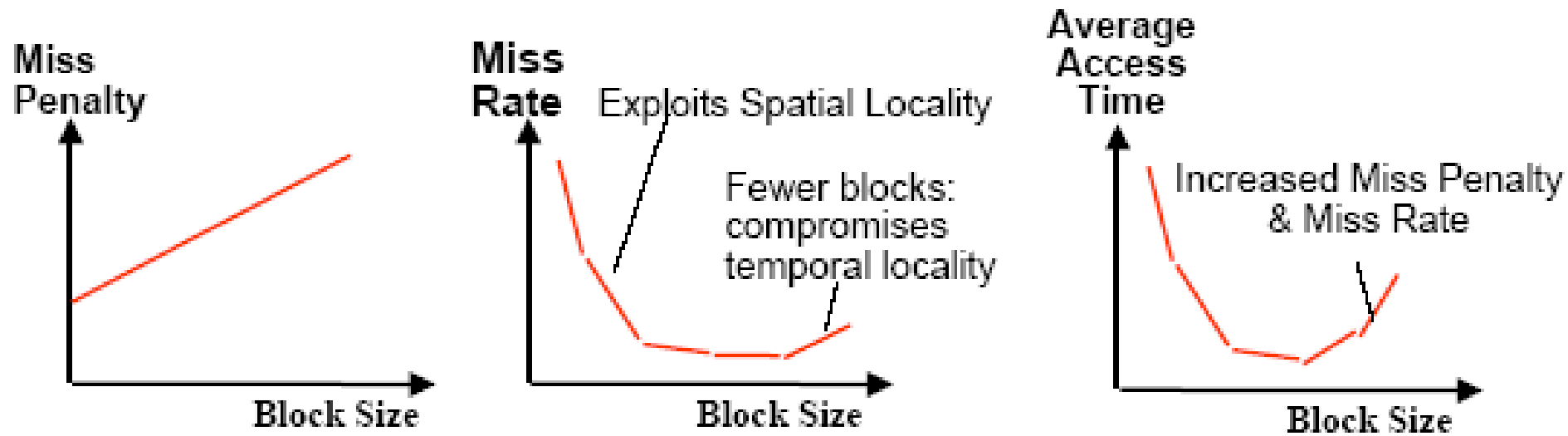
Mapeamento Conjunto Associativo – Níveis de Associatividade e Miss Rate



Mapeamento Conjunto Associativo – Miss Rate versus Tamanho do Bloco



Mapeamento Conjunto Associativo – O Efeito do Tamanho do Bloco



Exercícios:

1. Analise o comportamento das três curvas
2. Porque, a partir de um determinado tamanho, quando o bloco aumenta também aumenta o tempo de acesso?

Exercícios

1. Cite alguns problemas básicos do uso de memória cache e comente
2. Porque não é necessário para o processador saber onde estão fisicamente os dados quando gera o endereçamento?
3. Quais são as três formas básicas de mapeamento de memórias cache? Comente sobre vantagens e desvantagens das mesmas
4. Explique como funciona o mapeamento direto. Para que serve a tag, e para que serve o bit de validade?
5. Comente sobre o passo 4 (buscar dado no nível inferior) do algoritmo de acesso à memória cache com mapeamento direto. Fale com relação a tempo para sua execução. Detalhe melhor este passo
 1. Calcular o módulo do endereço pelo número de posições da *cache* (ou usar os bits menos significativos do endereço);
 2. Verificar o bit de validade da posição da *cache* correspondente e se for inválido acusar *miss* (ir para 4), senão verificar o Tag;
 3. Se Tag diferente do endereço procurado acusar *miss* (ir para 4), senão ocorre *hit*. Ler a posição (fim);
 4. Buscar dado no nível inferior. Colocar na posição e efetuar a leitura (fim)
6. Porque o mapeamento direto é tão rápido/simples para acessar um dado? Comente, descrevendo os mecanismos necessários para o seu funcionamento

