UFRRJ
UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO

Arquitetura de Comptuadores II

Sub-Sistema de Memória

Marcelo Zamith
zamith.marcelo@gmail.com

Sub-Sistema de Memória

- Componentes
- Mecanismo de acesso do processador
- Dispositivos de memória
  - Principais tipos
  - Características
- Conceitos de memória
  - Principal
  - Cache
  - Virtual

Localidade de Referência

- Este princípio estabelece que os programas acessam uma parte relativamente pequena do seu espaço de endereçamento em um dado instante de tempo
- Dois tipos diferentes de localidade
  - Temporal
  - Espacial

Localidade de Referência

- Localidade Temporal
  - Se um item é referenciado, ele tende a ser referenciado novamente dentro de um espaço de tempo pequeno
- Localidade Espacial
  - Se um item é referenciado, outros itens cujos endereços estejam próximos dele tendem a ser referenciados rapidamente

Localidade de Referência

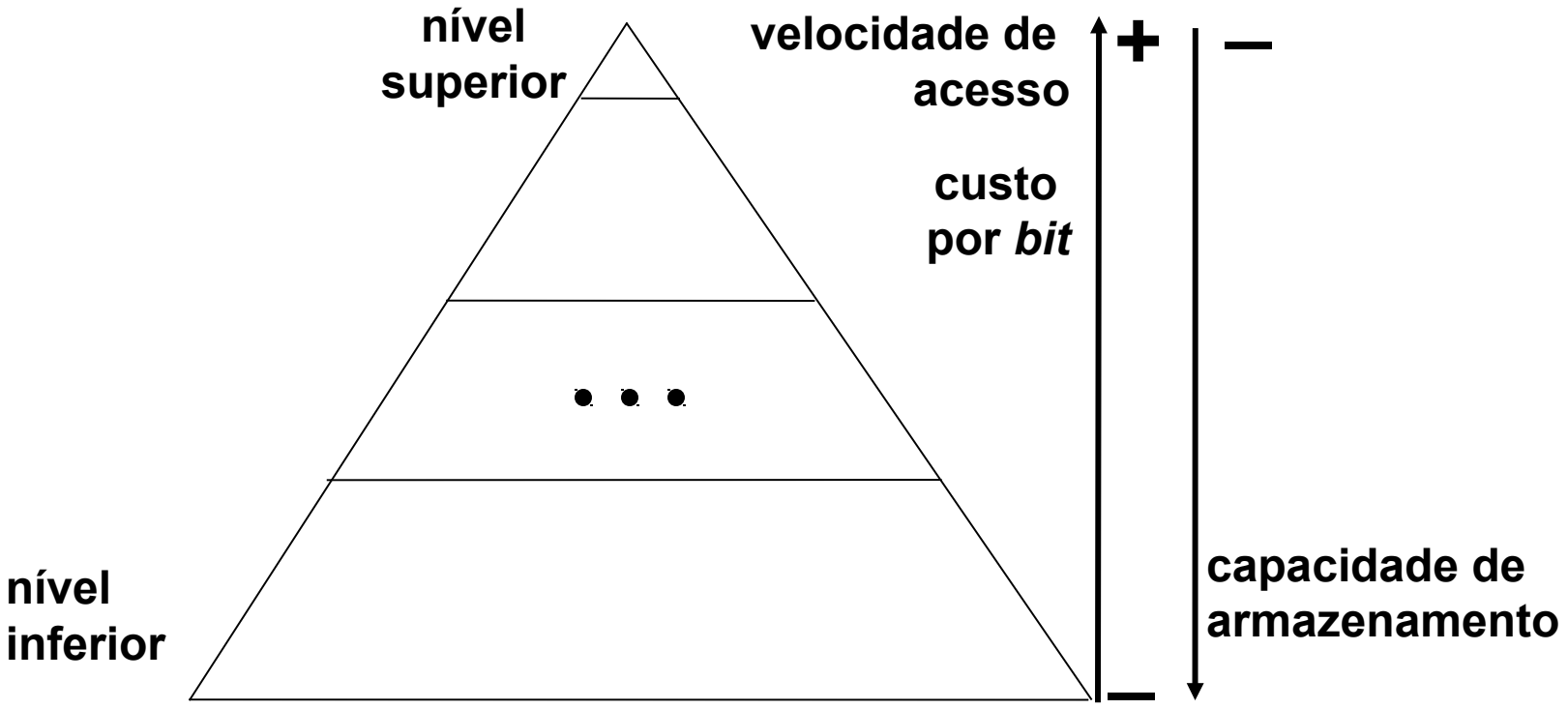
- O princípio da localidade é utilizado para implementar o sub-sistema de memória de um computador
- Esta organização é denominada **hierarquia de memória**

Hierarquia de Memória

- Vários níveis de memória
  - Capacidade de armazenamento
  - Velocidade de acesso
  - Custo por bit

Hierarquia de Memória

- Vários níveis de memória



Hierarquia de Memória

- Os dados são copiados entre dois níveis adjacentes
- Os níveis superiores são os mais próximos ao processador e os níveis inferiores são mais distantes do processador
- A unidade mínima de transferência de dados entre dois níveis adjacentes é chamada de **bloco**

Hierarquia de Memória

- **Acerto** (*hit*)
  - Os dados solicitados pelo processador aparecem em algum bloco no nível superior
- **Falha** (*miss*)
  - Os dados estão ausentes no nível superior
  - O nível inferior da hierarquia é acessado para que seja possível recuperar o bloco com a informação solicitada

Hierarquia de Memória

- **Taxa de acertos**
  - Fração dos acessos à memória encontrados no nível superior
  - Medida de desempenho da hierarquia de memória
- **Taxa de falhas** (1 – taxa de acertos)
  - Proporção dos acessos à memória não encontrados no nível superior

Hierarquia de Memória

- **Tempo de acerto**
  - Tempo para acessar o nível superior da hierarquia de memória, que inclui o tempo necessário para determinar se o acesso é um acerto ou uma falha
- **Penalidade de falha**
  - Somatório dos tempos necessários para acessar o nível inferior e trazer o bloco para o nível superior da hierarquia

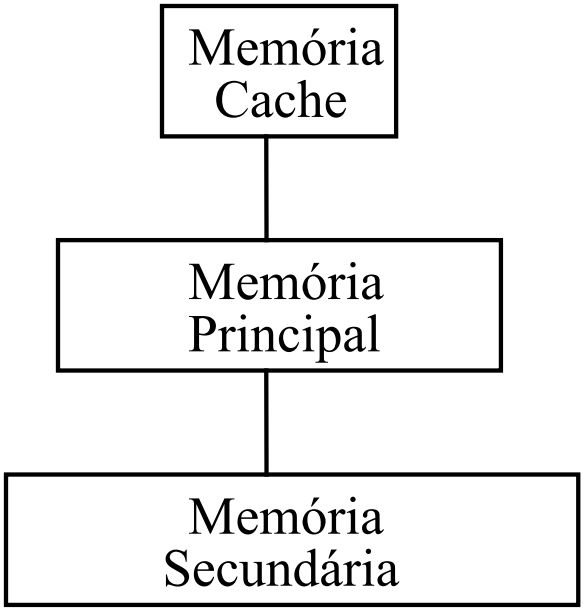
Hierarquia de Memória

- Num sistema de computador tanto a CPU quanto o sub-sistema de E/S interagem com a memória

- Cada conteúdo (palavra ou byte) armazenado na memória possui seu próprio endereço
  - Interação é feita através de uma seqüência de leituras e escritas a endereços de memória específicos

Hierarquia de Memória

- Um sistema de computador possui tipos de memória com diferentes características que formam uma hierarquia



Hierarquia de Memória

- No nível mais inferior, as memórias secundárias são capazes de armazenar uma grande quantidade de informação, seu custo por bit é menor e o seu tempo de acesso é relativamente maior do que as memórias dos níveis superiores

Hierarquia de Memória

- No segundo nível, a memória principal é capaz de armazenar uma quantidade menor de informação, seu custo por bit é maior e seu tempo de acesso é menor do que as memórias secundárias

Hierarquia de Memória

- No nível mais superior, as memórias *cache* são as mais rápidas e com o maior custo por bit
  - Por serem muito caras as memórias *cache* são as que têm menor capacidade de armazenamento em relação as demais

Gerenciamento de Memória

- Regras ou políticas de gerenciamento
  - Busca: quando um bloco de informação deve ser transferido de um nível de memória inferior para um nível de memória superior
  - Armazenamento: onde o bloco de informação deve ser colocado naquele nível de memória
  - Substituição: qual bloco de informação deve ser substituído por um novo bloco

Gerenciamento de Memória

- Existem muitos esquemas diferentes de gerenciamento de memória
- A seleção de um esquema, em especial, depende do suporte de hardware do sistema

Iteração entre Processador e Memória Principal

- O componente básico da memória principal é chamado **célula de bit**
  - Circuito eletrônico que armazena um bit de informação
- Acesso do processador
  - Conjunto de células de bit

Iteração entre Processador e Memória Principal

- O menor conjunto de células de bit que é acessado pelo processador é chamado **locação de memória**

- Na maioria dos computadores, uma locação de memória é formada por 8 células de bit
  - **Byte de memória**

