Pipelines, Memória

MC404

Pipelines

- Considere um processador com os seguintes passos de execução de uma instrução:
  - Busca de instrução (*fetch*)
  - Decodificação (*decode*)
  - Execução (*execute*)
  - Acesso a memória (*memory access*)
  - Escrita de registradores (*writeback*)

Pipelines

- Para agilizar a execução, cada estágio é construído de forma que possa executar de forma independente.
- Dessa forma, todos os estágios podem executar concorrentemente

Pipelines

```
@ exemplo de código

    add     r2,r1,r3
    sub     r4,r1,r3
    mov     r3,r1
    orr     r5,#1
    str     r3,contador
```

	1	2	3	4	5	6	7	8	9
Busca	add	sub	mov	orr	str				
Decod		add	sub	mov	orr	str			
Exec			add	sub	mov	orr	str		
Mem				add	sub	mov	orr	str	
Escrita					add	sub	mov	orr	str

Pipelines

- Muito problemas práticos para serem resolvidos:
  - Dependências
  - Desvios
  - Exceções, interrupções
  - Interações com o cache

Pipelines

```
@ exemplo de código com dependência

    add     r2,r1,r3
    sub     r4,r2
    and     r3,r2
    orr     r5,#1
    str     r3,contador
```

	1	2	3	4	5	6	7	8	9
Busca	add	sub	mov	orr	str				
Decod		add	sub	and	orr	str			
Exec			add	sub	and	orr	str		
Mem				add	sub	and	orr	str	
Escrita					add	sub	and	orr	str

Pipelines

- O que fazer no caso de dependências?
- Instrução seguinte pode ter que ser atrasada
  - Causa “bolhas" na pipeline
- Uma solução (nem sempre aplicável):
  - rearrumar a ordem das instruções
    - montador realiza esse trabalho

Pipelines

- O que fazer no caso de Exceções e interrupções?
  - Uma solução (simples mas cara):
    - Esgotar a pipeline antes de aceitar a interrupção