Exercícios

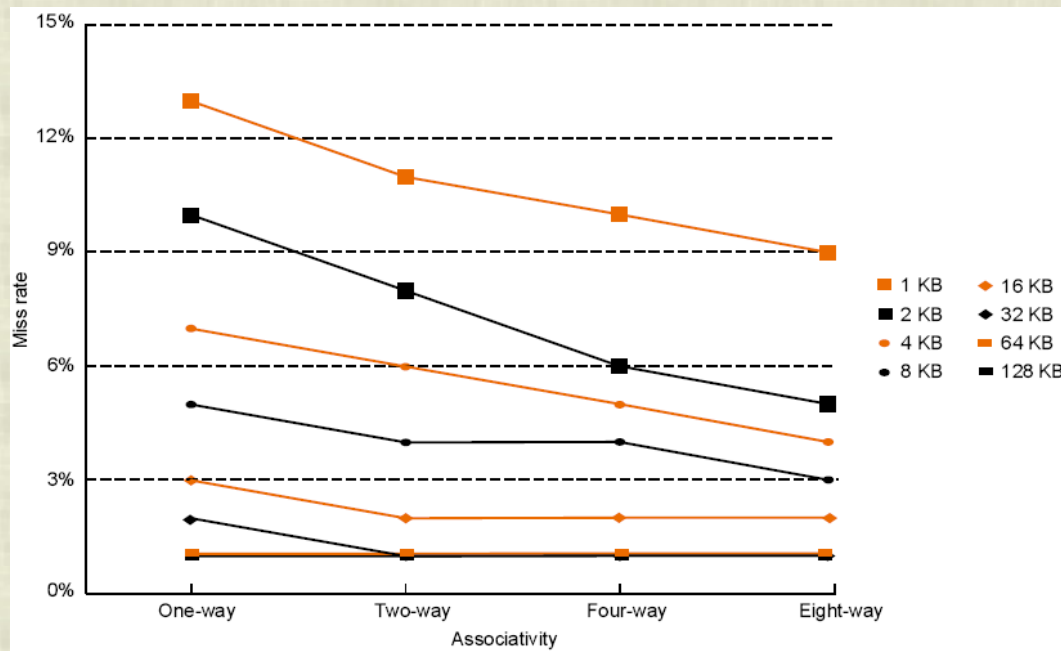
7. Qual a necessidade do multiplexador no mapeamento direto com blocos?
8. Comente a seguinte frase: “Depois de realizado vários testes, verificamos que cache, com mapeamento direto, somente é interessante se os dados que compartilham as mesmas áreas de cache estiverem bastante distantes na memória física”
9. Comente a afirmativa: “Para ter melhor desempenho, a escolha do modo de endereçamento depende da aplicação”. Descreva aplicações onde o mapeamento direto pode ser interessante
10. Existe como a abordagem de mapeamento direto ser realizada dinamicamente? Ou seja, os dados não têm endereços fixos conhecido em tempo de projeto, mas sim durante a execução do programa este endereço é calculado
11. Considerando possível a pergunta acima, quais as consequências desta modalidade de mapeamento direto? Comente analisando características como: velocidade de acesso, custos de implementação, flexibilidade da memória cache.
12. Qual a característica básica de uma memória associativa? Responda comentando sobre a organização deste tipo de memória
13. Descreva o funcionamento do mapeamento associativo. Cite uma grande vantagem? Cite uma grande desvantagem?
14. Porque no mapeamento associativo pode não ser necessário o uso de bit de validade?