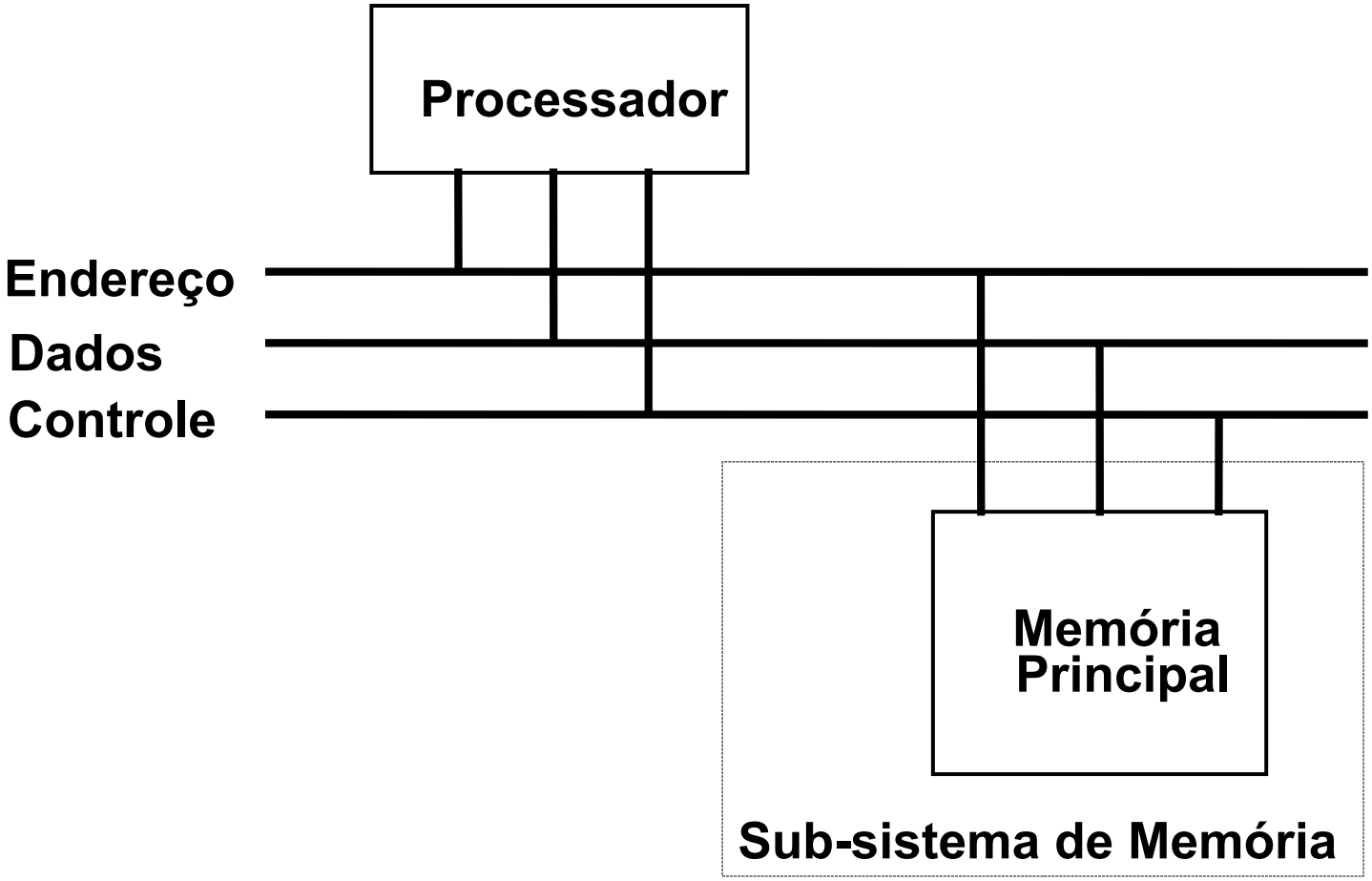
Iteração entre Processador e Memória Principal

- Em um acesso, o processador deve fornecer:
  - Identificação da locação, onde será feito o acesso, que é o **endereço de memória**
  - Sentido do acesso, que pode ser de **leitura** ou de **escrita**

Iteração entre Processador e Memória Principal

- Iteração através de três barramentos distintos:
  - Endereço
  - Dado
  - Controle

Iteração entre Processador e Memória Principal



Iteração entre Processador e Memória Principal

- Barramento de endereço
  - Via através da qual o processador transmite para a memória principal o endereço da locação onde será feito o acesso
- Barramento de dados
  - Via através da qual o conteúdo da locação é transferido entre o processador e a memória principal

Iteração entre Processador e Memória Principal

- Barramento de controle
  - É formado por diversos sinais através dos quais o processador controla o acesso à memória, indicando, por exemplo, se o acesso é de leitura ou de escrita

Barramento

- Largura do barramento, em número de bits, é determinado basicamente pelo processador usado no sistema

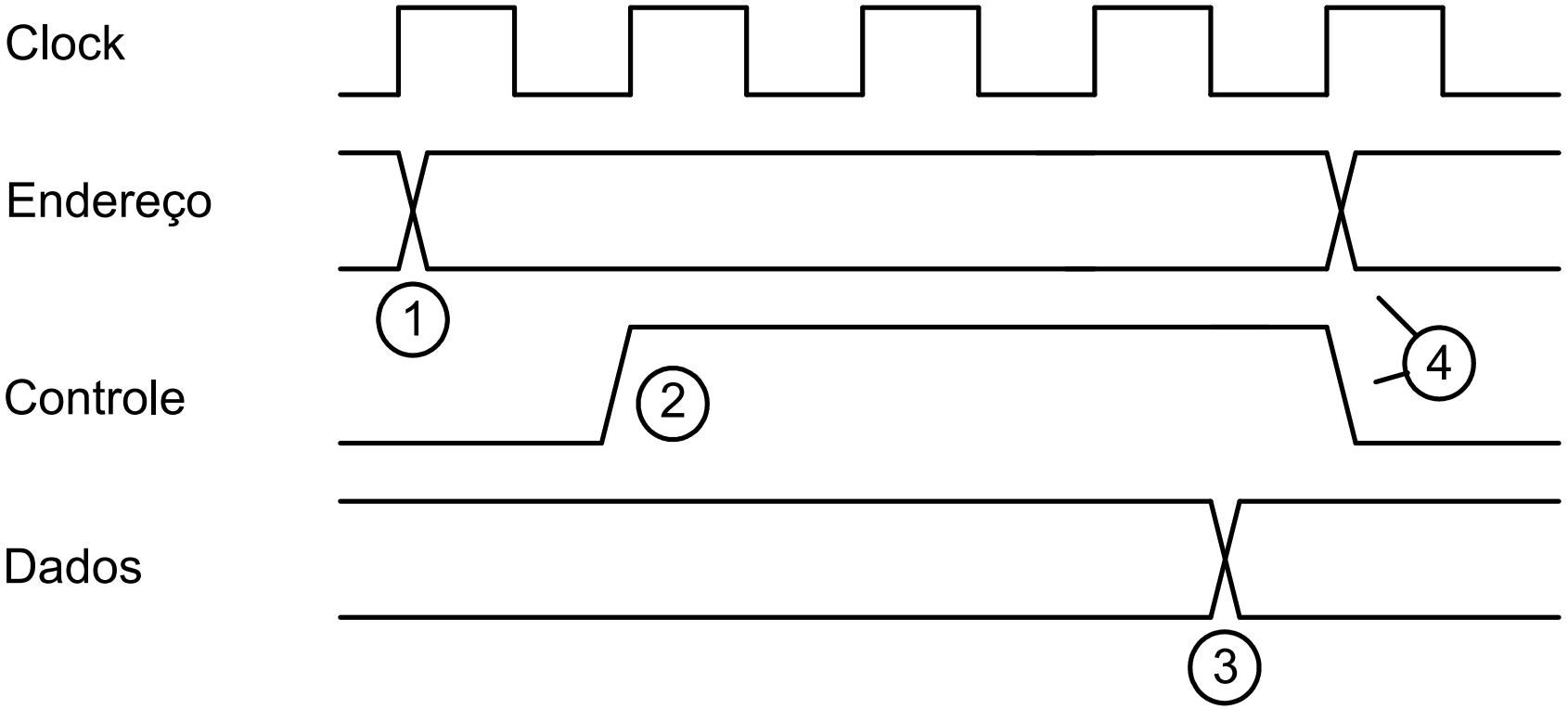
- Barramento de endereço com largura de **n** bits pode endereçar até $\mathbf{2^n}$ locações de memória distintas
  - Não necessariamente a memória toda é instalada, depende da relação custo/benefício

Ciclo de Barramento

- A sequência de eventos que acontecem durante um acesso do processador à memória principal é denominado de **ciclo de barramento** (*bus cycle*)
  - Definição da locação de memória
  - Ativação de sinais de controle
  - Troca de informações entre o processador e a memória principal