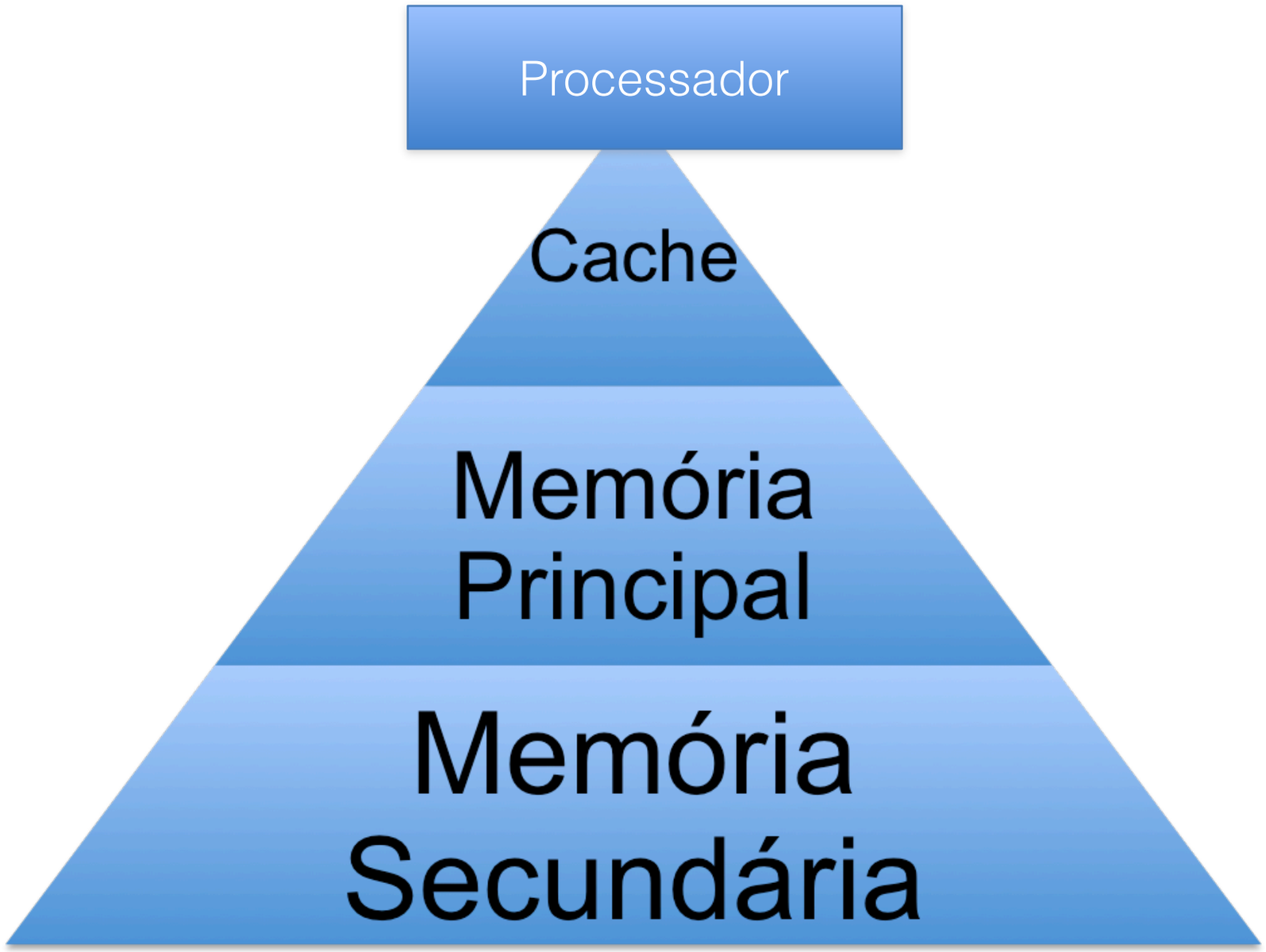
Pipelines

- O que fazer no caso de desvios?
    - Instruções seguintes *podem* ter que ser anuladas
    - Soluções:
        - Instrução adiada
            - sempre executa a próxima instrução após o desvio
            - montador pode rearrumar a ordem das instruções
        - Abrir dois fluxos
        - Predição de desvio

Hierarquia de Memória