Exercícios

15. Cite as três políticas básicas para substituição de dados em uma memória associativa. Explique como funciona cada uma delas. Diga vantagens e desvantagens. Sugira alguma outra política e compare com as anteriores
16. Diga qua(l/is) a(s) diferença(s) básica(s) entre o acesso a dados em cache com mapeamento direto e mapeamento associativo
17. Comente a afirmação: "A grande vantagem do mapeamento associativo frente ao mapeamento direto está no fato que toda a memória pode ser utilizada eficientemente como cache. Já no mapeamento direto esta eficiência é praticamente inalcançável"
18. Descreva o funcionamento do mapeamento conjunto associativo
19. Dado S como número de conjuntos de um mapeamento conjunto associativo. Porque se $S = 1$ o mapeamento conjunto associativo se assemelha ao mapeamento associativo, e se $S =$ número de linhas da cache o mapeamento conjunto associativo se assemelha ao mapeamento direto?
20. Descreva como funcionam os níveis de associatividade
21. Qual a diferença básica entre o mapeamento associativo e o mapeamento conjunto associativo? Apresente vantagens e desvantagens.

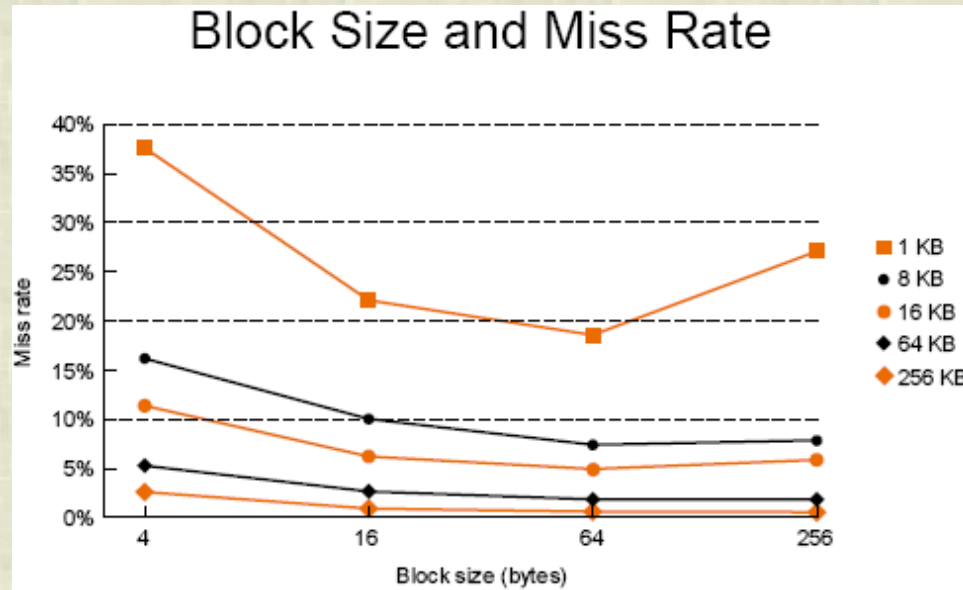
Exercícios

22. Compare com relação a vantagens e desvantagens, o mapeamento direto e o mapeamento conjunto associativo
23. O que diferencia a memória associativa das memórias convencionais? Qual a aplicação das memórias associativas? Faça um diagrama da arquitetura de uma memória associativa, explicando o seu funcionamento
24. Na figura abaixo, que relaciona taxa de associatividade com miss-rate, é mostrado que para, caches pequenas, o miss-rate reduz à medida que aumenta a associatividade (número de vias). Porque isto acontece? Porque o mesmo efeito não é verificado com caches maiores?