Ciclo de Barramento

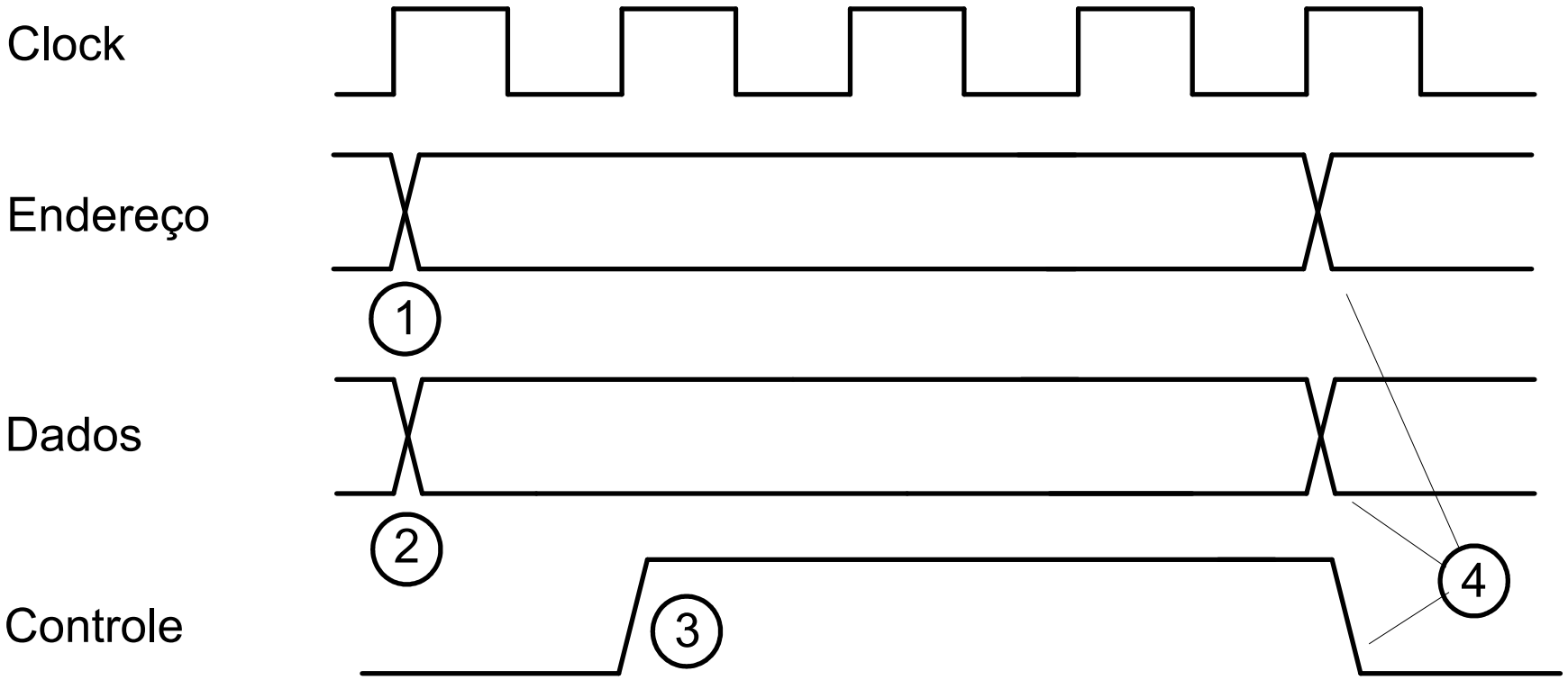
- Representação gráfica por diagrama de tempo



Leitura

Ciclo de Barramento

- Representação gráfica por diagrama de tempo



Escrita

Ciclo de Barramento

- Nos diagramas mostrados, o número de ciclos de *clock* que separam os eventos ao longo de um ciclo de barramento são hipotéticos
- O número de ciclos de *clock* consumidos no ciclo de barramento varia entre os processadores

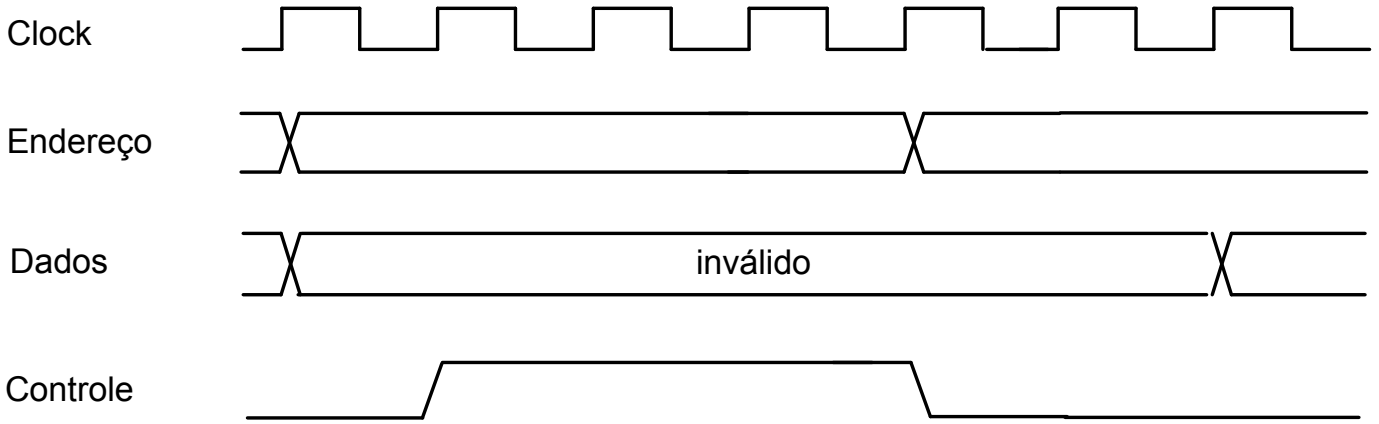
Ciclo de Barramento

- O tamanho do ciclo de barramento, em termos do número de ciclos de *clock*, é a princípio determinado pelo processador

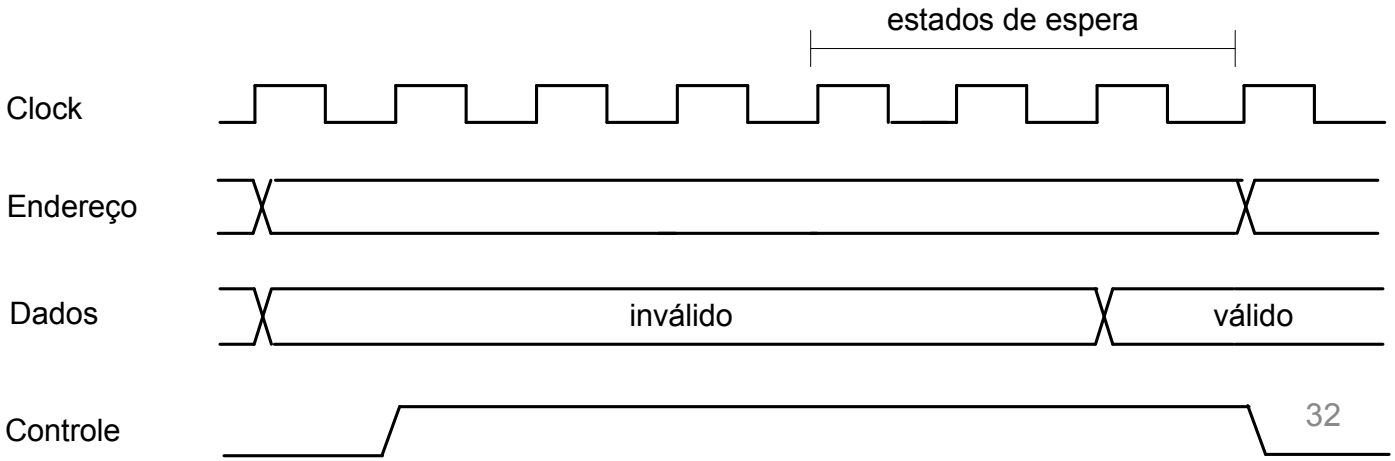
- Caso não haja uma solicitação externa quanto ao tamanho apropriado do ciclo de barramento, o processador sempre executará um **ciclo de barramento padrão** com um certo número de ciclos

Estados de Espera

- Tempo de acesso



Leitura



Estados de Espera

- Em geral, os processadores oferecem um mecanismo que permite compatibilizar o tamanho do ciclo de barramento ao tempo de acesso da memória

- O processador possui um sinal de entrada que, ao ser ativado, faz com que o tamanho do ciclo de barramento seja estendido pela introdução de ciclos de *clock* extras (**estados de espera**)

Estados de Espera

- Quando a memória detecta o início de um ciclo de barramento, a memória principal envia um sinal solicitando ao processador que o ciclo de barramento seja estendido

- Ao colocar a informação endereçada no barramento de dados, a memória principal retira a solicitação de espera

Tipos de Dispositivo

- ROM (*Read Only Memory*): Memória somente de leitura e não volátil
  - PROM (*Programmable ROM*)
  - EPROM (*Erasable Programmable ROM*)
  - EEPROM (*Electrically Erasable Programmable ROM*)
- RAM (*Random Access Memory*)
  - SRAM (*Static RAM*)
  - DRAM (*Dynamic RAM*)

Tipos de Dispositivo

- PROM (*Programmable ROM*)
  - Memória programável somente de leitura
  - Permite apenas uma única programação
  - As informações armazenadas não podem ser modificadas após o seu armazenamento

Tipos de Dispositivo

- EPROM (*Erasable Programmable ROM*)
  - Memória programável e apagável somente de leitura
  - O processador realiza apenas acessos de leitura a uma EPROM
  - O conteúdo de uma memória EPROM pode ser apagado (através de luz ultra-violeta) e a memória pode ser novamente usada para armazenar um novo conjunto de informações

Tipos de Dispositivo

- EEPROM (*Electrically Erasable Programmable ROM*)
  - Memória programável e apagável eletricamente somente de leitura
  - Pode ser reprogramada sem ser retirada do circuito, desde que o circuito seja construído com esta função

Tipos de Dispositivo

- A memória ROM é usada para armazenar o *firmware* do computador
  - O *firmware* é formado por sub-rotinas usadas pelo sistema operacional e que interagem diretamente com o hardware do computador
  - BIOS (*Basic Input/Output System*) é um exemplo de *firmware*, em que as rotinas são armazenadas em memórias do tipo ROM

Tipos de Dispositivo

- RAM (*Random Access Memory*)
  - O processador pode efetuar acessos de leitura e escrita a uma memória RAM
  - As memórias deste tipo são usadas para armazenar as instruções e dados de um programa em execução
  - As memórias SRAM (*Static RAM*) e DRAM (*Dynamic RAM*) diferem quanto à capacidade de armazenamento e ao tempo de acesso