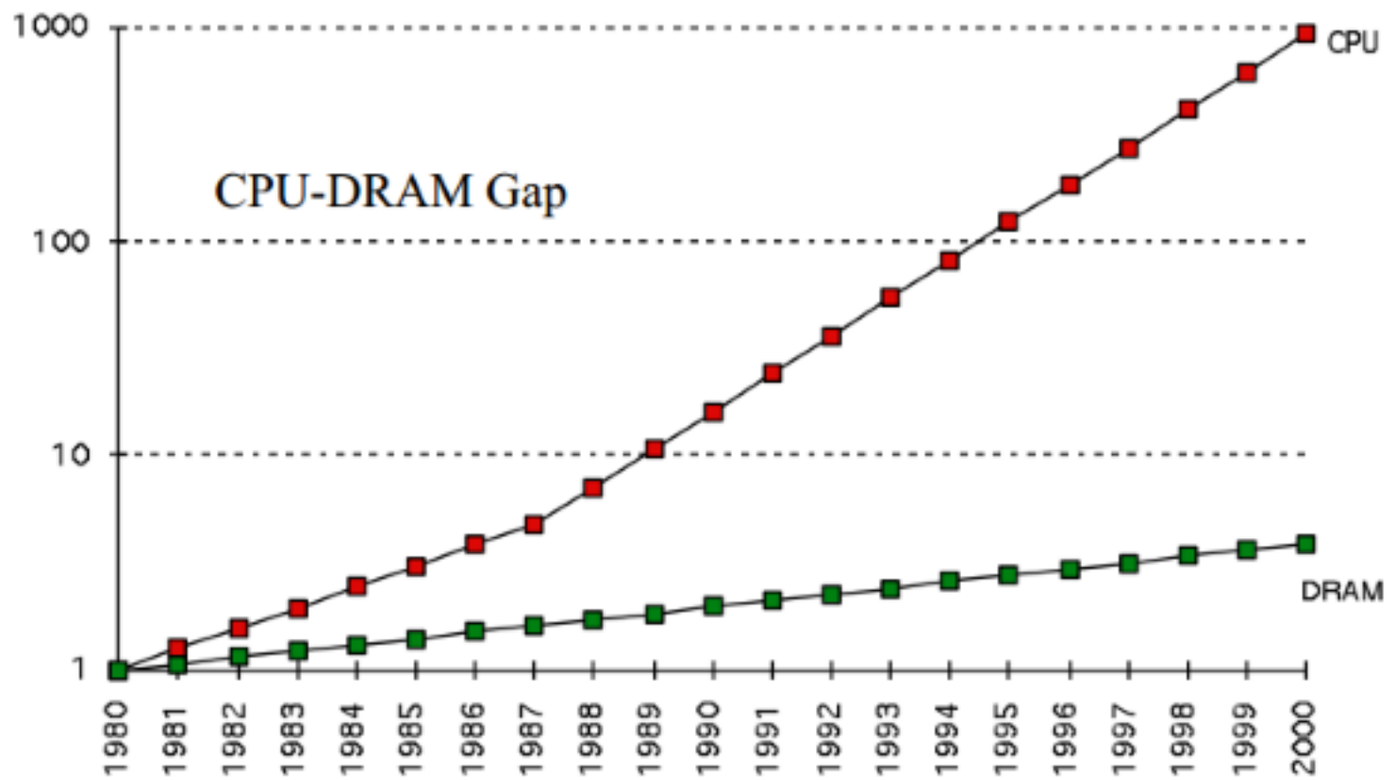
- A divisão Processador x Memória utilizada até aqui como modelo não é real
- Na realidade há uma hierarquia de memórias, de tempos de acesso crescente quanto maior a “distância” da memória para o processador
- As memórias mais próximas do processador são chamadas de “cache”
  - Atualmente é comum estarem no mesmo chip do processador