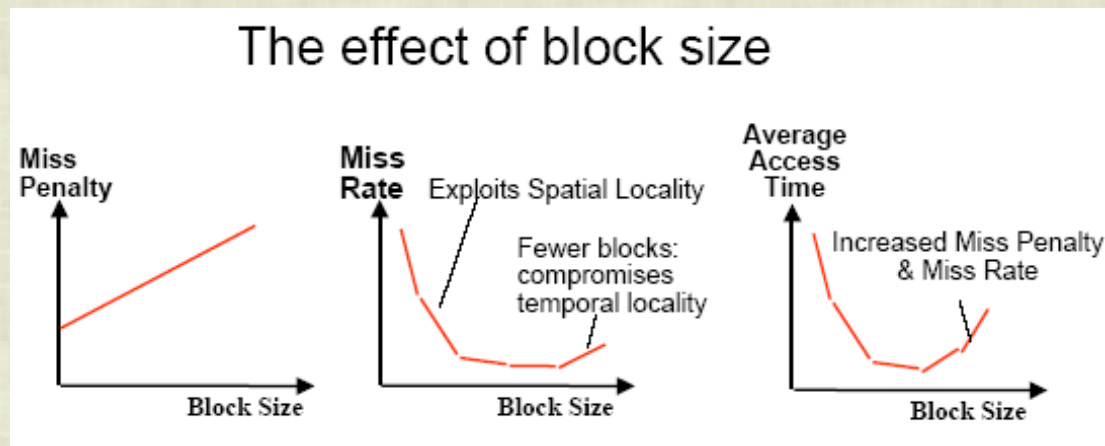


Exercícios

25. Comente as curvas que relacionam tamanho do bloco com miss-rate e tamanho de cache

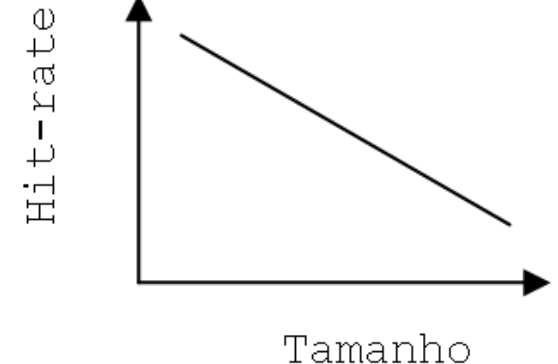
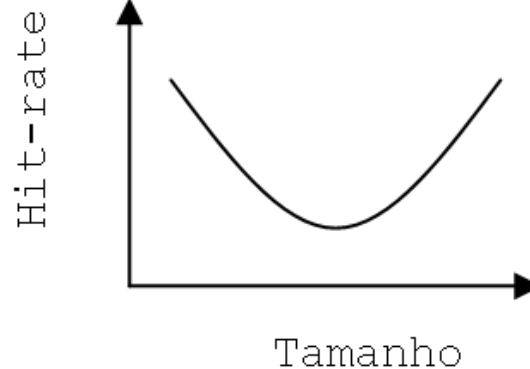
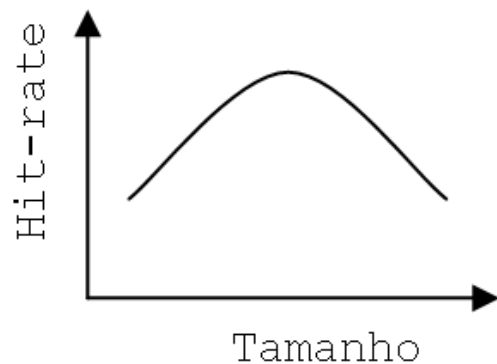
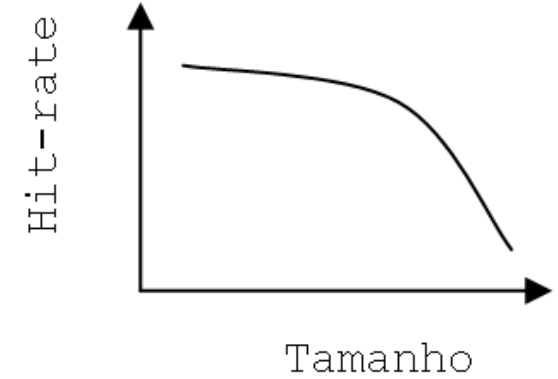
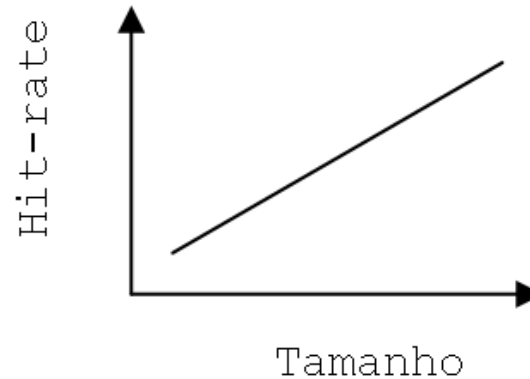
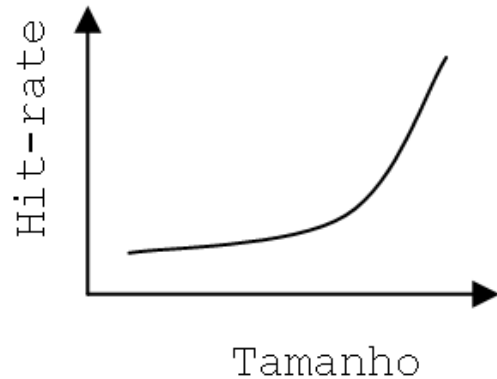