Tipos de Dispositivo

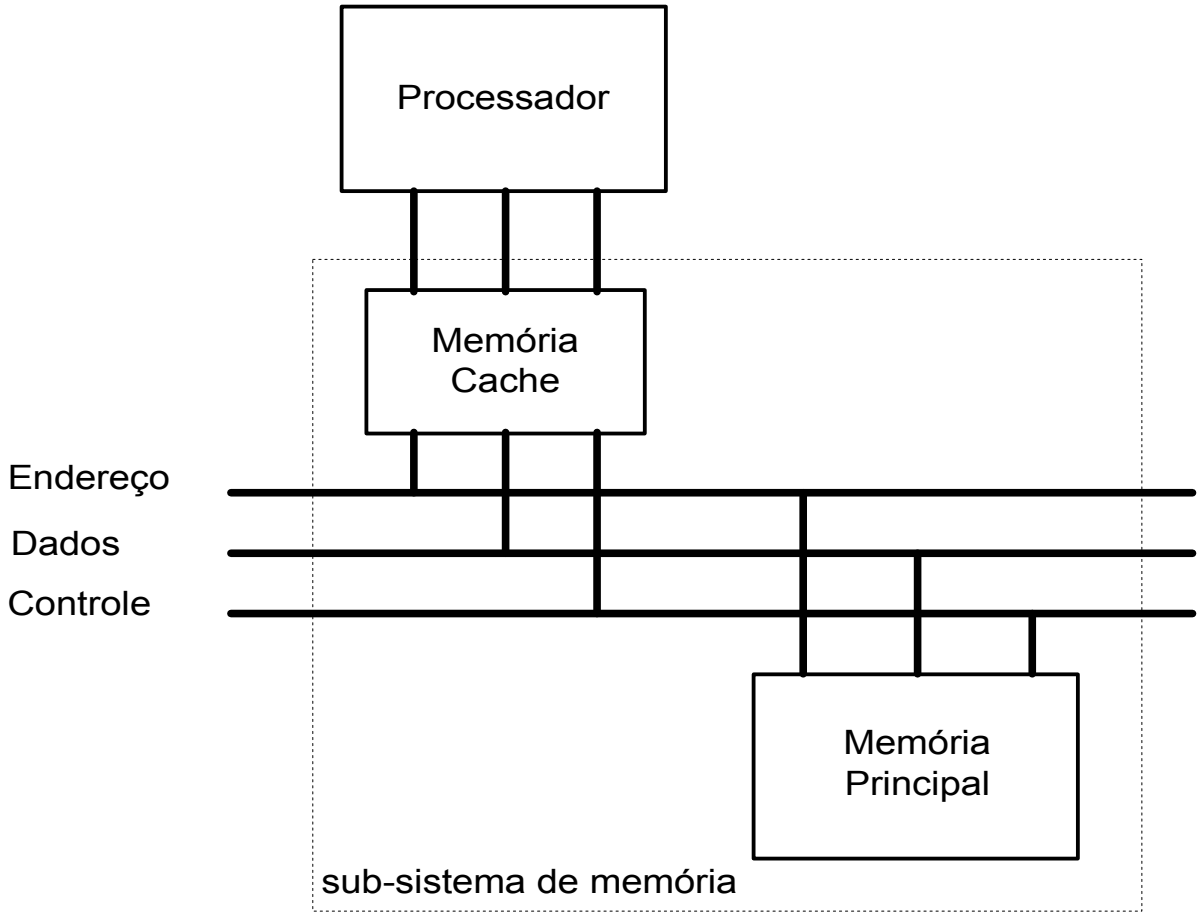
- SRAM x DRAM
  - A capacidade de um dispositivo DRAM é maior que a de um dispositivo SRAM
    - Nas memórias DRAM, a célula de bit é implementada por um circuito eletrônico que ocupa uma área de integração menor que a ocupada pelo circuito usado nas memórias SRAM
    - Como a área por célula de bit é menor, para a mesma área total de integração o número total de células de bit em uma DRAM é maior
  - O tempo de acesso de uma SRAM é menor que o tempo de acesso de uma DRAM

Hierárquia de Memória

- A solução para obter o compromisso desejado entre a capacidade de armazenamento, desempenho e custo encontra-se no uso apropriado de tecnologias de memória com diferentes relações entre estes três fatores

Memória Cache

- Localização da memória cache



Memória Cache

- Implementação => SRAM
- Cache
  - Hit => Acerto
  - Miss => Falha
- Propriedade da localidade de referência
  - Taxa de acerto ≈ 90% a 98%

Memória Cache Mapeamento

- Cache
  - Organizada em linhas
  - Cada linha possui:
    - **Bloco de dados** proveniente da memória principal
    - **Rótulo (*tag*)** para identificar se uma palavra na cache corresponde à palavra requisitada
    - **Bit de validade** para indicar se uma entrada da cache contém um endereço válido
- Mapeamento
  - Direto
  - Por associatividade

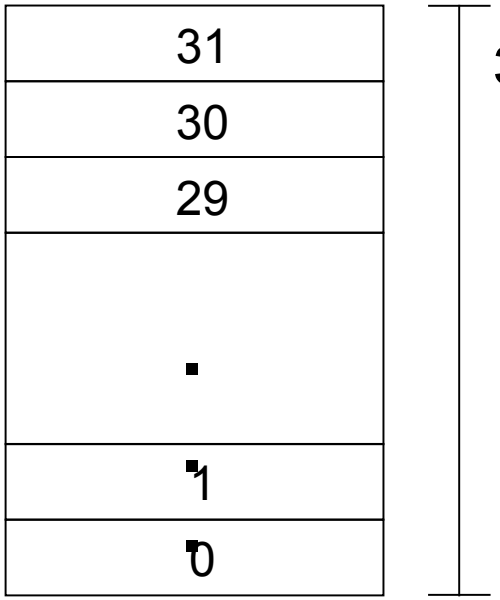
Cache Mapeada Diretamente

- Organização mais simples
- Uma palavra só pode ocupar uma única posição na cache
- Processo de mapeamento

