

# ARQUITETURA DE COMPUTADORES

Sistemas de memória

# Conteúdos

2

- Memória:
  - ▣ Classificação
  - ▣ Terminologia
  - ▣ Hierarquia
  - ▣ RAM
  - ▣ ROM
  - ▣ Cache
  - ▣ Externa

# Classificação da Memória

3

- Podemos classificar a memória quanto:
  - ▣ Localização
  - ▣ Volatilidade
  - ▣ Apagável/Não apagável

# Memória: Localização

4

## □ Interna

- Diretamente acessível pelo processador
- Registos do processador
- Memória principal
- Memória interna da unidade de controlo do processador

## □ Externa

- Acessível ao processador através de controladores I/O
- Disco rígido
- Dispositivos de armazenamento periféricos

# Memória: Volatilidade

5

## □ Volátil

- A informação perde-se ao desligar a corrente elétrica
- Memória principal, registos e memória cache

## □ Não volátil

- A informação mantém-se até ser alterada deliberadamente
- Nenhuma energia é requerida para manter os dados armazenados
- Discos rígidos, dispositivos externos

# Memória: Apagável / Não Apagável

6

## □ Apagável

- ▣ Conteúdo deste tipo de memórias pode ser alterado sem que se destrua a unidade
- ▣ Memórias RAM

## □ Não apagável

- ▣ O conteúdo deste tipo de memórias não pode ser alterado a menos que se destrua a unidade
- ▣ Memórias ROM

# Memória: Terminologia

7

- Taxa de transferência
- Tempo de acesso
- Tempo de ciclo
- Método de acesso

# Memória: Terminologia

8

- **Taxa de transferência:**

- Velocidade (taxa) a que os dados são transferidos de ou para a memória

- **Tempo de acesso:**

- Tempo para aceder a uma posição de memória

- **Tempo de ciclo:**

- Tempo de acesso, mais tempo necessário até que um segundo endereço possa ser acedido

- **Método de acesso:**

- Forma como o conteúdo da memória é acedido



# Memória: Taxa de transferência

9

- Velocidade a que os dados são transferidos de ou para a memória
- A taxa de transferência é diferente consoante o tipo de acesso da memória (aleatório ou não aleatório)
- Numa memória de acesso aleatório
  - $T = 1 / \text{tempo de ciclo}$
  - Tempo de ciclo – tempo de acesso, mais o tempo necessário até que um segundo endereço de memória possa ser acedido
- Numa memória de acesso não aleatório:
  - $T_N = T_A + (N/R)$
  - $T_N$  – tempo médio para ler  $N$  bits
  - $T_A$  – tempo médio de acesso
  - $N$  – número de bits
  - $R$  – taxa de transferência em bits/seg

# Memória: Tempo de acesso

10

- Tempo para aceder a uma posição de memória
- O tempo de acesso é diferente consoante o tipo de acesso da memória (aleatório ou não aleatório)
- Numa memória de acesso aleatório o tempo de acesso é o tempo entre a apresentação do endereço e a obtenção dos dados
- Numa memória de acesso não aleatório o tempo de acesso é dado pelo tempo gasto para posicionar o mecanismo de leitura-escrita na posição desejada mais o tempo necessário até que se possa realizar um próximo acesso

# Memória: Métodos de acesso

11

- Sequencial
- Direto
- Aleatório
- Associativo

# Memória: Método de acesso sequencial

12

- Os dados são organizados em unidades chamadas registros e não têm um endereço único
- Para além dos dados são também armazenadas informações de endereçamento
- É utilizado um mecanismo partilhado de leitura e escrita, que a cada operação, é movido da posição atual para a posição desejada
- O tempo de acesso depende da localização dos dados e da localização em que se encontrava antes
- Exemplo: unidade de tape

# Memória: Método de acesso direto

13

- Cada bloco ou registo tem um endereço único baseado na sua localização física
- É utilizado um mecanismo partilhado de leitura e escrita
- O acesso é feito saltando para um local na vizinhança do registo pretendido e depois a partir daí é feito sequencialmente até à posição desejada
- O tempo de acesso depende da localização pretendida e da localização anterior
- Exemplo: discos magnéticos

# Memória: Método de acesso aleatório

14

- Cada posição de memória possui um mecanismo de endereçamento único
- Qualquer posição é selecionada aleatoriamente
- O tempo de acesso é independente da localização e do acesso anterior
- Exemplo: memória RAM