26. Comente as três figuras que relacionam o efeito do tamanho do bloco no tempo médio de acesso e no miss-rate



Exercícios

27. Diferencie localidade espacial de localidade temporal. Explique porque os sistemas de hierarquia de memórias são baseados no princípio de localidade
28. Aponte qual é o gráfico que melhor representa o efeito do tamanho da memória cache no hit-rate e diga por quê



Exercícios

29. Considere a seguinte estrutura de memória:

- Memória principal: 1 MByte
- Memória cache: 16 Kbytes
- Tamanho do bloco na cache: 32 palavras
- Tamanho da palavra: 1 Byte
 - Quantos blocos têm a memória cache? Mostre o cálculo e explique
 - Como é formado o endereço para o mapeamento direto? Explique com diagramas
 - Qual a área necessária para armazenar os TAGs, em Bytes?
 - Qual a desvantagem do mapeamento direto?
 - Como seria a formação para o mapeamento conjunto associativo 4-way (4 blocos em cada conjunto)? Explique com diagramas

30. O que pode ser alterado na arquitetura de memórias cache para diminuir o miss-rate? Explique as possibilidades que forem citadas

Exercícios

31. (POSCOMP 2005 - 23) Das afirmações a seguir, sobre memória cache, quais são verdadeiras?

- I. Numa estrutura totalmente associativa, um bloco de memória pode ser mapeado em qualquer slot do cache.**
 - II. O campo tag do endereço é usado para identificar um bloco válido no cache, junto com o campo de índice.**
 - III. Um cache de nível 2 serve para reduzir a penalidade no caso de falta no nível 1.**
 - IV. O esquema de substituição LRU é o mais usado para a estrutura de mapeamento direto.**
- a. Somente as afirmações (I), (III) e (IV).
 - b. Somente as afirmações (II), (III) e (IV).
 - c. Somente as afirmações (I) e (II).
 - d. Somente as afirmações (I), (II) e (III).
 - e. Somente as afirmações (II) e (III).

Resposta de Exercícios

31. (POSCOMP 2005 - 23) Das afirmações a seguir, sobre memória cache, quais são verdadeiras?

- I. Numa estrutura totalmente associativa, um bloco de memória pode ser mapeado em qualquer slot do cache.
 - II. O campo tag do endereço é usado para identificar um bloco válido no cache, junto com o campo de índice.
 - III. Um cache de nível 2 serve para reduzir a penalidade no caso de falta no nível 1.
 - IV. O esquema de substituição LRU é o mais usado para a estrutura de mapeamento direto.
- a. Somente as afirmações (I), (III) e (IV).
 - b. Somente as afirmações (II), (III) e (IV).
 - c. Somente as afirmações (I) e (II).
 - d. Somente as afirmações (I), (II) e (III).
 - e. Somente as afirmações (II) e (III).