  (endereço do bloco) % (num. de blocos da cache)
- Número de comparações: 1

Mapeamento Direto

Memória Principal

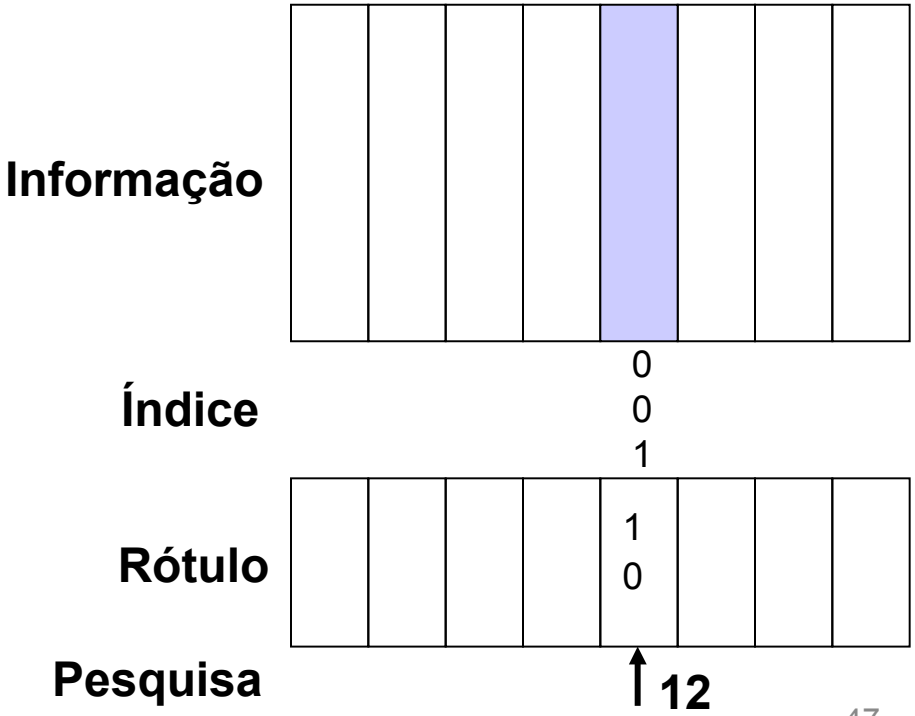


Faixa de endereços

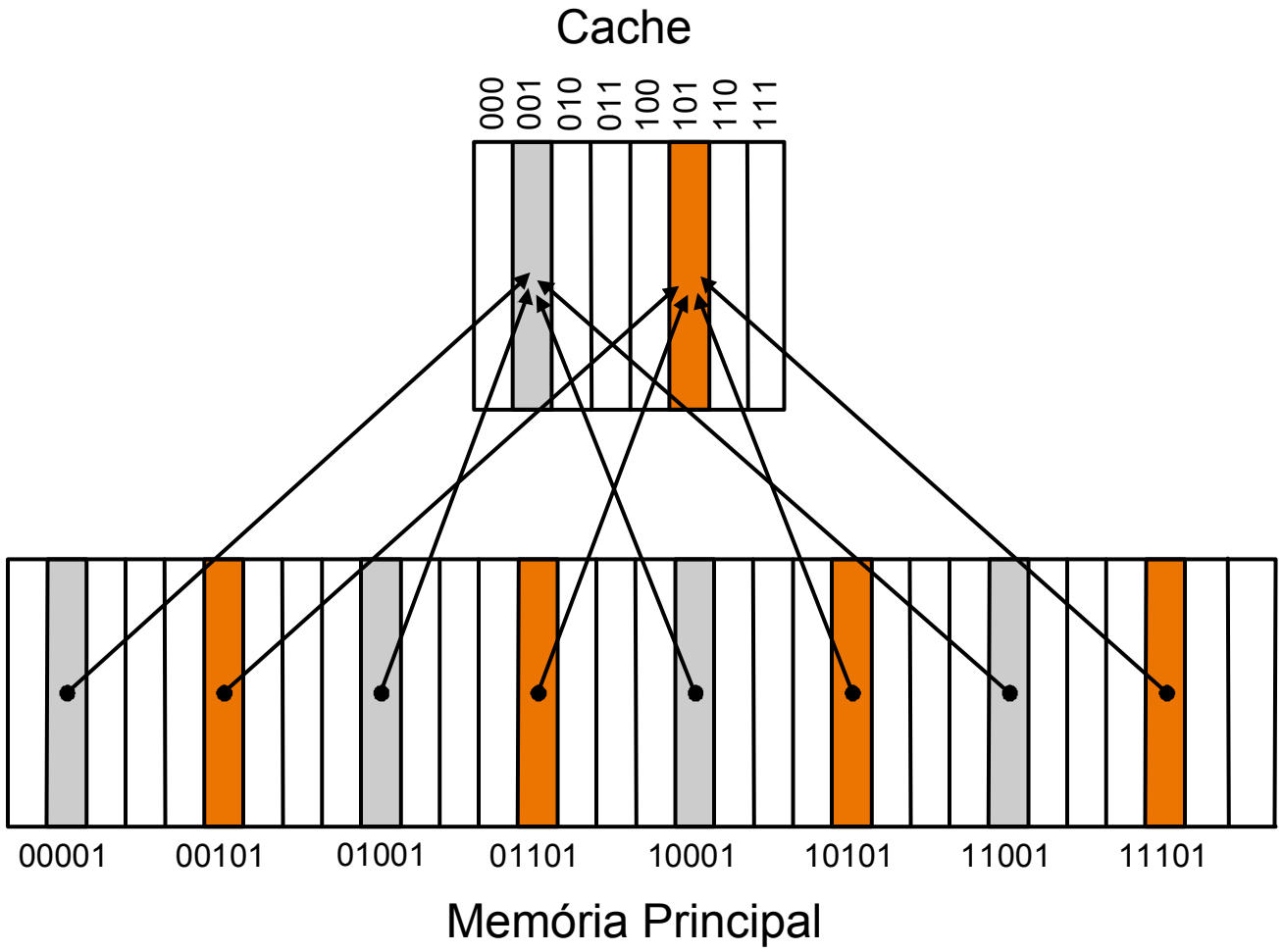
0 0 0 0 0
0 0 0 0 1
.....
1 1 1 1 1

12 % 8 = 4

0 1 1 0 0



Mapeamento Direto



Mapeamento Direto

- Considere uma cache com 64 blocos e um tamanho de bloco de 16 bytes. Para qual número de bloco o endereço em bytes 1200 é mapeado?

  Endereço do bloco é:

$$\left\lfloor \frac{\text{Endereço em bytes}}{\text{Bytes por bloco}} \right\rfloor = \left\lfloor \frac{1200}{16} \right\rfloor = 75$$

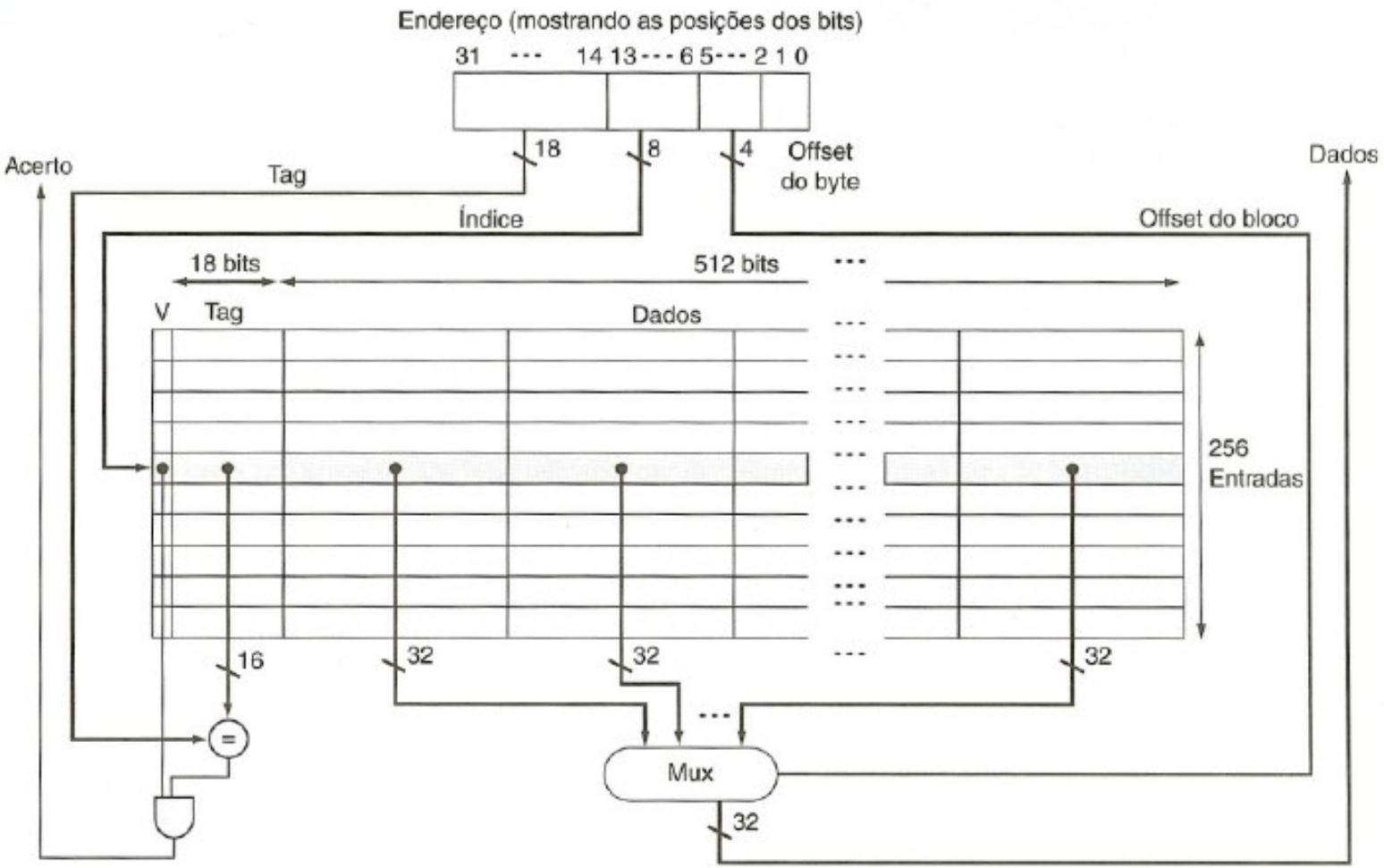
  Número de bloco de cache é:

$$75 \,\%\, 64 = 11$$

Mapeamento Direto

- Exemplo de cache
  - O **Intrinsity FastMATH** é um microprocessador que usa a arquitetura MIPS
  - Implementação de cache simples
  - Cache de instruções e de dados
  - Cada cache de 16KB, ou 4K palavras, contém 256 blocos com 16 palavras por bloco

Mapeamento Direto



Mapeamento Direto

- Considerando o endereço em bytes de 32 bits, uma cache diretamente mapeada de 2^n blocos de tamanho com 2^m palavras (2^{m+2} bytes) por bloco exigirá um campo **tag (rótulo)** cujo tamanho é:

$$32 – (n + m + 2) \text{ bits}$$

n bits para o índice
m bits para a palavra dentro do bloco
2 bits para a parte do byte do endereço

Mapeamento Direto

- Vantagens
  - Não há necessidade de algoritmo de substituição
  - Hardware simples e de baixo custo
  - Alta velocidade de operação
- Desvantagens
  - Redução no desempenho se acessos consecutivos são feitos a palavras com mesmo índice
  - Taxa de acerto (*hit ratio*) inferior ao de caches com mapeamento associativo

Memória Cache Mapeamento

- Bloco com tamanho maior do que uma palavra aproveita melhor as vantagens da localidade espacial
- Blocos maiores diminuem a taxa de falhas e melhoram a eficiência da cache até um determinado limite
  - Um bloco maior aumenta a penalidade nas falhas, pois ela cresce exponencialmente com o tamanho do bloco transferido

Mapeamento Associativo

- Classificação:
  - **Por conjunto**
    - Existe um número fixo de locais (pelo menos dois) onde cada bloco pode ser colocado
    - Uma cache associativa por conjunto com **n** locais para um bloco é chamado de **cache associativa por conjunto de n vias**
  - **Totalmente associativo**
    - Um bloco na memória pode ser associado com qualquer entrada da cache

Associativo por Conjunto

- Processo de mapeamento

  (endereço do bloco) % (num. de conjuntos da cache)

- Número de comparações:
  - Quantidade de elementos por conjunto
  - Palavra pode estar em qualquer elemento do conjunto

Associativo por Conjunto

- Endereço com três campos:
  - **Byte**
    - Bits menos significativos
    - Seleciona um byte específico dentro de um bloco
  - **Conjunto**
    - **n** bits seguintes
    - Seleciona um dos conjuntos
  - **Rótulo**
    - **m** bits mais significativos
    - Verifica se o dado referenciado se encontra em algum elemento do conjunto selecionado