# Memória: Método de acesso associativo

15

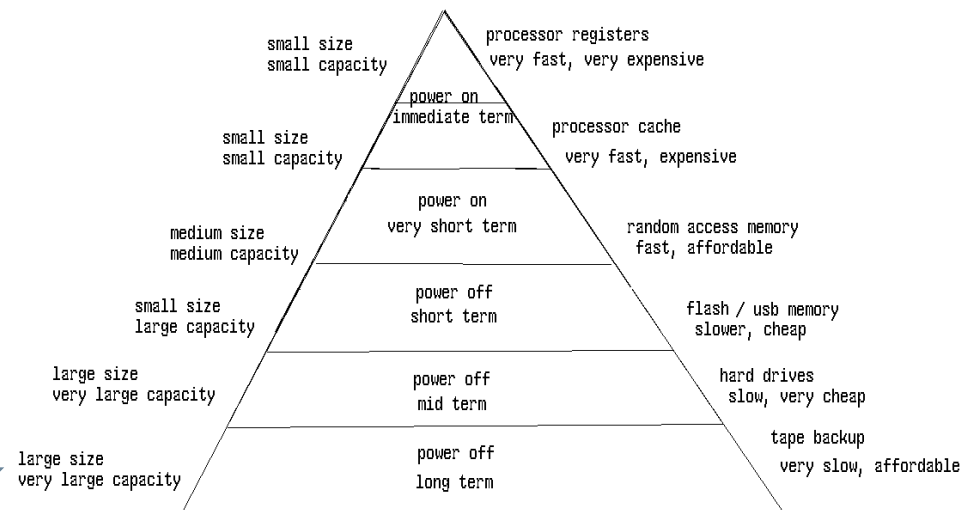
- A informação é acedida com base no conteúdo da memória e não no seu endereço
- Verifica a existência de um padrão da memória
- O tempo de acesso é independente da localização e do acesso anterior
- Todas as localizações são verificadas em paralelo – muito rápido para memórias com grande capacidade
- Exemplo: memória cache

# Memória: Hierarquia

16

- A memória está organizada segundo uma hierarquia e não num único componente ou tecnologia de memória
- O custo por bit diminui
- A capacidade aumenta
- O tempo de acesso aumenta
- A frequência de acesso à memória pelo processador diminui (chave de sucesso da hierarquia)

## Computer Memory Hierarchy



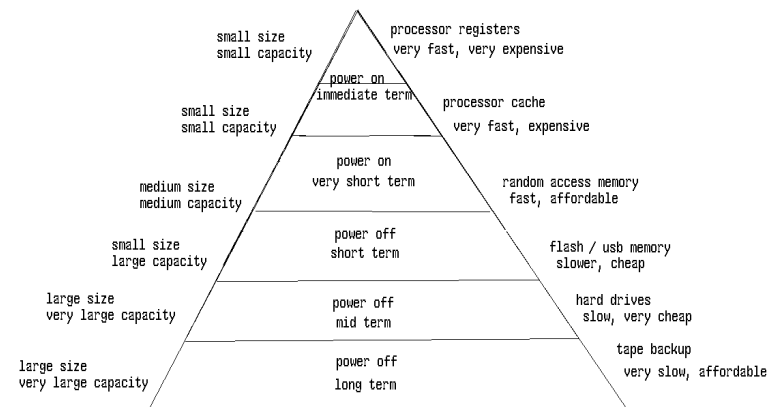


# Memória: Hierarquia

17

- A chave do sucesso da hierarquia da memória advém do facto do número de acessos diminuir de cima para baixo na pirâmide hierárquica
- Aquilo que o processador precisa está nas memórias mais próximas de si
- Isto deve-se a um princípio chamado de Localidade de Referência

Computer Memory Hierarchy  
by Dan Lash (.com)

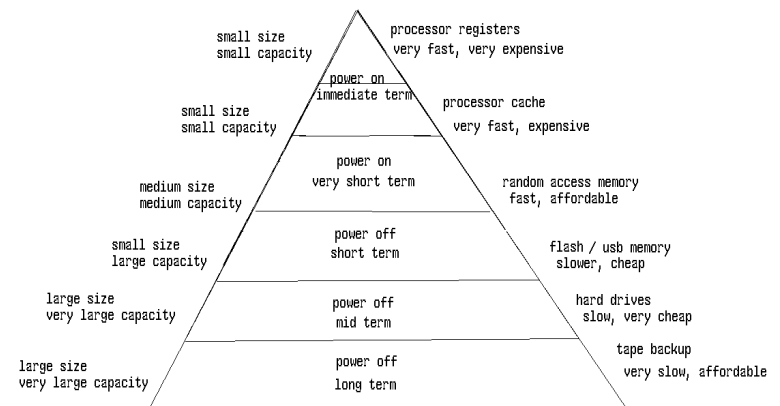


# Memória: Hierarquia

18

- Durante a execução de um programa, as referências de memória tendem a estar juntas
- Deste modo é possível organizar os dados por forma a que a percentagem de acessos de um certo nível seja bem menor que o nível imediatamente superior
- Um conjunto de dados mais usados a cada instante é passado para o nível superior
- De tempos a tempos esse conjunto é substituído por outro
- A maioria dos acessos é feito à memória de nível 1

## Computer Memory Hierarchy



# Memória: RAM

19

- A memória RAM é uma memória de acesso aleatório
- DRAM
- SRAM

# Memória: RAM

20

## □ DRAM – Dynamic RAM

### □ Vantagens:

- Barata
- Baixo consumo elétrico
- Alta densidade (pequenas)

### □ Desvantagens:

- Necessita de atualização (perde corrente)
- Lenta

# Memória: RAM

21

## □ SRAM – Static RAM

### □ Vantagens:

- Rápida
- Não necessita de atualização

### □ Desvantagens:

- Cara
- Consome mais energia (aquece mais)
- Baixa densidade (ocupa mais espaço)

# Memória: ROM

22

- A memória ROM é uma memória apenas de leitura
- Permite o armazenamento permanente de dados, o qual não pode ser alterado
- PROM
- EPROM
- EEPROM

# Memória: ROM

23

## □ PROM - Programmable ROM

- Memória apenas de leitura
- Não volátil
- Os dados apenas podem ser gravados uma vez
- A gravação é efetuada depois do fabrico da pastilha
- Precisa de equipamento especial para a sua programação