Exercícios

32. (POSCOMP 2008 - 55) Analise as seguintes afirmativas

- I. O processador que apresenta o melhor desempenho é sempre aquele que tem a frequência de relógio mais alta.
- II. A técnica de pipeline é utilizada para aumentar o desempenho em processadores. Dessa forma, o pipeline alivia o tempo de latência das instruções.
- III. A maneira mais simples de aumentar a taxa de acertos em memória cache é aumentar a sua capacidade.
- IV. Em arquiteturas superescalares, os efeitos das dependências e antidependências de dados são reduzidos na etapa de renomeação de registradores.

A análise permite concluir que:

- a) Todas as afirmativas são verdadeiras.
- b) Somente as afirmativas II e III são verdadeiras.
- c) Somente as afirmativas III e IV são verdadeiras.
- d) Somente as afirmativas II, III e IV são verdadeiras.
- e) Nenhuma das afirmativas é verdadeira.

Resposta de Exercícios

32. (POSCOMP 2008 - 55) Analise as seguintes afirmativas

- I. O processador que apresenta o melhor desempenho é sempre aquele que tem a frequência de relógio mais alta.
- II. A técnica de pipeline é utilizada para aumentar o desempenho em processadores. Dessa forma, o pipeline alivia o tempo de latência das instruções.
- III. A maneira mais simples de aumentar a taxa de acertos em memória cache é aumentar a sua capacidade.
- IV. Em arquiteturas superescalares, os efeitos das dependências e antidependências de dados são reduzidos na etapa de renomeação de registradores.

A análise permite concluir que:

- a) Todas as afirmativas são verdadeiras.
- b) Somente as afirmativas II e III são verdadeiras.
- c) Somente as afirmativas III e IV são verdadeiras.
- d) Somente as afirmativas II, III e IV são verdadeiras.
- e) Nenhuma das afirmativas é verdadeira.

Exercícios

33. (POSCOMP 2010 - 33) Um computador apresenta um sistema de memória organizado em quatro níveis: memórias cache níveis 1 e 2, memórias RAM principal e secundária. Programas prontos para execução são trazidos da memória secundária e transformados em processos na memória principal. Uma instrução para acessar dados na memória fornece o endereço real de memória onde se localiza a informação desejada. A informação é então buscada na cache nível 1. Se lá não for encontrada, ela é buscada no segundo nível de cache. Não sendo encontrada, a informação é finalmente buscada na memória principal.

Qual o modo de endereçamento utilizado?

- a) Imediato
- b) Indireto
- c) Direto
- d) Implícito
- e) Relativo

Resposta de Exercícios

33. (POSCOMP 2010 - 33) Um computador apresenta um sistema de memória organizado em quatro níveis: memórias cache níveis 1 e 2, memórias RAM principal e secundária. Programas prontos para execução são trazidos da memória secundária e transformados em processos na memória principal. Uma instrução para acessar dados na memória fornece o endereço real de memória onde se localiza a informação desejada. A informação é então buscada na cache nível 1. Se lá não for encontrada, ela é buscada no segundo nível de cache. Não sendo encontrada, a informação é finalmente buscada na memória principal.

Qual o modo de endereçamento utilizado?

- a) Imediato
- b) Indireto
- c) Direto
- d) Implícito
- e) Relativo

Exercícios

Considere uma memória principal de 256 bytes, endereçado a byte, e caches de 4 linhas, cada linha contendo um bloco de 8 bytes, além do espaço para colocar os bits de controle (bit de validade e tag), se forem necessários. Dadas as sequências de acesso à memória principal tabeladas abaixo, preencha a cache para o caso de **mapeamento conjunto associativo**, com **2 conjuntos**. Considere a política para substituição é LRU. Diga quantos miss e quantos hits aconteceram

Sequência de acessos (posição representada em hexadecimal)

Acesso	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Posição	16	17	18	19	16	17	18	19	96	97	98	99	1A	1B	C8	C9	CA	1B	1C	1D