Associativo por Conjunto

Endereço

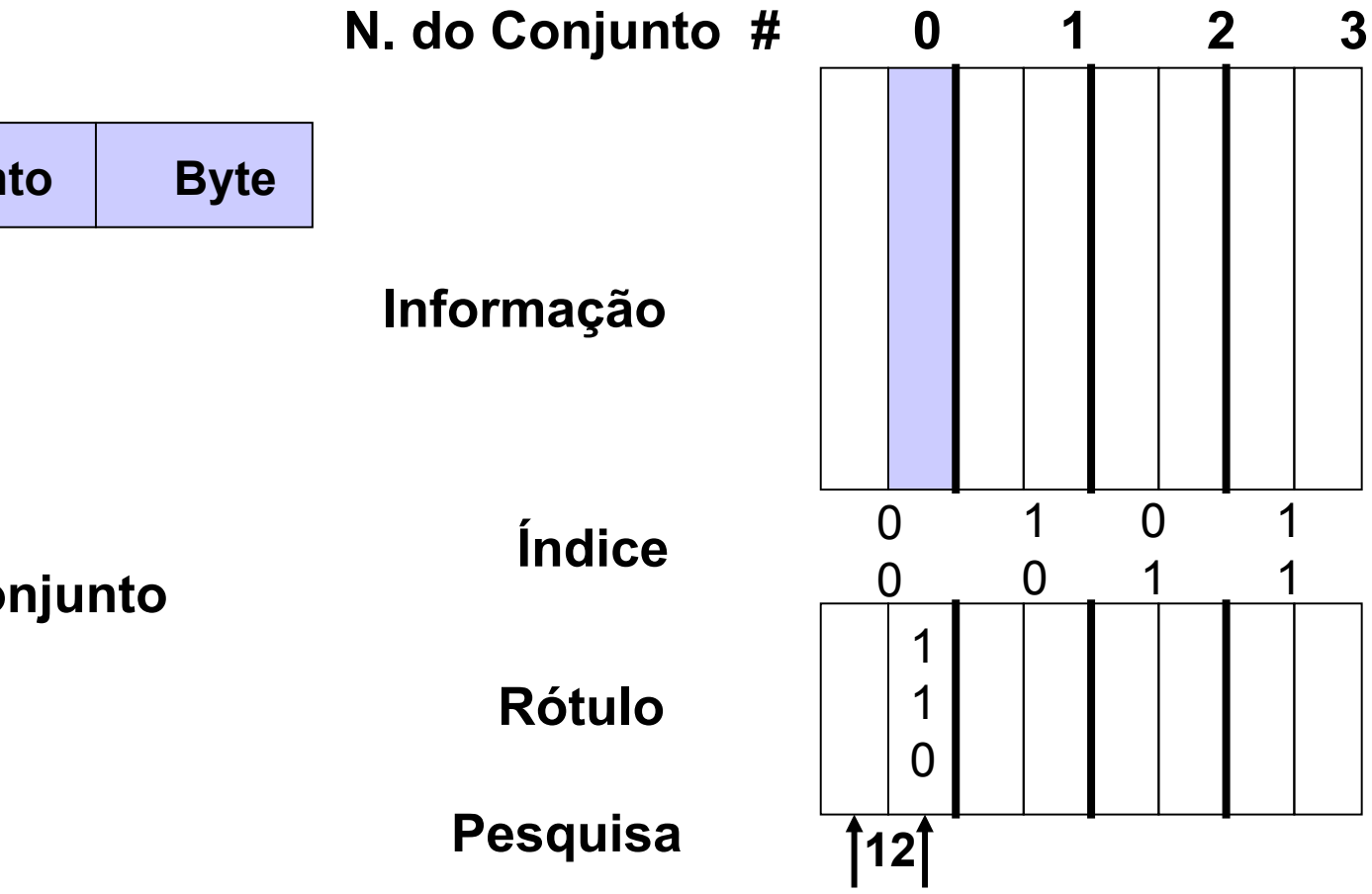
Rótulo	Conjunto	Byte

4 conjuntos

2 blocos por conjunto

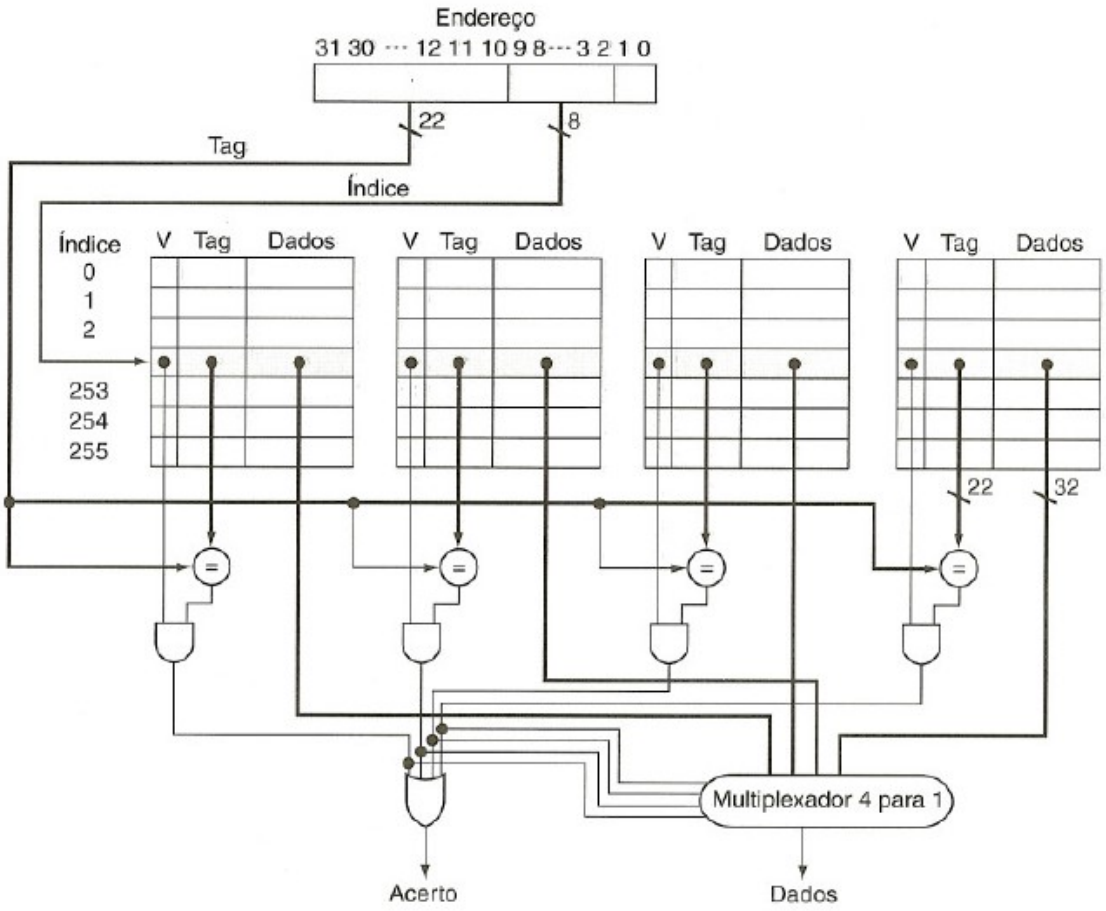
12 % 4 = 0

0 1 1 0 0



Associativo por Conjunto

- Cache associativa por conjunto de 4 vias



Associativo por Conjunto

- Vantagens em relação ao mapeamento totalmente associativo
  - Comparadores são compartilhados por todos os conjuntos
  - Algoritmo de substituição só precisa considerar blocos dentro de um conjunto

Totalmente Associativo

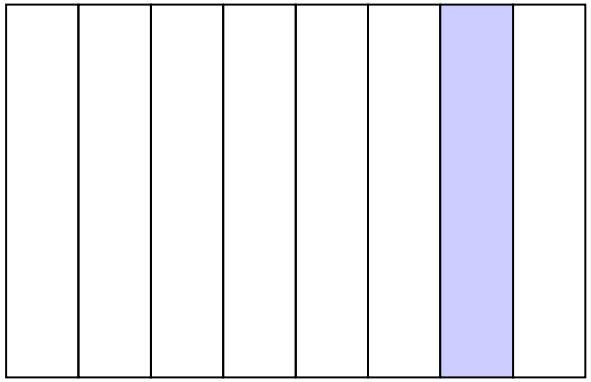
- Pode ser vista como tendo um único conjunto
- Processo de mapeamento
  - **Palavra pode ser colocada em qualquer elemento do conjunto**
- Utilizado apenas em memórias associativas de tamanho pequeno

Totalmente Associativo

1 conjunto

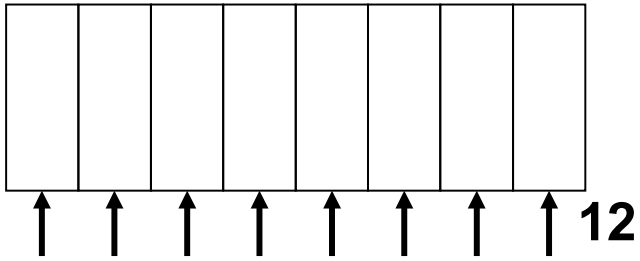
8 elementos no conjunto

Informação



Rótulo

Pesquisa



Totalmente Associativo

- Vantagem
  - Máxima flexibilidade no posicionamento de qualquer bloco da memória principal em qualquer bloco da cache
- Desvantagens
  - Custo em hardware da comparação simultânea de todos os endereços armazenados na cache
  - Algoritmo de substituição (em hardware) para selecionar uma posição da cache na ocorrência de uma falha (*miss*)