# Memória: ROM

24

- EPROM – Erasable Programmable ROM
  - ▣ Memórias principalmente de leitura
  - ▣ Apagada por exposição a radiações ultravioleta
  - ▣ Os dados podem ser lidos e gravados eletricamente
  - ▣ Antes de qualquer operação de gravação todos os dados têm de ser apagados



# Memória: ROM

25

## □ EEPROM – Electrically Erasable Programmable ROM

- Demora muito mais tempo a fazer uma escrita do que uma leitura
- Novos dados podem ser gravados sem que seja necessário apagar todo o seu conteúdo
- Toda a memória é apagada eletricamente
- É possível apagar blocos da memória e não a memória toda

# Memória: Erros

26

## □ Erro grave

- ▣ Defeito permanente: a célula de memória não é capaz de armazenar os dados com segurança podendo permanecer sempre com o valor lógico 0 ou 1 ou variar aleatoriamente entre os valores lógicos 0 e 1
- ▣ Podem ser causadas por defeito de fabrico ou uso excessivo

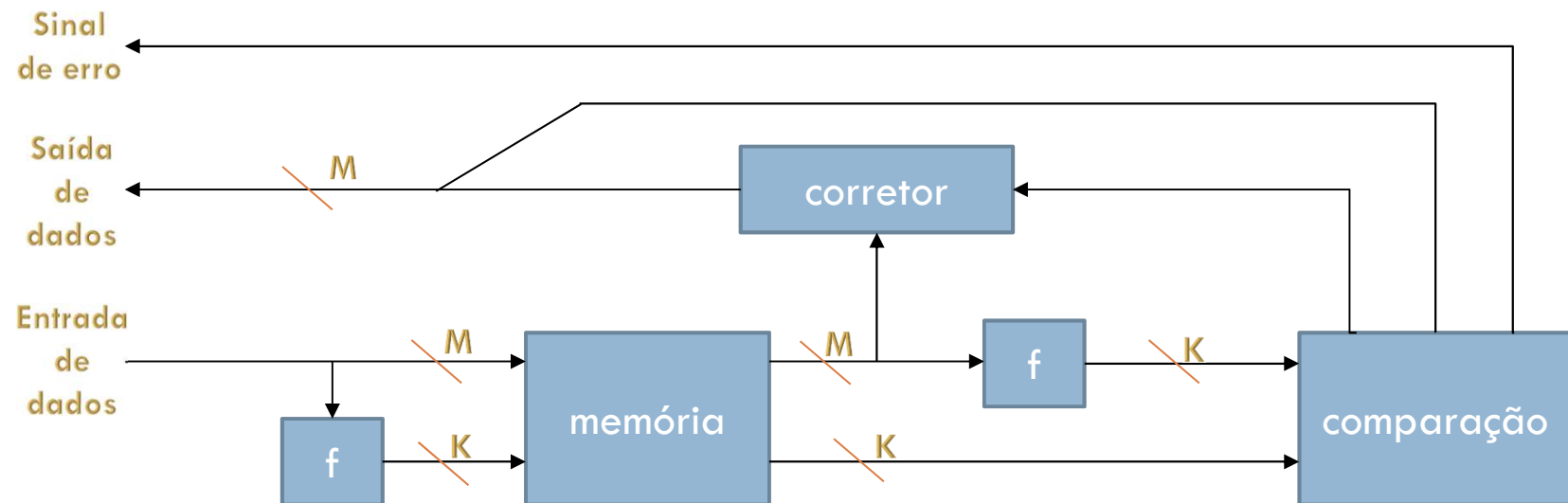
## □ Erro moderado

- ▣ Evento aleatório não destrutivo
- ▣ Altera o conteúdo de uma célula de memória sem danificar a memória
- ▣ Podem ser causados por problemas de fornecimento de energia ou pela presença de partículas alfa (resíduos radioativos)

# Memória: Detecção e Correção de erros

27

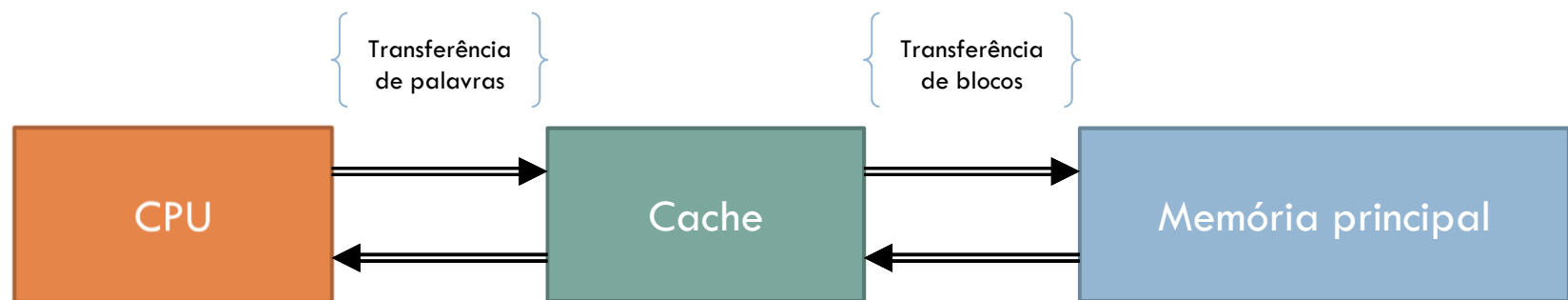
- Quando um dado ( $M$ ) é armazenado na memória é feito um cálculo ( $f$ ) envolvendo esse dado para a produção de um código ( $K$ )
- Dado armazenado =  $M + K$
- Quando a palavra armazenada é lida é gerado um novo código ( $K$ ) que é comparado com o valor  $K$  anterior