Hierarquia de memória



Cache

- Processor vs Memory Performance



1980: no cache in microprocessor;

1995 2-level cache

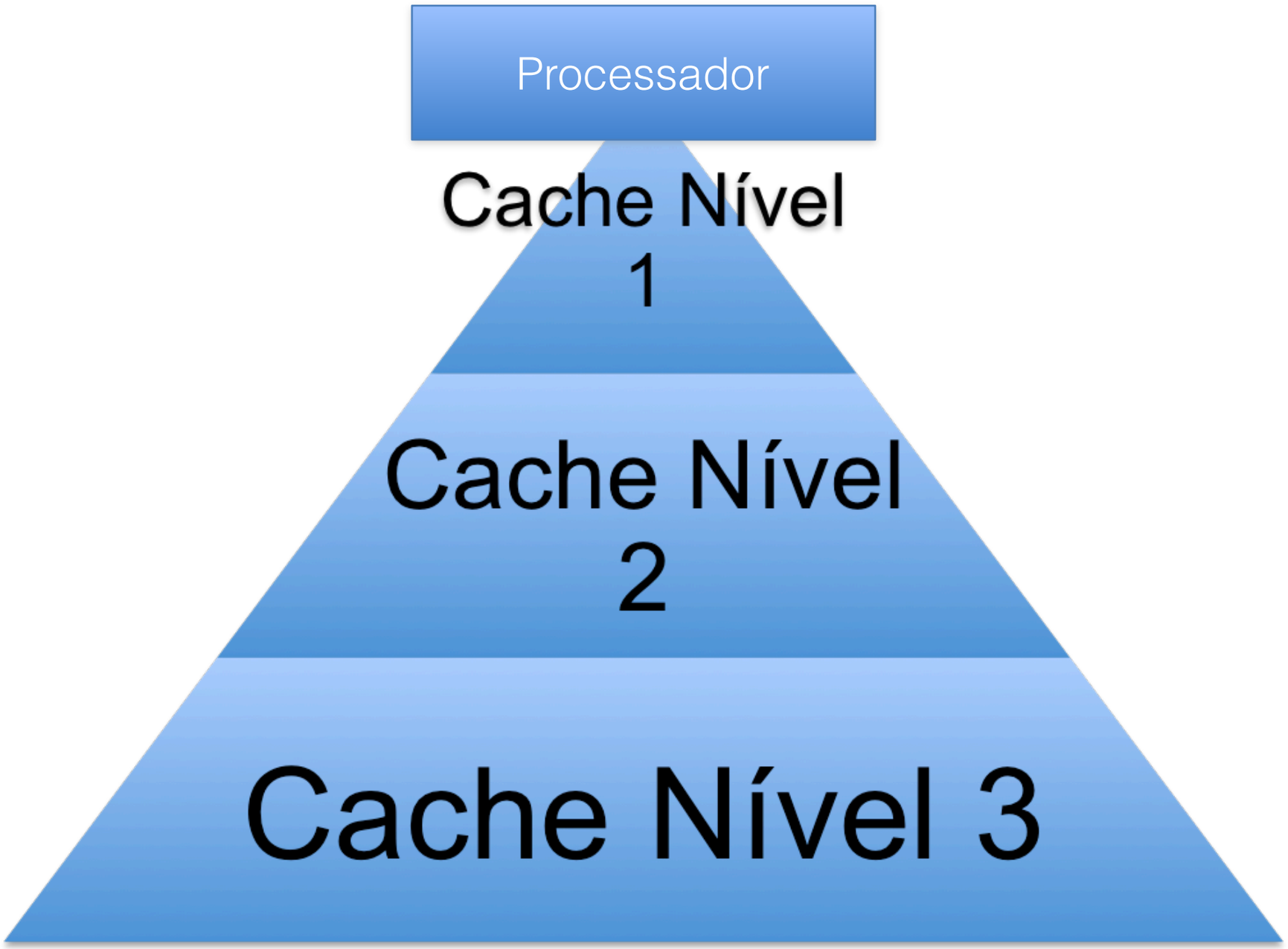
Cache

- Pequeno conjunto de memória muito rápida que contém a informação que “mais provavelmente" o processador vai usar em seguida.