Hits = , Miss =

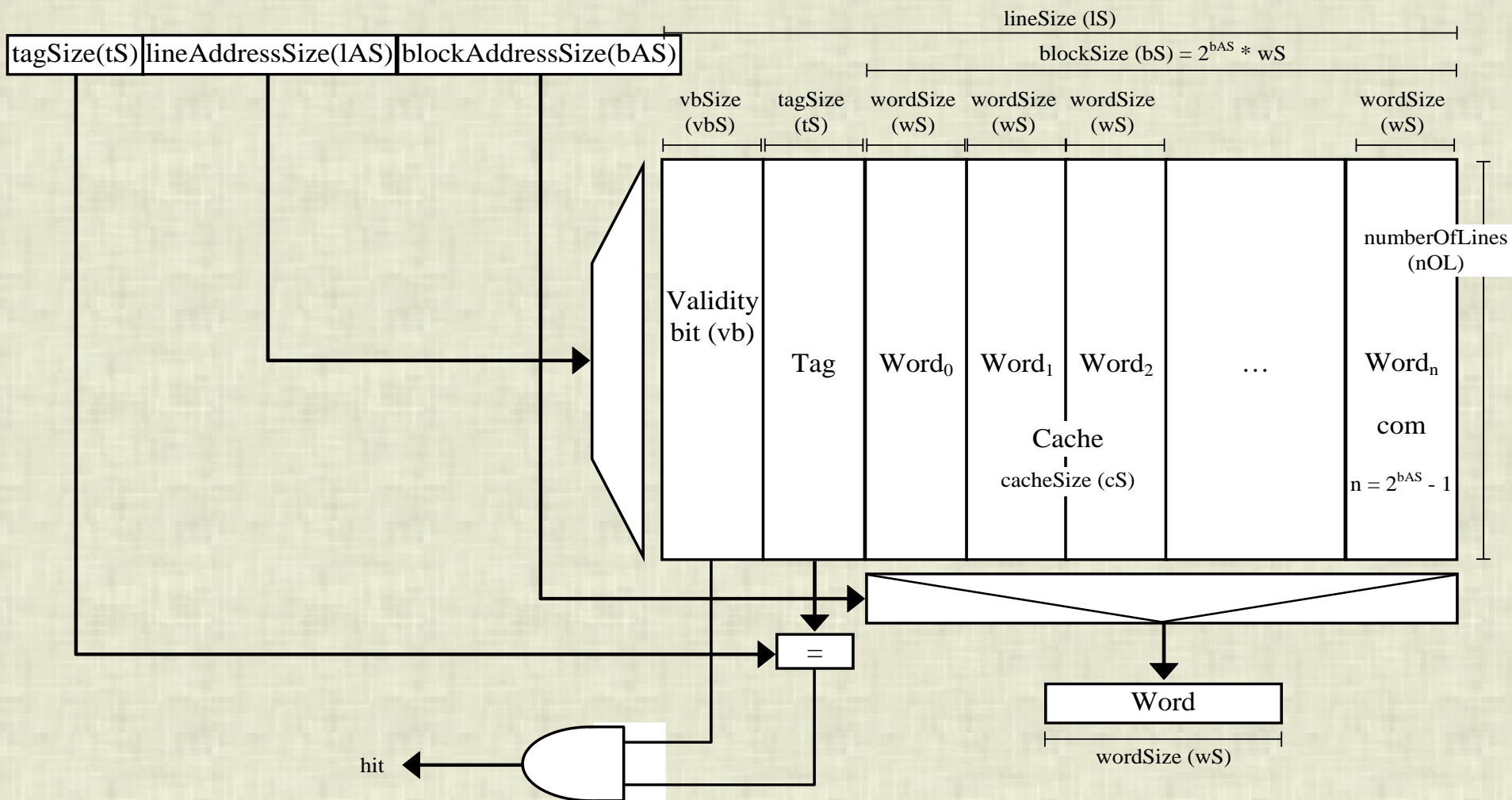
[illegible]

SLIDES EXTRAS

Mapeamento Direto - Exercícios

3. Supor as seguintes características: (i) Memória principal de 2 GBytes, com palavras de 32 bits; (ii) Cache com 64 Kbits; (iii) Blocos de 16 palavras