# Memória: Cache

28

- A memória cache é uma memória mais pequena mas mais rápida que a memória principal
- A memória cache situa-se entre o processador e a memória principal



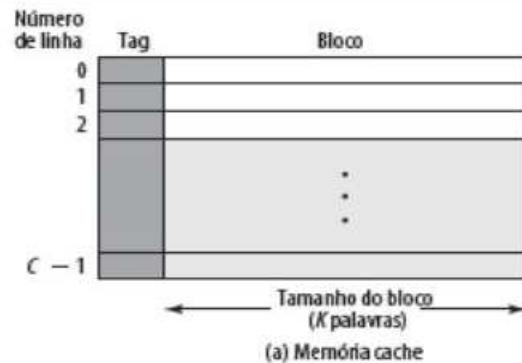
# Memória: Cache

29

- O CPU pede o conteúdo de uma posição de memória
- Em primeiro é verificado se a memória cache contém esses dados
  - ▣ Em caso afirmativo, os dados são obtidos diretamente da memória cache sem necessidade de aceder à memória principal
  - ▣ Caso contrário, um bloco de dados da memória principal, é copiado para a memória cache e posteriormente a entrega dos dados é feita da cache para o CPU

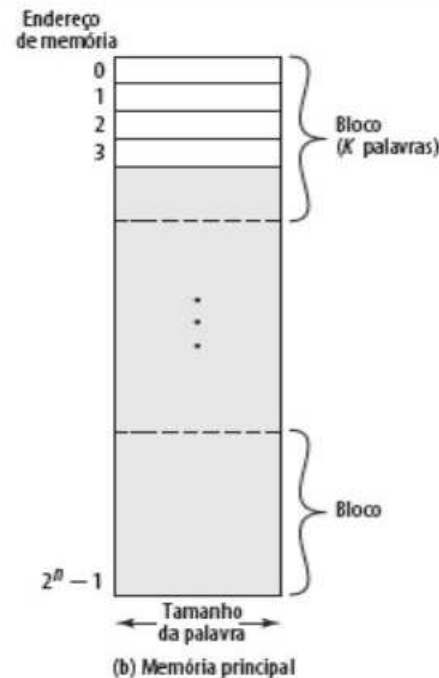
# Memória: Cache

30



## □ (a) Memória cache

- C linhas com K palavras
- $C = K \text{ palavras} + \text{TAG}$
- Bloco = K palavras