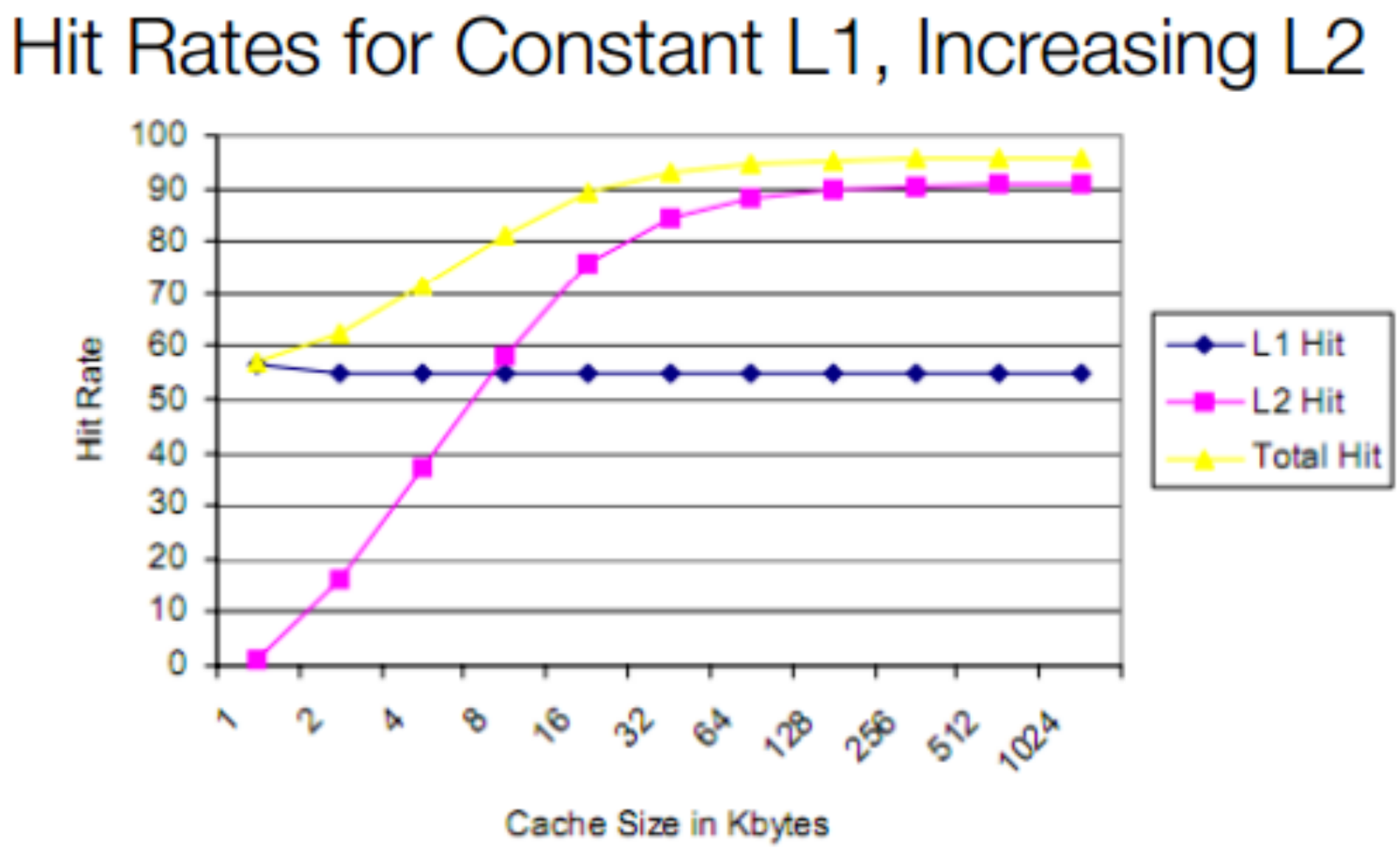
Cache

- Tipos de caches
  - Dados e Instruções no mesmo cache
  - Dados e instruções em caches separados
    - Arquitetura Harvard, mais comum agora
  - Caches em Níveis
    - Se não acha em um nível, procura em outro
    - Inclusiva – nível mais baixo inclui conteúdo do mais alto
    - Exclusiva – cada cache tem conteúdo distinto
    - AMD Athlon64: L1=64MB, L2=1 MB

Cache em níveis



Cache em níveis



Cache

- Implementação:
  - Necessita acesso muito rápido
  - Obviamente, nem toda palavra da memória cabe no cache: usa *memória associativa*
  - Esse tipo de memória não é organizada como um “vetor”, é organizada como um “dicionário"
  - Se a palavra do endereço procurado está no cache: HIT
  - Se a palavra do endereço procurado não está no cache: MISS

Cache

- Problemas a resolver
  - O que fazer quando a palavra procurada não está presente?
  - Políticas de atualização de cache
    - FIFO (first in, first out)
    - LRU (least recently used)
    - MRU (most recently used)
    - … qual a “ótima?”
  -

Cache

- Problemas a resolver
  - Coerência: se memória é atualizada (com uma instrução store), hierarquia deve ser atualizada
    - Sempre, todos os níveis?
    - Apenas quando for “necessário"?

Outros tipos de Cache

- Cache de “traços” de execução
  - No Pentium 4, usado para evitar decodificar as instruções
  - No PowerPC, usado para emular outros processadores de maneira eficiente (evita tradução que já foi feita)

Memória Primária

Memória primária

- É a memória com a qual estamos acostumados, onde os programas (código e dados) são armazenados enquanto estão em execução. Podem ser:

  - Leitura e escrita: RAM (nome vem de acesso “aleatório”, significando que qualquer endereço tem o mesmo tempo de acesso),

  - Leitura apenas: ROM (por exemplo trechos do sistema operacional, como a BIOS)

RAM

- Várias tecnologias, mas podem ser basicamente de dois tipos:
- RAM estática
  - Mais rápidas, mais caras, não precisam de "refresh"
- RAM dinâmica
  - Mais lentas, mais baratas, precisam de re-escrita milhares de vezes por segundo (um circuito auxiliar, dentro da memória, faz isso, é transparente para o usuário)