Mostre como ficará distribuída a cache e o formato da palavra de endereço



Mapeamento Direto

Obtendo o formato do Endereçamento

O formato de instrução pode ser obtido com as relações que seguem:

- $\text{cacheSize} \geq \text{lineSize} \times \text{numberOfLines}$
 - $\text{lineSize} = \text{numberOfWordInsideBlocks} \times \text{wordSize} + \text{tagSize} + \text{bitValiditySize}$
 - $\text{numberOfLines} = 2^{\text{lineAddressSize}}$
 - $\text{validityBitSize} = 1$
 - **$\text{cacheSize} \geq (\text{numberOfWordInsideBlocks} \times \text{wordSize} + \text{tagSize} + 1) \times 2^{\text{lineAddressSize}}$**
- $\text{addressSize} = \text{tagSize} + \text{lineAddressSize} + \text{blockAddressSize}$
 - $\text{addressSize} = \log_2(\text{memorySize})$
 - $\text{blockAddressSize} = \log_2(\text{numberOfWordInsideBlocks})$
 - $\log_2(\text{memorySize}) = \text{tagSize} + \text{lineAddressSize} + \log_2(\text{numberOfWordInsideBlocks})$
 - **$\text{tagSize} = \log_2(\text{memorySize}) - \text{lineAddressSize} - \log_2(\text{numberOfWordInsideBlocks})$**

Mapeamento Direto

Obtendo o formato do Endereçamento

Voltando ao problema:

- Memória principal de 8GBytes, com palavras de 32 bits (word = 4 bytes)
 - **memorySize = 2GWords**
- Cache com 64Kbits
 - **cacheSize = 64KBits = $64 \times 1024 = 65536$ bits**
- Blocos de 16 palavras
 - **numberOfWordInsideBlocks = 16**
 - **blockAddressSize = $\log_2(16) = 4$**

Aplicando as relações, teremos:

- **$\text{cacheSize} \geq (\text{numberOfWordInsideBlocks} \times \text{wordSize} + \text{tagSize} + 1) \times 2^{\text{lineAddressSize}}$**
 - $65536 \geq (16 \times 32 + \text{tagSize} + 1) \times 2^{\text{lineAddressSize}}$
 - $65536 \geq (512 + \text{tagSize} + 1) \times 2^{\text{lineAddressSize}}$
 - **$65536 \geq (513 + \text{tagSize}) \times 2^{\text{lineAddressSize}}$**
- **$\text{tagSize} = \log_2(\text{memorySize}) - \text{lineAddressSize} - \log_2(\text{numberOfWordInsideBlocks})$**
 - $\text{tagSize} = \log_2(2 \text{ Giga}) - \text{lineAddressSize} - 4$
 - $\text{tagSize} = 31 - \text{lineAddressSize} - 4$
 - **$\text{tagSize} = 27 - \text{lineAddressSize}$**

Mapeamento Direto

Obtendo o formato do Endereçamento

Finalizando:

Substituindo o termo da equação na inequação, teremos:

- $65536 \geq (513 + 27 - \text{lineAddressSize}) \times 2^{\text{lineAddressSize}}$
- $65536 \geq (540 - \text{lineAddressSize}) \times 2^{\text{lineAddressSize}}$

A solução da inequação pode ser feita através de tentativa e erro, de forma a chegar ao valor que mais se aproxima da solução:

- $\text{lineAddressSize} = 6 \Rightarrow$
 - $65536 \geq (540 - 6) \times 26$
 - $65536 \geq 34176$
- $\text{tagSize} = 27 - \text{lineAddressSize} \Rightarrow$
 - $\text{tagSize} = 27 - 6$
 - $\text{tagSize} = 21$

Assim, o formato do endereçamento fica:

tagSize = 21	lineAddressSize = 6	blockAddressSize = 4
--------------	---------------------	----------------------