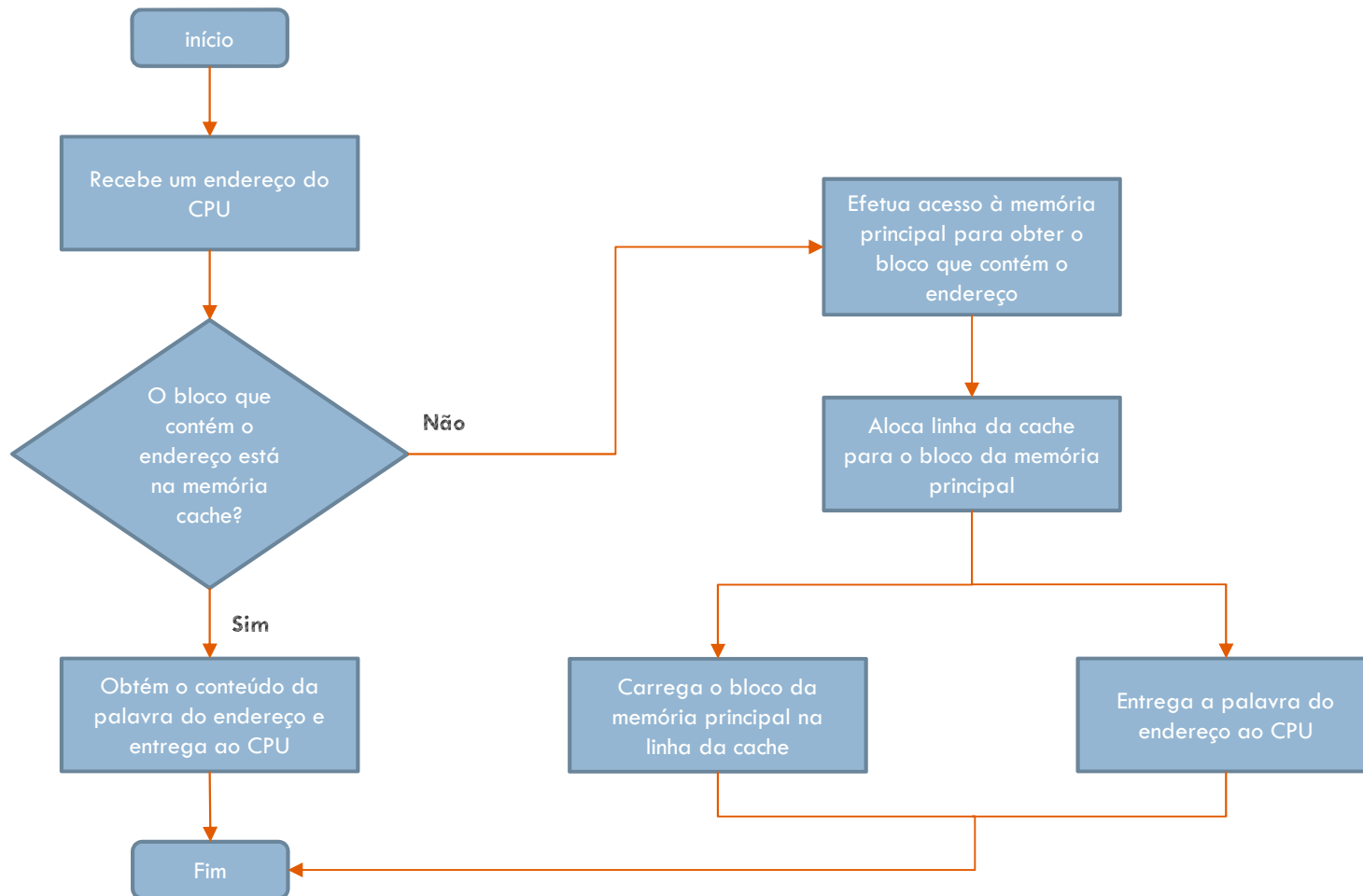


## □ (b) Memória principal

- $2^n$  palavras endereçáveis cada qual com um endereço distinto de n bits
- Para fins de mapeamento na memória cache a memória principal é constituída por um determinado número de blocos de tamanho fixo, cada qual com K palavras
- N° de blocos =  $2^n/K$

# Memória: Cache

31



# Memória: Cache

32

- Quando se projeta uma memória cache devemos ter em conta:
  - ▣ Tamanho
  - ▣ Mapeamento
  - ▣ Algoritmo de substituição
  - ▣ Política de escrita
  - ▣ Tamanho do bloco
  - ▣ Número de memórias cache



# Memória: Cache

33

- Tamanho da memória cache:
  - ▣ Deve ser suficientemente pequeno, para que o custo total médio por bit seja próximo do custo da memória principal
  - ▣ Deve ser suficientemente grande, para que o tempo médio de acesso à memória principal seja próximo do tempo de acesso da memória cache
  - ▣ Atenção porque quanto maior for a cache, maior o número de portas lógicas envolvidas, o que implica que mais lenta se torna a memória cache

# Memória: Cache

34

- Mapeamento da memória cache:
  - ▣ O mapeamento da memória cache é a forma como se copiam blocos da memória principal para a memória cache
  - ▣ Para que seja possível copiar blocos da memória principal para a memória cache são necessários dois algoritmos:
    - Algoritmo para mapear os blocos da memória principal em linhas da memória cache
    - Algoritmo para determinar em que linha da memória cache o bloco da memória principal vai ficar
  - ▣ Tipos de mapeamento:
    - Mapeamento direto (Direct Mapping)
    - Mapeamento associativo (Associative Mapping)
    - Mapeamento associativo por conjuntos (Set Associative Mapping)

# Projeto

35

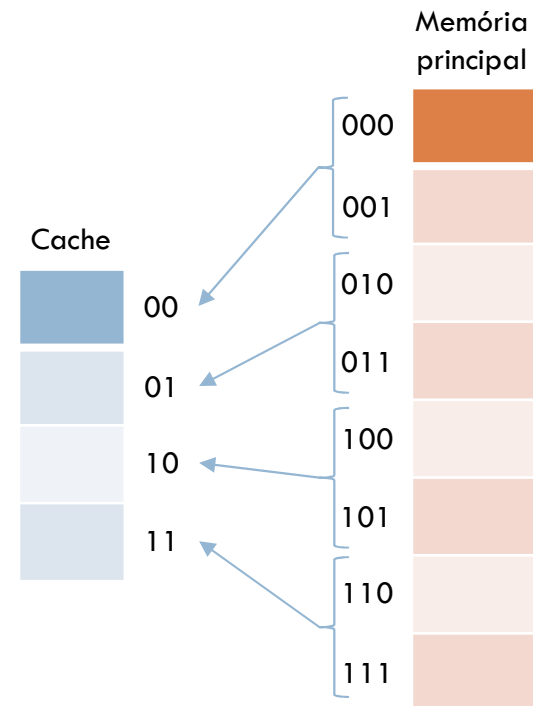
- Mapeamento direto
  - ▣ Cada bloco da memória principal é mapeado numa única linha da cache
  
- Mapeamento associativo
  - ▣ Um bloco da memória principal pode ser mapeado em qualquer linha da cache
  
- Mapeamento associativo por conjuntos
  - ▣ Um bloco da memória principal é mapeado em qualquer linha de um determinado conjunto

# Memória: Cache

36

## □ Mapeamento direto

- Simples
- Menos dispendioso
- Localização fixa para um dado bloco
- Cada bloco da memória principal é mapeado apenas para uma linha de cache



# Memória: Cache

37

## □ Mapeamento associativo

- Evita as desvantagens do mapeamento direto uma vez que cada bloco da memória principal pode ser escrito em qualquer linha da memória cache
- Permite o uso de algoritmos de substituição para maximizar a taxa de acertos
- Apresenta como desvantagem a complexidade do conjunto de circuitos necessários para a comparação de todos os blocos existentes em simultâneo

# Memória: Cache

38

- O mapeamento associativo por conjuntos combina as vantagens dos dois métodos anteriores e minimiza as suas desvantagens.