Coerência de Cache

- Esquemas
  - **Write-through**: a cada escrita na cache, escreve, também, na memória principal
  - **Write-back**: só escreve na memória principal quando um bloco for substituído ou no término da aplicação
    - Necessita de um bit de controle

Políticas de Substituição

- Políticas
  - **LRU**: o bloco a ser substituído é aquele que não é acessado a mais tempo
  - **Aleatória**: qualquer bloco pode ser escolhido para ser substituído

Cache Multinível

- Minimizar o efeito do cache *miss*
- **Memória cache em dois níveis (*two-level* cache)**
  - **Primária**
    - Capacidade de armazenamento menor
    - Minimização do tempo para tratamento de acertos
  - **Secundária**
    - Capacidade de armazenamento maior
    - Redução da penalidade de falhas

Cache Multinível

- Informação na cache de segundo nível
  - Penalidade de falha será igual ao tempo de acesso à cache de segundo nível
- Nenhum dos dois níveis da cache contiver a informação requisitada
  - Acessar a memória principal
  - Penalidade de falha maior

Memória Virtual

- Técnica originalmente criada para permitir a execução de programas cujas exigências quanto ao tamanho da memória sejam maiores do que a capacidade da memória principal instalada no sistema

Conceito de Memória Virtual

- O endereço gerado pelo programa é o **endereço virtual**
- O **endereço real** é usado para acessar a memória principal
- Os possíveis endereços virtuais que podem ser gerados pelo programa formam o **espaço de endereçamento virtual**
- Os endereços na memória principal formam o **espaço de endereçamento real**

Conceito de Memória Virtual

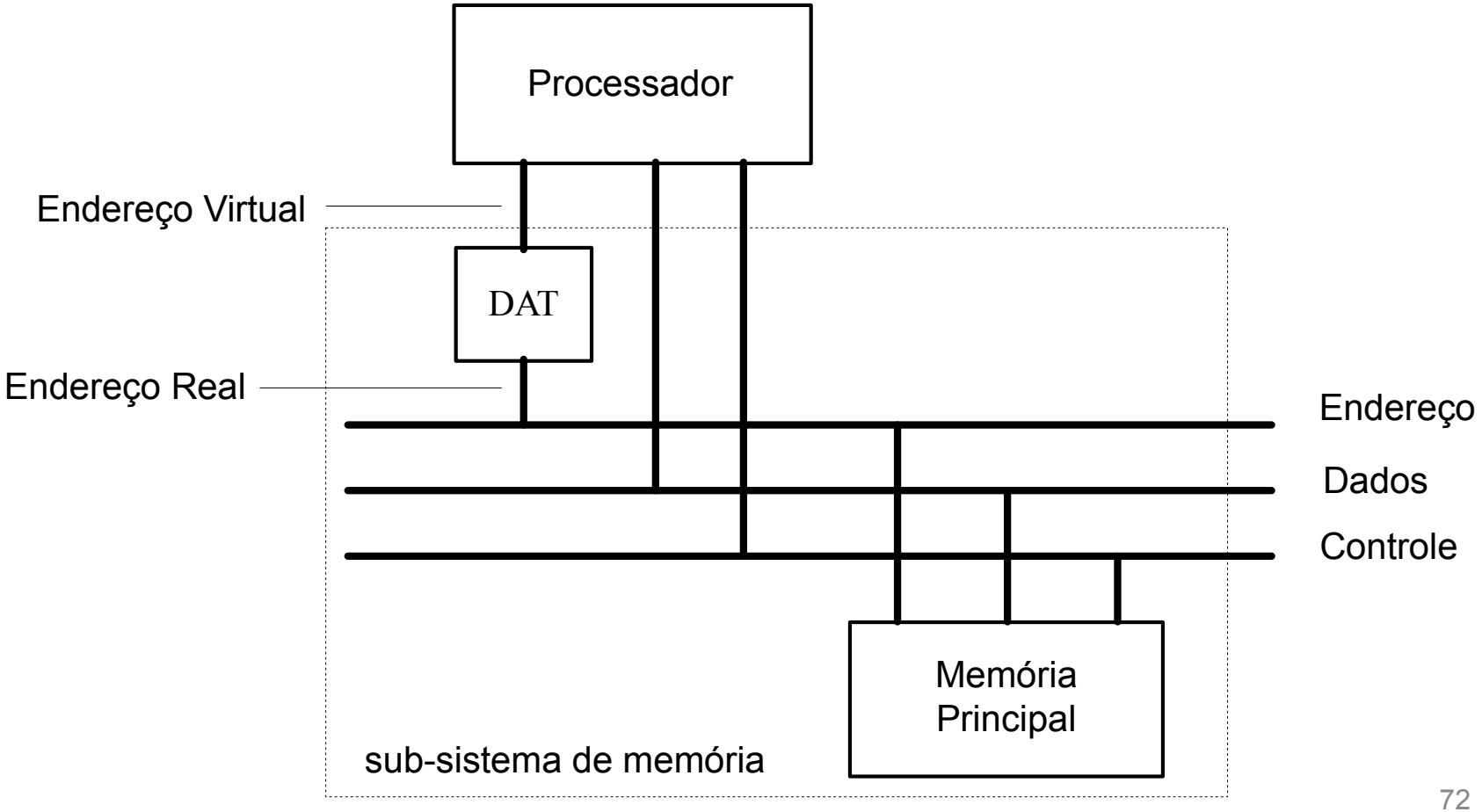
- Do ponto de vista de um programa a memória disponível é aquela representada pelo espaço de endereçamento virtual
- Espaço de endereçamento virtual é uma abstração
- Distinção entre endereços e espaços de endereçamento exige um mecanismo de correspondência entre o endereço virtual e o endereço real

Conceito de Memória Virtual

- Técnica de memória virtual
  - Permite que instruções e dados não estejam na memória principal até que sejam referenciados
  - Necessidade de um mecanismo para o carregamento automático das instruções e dados na memória principal

Mecanismo de Memória Virtual

- Sistema com memória virtual



Mecanismo de Memória Virtual

- Mapeamento realizado pelo **tradutor dinâmico de endereços** ou **DAT** (***Dynamic Address Translator***)
  - Mapeia o endereço virtual em endereço real
  - Usa uma **tabela de mapeamento**
  - Tabela localiza-se na memória principal durante toda a execução do processo
  - Para manter um tamanho de tabela aceitável o mapeamento é feito no nível de blocos

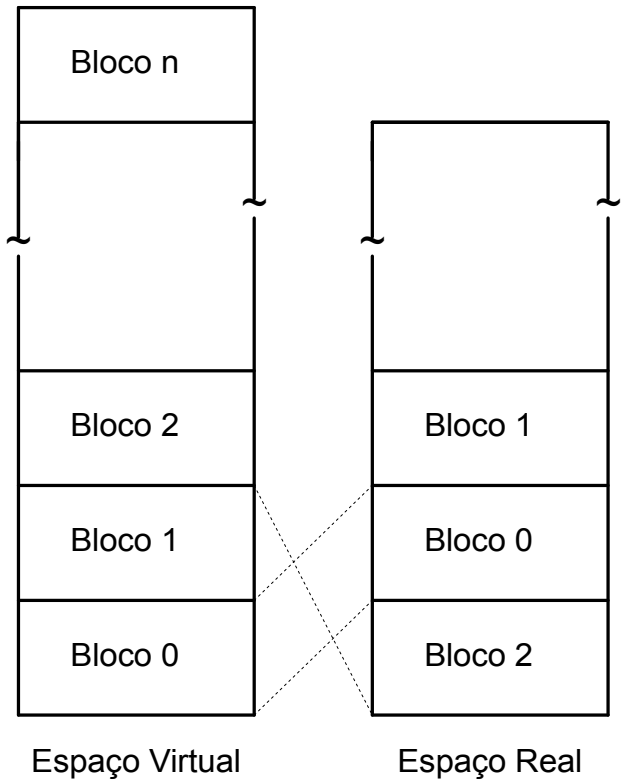
Mecanismo de Memória Virtual

- O espaço de endereçamento virtual é logicamente dividido em blocos, que são mapeados para o espaço de endereçamento real pelo DAT
- Cada entrada na tabela de mapeamento contém o endereço-base de um bloco
  - Endereço real a partir do qual o bloco está armazenado na memória principal