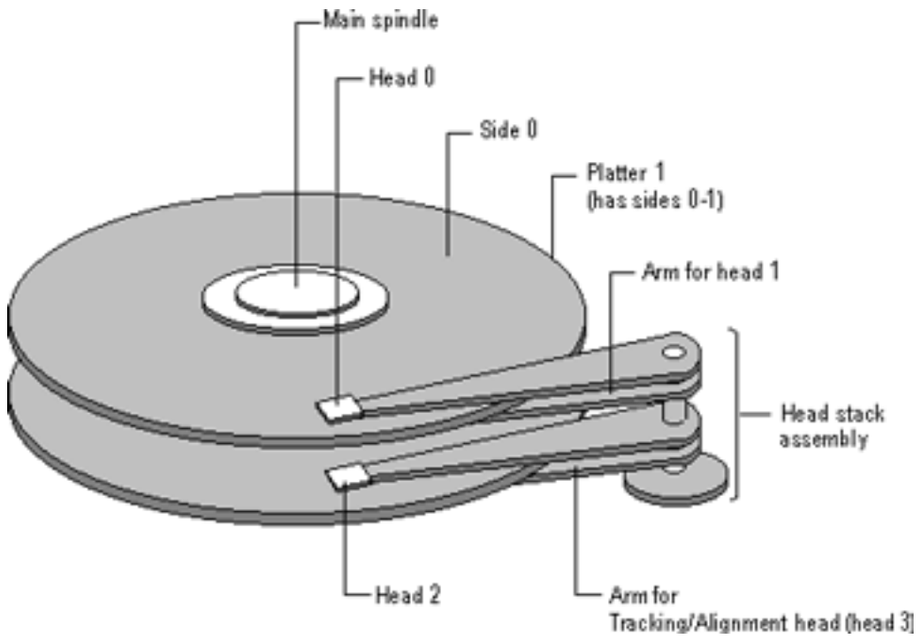
Memória Secundária

Memória Secundária

- Não é acessível diretamente pelo processador, que tem que se comunicar, via E/S, com o periférico para acessar o seu conteúdo
- Exemplos: discos, cartões de memória, pen-drives

Discos

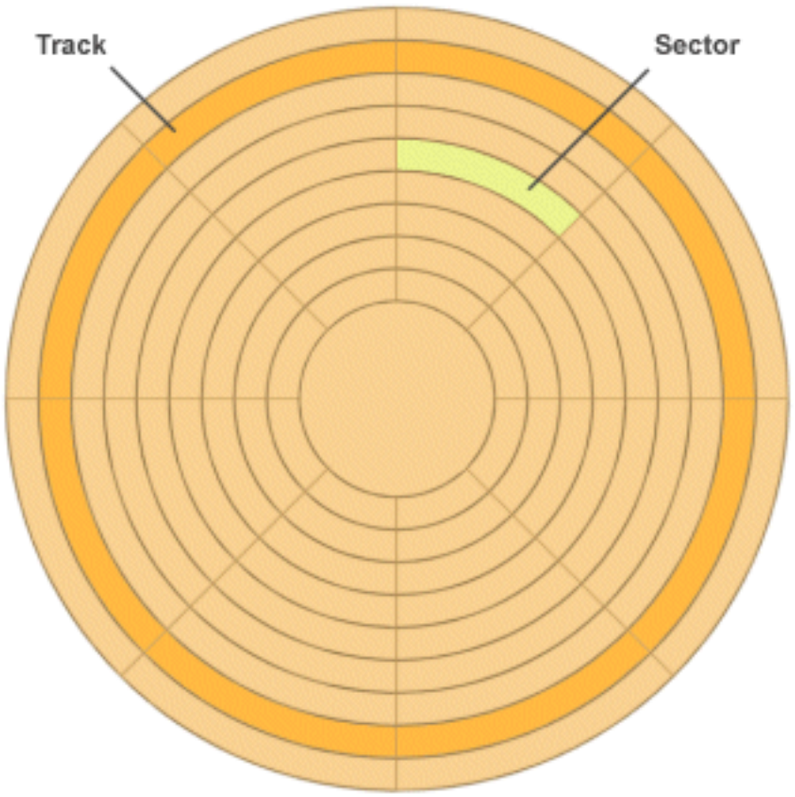
- O tipo normalmente presente em computadores ainda tem uma componente mecânica: disco magnético que gira a velocidades variando de 4500 a 7200 rpm.



- Discos SSD (Solid State Disk) não têm partes móveis, são implementados apenas com memória não volátil.

Discos

- Cada “disco" é dividido em
  - Faces (2)
- Cada Face é dividida em
  - trilhas (track)
- Cada Trilha é dividida em
  - setores (sector)



Discos

- Cada setor armazena tipicamente entre 512 e 2048 bytes (4096 para discos mais recentes), mais alguns bytes de cabeçalho (identificação do setor, códigos de correção de erros, etc).

- A velocidade do disco é o tempo médio de acesso (depende basicamente da velocidade de rotação, mas também de tamanhos de buffers)

Discos SSD

- Usam memória “flash" não para implementar um dispositivo de armazenamento
- memória flash é uma memória de acesso aleatório com tecnologia não volátil (não perde os dados sem energia), usada em pen-drives e SSDs.
  - bem mais lenta do que RAM estática

Memória Terciária

Memória Terciária

- Para fins de backup, ou de necessidade de histórico/auditorias, são usadas fitas magnéticas
    - mais baratas por MB do que discos
    - maior durabilidade
    - muito mais lentas (acesso aos dados é sequencial), mecanismo tem que fazer fita ir para frente ou para trás