# Memória: Cache

39

## □ Algoritmos de substituição

- Quando um novo bloco é trazido da memória principal para a memória cache, é necessário substituir um dos blocos existentes na memória cache
- No **mapeamento direto** cada bloco como cada bloco é mapeado numa determinada linha então automaticamente sabemos sempre qual o bloco a ser substituído
- No **mapeamento associativo** ou **mapeamento associativo por conjuntos** é necessário a utilização de um algoritmo de substituição:
  - **LRU** (*least recently used*): o bloco a ser substituído é o bloco que não é usado à mais tempo
  - **FIFO** (*first-in-First-out*): o bloco a ser substituído é o bloco que está na cache á mais tempo
  - **LFU** (*least frequently used*): o bloco a ser substituído é o bloco que menos vezes foi utilizado
  - **Substituição aleatória**

# Memória: Cache

40

## □ Políticas de atualização

- Antes que um bloco residente na memória cache possa ser substituído por outro é necessário verificar se ele foi alterado na memória cache
- Se o bloco na memória cache não foi alterado então o bloco pode ser substituído
- Se o bloco na memória cache foi alterado então é preciso copiá-lo para a memória principal antes de ser substituído:
  - Escrita direta (*write trough*):
    - Todas as operações de escrita são feitas tanto na memória cache como na memória principal
    - Tem como desvantagem o tráfego gerado
  - Escrita de volta (*write back*):
    - Apenas a memória cache é atualizada
    - Quando é feita a atualização é atribuído o valor 1 a um bit de atualização para quando for preciso substituir o bloco sabermos que é necessário copiá-lo para a memória principal
    - Tem como desvantagem existir apenas um local onde a informação atualizada está armazenada



# Memória: Cache

41

## □ Tamanho do bloco

- Pelo princípio da localidade de referência o facto das palavras serem trazidas em blocos aumenta a taxa de acerto
- No entanto se o tamanho do bloco for muito grande a taxa de acerto tende a diminuir porque cada palavra adicional está mais distante da palavra usada, logo, pelo princípio da localidade de referência, é mais improvável que seja utilizada

# Memória: Cache

42

## □ Número de memórias cache

### □ Vantagens da utilização de uma única memória cache:

- A taxa de acerto é maior
- Apenas uma memória cache precisa de ser projetada e implementada

### □ Atualmente usam-se memórias cache separadas para instruções e dados:

- Elimina a disputa por acesso à memória principal entre o processador e a unidade de execução
- Particularmente útil em processadores escaláveis, que antecipam a execução de instruções

# Memória: Externa

43

- Disco magnético
- Memória ótica
  - ▣ CD-ROM; CDR; CDRW; DVD;
- Fita magnética
- Etc...

# Memória: Externa

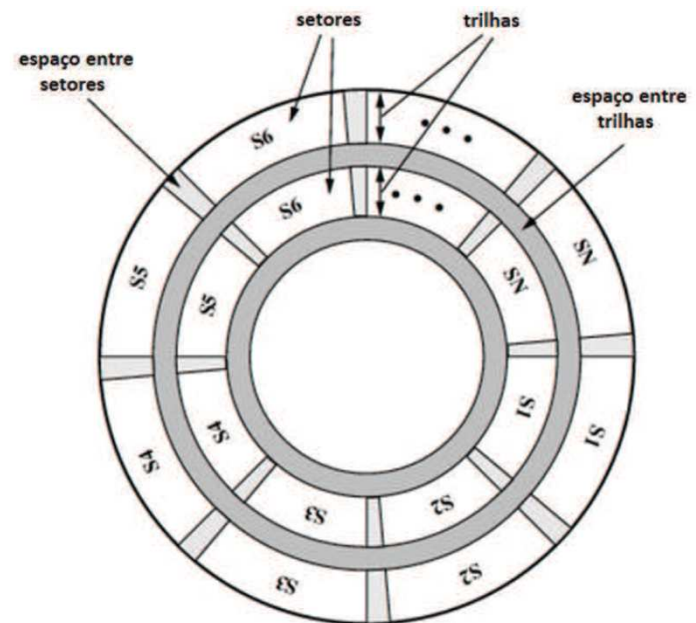
44

- O disco magnético é um disco de metal ou plástico, coberto com um material que pode ser magnetizado
- Os dados são gravados e posteriormente lidos do disco através de uma bobine condutora denominada cabeça de leitura/escrita, à qual são aplicados impulsos de corrente elétrica
- A cabeça de leitura/escrita permanece fixa e o disco gira

# Memória: Externa

45

- O disco magnético é constituído por:
  - ▣ Anéis concêntricos ou pistas ou trilhas
  - ▣ Cada pista tem a mesma largura da cabeça de leitura/escrita
  - ▣ Pistas adjacentes são separadas por espaços por forma a diminuir a taxa de erros por falta de alinhamento da cabeça de leitura/escrita ou devido a interferências eletromagnéticas