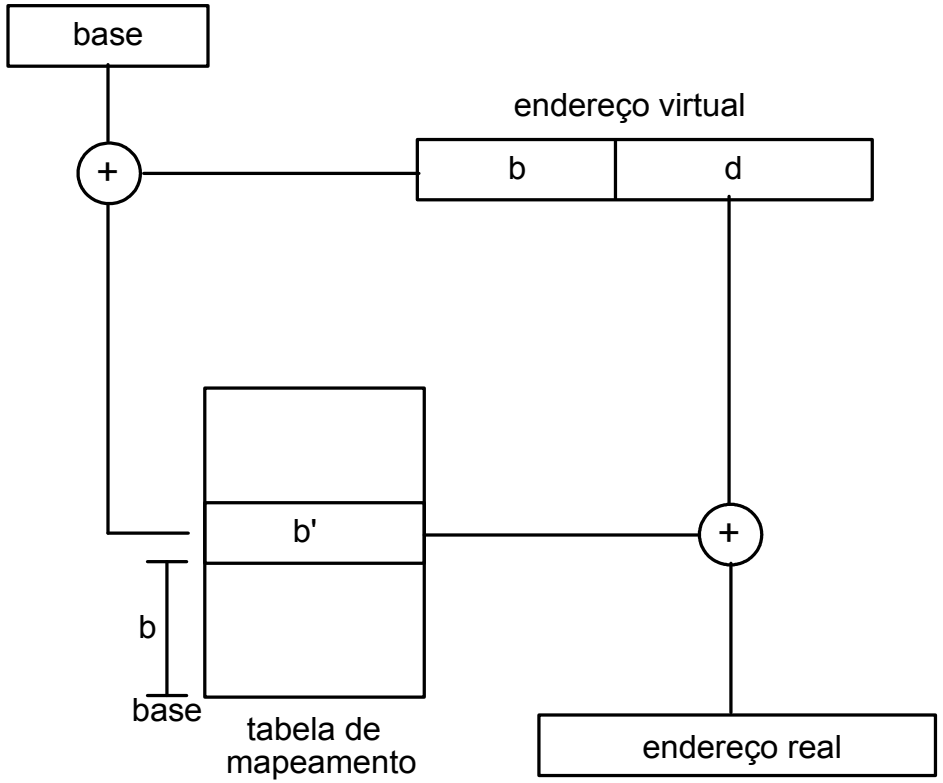
Mecanismo de Memória Virtual

- Memória virtual pode ser:
  - **Segmentada**: os blocos são chamados de **segmentos**, e podem ter **tamanhos variáveis**
  - **Paginada**: os blocos são chamados de **páginas** e possuem **tamanho fixo**

Mecanismo Básico de Mapeamento



(a)



(b)

Mecanismo Básico de Mapeamento

- Endereço virtual poder visto como constituído por dois campos
  - **campo b**: bits mais significativos
    - Indica o número do bloco onde se encontra a locação de memória referenciada
  - **campo d**: bits menos significativos
    - Indica o deslocamento da locação de memória em relação ao início do bloco

Mecanismo Básico de Mapeamento

- DAT soma o valor do **campo b** ao **endereço-base** da tabela de mapeamento
  - Endereço de uma entrada da tabela
- O endereço-base da tabela é mantido internamente no próprio DAT
  - Registrador TBR (*Table Base Register*)
- DAT acessa a tabela de mapeamento e obtém o endereço-base do bloco
- Endereço real da locação de memória
  - Valor do **campo d** + **endereço-base** do bloco

Mecanismo Básico de Mapeamento

- Para cada referência à memória realizada pelo programa é necessário um acesso adicional para consultar a tabela de mapeamento
  - Dois acessos à memória
  - Compromete o desempenho

Mecanismo Básico de Mapeamento

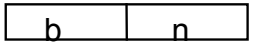
- Para solucionar este problema, o DAT possui internamente uma pequena memória, denominada **TLB** (***Translation Lookaside Buffer***)
  - Comporta-se como uma memória cache
  - Armazena os pares de endereço virtual/real, que foram usados nos acessos mais recentes
  - Em **um acerto** reduz o tempo de mapeamento

Memória Secundária

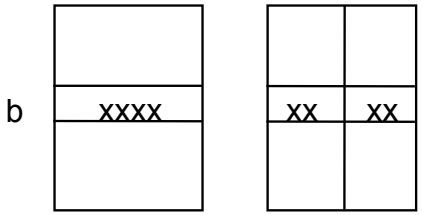
- Blocos de instruções e de dados de um programa em execução ficam armazenados na **memória secundária**
  - Unidade de disco
- A ausência de um bloco referenciado pelo programa é detectada pelo DAT no momento do mapeamento
- O bloco é então carregado da memória secundária para a memória principal

Memória Secundária

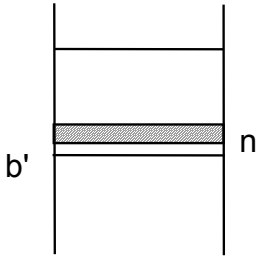
- Carregamento de blocos na MP



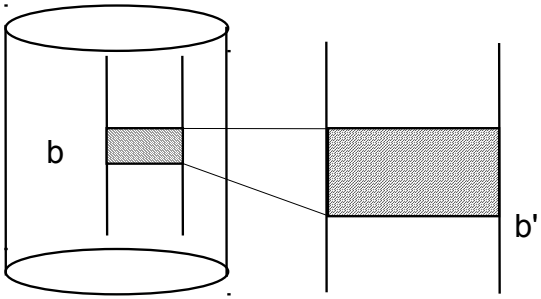
1. programa referencia locação n no bloco de número b



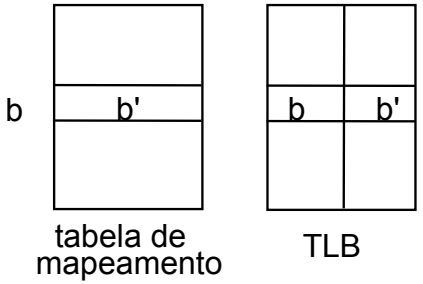
2. tabela de mapeamento e TLB não mapeiam o bloco b: o bloco referenciado não se encontra na memória principal



5. acesso à locação n



3. o bloco de número b é transferido da memória secundária para a memória principal



4. tabela de mapeamento e TLB são atualizadas

controlado pelo sistema operacional

Memória Secundária

- Ao receber o endereço virtual, o DAT verifica o conteúdo da TLB e da tabela de mapeamento
  - Um bit de controle na entrada da tabela indica ao DAT que não existe um mapeamento válido para o bloco
  - Bloco não se encontra na memória principal
  - **Falha de página** (memória virtual paginada) ou **falha de segmento** (memória virtual segmentada)
  - DAT transfere o controle para o S.O.

Memória Secundária

- O sistema operacional obtém do DAT o número do bloco que foi referenciado
  - Localiza este bloco na memória secundária, aloca um espaço na memória principal e transfere o bloco da memória secundária para a memória principal
  - Atualiza a tabela de mapeamento e a TLB e devolve o controle para o programa que estava em execução

Memória Secundária

- O controle é devolvido pelo sistema operacional exatamente para o ponto onde se encontra o acesso que provocou o carregamento do bloco
- O acesso é então re-executado pelo programa, sendo agora completado com sucesso porque o bloco referenciado já se encontra na memória principal

Estrutura Comum para Hierarquia de Memória

- Aspectos operacionais comuns aos sistemas hierárquicos da memória
- Quatro questões para tratar
  - **Onde colocar** um bloco
  - **Como localizar** um bloco
  - **Como substituir** um bloco
  - **Como modificar** um bloco

Onde Colocar um Bloco

- Diversos esquemas
  - Mapeamento direto
  - Mapeamento associativo
    - Por conjunto
    - Totalmente Associativo
- Podem ser expressos com a variação do número de:
  - Conjuntos (NC)
  - Blocos por conjunto (NB)

Onde Colocar um Bloco

- Esquema de mapeamento de blocos

Esquema	NC	NB
Direto	Número de blocos	1
Associativo por conjunto	$\frac{\text{Número de blocos}}{\text{Associatividade}}$	Associatividade
Totalmente Associativo	1	Número de blocos

Esquema de Mapeamento de Blocos

- Vantagem de aumentar associatividade
  - Melhora a localidade espacial
  - Reduz a taxa de falhas
- Desvantagens
  - Custo mais alto
  - Tempo de acesso mais longo
- Cache
  - Em geral, associativa por conjunto 2, 4, 8 e 16
- Memória Virtual
  - Totalmente associativa

Como Localizar um Bloco

- Depende do esquema utilizado para a colocação do bloco
- Escolha do mapeamento
  - Custo de uma falha comparado com o custo de implementar a associatividade
  - Termos de tempo e hardware extra
- Caches e TLB's
  - Direto
  - Associativo por conjunto

Esquema de Localização de Blocos

Esquema	Método de Localização	Número de Comparações
Direto	**índice**	**1**
Associativo por conjunto	**indexar o conjunto, pesquisar entre os elementos**	**grau de associatividade**
Totalmente associativo	**pesquisar todas as entradas tabela separada para busca**	**igual ao tamanho 0**

Como Localizar um Bloco

- Caches e TLB's totalmente associativo
  - **Proibido**
  - Exceto para caches e TLB´s muito pequenas
- Memória Virtual
  - Totalmente associativo

Como Localizar um Bloco

- A utilização de uma tabela totalmente associativa é motivada por quatro fatores
  - Alto custo das falhas e a associatividade é benéfica
  - A associatividade total permite algoritmos de substituição de páginas mais sofisticados
    - Reduzir a taxa de falhas
  - Mapeamento completo: facilidade de indexação
    - Não necessita de hardware extra
  - Quanto maior o tamanho da página
    - Menor overhead relativo ao tamanho da tabela de páginas

Como Substituir um Bloco

- Quando não há mais espaço livre
  - Substituição de um bloco
- Bloco candidato
  - Direto: Não há escolha
  - Associativo por conjunto: Um dos blocos do conjunto
  - Totalmente associativo: Qualquer bloco
- Estratégias de substituição mais comuns
  - Aleatória: Cache
  - LRU: Cache e Memória Virtual

Como Modificar um Bloco

- Duas opções básicas
  - *Write-through*: Cache
  - *Write-back*: Cache e Memória Virtual

Modelo dos três C's

- Explica as fontes de falhas
  - Compulsórias: causadas pelo primeiro acesso a um bloco que nunca está na cache
  - Capacidade: cache não pode armazenar todos os blocos, e eventualmente os blocos precisam ser substituídos
  - Conflitos ou colisão: ocorrem nas caches associativas por conjunto ou mapeadas diretamente quando diversos blocos competem pelo mesmo conjunto

Perguntas ?

Marcelo Zamith
zamith.marcelo@gmail.com