# Memória: Externa

46

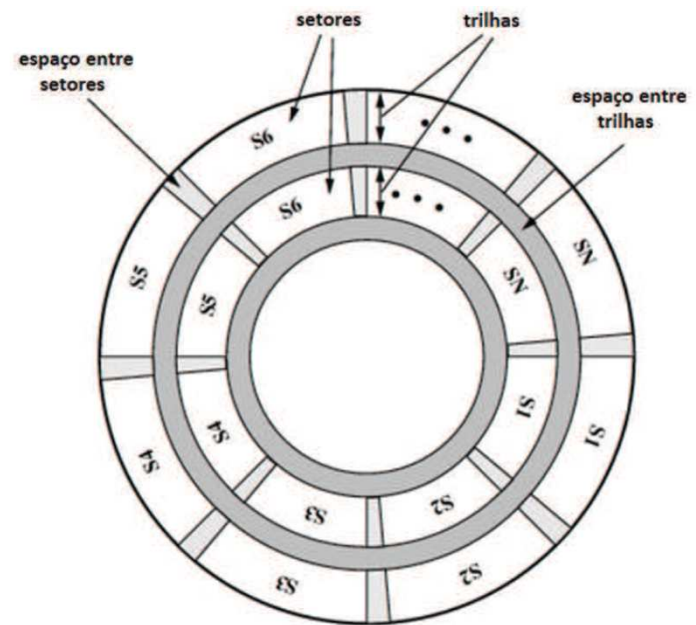
□ O disco magnético é constituído por:

□ ...

□ Cada pista tem o mesmo número de bits

□ Os dados são transferidos em blocos e armazenados em setores

□ Os setores adjacentes são separados por espaços



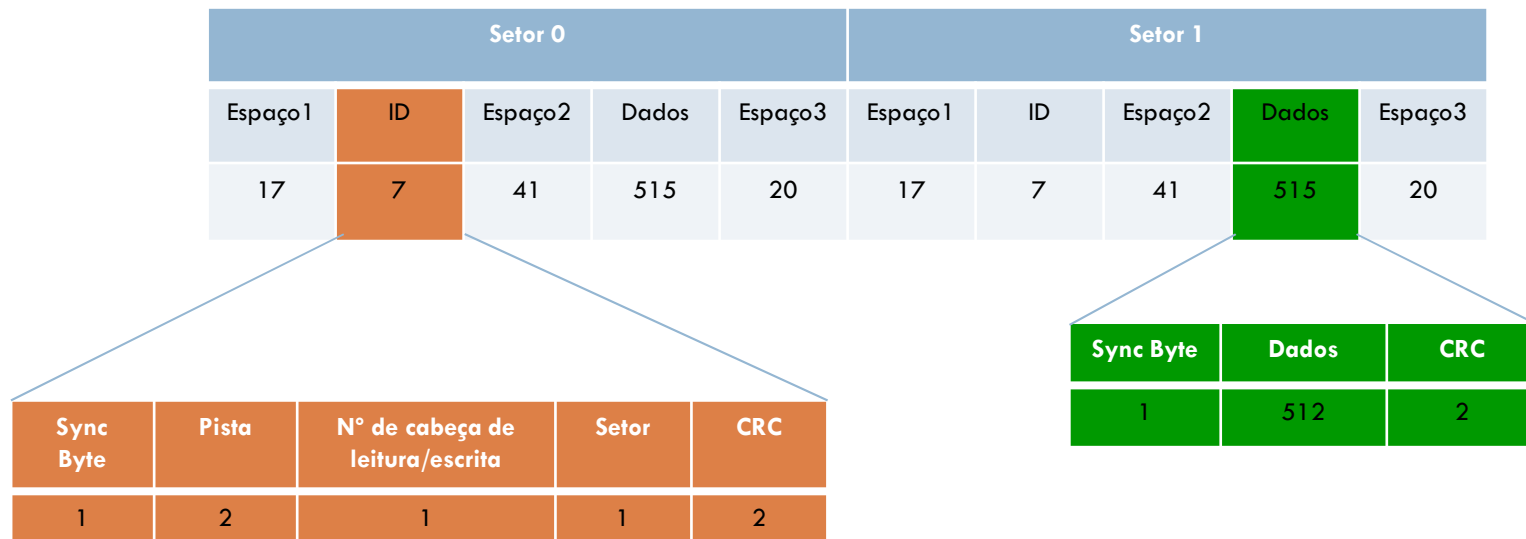
# Memória: Externa

47

- Para poder aceder á informação de um disco magnético é necessário identificar o início de uma pista e de um setor
- A identificação de uma pista e de um setor tem a ver com a formatação do disco:
  - ▣ Esta informação não está disponível ao utilizador
  - ▣ É a formatação do disco magnético que marca as pistas e os setores

# Memória: Externa

48

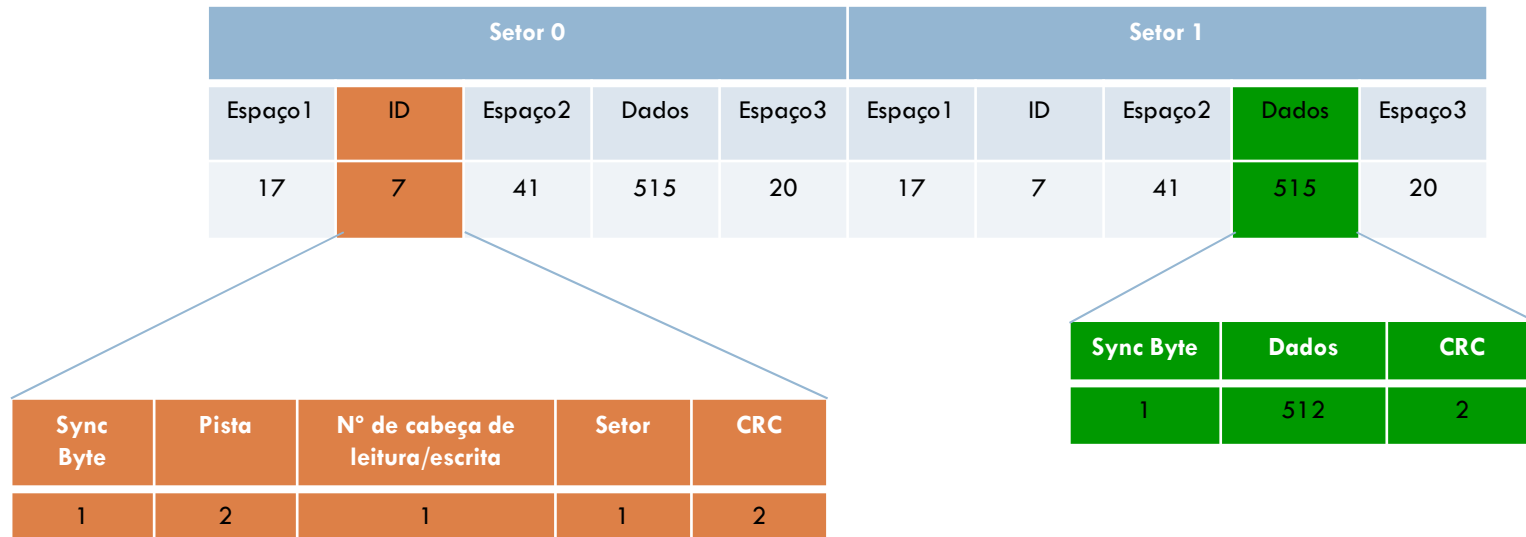


- ❑ Cada pista tem N sectores
- ❑ Cada setor tem vários bytes
  - ❑ No exemplo da figura cada setor tem 600 bytes (17+7+41+515+20)



# Memória: Externa

49



- No exemplo da figura:
  - ▣ Existem 512 bytes de dados
  - ▣ O campo ID (identificador do setor) é constituído por vários bits: Sync (início do campo), Pista (identificador da pista), Nº da cabeça de leitura/escrita (quando existem várias cabeças de leitura/escrita) e CRC (bytes de correção de erros)

# Memória: Externa

50

- Podemos classificar um disco magnético quanto:
  - ▣ Cabeça de leitura/escrita fixa ou móvel
  - ▣ Removível ou fixo
  - ▣ Um só lado ou com dois lados (mais comum)
  - ▣ Um só prato ou múltiplos pratos
  - ▣ Mecanismo de leitura/escrita

# Memória: Externa

51

- Cabeça de leitura/escrita fixa ou móvel
  - ▣ Cabeça de leitura/escrita fixa
    - Uma cabeça de leitura/escrita por pista
    - Cabeças montadas num braço fixo
  - ▣ Cabeça de leitura/escrita móvel
    - Uma cabeça de leitura/escrita
    - Montadas num braço que pode ser estendido ou retraído

# Memória: Externa

52

- Removível ou fixo

- ▣ Disco removível

- Pode ser retirado da *drive* e substituído por outro disco
    - Permite uma capacidade de armazenamento ilimitada
    - Facilidade na transferência de dados entre sistemas

- ▣ Disco não removível

- Permanentemente montado na *drive*

# Memória: Externa

53

## □ Um só lado ou com dois lados

### □ Único lado

- A cobertura magnetizável é aplicada apenas a um dos lados do prato
- Sistemas mais baratos

### □ Duplo lado

- A cobertura magnetizável é aplicada aos dois lados do prato
- Sistemas mais caros

# Memória: Externa

54

## □ Mecanismo de leitura escrita

### □ Contacto físico (Disquetes)

### □ Intervalo fixo

- O intervalo existente refere-se à camada de ar existente entre a cabeça de leitura/escrita e o prato

### □ Intervalo aerodinâmico (*Winchester*)

- A cabeça de leitura/escrita repousa sobre a superfície do disco
- Quando o disco gira as cabeças de leitura/escrita levantam-se ligeiramente acima da superfície do disco

# Memória: Externa

55

- Terminologia dos discos magnéticos:
  - ▣ Tempo de procura (*Seek Time*)
    - Representa o tempo necessário para mover a cabeça de leitura/escrita para a pista correta
  - ▣ Latência rotacional
    - Representa o tempo gasto enquanto os dados não começam a passar debaixo da cabeça de leitura/escrita
  - ▣ Tempo de acesso
    - Representa a soma do Tempo de procura com a Latência rotacional
  - ▣ Tempo de transferência
    - Representa o tempo necessário para ler ou escrever os dados

# Memória: Externa

56

- É possível agrupar um conjunto de discos independentes sem qualquer relação hierárquica entre eles por forma a construir sistemas de RAID
- Os sistemas RAID são constituídos por um conjunto de discos físicos que são vistos pelo sistema operativo como uma única unidade de disco lógica
- Nos sistemas RAID:
  - ▣ Os dados são distribuídos pelas várias unidades de disco físicas do agrupamento (*Disk Striping*)
  - ▣ O restante espaço é utilizado para armazenar informação de paridade, que é utilizada, se necessário para recuperar os dados